



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Tecnológico Nacional de México

**Centro Nacional de Investigación
y Desarrollo Tecnológico**

Tesis de Maestría

**Método para Combinar Patrones de Análisis
mediante Ontologías Temporales**

presentada por

Lic. Valentino Velázquez Olivera

como requisito para la obtención del grado de
Maestro en Ciencias de la Computación

Director de tesis

Dr. Moisés González García

Cuernavaca, Morelos, México. Febrero de 2020.



"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

Cuernavaca, Mor., **28/enero/2020**

OFICIO No. DCC/023/2020

Asunto: Aceptación de documento de tesis
CENIDET-AC-004-M14-OFFICIO

C. DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial del **C. Lic. Valentino Velázquez Olivera**, con número de control M18CE021, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis de grado titulado "**Método para Combinar Patrones de Análisis mediante Ontologías Temporales**" y hemos encontrado que se han atendido todas las observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

Dr. Moisés González García
Doctor en Ciencias en la Especialidad de
Ingeniería Eléctrica
7501724
Director de tesis

Dr. René Santaolaya Salgado
Doctor en Ciencias de la Computación
4454821
Revisor 1

Dr. Noé Alejandro Castro Sánchez
Doctor en Ciencias de la Computación
08701806
Revisor 2

C.c.p. Depto. Servicios Escolares
Expediente / Estudiante
JGGS/lmz



"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

Cuernavaca, Morelos, **30/enero/2020**

OFICIO No. SAC/100/2020
Asunto: Autorización de impresión de tesis

LIC. VALENTINO VELÁZQUEZ OLIVERA
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS
DE LA COMPUTACIÓN
P R E S E N T E

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado "*Método para Combinar Patrones de Análisis mediante Ontologías Temporales*", ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

ATENTAMENTE
Excelencia en Educación Tecnológica®
"Conocimiento y tecnología al servicio de México"

DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO

SEP TecNM
CENTRO NACIONAL
DE INVESTIGACIÓN
Y DESARROLLO
TECNOLÓGICO
SUBDIRECCIÓN
ACADÉMICA

C.p. M.E. Guadalupe Garrido Rivera. Jefa del Departamento de Servicios Escolares.
Expediente.

GVGR/CHG

Dedicatoria

*“La familia no es algo importante. Lo es todo.”
(Michael J. Fox)*

A mis padres, **Humberto Velázquez Zárate** y **Micaela Olivera Mendoza**, por apoyarme en todo momento, por sus consejos y palabras de aliento, por sus regaños y por la constante motivación, por la educación que me inculcaron, pero sobre todo por su amor incondicional. Con mucho amor y cariño, este logro es también de ustedes.

A mi esposa e hijo, **Gabriela Márquez Zárate** y **Axel Gabriel Velázquez Márquez**, por llenar mi vida de alegría y por ser la motivación y fuente de inspiración para superarme. Gaby, eres una mujer maravillosa y soy muy afortunado de tenerte a mi lado, gracias por tu amor incondicional. Axel, te quiero mucho y siempre estaré para ti, gracias por enseñarme que el único amor perfecto en este mundo, es aquel del padre por su hijo.

A mis hermanos, **Bruno**, **Alfredo**, **Daniel**, **Oscar**, **Fabiola** y **Marcos Hugo**, por su apoyo y cariño, porque pese a tantos momentos difíciles siempre hemos salido adelante, por las tantas experiencias vividas y por las que aún nos faltan vivir, les agradezco de todo corazón y es un orgullo para mí ser su hermano.

Agradecimientos

Al **Consejo Nacional de Ciencia y Tecnología (CONACYT)** por el apoyo económico para realizar este trabajo de investigación.

Al **Tecnológico Nacional de México (TecNM) / Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET)**, por darme la oportunidad de realizar mis estudios de posgrado en esta destacada institución.

A mi director de tesis, **Dr. Moisés González García**, por su tiempo, disponibilidad, paciencia y conocimientos para la culminación de este trabajo. Gracias por su confianza y por su gran aporte a mi formación profesional.

A los doctores, **René Santaolaya Salgado** y **Noé Alejandro Castro Sánchez**, por sus conocimientos, y valiosas aportaciones para enriquecer este trabajo, por su disponibilidad y tiempo en la revisión de este trabajo de tesis, sin su ayuda esto no hubiese sido posible.

Al Cuerpo Académico de Ingeniería de Software (CAIS) por las enseñanzas transmitidas en cada una de las clases y asesorías brindadas. Gracias por aportar su granito de arena en la culminación de mis estudios de posgrado satisfactoriamente.

A mis amigos, **Antonio Juan, Orlando, Juan Vicente** y **Julia**, por todos los momentos que pasamos juntos en estos dos años de preparación, por los buenos y malos momentos, pero sobre todo por su amistad.

A todas aquellas personas que forman parte de mi vida y que han sido partícipes de este logro, muchas gracias.

Resumen

Hoy en día, la demanda en la construcción de sistemas de software va en aumento, debido a esto, es importante considerar la reutilización en las fases tempranas del proceso de desarrollo de software, para disminuir los recursos requeridos y el esfuerzo total realizado. Los patrones de análisis, son elementos de una técnica de reutilización utilizada en la comunidad de ingeniería de software debido a que agilizan el proceso de análisis y mejoran la calidad en el desarrollo de sistemas de software. Actualmente, con la Metodología para la Construcción de Ontologías Temporales de Patrones de Análisis (MECOTPA), es posible construir ontologías temporales de patrones de análisis que incluyen en su definición aspectos estructurales y de comportamiento dinámico, utilizando la lógica descriptiva y la lógica temporal, respectivamente. Esto permite tener una descripción más completa de los patrones de análisis. Sin embargo, es posible que en algún momento se requiera la combinación de dos o más patrones de análisis para representar problemas más grandes que ocurren en la fase de análisis del proceso de desarrollo de software. Por tal motivo, en este trabajo de investigación, se presenta el método para la Combinación de Ontologías Temporales de Patrones de Análisis (COMOTPA). Con este método, es posible combinar dos patrones de análisis representados mediante ontologías temporales que son obtenidas con la metodología MECOTPA y obtener una nueva ontología temporal que representa un patrón de análisis compuesto. Además, en este trabajo de investigación, se presenta también la construcción de la ontología temporal del Patrón de Análisis para Inventarios (PAI) y la combinación de ésta con la ontología temporal del Patrón de Análisis de Entidades Reutilizables (PAER), como caso de estudio para obtener un patrón de análisis compuesto, utilizando el método COMOTPA. Para verificar la validez y consistencia de la ontología temporal de patrón de análisis compuesto, se realiza una verificación de satisfacibilidad y de consistencia semántica temporal. Con el trabajo realizado en esta investigación, se pretende facilitar el reuso y disminuir las dificultades para adoptar los patrones de análisis en dominios diferentes.

Palabras clave:

Patrones de Análisis, Estructura estática, Comportamiento dinámico, Ontología, Ontología temporal, Lógica descriptiva, Lógica temporal, Lógicas de Descripción Temporal.

Abstract

Nowadays, the demand in the construction of software systems is increasing, due to this, it is important to consider reuse in the early stages of the software development process, to decrease the resources required and the total effort made. The analysis patterns are elements of a reuse technique used in the software engineering community, because they speed up the analysis process and improve the quality of software system development. Currently, with the Methodology for Temporal Ontologies of Analysis Patterns Construction (MECOTPA, Metodología para la Construcción de Ontologías Temporales de Patrones de Análisis), it is possible to construct temporal ontologies of analysis patterns that include structural and dynamic behavior aspects in their definition, using descriptive logic and temporal logic, respectively. This allows for a more complete description of the analysis patterns. However, at some point it may require the combination of two or more analysis patterns to represent larger problems that occur in the analysis phase of the software development process. For this reason, in this research work, the method for the Combination of Temporal Ontologies of Analysis Patterns (COMOTPA, Combinación de Ontologías Temporales de Patrones de Análisis) is presented. With this method, it is possible to combine two analysis patterns represented by temporal ontologies that are obtained with the MECOTPA methodology and obtain a new temporal ontology that represents a composite analysis pattern. In addition, in this research work, the construction of the temporal ontology of the Inventory Analysis Pattern (PAI, Patrón de Análisis para Inventarios) and the combination of this with the temporal ontology of the Reusable Entities Analysis Pattern (PAER, Patrón de Análisis de Entidades Reutilizables), are presented as a case study to obtain a composite analysis pattern, using the COMOTPA method. To verify the validity and consistency of the temporal ontology of the composite analysis pattern, a verification of satisfiability and temporal semantic consistency is performed. With the work done in this research, it is intended to facilitate reuse and reduce difficulties in adopting the analysis patterns in different domains.

Key words:

Analysis Patterns, Static Structure, Dynamic Behavior, Ontology, Temporal Ontology, Description Logic, Temporal Logic, Temporal Description Logic.

Contenido

Índice de figuras	IV
Índice de tablas	VII
Abreviaturas y siglas	XI
Capítulo I. Introducción	1
1.1. Introducción	1
1.2. Descripción del problema	2
1.3. Objetivos	3
1.3.1. Objetivo general	3
1.3.2. Objetivos específicos.....	3
1.4. Justificación y beneficios.....	3
1.4.1. Justificación.....	3
1.4.2. Beneficios.....	4
1.5. Propuesta de solución	4
1.6. Alcances y limitaciones	5
1.6.1. Alcances	5
1.6.2. Limitaciones	5
1.7. Organización del documento de tesis	5
Capítulo II. Marco teórico	7
.....	7
2.1. Patrón de análisis	8
2.2. Patrón de análisis estable	8
2.3. Plantilla de patrones de análisis estables	8
2.4. Ontología	9
2.5. Lógica de Descripción	9
2.6. Lógica Temporal.....	10
2.7. Lógicas de Descripción Temporal	11
2.8. Verificación de Consistencia Semántica Temporal	11
2.9. Verificación de Modelos.....	11
Capítulo III. Estado del arte	14
3.1. Antecedentes	15

3.2. Trabajos relacionados	20
Capítulo IV. Método para la Combinación de Ontologías Temporales de Patrones de Análisis.....	37
4.1. Requisitos para la implementación del método COMOTPA.....	39
4.1.1. Requisitos de entrada	39
4.1.2. Requisitos de salida.....	39
4.2. Fases del método COMOTPA	40
4.2.1. Combinación de términos del dominio (Conceptualización).....	43
4.2.2. Especificación de conceptos y relaciones (Formalización).....	49
4.2.3. Actualización de restricciones.....	52
4.2.4. Actualización de la KB con aspectos de temporalidad	55
4.2.5. Combinación de fórmulas ALC_{LTL}	59
4.2.6. Interpretación semántica temporal	63
4.2.7. Verificación de la formalización.....	70
4.2.8. Instanciación de la ontología.....	76
Capítulo V. Creación de la Ontología Temporal del Patrón de Análisis para Inventarios	78
5.1. Introducción	79
5.2. Metodología para la Construcción de Ontologías Temporales de Patrones de Análisis (MECOTPA).....	79
5.2.1. Refinamiento del patrón de análisis	80
5.2.2. Refinamiento de los requerimientos.....	82
5.2.3. Determinación del ámbito de la ontología (Especificación)	85
5.2.4. Búsqueda de ontologías relacionadas.....	87
5.2.5. Identificación de términos del dominio (Conceptualización)	87
5.2.6. Especificación de conceptos y relaciones (Formalización).....	88
5.2.7. Definición de restricciones.....	90
5.2.8. Verificación y validación de la base de conocimiento (Semántica).....	91
5.2.9. Formalización complementaria mediante aspectos de temporalidad.....	100
5.2.10. Instanciación de la ontología.....	118
Capítulo VI. Caso de estudio: Combinación de la OTPAE del PAI con la OTPAE del PAER	120
6.1. Combinación de la Ontología Temporal del PAI con la Ontología Temporal del PAER	121

6.1.1 Combinación de términos del dominio (Conceptualización).....	122
6.1.2. Actualización de la especificación de conceptos y relaciones (Formalización) ...	127
6.1.3. Actualización de restricciones.....	130
6.1.4. Actualización de la KB con aspectos de temporalidad	135
6.1.5. Combinación de fórmulas ALC_{LTL}	148
6.1.6. Interpretación semántica temporal	153
6.1.7. Verificación de la formalización.....	156
6.1.8. Instanciación de la ontología.....	161
Capítulo VII. Conclusiones.....	163
7.1. Conclusiones	164
7.2. Aportaciones	164
7.3. Trabajos futuros	165
Anexos.....	169
Anexo A. Ontología OPAE del patrón de análisis PAI	170
Anexo B. Ontología OTPAE del patrón de análisis PAI.....	185
Anexo C. Ontología OPAE del patrón de análisis PAIyER	242
Anexo D. Ontología OTPAE del patrón de análisis PAIyER.....	272
Anexo E. Documentación TSCC	403
Anexo F. Documentación SC	407
Anexo G. Contenido del disco.....	411

Índice de figuras

FIGURA 1. LENGUAJE DE PATRONES PARA LA RESERVACIÓN DE VUELOS.	23
FIGURA 2. A) ONTOLOGÍA DE EJEMPLO1 OT, B) ONTOLOGÍA DE EJEMPLO2 OS Y C) ONTOLOGÍA RESULTADO DE LA FUSIÓN DE OT Y OS.	28
FIGURA 3. FASES DEL MÉTODO COMOTPA.	40
FIGURA 4. ENTRADAS Y SALIDAS DEL MÉTODO COMOTPA.	41
FIGURA 5. ACTIVIDADES PRINCIPALES EN LAS FASES DEL MÉTODO COMOTPA.	42
FIGURA 6. EJEMPLO DE CLASIFICACIÓN DE ENTIDADES ESTÁTICAS Y DINÁMICAS DE UNA OTPAE.	43
FIGURA 7. RELACIÓN DE SIMILITUD ENTRE LAS CLASES ESTÁTICAS DE LA OTPAE DEL PAER Y LA OTPAE DEL PAI.	45
FIGURA 8. DEFINICIÓN DE RESTRICCIONES EXISTENCIALES EN LA OTPAE DEL PAIYER CON PROTÉGÉ.	53
FIGURA 9. DEFINICIÓN DE RESTRICCIONES UNIVERSALES EN LA OTPAE DEL PAIYER CON PROTÉGÉ.	54
FIGURA 10. VENTANA PARA COMBINAR LA ONTOLOGÍA OPAE CON LA ONTOLOGÍA OWL-TIME.	56
FIGURA 11. TRIPLES AFECTADOS EN LA CONVERSIÓN DE UNA RELACIÓN BINARIA A TEMPORAL.	57
FIGURA 12. RELACIÓN BINARIA A PARTIR DE LA PROPOSICIÓN "EL CLIENTE RESERVA UNA ENTIDAD".	57
FIGURA 13. RELACIÓN TEMPORAL DE LA PROPOSICIÓN "EL CLIENTE RESERVA UNA ENTIDAD A TRAVÉS DE UN EVENTO".	58
FIGURA 14. ESTABLECER RELACIÓN.	66
FIGURA 15. SELECCIÓN DEL INTERVALO EN EL QUE SE REALIZARÁ LA RELACIÓN.	66
FIGURA 16. RESUMEN DE LA ASERCIÓN DE LA PROPIEDAD DE OBJETO.	67
FIGURA 17. ASERCIÓN DE LA PROPIEDAD DE OBJETO REFLEJADA EN LA INTERFAZ DE CHRONOS.	67
FIGURA 18. AUTÓMATA PARA LA TRAYECTORIA DE ESTADOS π' , DE ÉXITO.	71
FIGURA 19. AUTÓMATA PARA LA TRAYECTORIA DE ESTADOS π'' , DE FRACASO.	72
FIGURA 20. FASES DE LA METODOLOGÍA MECOTPA.	80
FIGURA 21. VALORES DE PROPIEDADES PARA LA CLASE "PROBLEMA" CORRESPONDIENTE AL PUNTO 5 DE LA PLANTILLA PPAE.	81
FIGURA 22. DIAGRAMA GRL DEL RF. SEPARAR DIFERENTES TIPOS DE EXISTENCIAS.	84
FIGURA 23. DIAGRAMA UCM DEL RF. SEPARAR DIFERENTES TIPOS DE EXISTENCIAS.	84
FIGURA 24 . REPRESENTACIÓN GRÁFICA DE LOS TA DEL PAI.	93
FIGURA 25 . TBOX PARA EL DOMINIO DE ESTUDIO.	94
FIGURA 26 . MODELADO DE DATOS DEL PAI QUE INCLUYE LOS AXIOMAS TA Y CA.	95
FIGURA 27. DOMINIO Y FUNCIÓN DE INTERPRETACIÓN SEMÁNTICA DEL PATRÓN DE ANÁLISIS PAI.	99
FIGURA 28. ESTRUCTURA DEL MODELO DE DOMINIO DEL PATRÓN DE ANÁLISIS PAI.	99
FIGURA 29. REPRESENTACIÓN GRÁFICA DE LA TRAYECTORIA DE ESTADOS π' DE CASOS DE ÉXITO DEL RF REALIZAR EL SEGUIMIENTO DE CANTIDADES DE ARTÍCULOS EN EXISTENCIAS.	112
FIGURA 30 . REPRESENTACIÓN GRÁFICA DE LA TRAYECTORIA DE ESTADOS π'' DE CASOS DE ÉXITO Y FRACASO DEL RF REALIZAR EL SEGUIMIENTO DE CANTIDADES DE ARTÍCULOS EN EXISTENCIAS.	112
FIGURA 31. AUTÓMATA PARA LA TRAYECTORIA DE ESTADOS π'	113
FIGURA 32 . AUTÓMATA PARA LA TRAYECTORIA DE ESTADOS π''	115
FIGURA 33. RELACIÓN DE SIMILITUD ENTRE LAS CLASES ESTÁTICAS DE LA OTPAE DEL PAER Y LA OTPAE DEL PAI.	123
FIGURA 34. DEFINICIÓN DE RESTRICCIONES EXISTENCIALES EN PROTÉGÉ.	131
FIGURA 35. DEFINICIÓN DE RESTRICCIONES UNIVERSALES EN PROTÉGÉ.	132
FIGURA 36. ONTOLOGÍA OPAE DEL PATRÓN DE ANÁLISIS PAIYER, CON CLASES, POS Y PDS.	133
FIGURA 37. RELACIÓN ENTRE CLASES DE LA OPAE DEL PAIYER.	134
FIGURA 38. PROPIEDADES DE LAS CLASES DE LA OPAE DEL PAIYER.	134
FIGURA 39. ONTOLOGÍA OTPAE DEL PAIYER CON ELEMENTOS SEPARADOS (ESTÁTICOS Y TEMPORALES).	135
FIGURA 40. VERIFICACIÓN DE LA CONSISTENCIA DE LA OTPAE DEL PAIYER UTILIZANDO EL RAZONADOR PELLET.	148

FIGURA 41. AUTÓMATA PARA LA TRAYECTORIA DE ESTADOS π' DEL RF SEPARAR DIFERENTES TIPOS DE EXISTENCIAS.....	157
FIGURA 42 . AUTÓMATA PARA LA TRAYECTORIA DE ESTADOS π'' DEL RF SEPARARA DIFERENTES TIPOS DE EXISTENCIAS.....	158
FIGURA A.1 JERARQUÍA DE CLASES EN LA KB DEL PATRÓN DE ANÁLISIS PAI.....	180
FIGURA B.1 JERARQUÍA DE CLASES ESTÁTICAS Y TEMPORALES EN LA KB DEL PATRÓN DE ANÁLISIS PAI.....	198
FIGURA B.2 SOIS EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF SEPARAR DIFERENTES TIPOS DE EXISTENCIAS.....	205
FIGURA B.3 CASO DE ÉXITO 1 EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF SEPARAR DIFERENTES TIPOS DE EXISTENCIAS.....	206
FIGURA B.4 SOIS EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF SEPARAR DIFERENTES TIPOS DE EXISTENCIAS.....	212
FIGURA B.5 CASO DE FRACASO 1 EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF SEPARAR DIFERENTES TIPOS DE EXISTENCIAS.....	212
FIGURA B.6 SOIS EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF REALIZAR SEGUIMIENTO DE CANTIDADES DE ARTÍCULOS EN EXISTENCIAS.....	220
FIGURA B.7 CASO DE ÉXITO 1 EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF REALIZAR SEGUIMIENTO DE LAS CANTIDADES DE ARTÍCULOS EN EXISTENCIAS.....	220
FIGURA B.8 SOIS EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF REALIZAR SEGUIMIENTO DE CANTIDADES DE ARTÍCULOS EN EXISTENCIAS.....	227
FIGURA B.9 CASO DE FRACASO 1 EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF REALIZAR ELSEGUIMIENTO DE LAS CANTIDADES DE ARTÍCULOS EN EXISTENCIAS.....	227
FIGURA B.10 SOIS EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF REALIZAR SEGUIMIENTO DE LAS UBICACIONES DE LOS ARTÍCULOS.....	234
FIGURA B.11 CASO DE ÉXITO 1 EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF REALIZAR SEGUIMIENTO DE LAS UBICACIONES DE LOS ARTÍCULOS.....	234
FIGURA B.12 SOIS EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF REALIZAR SEGUIMIENTO DE LAS UBICACIONES DE LOS ARTÍCULOS.....	241
FIGURA B.13 CASO DE FRACASO 1 EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF REALIZAR SEGUIMIENTO DE LAS UBICACIONES DE LOS ARTÍCULOS.....	241
FIGURA C.1 JERARQUÍA DE CLASES EN LA KB DEL PATRÓN DE ANÁLISIS PAIYER.....	263
FIGURA D.1 JERARQUÍA DE CLASES ESTÁTICAS Y TEMPORALES EN LA KB DEL PATRÓN DE ANÁLISIS PAIYER.....	294
FIGURA D.2 SOIS EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF1. HACER UNA RESERVACIÓN.....	302
FIGURA D.3 CASO DE ÉXITO 1 EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF1. HACER UNA RESERVACIÓN.....	303
FIGURA D.4 SOIS EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF1. HACER UNA RESERVACIÓN.....	310
FIGURA D.5 CASO DE FRACASO 1 EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF1. HACER UNA RESERVACIÓN.....	311
FIGURA D.6 SOIS EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF2. USAR UNA ENTIDAD RESERVADA.....	315
FIGURA D.7 CASO DE ÉXITO 1 EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF2. USAR UNA ENTIDAD RESERVADA.....	316
FIGURA D.8 SOIS EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF2. USAR UNA ENTIDAD RESERVADA.....	320
FIGURA D.9 CASO DE FRACASO 1 EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF2. USAR UNA ENTIDAD RESERVADA.....	321
FIGURA D.10 SOIS EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF3. MODIFICAR UNA RESERVACIÓN.....	328
FIGURA D.11 CASO DE ÉXITO 1 EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF3. MODIFICAR UNA RESERVACIÓN.....	329
FIGURA D.12 SOIS EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF3. MODIFICAR UNA RESERVACIÓN.....	336

FIGURA D.13 CASO DE FRACASO 1 EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF3. MODIFICAR UNA RESERVACIÓN.	337
FIGURA D.14 SOIS EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF4. CANCELAR UNA RESERVACIÓN.	344
FIGURA D.15 CASO DE ÉXITO 1 EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF4. CANCELAR UNA RESERVACIÓN.	344
FIGURA D.16 SOIS EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF4. CANCELAR UNA RESERVACIÓN.	351
FIGURA D.17 CASO DE FRACASO 1 EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF4. CANCELAR UNA RESERVACIÓN.	352
FIGURA D.18 SOIS EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF5. CREAR UN CONTRATO.	358
FIGURA D.19 CASO DE ÉXITO 1 EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF5. CREAR UN CONTRATO.	358
FIGURA D.20 SOIS EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF5. CREAR UN CONTRATO.	364
FIGURA D.21 CASO DE FRACASO 1 EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF5. CREAR UN CONTRATO.	364
FIGURA D.22 SOIS EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF6. SEPARAR DIFERENTES TIPOS DE EXISTENCIAS.	370
FIGURA D.23 CASO DE ÉXITO 1 EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF6. SEPARAR DIFERENTES TIPOS DE EXISTENCIAS.	371
FIGURA D.24 SOIS EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF6. SEPARAR DIFERENTES TIPOS DE EXISTENCIAS.	376
FIGURA D.25 CASO DE FRACASO 1 EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF6. SEPARAR DIFERENTES TIPOS DE EXISTENCIAS.	377
FIGURA D.26 SOIS EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF7. REALIZAR SEGUIMIENTO DE CANTIDADES DE ENTIDADES EN EXISTENCIAS.	383
FIGURA D.27 CASO DE ÉXITO 1 EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF7. REALIZAR SEGUIMIENTO DE LAS CANTIDADES DE ENTIDADES EN EXISTENCIAS.	384
FIGURA D.28 SOIS EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF7. REALIZAR SEGUIMIENTO DE CANTIDADES DE ENTIDADES EN EXISTENCIAS.	390
FIGURA D.29 CASO DE FRACASO 1 EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF7. REALIZAR EL SEGUIMIENTO DE LAS CANTIDADES DE ENTIDADES EN EXISTENCIAS.	390
FIGURA D.30 SOIS EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF8. REALIZAR SEGUIMIENTO DE LAS UBICACIONES DE LAS ENTIDADES.	396
FIGURA D.31 CASO DE ÉXITO 1 EN LA ABOX DE CASOS DE ÉXITO DE ACUERDO AL RF8. REALIZAR SEGUIMIENTO DE LAS UBICACIONES DE LAS ENTIDADES.	396
FIGURA D.32 SOIS EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF8. REALIZAR SEGUIMIENTO DE LAS UBICACIONES DE LAS ENTIDADES.	402
FIGURA D.33 CASO DE FRACASO 1 EN LA ABOX DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF8. REALIZAR SEGUIMIENTO DE LAS UBICACIONES DE LAS ENTIDADES.	402

Índice de tablas

TABLA 1. COMPARATIVA DE TRABAJOS ANTECEDENTES.....	19
TABLA 2. TABLA COMPARATIVA DE TRABAJOS RELACIONADOS.....	34
TABLA 3. EVOLUCIÓN DEL TRABAJO DE INVESTIGACIÓN ACTUAL.....	38
TABLA 4. CONJUNTO DE CLASES ESTÁTICAS DE LAS OTPAE DE DE ENTRADA Y SALIDA.....	46
TABLA 5. RELACIÓN ENTRE PROPIEDADES DE OBJETOS DE LA OTPAE DEL PAER CON LA OTPAE DEL PAI.....	47
TABLA 6. RELACIÓN ENTRE PROPIEDADES DE DATOS DE LA OTPAE DEL PAER CON LA OTPAE DEL PAI.....	49
TABLA 7. ESPECIFICACIÓN DE LA JERARQUÍA DE CLASES PARA LA OTPAE DEL PAIYER.....	50
TABLA 8. FRAGMENTO DE LA ESPECIFICACIÓN DE RELACIONES ENTRE CLASES EN LA OTPAE DEL PAIYER.....	50
TABLA 9. FRAGMENTO DE LA ESPECIFICACIÓN DE LAS PROPIEDADES DE LAS CLASES DE LA OTPAE DEL PAIYER.....	51
TABLA 10. RESTRICCIONES EXISTENCIALES EN LA OTPAE DEL PAIYER.....	52
TABLA 11. RESTRICCIONES UNIVERSALES EN LA OTPAE DEL PAIYER.....	53
TABLA 12. REGLAS SWRL PARA CARACTERÍSTICAS FUNCIONAL E INVERSA FUNCIONAL.....	58
TABLA 13. FÓRMULA $ALC_{LTL}(\varphi)$ DEL RF. HACER UNA RESERVACIÓN DEL PAER.....	61
TABLA 14. FÓRMULA $ALC_{LTL}(\varphi)$ DEL RF. HACER UNA RESERVACIÓN DEL PAIYER.....	62
TABLA 15. REDUCCIÓN DE LA FÓRMULA $ALC_{LTL}(\varphi)$ DEL RF. HACER UNA RESERVACIÓN A UNA FÓRMULA LTL (φ').....	62
TABLA 16. NOMENCLATURA PROPUESTA PARA LA CLASE EVENT.....	64
TABLA 17. NOMENCLATURA PROPUESTA PARA LA CLASE DATETIMEINTERVAL.....	64
TABLA 18. NOMENCLATURA PROPUESTA PARA LA CLASE INSTANT PARA REPRESENTAR EL INSTANTE DE INICIO.....	64
TABLA 19. NOMENCLATURA PROPUESTA PARA LA CLASE INSTANT PARA REPRESENTAR EL INSTANTE DE FIN.....	64
TABLA 20. EJEMPLO DE SECUENCIAS DE INTERPRETACIONES PARA CASOS DE ÉXITO EN EL REQUERIMIENTO FUNCIONAL RF. HACER UNA RESERVACIÓN.....	64
TABLA 21. VALORES DE VERDAD DE LAS PROPOSICIONES DE ACUERDO A LA SECUENCIA DE INTERPRETACIONES PARA EL RF. HACER UNA RESERVACIÓN.....	69
TABLA 22. REPRESENTACIÓN GRÁFICA DE LA TRAYECTORIA DE ESTADOS EN LA SECUENCIA DE INTERPRETACIONES PARA EL RF. HACER UNA RESERVACIÓN.....	70
TABLA 23. CÓDIGO FUENTE DEL PROGRAMA REQ1-OTPAEPAIYER-E.SMV.....	71
TABLA 24. CÓDIGO FUENTE DEL PROGRAMA REQ1-OTPAEPAIYER-EF.SMV.....	73
TABLA 25. COMANDOS PARA LA EJECUCIÓN DEL AUTÓMATA.....	76
TABLA 26. COMANDO PARA REALIZAR LA VERIFICACIÓN SC DE UNA FÓRMULA LTL.....	76
TABLA 27. EJEMPLO DE ASERCIONES DE CLASES Y DE PROPIEDADES DE OBJETOS EN EL RF HACER UNA RESERVACIÓN DEL PAIYER.....	77
TABLA 28. FRAGMENTO DE LA TABLA DE RAZONAMIENTO TEMPORAL EN EL PATRÓN DE ANÁLISIS PAI (SECCIÓN PPAE).....	81
TABLA 29. FÓRMULAS LTL PROPUESTAS EN LOS RFS 1 Y 2 DEL PAI.....	82
TABLA 30. FRAGMENTO DE LA TABLA DE RAZONAMIENTO TEMPORAL DEL PATRÓN DE ANÁLISIS PAI (SECCIÓN MECOTPA).....	83
TABLA 31. FÓRMULAS LTL PROPUESTAS PARA LAS ESPECIFICACIONES DE LOS DIAGRAMAS GRL DE LOS RFS 2 Y 3 DEL PAI.....	85
TABLA 32. FÓRMULAS LTL PROPUESTAS PARA LAS ESPECIFICACIONES DE LOS DIAGRAMAS UCM DE LOS RFS 2 Y 3 DEL PAI.....	85
TABLA 33. CONCEPTOS Y ROLES CONCRETOS EN EL PAI.....	87
TABLA 34. ROLES ABSTRACTOS SELECCIONADOS POR RF DEL PATRÓN DE ANÁLISIS PAI.....	88
TABLA 35. RELACIONES ENTRE CONCEPTOS DEL PATRÓN DE ANÁLISIS PAI.....	89
TABLA 36. PROPIEDADES DE LOS CONCEPTOS DEL PATRÓN DE ANÁLISIS PAI.....	89
TABLA 37. AXIOMAS EN EL COMPONENTE INTENCIONAL DEL PATRÓN DE ANÁLISIS PAI.....	93
TABLA 38. AXIOMAS EN EL COMPONENTE EXTENSIONAL DEL PATRÓN DE ANÁLISIS PAI.....	94

TABLA 39 . REQUERIMIENTOS FUNCIONALES (RFs) DEL PATRÓN DE ANÁLISIS PAI	100
TABLA 40 . ESPECIFICACIONES EC DE LOS RFs DEL PATRÓN DE ANÁLISIS PAI.....	101
TABLA 41 . CLARIFICACIÓN DE ORACIONES DE LA EC DE LOS RFs DEL PATRÓN DE ANÁLISIS PAI.	102
TABLA 42 . ESPECIFICACIONES EC DE LOS RFs DEL PATRÓN DE ANÁLISIS PAI A PARTIR DE LA UNIÓN DE ORACIONES CLARIFICADAS.	103
TABLA 43 . ESPECIFICACIONES ECC DE LOS RFs DEL PATRÓN DE ANÁLISIS PAI.	103
TABLA 44 . ESPECIFICACIONES ECA DE LOS RFs DEL PATRÓN DE ANÁLISIS PAI.....	104
TABLA 45 . ESPECIFICACIONES ECAP DE LOS RFs DEL PATRÓN DE ANÁLISIS PAI.	105
TABLA 46 . PAS, Cs Y ARs EN LA ESPECIFICACIÓN ECAP DE LOS RFs DEL PATRÓN DE ANÁLISIS PAI.	106
TABLA 47 . CIs DE ROLES ABSTRACTOS Y SU VERSIÓN TEMPORAL EN LOS RFs DEL PATRÓN DE ANÁLISIS PAI.....	107
TABLA 48 . ESPECIFICACIONES FORMALES A PARTIR DE LAS ESPECIFICACIONES ECAP DE LOS RFs DEL PATRÓN DE ANÁLISIS PAI.	109
TABLA 49 . REDUCCIÓN DE ESPECIFICACIONES FORMALES A FÓRMULAS DE LA LTL.	110
TABLA 50. RESULTADOS DE LA TSCC DE LA OTPAE DEL PAI.	111
TABLA 51 . CÓDIGO FUENTE DEL PROGRAMA REQ1-OTPAEPAI-E.SMV.....	114
TABLA 52 . CÓDIGO FUENTE DEL PROGRAMA REQ1-OTPAEPAI-EF.SMV.....	115
TABLA 53 . RESULTADOS DE LA VERIFICACIÓN DE SATISFACIBILIDAD DE LAS FÓRMULAS LTL.	117
TABLA 54. EJEMPLO DE ASERCIONES DE CLASES Y DE PROPIEDADES DE OBJETOS EN EL RF SEPARAR DIFERENTES TIPOS DE EXISTENCIAS.	118
TABLA 55. CONJUNTO DE CLASES ESTÁTICAS CORRESPONDIENTES A LAS OTPAES DE ESTUDIO.	123
TABLA 56. RELACIÓN DE SIMILITUD ENTRE PROPIEDADES DE OBJETOS DE LA OTPAE DEL PAER CON LA OTPAE DEL PAI.....	124
TABLA 57. RELACIÓN DE SIMILITUD ENTRE PROPIEDADES DE DATOS DE LA OTPAE DEL PAER CON LA OTPAE DEL PAI.....	126
TABLA 58. ESPECIFICACIÓN DE LA JERARQUÍA DE CLASES PARA EL PAIYER.	127
TABLA 59. RELACIONES ENTRE CLASES DEL PATRÓN DE ANÁLISIS PAIYER.	128
TABLA 60. PROPIEDADES DE LAS CLASES DEL PATRÓN DE ANÁLISIS PAIYER.	129
TABLA 61. RESTRICCIONES EXISTENCIALES DE LA OPAE DEL PAIYER.	130
TABLA 62. RESTRICCIONES UNIVERSALES DE LA OPAE DEL PAIYER.	131
TABLA 63. CONVERSIÓN DE PROPIEDADES ESTÁTICAS A TEMPORALES EN LA OTPAE DEL PAIYER.	136
TABLA 64. REGLAS SWRL EN CARACTERÍSTICAS FUNCIONAL E INVERSA FUNCIONAL DE PROPIEDADES DE OBJETOS EN LA OTPAE DEL PAIYER.....	143
TABLA 65. ELEMENTOS DE ENTRADA PARA REALIZAR LA COMBINACIÓN DE FÓRMULAS ALC_{LTL}	148
TABLA 66. CONJUNTO DE FÓRMULAS ALC_{LTL} DE LAS ESPECIFICACIONES ECAP DEL PATRÓN DE ANÁLISIS PAIYER.	150
TABLA 67. REDUCCIÓN DE LAS ESPECIFICACIONES FORMALES DE LA OTPAE COMPUESTA A FÓRMULAS DE LA LTL.	152
TABLA 68. RESULTADOS POR REQUERIMIENTO FUNCIONAL EN LA VERIFICACIÓN TSCC DE LA OTPAE DEL PAIYER.	154
TABLA 69. REPRESENTACIÓN GRÁFICA DE LAS TRAYECTORIAS DE ESTADOS DEL RF USAR UNA ENTIDAD RESERVADA.	155
TABLA 70 . CÓDIGO FUENTE DEL PROGRAMA REQ6-OTPAEPAIYER-E.SMV	157
TABLA 71 . CÓDIGO FUENTE DEL PROGRAMA REQ6-OTPAEPAIYER-EF.SMV.....	158
TABLA 72 . RESULTADOS DE LA VERIFICACIÓN DE SATISFACIBILIDAD DE LAS FÓRMULAS LTL.	160
TABLA 73. EJEMPLO DE ASERCIONES DE CLASES Y DE PROPIEDADES DE OBJETOS EN EL RF HACER UNA RESERVACIÓN DEL PAIYER.	161
TABLA A.1 TBOX DE LA KB DEL PATRÓN DE ANÁLISIS PAI (CONT.)	172
TABLA A.2 ABOX DE LA KB DEL PATRÓN DE ANÁLISIS PAI (CONT.).....	181
TABLA B.3 REGLAS SWRL EN CARACTERÍSTICAS FUNCIONAL E INVERSA FUNCIONAL DE PROPIEDADES DE OBJETO EN EL PATRÓN DE ANÁLISIS PAI (CONT.)	188
TABLA B.4 TBOX DE LA KB DEL PATRÓN DE ANÁLISIS PAI CON ASPECTOS TEMPORALES (CONT.).....	190
TABLA B.5 FÓRMULA ALC_{LTL} DEL RF SEPARAR DIFERENTES TIPOS DE EXISTENCIAS.	199

TABLA B.6 ABOX DE CASOS DE ÉXITO EN EL RF SEPARAR DIFERENTES TIPOS DE EXISTENCIAS (CONT.)	199
TABLA B.7 ABOX DE CASOS DE ÉXITO Y FRACASO EN EL RF SEPARAR DIFERENTES TIPOS DE EXISTENCIAS (CONT.)	206
TABLA B.8 REGLAS SWRL PARA LA SOIS DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF SEPARAR DIFERENTES TIPOS DE EXISTENCIAS. (CONT.)	212
TABLA B.9 FÓRMULA ALC _{LT} DEL RF REALIZAR SEGUIMIENTO DE CANTIDADES DE ARTICULOS EN EXISTENCIAS.	213
TABLA B.10 ABOX DE CASOS DE ÉXITO EN EL RF REALIZAR SEGUIMIENTO DE CANTIDADES DE ARTICULOS EN EXISTENCIAS. (CONT.).....	213
TABLA B.11 ABOX DE CASOS DE ÉXITO Y FRACASO EN EL RF REALIZAR SEGUIMIENTO DE CANTIDADES DE ARTÍCULOS EN EXISTENCIAS (CONT.)	221
TABLA B.12 REGLAS SWRL PARA LA SOIS DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF REALIZAR SEGUIMIENTO DE CANTIDADES DE ARTÍCULOS EN EXISTENCIAS (CONT.)	226
TABLA B.13 FÓRMULA ALC _{LT} DEL RF REALIZAR SEGUIMIENTO DE LAS UBICACIONES DE LOS ARTICULOS.....	228
TABLA B.14 ABOX DE CASOS DE ÉXITO EN EL RF REALIZAR SEGUIMIENTO DE LAS UBICACIONES DE LOS ARTÍCULOS. (CONT.)	228
TABLA B.15 ABOX DE CASOS DE ÉXITO Y FRACASO EN EL RF REALIZAR SEGUIMIENTO DE CANTIDADES DE ARTÍCULOS EN EXISTENCIAS (CONT.).....	235
TABLA B.16 REGLAS SWRL PARA LA SOIS DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF REALIZAR SEGUIMIENTO DE LAS UBICACIONES DE LOS ARTÍCULOS (CONT.).....	240
TABLA C.1 TBOX DE LA KB DEL PATRÓN DE ANÁLISIS PAIYER(CONT.).....	244
TABLA C.2 ABOX DE LA KB DEL PATRÓN DE ANÁLISIS PAIYER (CONT.).....	264
TABLA D.3 REGLAS SWRL EN CARACTERÍSTICAS FUNCIONAL E INVERSA FUNCIONAL DE PROPIEDADES DE OBJETO EN EL PATRÓN DE ANÁLISIS PAIYER (CONT.).....	275
TABLA D.4 TBOX DE LA KB DEL PATRÓN DE ANÁLISIS PAIYER CON ASPECTOS TEMPORALES (CONT.)	280
TABLA D.5 FÓRMULA ALC _{LT} DEL RF1. HACER UNA RESERVACIÓN.	295
TABLA D.6 ABOX DE CASOS DE ÉXITO EN EL RF1. HACER UNA RESERVACIÓN (CONT.).....	296
TABLA D.7 ABOX DE CASOS DE ÉXITO Y FRACASO EN EL RF1. HACER UNA RESERVACIÓN (CONT.)	303
TABLA D.8 REGLAS SWRL PARA LA SOIS DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF HACER UNA RESERVACIÓN. (CONT.).....	310
TABLA D.9 FÓRMULA ALC _{LT} DEL RF2. USAR UNA ENTIDAD RESERVADA.	311
TABLA D.10 ABOX DE CASOS DE ÉXITO EN EL RF2. USAR UNA ENTIDAD RESERVADA (CONT.)	312
TABLA D.11 ABOX DE CASOS DE ÉXITO Y FRACASO EN EL RF2. USAR UNA ENTIDAD RESERVADA (CONT.).....	316
TABLA D.12 REGLAS SWRL PARA LA SOIS DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF2. USAR UNA ENTIDAD RESERVADA (CONT.).....	320
TABLA D.13 FÓRMULA ALC _{LT} DEL RF3. MODIFICAR UNA RESERVACIÓN.	321
TABLA D.14 ABOX DE CASOS DE ÉXITO EN EL RF3. MODIFICAR UNA RESERVACIÓN (CONT.)	322
TABLA D.15 ABOX DE CASOS DE ÉXITO Y FRACASO EN EL RF3. MODIFICAR UNA RESERVACIÓN (CONT.).....	329
TABLA D.16 REGLAS SWRL PARA LA SOIS DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF3. MODIFICAR UNA RESERVACIÓN. (CONT.).....	336
TABLA D.17 FÓRMULA ALC _{LT} DEL RF4. CANCELAR UNA RESERVACIÓN.	337
TABLA D.18 ABOX DE CASOS DE ÉXITO EN EL RF4. CANCELAR UNA RESERVACIÓN (CONT.)	337
TABLA D.19 ABOX DE CASOS DE ÉXITO Y FRACASO EN EL RF4. CANCELAR UNA RESERVACIÓN (CONT.).....	345
TABLA D.20 REGLAS SWRL PARA LA SOIS DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF4. CANCELAR UNA RESERVACIÓN (CONT.).....	351
TABLA D.21 FÓRMULA ALC _{LT} DEL RF5. CREAR UN CONTRATO.	352
TABLA D.22 ABOX DE CASOS DE ÉXITO EN EL RF5. CREAR UN CONTRATO. (CONT.).....	353
TABLA D.23 ABOX DE CASOS DE ÉXITO Y FRACASO EN EL RF5. CREAR UN CONTRATO. (CONT.).....	359
TABLA D.24 REGLAS SWRL PARA LA SOIS DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF5. CREAR UN CONTRATO (CONT.).....	364
TABLA D.25 FÓRMULA ALC _{LT} DEL RF6. SEPARAR DIFERENTES TIPOS DE EXISTENCIAS.	365
TABLA D.26 ABOX DE CASOS DE ÉXITO EN EL RF6. SEPARAR DIFERENTES TIPOS DE EXISTENCIAS (CONT.)	365

TABLA D.27 ABOX DE CASOS DE ÉXITO Y FRACASO EN EL RF6. SEPARAR DIFERENTES TIPOS DE EXISTENCIAS (CONT.).....	371
TABLA D.28 REGLAS SWRL PARA LA SOIS DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF6. SEPARAR DIFERENTES TIPOS DE EXISTENCIAS. (CONT.).....	376
TABLA D.29 FÓRMULA ALC _{LTL} DEL RF7. REALIZAR SEGUIMIENTO DE CANTIDADES DE ENTIDADES EN EXISTENCIAS.....	377
TABLA D.30 ABOX DE CASOS DE ÉXITO EN EL RF7. REALIZAR SEGUIMIENTO DE CANTIDADES DE ENTIDADES EN EXISTENCIAS. (CONT.).....	378
TABLA D.31 ABOX DE CASOS DE ÉXITO Y FRACASO EN EL RF7. REALIZAR SEGUIMIENTO DE CANTIDADES DE ENTIDADES EN EXISTENCIAS (CONT.).....	384
TABLA D.32 REGLAS SWRL PARA LA SOIS DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF REALIZAR SEGUIMIENTO DE CANTIDADES DE ARTÍCULOS EN EXISTENCIAS (CONT.)	389
TABLA D.33 FÓRMULA ALC _{LTL} DEL RF8. REALIZAR SEGUIMIENTO DE LAS UBICACIONES DE LAS ENTIDADES. ...	390
TABLA D.34 ABOX DE CASOS DE ÉXITO EN EL RF8. REALIZAR SEGUIMIENTO DE LAS UBICACIONES DE LAS ENTIDADES. (CONT.)	391
TABLA D.35 ABOX DE CASOS DE ÉXITO Y FRACASO EN EL RF8. REALIZAR SEGUIMIENTO DE LAS UBICACIONES DE LAS ENTIDADES. (CONT.)	398
TABLA D.36 REGLAS SWRL PARA LA SOIS DE CASOS DE ÉXITO Y FRACASO DE ACUERDO AL RF REALIZAR SEGUIMIENTO DE LAS UBICACIONES DE LAS ENTIDADES. (CONT.).....	402
TABLA G.1 DOCUMENTACIÓN GENERADA (CONT.).....	413
TABLA G.2 VERSIONES FINALES DE LAS ONTOLOGÍAS DEL PATRÓN DE ANÁLISIS PAI (CONT.)	415
TABLA G.3 VERSIONES FINALES DE LAS ONTOLOGÍAS DEL PATRÓN DE ANÁLISIS PAIYER (CONT.).....	415

Abreviaturas y siglas

A(In)S	Símbolos Independientes de la Aplicación por el inglés <i>Application Independent Symbols</i> .
ABox	Componente Asercional por el inglés <i>Assertional Box</i> .
AC	Conceptos Atómicos por el inglés <i>Atomic Concepts</i> .
AI	Individuos Abstractos por el inglés <i>Abstract Individuals</i> .
ALC	Lenguaje de Atributos con Complementos por el inglés <i>Attribute Language with Complements</i> .
ALC _{LT}	Un tipo de TDL.
APRI	Infraestructura para el Reúso de Patrones de Análisis por el inglés <i>Analysis Patterns Reuse Infrastructure</i> .
AR	Roles Abstractos por el inglés <i>Abstract Roles</i> .
AS	Símbolos Dependientes de Aplicación por el inglés <i>Application Dependent Symbols</i> .
ATL	Lenguaje de transformación Atlas por el inglés <i>Atlas Transformation Language</i> .
AUPAS	Almacenamiento y Uso de Patrones de Análisis apegados a su Semántica
BOs	Objetos de Negocio por el inglés <i>Business Objects</i> .
CI	Individuos Concretos por el inglés <i>Concrete Individuals</i> .
CI _s	Conceptos de Inclusión por el inglés <i>Concept Inclusions</i> .
CR	Roles concretos por el inglés <i>Concrete Roles</i> .
CTL	Lógica Computacional Ramificada por el inglés <i>Computation Tree Logic</i> .
DLs	Lógica de Descripción por el inglés <i>Description Logic</i> y DLs para el plural: Lógicas de Descripción.
EBTs	Temas Perdurables de Negocio por el inglés <i>Enduring Business Themes</i>
EC	Especificación de Comportamiento.
ECA	Especificación de Comportamiento Adaptada.
ECAP	Especificación de Comportamiento Adaptada con orden de Precedencia.
ECC	Especificación de Comportamiento Clarificada.
FOCOPAT	Formalización Complementaria de Patrones de Análisis con Aspectos de Temporalidad.
GRL	Lenguaje de Requerimientos Orientado a Metas por el inglés <i>Goal-Oriented Requirement Language</i> .
IOs	Objetos Industriales por el inglés <i>Industrial Objects</i>
KB	Base de Conocimiento por el inglés <i>Knowledge Base</i> .
LTL	Lógica Temporal Lineal por el inglés <i>Linear Temporal Logic</i> .
MC	Verificación de modelos por inglés <i>Model Checking</i> .
MECOPA	Metodología para la Construcción de Ontologías de Patrones de Análisis.
MECOTPA	Metodología para la Construcción de Ontologías Temporales de Patrones de Análisis.
MDA	Arquitectura Dirigida por Modelos por el inglés <i>Model-Driven Architecture</i> .

OPAE	Ontología del Patrón de Análisis Estable y OPAEs para el plural: Ontologías de Patrones de Análisis Estables.
OPPAE	Ontología de la Plantilla de Patrones de Análisis Estable.
OTPAE	Ontología Temporal de Patrón de Análisis Estable y OTPAEs para el plural: Ontologías Temporales de Patrones de Análisis Estables.
OWL	Lenguaje de Ontologías Web por el inglés <i>Ontology Web Language</i> .
PA	Patrón de Análisis y PAs para el plural: Patrones de Análisis.
PAC	Patrón de Análisis de Cuentas
PAE	Patrón de Análisis Estable
PAER	Patrón de Análisis de Entidades Reutilizables
PAI	Patrón de Análisis para Inventarios
PAIyER	Patrón de Análisis para Inventarios y Entidades Reutilizables
PPAE	Plantilla del Patrones de Análisis Estable
RDF	Marco de Descripción de Recursos por el inglés <i>Resource Description Framework</i> .
RF	Requerimiento Funcional.
SaaS	Software como un Servicio por el inglés <i>Software as a Service</i> .
SAP	Patrón de Análisis Semántico por el inglés <i>Semantic Analysis Pattern</i> .
SC	Verificación de Satisfacibilidad por el inglés <i>Satisfiability Checking</i> .
SMV	Verificador de Modelos Simbólico por el inglés <i>Symbolic Model Verifier</i> .
SoIs	Secuencias de Interpretaciones por el inglés <i>Sequences of Interpretations</i> .
SOWL	Lenguaje de Ontologías Web Espacio-Temporal por el inglés <i>Spatio-Temporal OWL</i> .
SSM	Modelo de Estabilidad del Software por el inglés <i>Software Stability Model</i> .
SUMO	Ontología combinada superior sugerida por el inglés <i>Suggested Upper Merged Ontology</i> .
SWRL	Lenguaje de Reglas para la Web Semántica por el inglés <i>Semantic Web Rule Language</i> .
TA	Axiomas terminológicos por el inglés <i>Terminological Axioms</i> .
TBox	Componente terminológico por el inglés <i>Terminological Box</i> .
TDL	Lógica de Descripción Temporal por el inglés <i>Temporal Description Logic</i> , y TDLs para el plural: Lógicas de Descripción Temporal.
TL	Lógica Temporal por el inglés <i>Temporal Logic</i> .
TSCC	Verificación de Consistencia Semántica Temporal por el inglés <i>Temporal Semantic Consistency Checking</i> .
UCM	Mapas de Casos de Uso por el inglés <i>Use Case Maps</i> .
UML	Lenguaje Unificado de Modelado por el inglés <i>Unified Modeling Language</i> .
URN	Notación de Requerimientos de Usuario por el inglés <i>User Requirements Notation</i> .
XML	Lenguaje de Marcas Extensible por el inglés <i>Extensible Markup Language</i> .

Capítulo I. Introducción

En este capítulo, se describe el problema a resolver en esta tesis, los objetivos planteados, la justificación y los beneficios a obtener. Además, se presenta la propuesta de solución y los alcances y limitaciones establecidos. Finalmente, se describe de forma breve la organización del presente documento de tesis.

1.1. Introducción

Los patrones de análisis (PAs) son elementos de una técnica de reutilización utilizada en la comunidad de ingeniería de software, debido a que agilizan el proceso de análisis y mejoran la calidad en el desarrollo de sistemas de software. Incluyen las mejores representaciones consensuadas y correctas de los dominios del problema, además, ayudan a reducir tiempos, costos y esfuerzo (Grier, Sam, Hennessey, State, & Valley, n.d.).

Por otro lado, representar los PAs usando ontologías, ha permitido describir formalmente las entidades y sus propiedades, relaciones, restricciones y comportamiento. Con ello se representa, intercambia y comparte el conocimiento utilizando un vocabulario común, ya sea entre personas o entre agentes de software (Barchini & Álvarez, 2010).

Con la finalidad de aumentar la reutilización y adopción de los PAs en diferentes dominios, en el departamento de Ciencias Computacionales del Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), se creó la Plantilla de Patrones de Análisis Estables (PPAE) y su ontología (OPPAE) (González et al., 2015). Posteriormente, en (López-Ramírez, 2015) se propuso la metodología para construir ontologías de patrones de análisis (MECOPA) y después, en (Juárez Hernández, 2016) se incorporó a esta metodología una nueva fase denominada MEVASE con el objetivo de formalizar la estructura de los patrones de análisis utilizando la lógica descriptiva. Finalmente, en (Piedrola Domínguez N., 2019) se propuso la metodología para la construcción de ontologías temporales de patrones de análisis (MECOTPA) que es una actualización de la metodología MECOPA, para facilitar a los ingenieros de software la inclusión de comportamiento dinámico en las ontologías de PAs mediante aspectos de temporalidad, para obtener una ontología temporal de un PAs estable (OTPAE).

Con la metodología MECOTPA es posible construir ontologías de PAs que incluyen en su definición aspectos estructurales utilizando la lógica descriptiva y aspectos de temporalidad utilizando la lógica temporal (OTPAEs). Complementar los patrones de análisis con aspectos de temporalidad permite razonar e inferir información tanto en aspectos estructurales como de comportamiento.

Actualmente, solamente existen dos OTPAEs documentadas de patrones de análisis creadas con la metodología MECOTPA: La ontología temporal del Patrón de Análisis de Entidades Reutilizables (PAER) y la ontología temporal del Patrón de Análisis de Cuentas (PAC). Contar con las dos ontologías, fue importante en esta investigación para cumplir con el objetivo.

Dada la complejidad de las aplicaciones, se requerirá la combinación de dos o más patrones de análisis para representar la problemática de una empresa o un problema más grande (Jiang & Fernandez, 2009). Sin embargo, no se cuenta con la descripción de cómo combinar ontologías temporales de patrones de análisis.

En este contexto, en este trabajo de investigación se describe un método para combinar patrones de análisis representados mediante ontologías temporales (OTPAEs), que resulta en un patrón de análisis compuesto representado también por una OTPAE.

Con el método descrito en este trabajo, es decir, el método para la Combinación de Ontologías Temporales de Patrones de Análisis (COMOTPA) es posible combinar OTPAEs obtenidos a partir de la metodología MECOTPA y obtener una nueva ontología temporal que representa un PA compuesto. Utilizando el método COMOTPA:

- Se combinó de forma manual la ontología temporal del Patrón de Análisis para Inventarios (PAI) con la ontología temporal del patrón de análisis PAER, dando como resultado la OTPAE consistente del Patrón de Análisis para Inventarios y Entidades Reutilizables (PAIyER).
- Se obtuvo la especificación de comportamientos dinámicos en la OTPAE del PAIyER representados mediante fórmulas de la TDL ALC_{LTL} .
- Se incluyeron en la ontología temporal del PAIyER representaciones de comportamientos dinámicos mediante secuencias de interpretaciones (descripciones de trayectorias de estado en las ontologías) basadas en casos de éxito y casos de éxito y fracaso (una de cada tipo por RF).
- Se comprobó la satisfacibilidad de las especificaciones en los comportamientos dinámicos incluidas en la OTPAE del PAIyER, mediante dos verificaciones de satisfacibilidad por cada especificación, es decir, una para las secuencias de interpretaciones basadas en casos de éxito y otra para las secuencias de interpretaciones basadas en casos de éxito y fracaso.

1.2. Descripción del problema

La representación de los PAs mediante ontologías ha facilitado la reutilización del conocimiento sobre un dominio determinado de una manera más fácil ya que proveen una base para la representación del conocimiento del dominio (Dzung, 2009; Kaiya & Saeki, 2006) y además ayudan a identificar las categorías semánticas (ejemplo: categorías de acción como *comprar*, *solicitar*, o categorías de objeto como *persona*, *libro*, entre otras) del dominio (Barchini & Álvarez, 2010).

Actualmente, en el Centro Nacional de Investigación y Desarrollo Tecnológico se está trabajando en la construcción de ontologías de patrones de análisis que incluyen la lógica descriptiva y la lógica temporal en su formalización. Este es un nuevo enfoque que se está trabajando con la finalidad de complementar y mejorar la formalización de los patrones de análisis (Konrad & Cheng, 2005).

Para representar la problemática de una empresa o un problema más grande es posible que se requiera la combinación de dos o más patrones de análisis (Jiang & Fernandez, 2009)

El problema radica en que no hay una forma de combinar los patrones de análisis representados mediante ontologías que utilicen la lógica descriptiva y la lógica temporal. La causa de este problema se debe a que no se han hecho combinaciones de OTPAEs y esto a su vez se debe a que no hay algo que explique cómo combinarlas.

1.3. Objetivos

1.3.1. Objetivo general

Describir un método para la combinación de patrones de análisis representados mediante ontologías que incluyan en su formalización la lógica descriptiva y la lógica temporal para aspectos estructurales y de comportamiento, respectivamente.

1.3.2. Objetivos específicos

1. Seleccionar los métodos a utilizar para realizar la combinación de ontologías que utilizan lógicas descriptiva y temporal en su formalización (OTPAEs).
2. Definir una forma de combinar los patrones de análisis representados mediante ontologías que incluyen en su formalización la lógica descriptiva y la lógica temporal.
3. Probar la combinación con una nueva ontología de un PA (tercer OTPAE disponible), utilizando el método propuesto.

1.4. Justificación y beneficios

1.4.1. Justificación

Con base en el estudio y análisis del estado del arte (descrito en el *Capítulo III. Estado del arte*) se ha notado que los patrones de análisis utilizados para la combinación no utilizan aspectos estructurales y de comportamiento en su definición.

Con la descripción de un método para combinar patrones de análisis representados mediante ontologías temporales (OTPAEs), será posible combinar dos o más ontologías de patrones de análisis que utilizan la lógica descriptiva y temporal en su formalización y con ello facilitar la representación de soluciones a problemas más grandes que ocurren en la fase de análisis del proceso de desarrollo de software.

1.4.2. Beneficios

Con el desarrollo de este trabajo de investigación los beneficios que se podrán obtener son los siguientes:

1. Se facilitará el reuso de patrones de análisis.
2. Se ayudará a representar soluciones de problemas más grandes con base en la combinación de problemas más pequeños que ocurren en la fase de análisis del proceso de desarrollo de software.
3. Se facilitará la combinación de ontologías de patrones de análisis que incluyen en su definición aspectos estructurales y de comportamiento.
4. Se obtendrá una formalización completa (con aspectos estructurales y de comportamiento) de un patrón de análisis compuesto.
5. El método propuesto generará conocimiento para el acervo científico del Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET). Con este método, se pretende facilitar el reuso y disminuir las dificultades para adoptar los patrones de análisis en dominios diferentes. Además, se contribuirá en el estado del arte para futuras investigaciones.

1.5. Propuesta de solución

Para el desarrollo del presente trabajo de investigación se propone la siguiente solución.

1. **Inclusión del protocolo de investigación y complemento del estado del arte.** Se incluirá la investigación del protocolo y se complementará el estado del arte con trabajos que sean de mayor interés para la presente investigación.
2. **Estudio y análisis de la lógica descriptiva y la lógica temporal.** Se realizará un estudio sobre el uso y funcionamiento de la lógica descriptiva y la lógica temporal en los trabajos realizados por (Piedrola Domínguez N., 2019).
3. **Análisis de la OTPAE del PAER.** Se analizará la OTPAE del PAER con la finalidad de detectar y entender el comportamiento de la lógica descriptiva y la lógica temporal en su definición.
4. **Análisis de la OTPAE del PAC.** Se analizará la OTPAE del PAC con la finalidad de detectar y entender el comportamiento de la lógica descriptiva y la lógica temporal en su definición.
5. **Revisión y estudio de técnicas y/o métodos para la combinación de lógicas descriptiva y temporal.** Se realizará un estudio de técnicas y/o métodos que permitan realizar la combinación de lógicas descriptiva y temporal con la finalidad de conocer el procedimiento para dicho proceso.
6. **Combinación de la OTPAE del PAER con la OTPAE del PAC.**

7. **Descripción del método utilizado en la combinación.** Se describirá el método utilizado para realizar la actividad del paso 6 con la finalidad de documentar el conjunto de actividades utilizadas para obtener un patrón de análisis compuesto.
8. **Creación de la OTPAE del Patrón de Análisis para Inventarios (PAI).** Se creará la OTPAE del PAI para utilizarla posteriormente en un caso de estudio.
9. **Combinación de la OTPAE del PAER con la OTPAE del PAI.** Se realizará un caso de estudio para la combinación de la OTPAE del PAER con la OTPAE del PAI (creada en el paso 8) con la finalidad de probar el método propuesto.

Por tal motivo, en este trabajo de investigación se describirá un método para combinar patrones de análisis representados mediante ontologías temporales (OTPAEs) creadas a partir de la metodología MECOTPA.

1.6. Alcances y limitaciones

A continuación, se presentan los alcances y las limitaciones que tendrá este trabajo de investigación.

1.6.1. Alcances

- Describir un método para combinar patrones de análisis representados mediante ontologías temporales.
- Obtener un patrón de análisis compuesto, representado mediante una ontología temporal, como caso de estudio.
- Realizar la verificación y validación del patrón de análisis compuesto.

1.6.2. Limitaciones

- Se realizará un caso de estudio para la combinación de solo dos ontologías temporales de patrones de análisis.
- Las ontologías a combinar serán la OTPAE del Patrón de Análisis de Entidades Reutilizables (PAER) y la OTPAE del Patrón de Análisis para Inventarios (PAI).
- La combinación de las ontologías se realizará de forma manual.
- La verificación y validación del patrón de análisis compuesto se realizará de forma manual.

1.7. Organización del documento de tesis

El presente documento de tesis se encuentra estructurado en siete capítulos y una sección de anexos. A continuación, se describen de forma general los capítulos restantes.

Capítulo II. Marco teórico. En este capítulo, se presentan los conceptos fundamentales que determinan el contexto en el cual se posiciona la investigación.

Capítulo III. Estado del arte. En este capítulo, se presentan los resultados del proceso de revisión y análisis del conocimiento existente acerca del tema de ontologías temporales de patrones de análisis y que sirven como punto de referencia y comparación con el trabajo propuesto.

Capítulo IV. Método para la Combinación de Ontologías Temporales de Patrones de Análisis. En este capítulo, se presenta la descripción de las fases y actividades del Método para la Combinación de Ontologías Temporales de Patrones de Análisis (COMOTPA). Se describen detalladamente las actividades que se realizan en cada una de las fases del método propuesto.

Capítulo V. Creación de la Ontología Temporal del Patrón de Análisis para Inventarios. En este capítulo, se presentan los resultados obtenidos en cada una de las fases de la metodología MECOTPA durante la creación de la ontología temporal del PAI.

Capítulo VI. Caso de estudio: Combinación de la OTPAE del PAI con la OTPAE del PAER. En este capítulo, se presenta la documentación y las diferentes actividades que se realizaron en la combinación de las ontologías temporales de los PAs de Inventarios y Entidades Reutilizables.

Capítulo VII. Conclusiones y trabajos futuros. En este último capítulo, se presenta las conclusiones obtenidas del presente trabajo, las aportaciones realizadas y sugerencias para trabajos futuros en el ámbito de estudio.

Anexos. En esta sección, se presentan como anexos: las ontologías OPAE y OTPAE de los patrones de análisis PAI y PAIyER que fueron obtenidos en este trabajo de investigación; la documentación utilizada en los capítulos V y VI para la Verificación de Consistencia Semántica Temporal (TSCC) y; Verificación de Satisfacibilidad (SC). Finalmente, se presenta el contenido del disco que acompaña este trabajo de tesis.

Capítulo II. Marco teórico

En este capítulo, se describen los conceptos fundamentales que determinan el ámbito contextual de esta investigación. Los conceptos descritos son los siguientes: Patrón de análisis, Patrón de análisis estable, Plantilla de patrones de análisis estables, Ontología, Lógicas de descripción, Lógica temporal, Lógicas de descripción temporal, Verificación de consistencia semántica temporal y Verificación de modelos.

2.1. Patrón de análisis

Martin Fowler introdujo este término para describir soluciones relacionadas con problemas que aparecen durante el análisis de requisitos y las fases de modelado conceptual de los datos. Para él “un patrón es una idea que fue útil en un contexto práctico y probablemente será útil en otros” (Fowler, 1996). La semántica de los patrones de análisis describe aspectos específicos de algún dominio de aplicación en concreto, permitiendo definir un conjunto de modelos conceptuales que juntos describen un lenguaje del dominio (Fowler, 1996).

En (Fernandez, 1998) se establece que un patrón de análisis contiene un conjunto de clases conceptuales y asociaciones que tienen significado en el contexto de una aplicación; es decir, es un modelo conceptual de una parte de la aplicación. Sin embargo, la estructura es válida en otras aplicaciones.

A diferencia de los patrones de diseño (utilizados en la fase de diseño del Proceso de Desarrollo de Software), los patrones de análisis no profundizan en detalles de implementación. Normalmente son especificados por medio de narración libre, plantillas o modelos conceptuales.

2.2. Patrón de análisis estable

Con el fin de aumentar la estabilidad y reuso, los patrones de análisis se describen usando conceptos basados en el Modelo de Estabilidad del Software (SSM, *Software Stability Model*) dando lugar a los patrones de análisis estables (PAEs).

Los PAEs permiten analizar los problemas en términos de temas perdurables de negocio (EBTs, *Enterprise Business Themes*), objetos de negocio (BOs, *Business Objects*) y objetos industriales (IOs, *Industrial Objects*) (López Ramírez, 2015). Los PAEs modelan el conocimiento central y en consecuencia se usan para modelar el mismo problema sin importar el contexto (Hamza & Fayad, 2002).

2.3. Plantilla de patrones de análisis estables

En (González et al., 2015), se propone una plantilla de patrones de análisis estables (PPAE) para describir los PAEs. En esta plantilla se integran los puntos propuestos en (Raminhos et al. 2006), clasificados en términos de EBTs, BOs e IOs para ajustarse a las características de estabilidad del modelo SSM y capturar el conocimiento central de un problema, de tal manera que se reutilice cada vez que se presente el mismo problema.

2.4. Ontología

Según Studer (Studer et al., 1998), las dos definiciones que expresan de mejor manera, el concepto de ontología son las de Gruber (Gruber, 1993): “Una ontología es una especificación explícita de una conceptualización” y la de Borst (Borst, 1997): “Una ontología es una especificación formal de una conceptualización compartida”; y propone una nueva definición haciendo una mezcla de ambas: “Una ontología es una especificación formal y explícita de una conceptualización compartida”.

Los elementos que componen esta definición, los explica el autor de la siguiente forma:

- a) Compartida: refleja la noción de que una ontología captura conocimiento consensual, que no es producto de un sólo individuo, sino aceptado por un grupo.
- b) Conceptualización: se refiere a un modelo abstracto de algún fenómeno en el mundo proveniente de la identificación de los conceptos relevantes de dicho fenómeno.
- c) Explícita: significa que el tipo de conceptos utilizados y las restricciones de su uso se definen explícitamente.
- d) Formal: se refiere al hecho de que la ontología debe ser legible por una computadora, lo cual se consigue con la utilización de un lenguaje para la representación de ontologías. El lenguaje de este tipo más utilizado, es OWL.

2.5. Lógica de Descripción

Las lógicas de descripción (DLs) son una familia de formalismos de representación del conocimiento que pueden usarse para representar el conocimiento terminológico de un dominio de aplicación de manera estructurada (Harmelen, Frank Van, Vladimir Lifschitz, 2008).

A estas lógicas se les suele clasificar de acuerdo a las posibilidades expresivas que poseen, es decir, de acuerdo a los constructores que proveen:

- Lenguaje de Atributos (AL por sus siglas en inglés, *Atributive Language*): soporta negación atómica, intersección de conceptos, restricciones universales y cuantificación existencial limitada.
- Lenguaje de Descripción Basado en Marcos (FL, por sus siglas en inglés, *Frame Based Description Language*): soporta intersección de conceptos, restricciones universales, cuantificación existencial limitada y restricción de roles.
- Lenguaje Existencial (EL por sus siglas en inglés, *Existencial Language*): soporta intersección de conceptos y restricciones existenciales.

Las lógicas de descripción pueden extenderse mediante una nomenclatura que consiste en añadir al final del nombre de la lógica de descripción una letra que indica un tipo de constructor adicional. Por ejemplo, una extensión de la DL AL es el Lenguaje de Atributos

con Complementos (ALC, por sus siglas en inglés, *Attributive Language with Complements*), la cual es compatible con la representación de conocimiento terminológico (axiomas del Componente Terminológico (TBox por su abreviación en inglés, *Terminological Component*) utilizando las: relaciones entre las subclases (\sqsubseteq), equivalencia (\equiv), conjunción (\sqcap), disyunción (\sqcup), negación (\neg), restricciones de propiedad (\forall , \exists), tautología (\top) y contradicción (\perp). La ALC también puede representar el conocimiento afirmativo (axiomas del componente de aseveraciones (ABox por su abreviación en inglés, *Assertion Component*), tales como las afirmaciones individuales y aseveraciones de propiedad (Harmelen, Frank Van, Vladimir Lifschitz, 2008).

2.6. Lógica Temporal

Las lógicas temporales (TLs por sus siglas en inglés, *Temporal Logics*) son un tipo de lógica modal para representar propiedades de dominios dependientes del tiempo. Permiten la especificación del orden relativo de eventos, por ejemplo, “Un auto se detiene una vez que el conductor pisa el freno” o “Un mensaje se recibe después de que éste se envía”. Sin embargo, una lógica temporal no soporta algún medio para referirse a los tiempos precisos de los eventos. Por lo tanto las modalidades de la lógica temporal son abstractas respecto al tiempo (Harmelen, Frank Van, Vladimir Lifschitz, 2008).

En la lógica temporal aparecen los mismos operadores que en una lógica de primer orden, junto con otros nuevos, entre los que se pueden encontrar: *siempre*, *algunas veces* y *nunca*. Las lógicas temporales se interpretan tradicionalmente en términos de estructuras de Kripke, es decir, un grafo cuyos nodos representan estados alcanzables del sistema y cuyas aristas representan transiciones de estados (Harmelen, Frank Van, Vladimir Lifschitz, 2008).

Algunos sistemas lógicos basados en lógica temporal con distinto nivel de expresividad y complejidad son:

- Lógica Temporal de Intervalos (ITL por sus siglas en inglés, *Interval Temporal Logic*).
- Lógica Lineal Temporal (LTL por sus siglas en inglés, *Linear Temporal Logic*).
- Lógica Computacional Ramificada (CTL por sus siglas en inglés, *Computational Tree Logic*).
- Lenguaje de Especificación de Propiedades (PSL por sus siglas en inglés, *Property Specification Language*).
- CTL* la cual generaliza LTL y CTL.
- Lógica Hennessy-Milner (HML por sus siglas en inglés, *Hennessy–Milner logic*).
- μ calculus, la cual incluye un subconjunto de HML y CTL*.
- Lógica Temporal de Intervalo Métrico (MITL por sus siglas en inglés, *Metric Interval Temporal*).

- Lógica Temporal de Señal (STL por sus siglas en inglés, *Signal Temporal Logic*).

2.7. Lógicas de Descripción Temporal

Las DLs se diseñaron para usarse como un formalismo en la representación y razonamiento del conocimiento en dominios de aplicación estáticos. Por lo tanto, no son capaces de representar y razonar sobre propiedades cualitativas o cuantitativas del tiempo, como lo hacen las TLs.

Frecuentemente, es deseable combinar la TL con la DL para representar y razonar sobre el conocimiento de dominios de aplicación dinámicos (o dependientes del tiempo). Esta combinación ha resultado en diversas propuestas de Lógicas de Descripción Temporal (TDLs, *Temporal Description Logics*). La propuesta de Schild en 1993, es una de las más recientes. Desde entonces ha experimentado un desarrollo constante en el sentido de que la TL y la DL han variado.

Un enfoque es combinar las DLs estándar tales como el Lenguaje de Atributos con Complementos (*ALC, Atributive Language with Complements*) y el Lenguaje de cuantificación existencial completa (*EL, full Existential quantification Language*), con TLs estándar como LTL, CTL y CTL*. Por ejemplo, la combinación de la DL *ALC* y la TL LTL, da como resultado la TDL ALC_{LTL} .

Para propósitos de esta investigación, se utiliza la lógica de descripción temporal ALC_{LTL} , en (Piedrola Domínguez N., 2019) se describe a detalle la sintaxis y semántica de la TDL ALC_{LTL} .

2.8. Verificación de Consistencia Semántica Temporal

Una Verificación de Consistencia Semántica Temporal (TSCC, *Temporal Semantic Consistency Checking*) asegura que la base de conocimiento (KB, *Knowledge Base*, conformada por una *TBox* y varias *ABox*) no contenga hechos contradictorios. Incluso, asegura que las secuencias de interpretaciones (SoIs, *Sequences of Interpretations*) que contiene, estén definidas en una trayectoria adecuada.

En (Piedrola Domínguez N., 2019) se describe brevemente al razonador Pellet como una herramienta útil en la TSCC.

2.9. Verificación de Modelos

La Verificación de Modelos (MC, *Model Checking*), es una técnica formal con el propósito de ejecutar abstracciones de software (por ejemplo, un sistema de transición de estados) y

analizar todos los posibles comportamientos (por ejemplo, trayectorias de estados) de estas abstracciones, antes de que el sistema se construya (Adesina, 2017).

El enfoque MC se basa en modelos representados como estructuras Kripke (por ejemplo, un sistema de transición de estados) y se fundamenta en el principio de la exploración exhaustiva de todos los estados posibles. MC requiere de un modelo y los requerimientos del sistema bajo análisis, expresados en el dialecto de un motor particular y un formalismo lógico tal como la TL LTL o la TL CTL. El modelo y sus requerimientos se pasan a un verificador de modelos para determinar si el modelo cumple con sus requerimientos.

Formalmente, MC se refiere al siguiente problema: Dado una propiedad deseada, expresada como una fórmula en lógica temporal p , y un modelo M con un estado inicial s ; decidir si $M, s \models p$, es decir, a partir del estado s de la estructura M se satisface la fórmula p . En caso contrario $M, s \not\models p$, se producirá un contraejemplo. Los contraejemplos se usan como retroalimentación para asistir al usuario en certificar o corregir el sistema bajo análisis. Los contraejemplos describen trayectorias de ejecución que conducen desde el estado inicial del sistema a los estados que violan el requerimiento.

Los motores de análisis dependen de una enumeración de estado explícito o de enfoques simbólicos. Con la enumeración de estado explícito, los estados del programa tienden a aumentar exponencialmente. Por otro lado, los enfoques simbólicos (por ejemplo, SAT (*Boolean Satisfiability Problem*), OBDDs (*Ordered Binary Decision Diagrams*), AIGs (*And-Inverter Graphs*) representan de manera eficiente conjuntos y relaciones como fórmulas booleanas.

Los enfoques MC tienen sólidos fundamentos matemáticos y ofrecen resultados en los que se puede confiar. Los beneficios de los enfoques incluyen la facilidad de integración con un ciclo de desarrollo existente y la verificación automática. Sin embargo, tienen una capacidad limitada para respaldar la verificación de sistemas intensivos en datos y sufren del problema de explosión del estado.

Algunas herramientas automáticas (o verificadores de modelos) que garantizan la satisfacibilidad de las TLs de los marcos lineal y temporal son: UPPAAL (resultado de un desarrollo en conjunto por las universidades de Uppsala y Aalborg en Suecia y Dinamarca, respectivamente), PRISM (*Probabilistic Symbolic Model Checker*), Spin (*Simple Promela Interpreter*) y NuSMV/nuXmv (*New [eXtended] Symbolic Model Verifier*).

En este trabajo de investigación, se utiliza el verificador de modelos simbólico nuXmv, el cual verifica la satisfacibilidad (SC) de una fórmula p de la TL LTL en una estructura de estados finita o infinita denominada modelo M , es decir, comprueba que existe al menos una trayectoria en el modelo M que satisface lo especificado por la fórmula p . En caso contrario, genera los respectivos contraejemplos.

En este capítulo, se presentaron los conceptos que sustentan la investigación reportada en esta tesis. Se incluyó información acerca de patrones de análisis debido a que esta investigación se enfoca principalmente al estudio y representación de aspectos estructurales y de comportamiento dinámico que describen los patrones de análisis. Además, se presentó información acerca de ontologías, debido a que en esta investigación, son utilizadas para representar formalmente el conocimiento estático y dinámicos de los patrones de análisis. También, se presentó información acerca de la verificación de la consistencia semántica temporal y verificación de modelos con la finalidad de conocer los elementos necesarios para obtener ontologías temporales de patrones de análisis válidas y consistentes en este trabajo de investigación.

Capítulo III. Estado del arte

En este capítulo, se presenta el estudio y análisis de los antecedentes (trabajos realizados en el CENIDET) y trabajos relacionados (trabajos realizados por otros grupos de investigación) con respecto al tema de ontologías de patrones de análisis y que sirven como punto de referencia y comparación con el trabajo propuesto.

3.1. Antecedentes

En esta sección, se describen brevemente los trabajos de investigación que se han realizado en el Departamento de Ciencias Computacionales del Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) que preceden y están relacionados a esta investigación.

3.1.1 Almacenamiento y uso de patrones de análisis apegados a su semántica (López, 2015)

En este trabajo, se propone facilitar la utilización de patrones de análisis como apoyo en la elaboración de software, incorporando semántica a la descripción de patrones de análisis, a partir de individuos de la ontología de la plantilla estable de patrones de análisis (OPPAE). Se define la Metodología para la Construcción de Ontologías de Patrones de Análisis (MECOPA), que es el resultado de una adaptación a METHONTOLOGY, para la creación de ontologías de patrones de análisis, describiendo el proceso de creación de la ontología del patrón de análisis de reservación y uso de entidades reutilizables (OPAE del PARE), mediante un proceso de análisis y síntesis a partir de la información y semántica representadas en las instancias de la ontología de la plantilla estable de patrones de análisis (OPPAE).

Otro resultado de esta investigación es la especificación de requerimientos y el diseño de alto nivel del sistema para el Almacenamiento y Uso de Patrones de Análisis apegados a su Semántica (AUPAS), que utiliza la estructura de un repositorio para almacenar los tres niveles de abstracción de las siguientes ontologías: 1) ontología de la plantilla estable de patrones de análisis (OPPAE), y sus instancias (descripciones de los PAs) , 2) ontologías de los patrones de análisis mismos (OPAE del PARE), y sus instancias (casos de uso de los PAs en dominios reales), y 3) diferencias entre los casos de uso y las ontologías de los patrones. De manera que las ontologías del nivel 2 se crean mediante un proceso de análisis y síntesis (descrito en MECOPA) a partir de la información y semántica representadas en las instancias del nivel 1; y las diferencias del nivel 3 se crean comparando las ontologías y las ontologías del nivel 2.

En la investigación reportada en esta tesis, se utiliza la OTPAE del PAER, que es una actualización de la ontología del patrón de análisis de entidades reutilizables (OPAE del PAER) creada en este trabajo. Por lo tanto, este trabajo fue de gran importancia debido a que proporcionó una primera descripción de la ontología del patrón de análisis PAER, lo cual fue de gran ayuda para obtener la ontología temporal del PAER utilizada en esta investigación.

3.1.2 Metodología de Validación de Consistencia de Patrones de Análisis (Juárez, 2016)

En este trabajo, se propuso una nueva metodología denominada Metodología de Validación de Consistencia de Patrones de Análisis (MEVASE) la cual se incluyó como una nueva fase en la

metodología MECOPA, de tal manera que mientras se construye una ontología se pueda 1) verificar la consistencia de un patrón de análisis, y 2) validar la ontología mediante el lenguaje matemático de las DLs, durante su ciclo de desarrollo.

En la investigación reportada en esta tesis, se utiliza la metodología MECOPA (con la metodología MEVASE incluida) para obtener la ontología (OPAE, Ontología del Patrón de Análisis Estable) válida y consistente, que representa la estructura estática del Patrón de Análisis para Inventarios (PAI), que denominamos OPAE del PAI.

3.1.3 Almacenamiento y manipulación de ontologías que describen patrones de análisis y su semántica (Candía, 2017)

Con el fin de promover y facilitar el reuso de los patrones de análisis en la representación de problemas específicos, en este trabajo diseñó y construyó la estructura de los tres niveles de abstracción propuesta en el diseño del sistema AUPAs (ver sección 3.1.1), mediante una instancia del repositorio de ontologías biomédicas BioPortal, que permite almacenar y acceder a las descripciones y semántica de los patrones de análisis a través de la web. Con la estructura definida, se almacenaron las descripciones y la semántica de los patrones de análisis de entidades reutilizables (PAER) y de cuentas (PAC).

Este repositorio ayuda a compartir información de los procesos de un dominio, permite la proliferación de patrones de análisis aumentando la diversidad de dominios y ayuda a representar la problemática que se desea resolver.

Para la investigación reportada en esta tesis, este trabajo puede ser de gran ayuda para almacenar las ontologías temporales compuestas, de tal manera que podrían facilitar y promover el reuso de los patrones de análisis.

3.1.4. FOCOPAT: Conversión de ontologías de patrones de análisis estáticas a temporales, para la especificación de comportamientos dinámicos y su validación (Piedrola Domínguez N., 2019)

En este trabajo, se incluye como una nueva fase en la metodología MECOPA (actualizada en (Juárez, 2016)), la realización del método para la Formalización Complementaria de Patrones de Análisis con Aspectos de Temporalidad (FOCOPAT). Esta nueva fase, complementa la formalización de los patrones de análisis con aspectos de temporalidad con la descripción de su comportamiento. Debido a esto, la metodología MECOPA se actualiza y se le cambia el nombre a Metodología para la Construcción de Ontologías Temporales de Patrones de Análisis (MECOTPA). Con MECOTPA es posible construir Ontologías Temporales de Patrones de Análisis Estables (OTPAEs) (González et al., 2015), que describen comportamientos satisfacibles que facilitan también el uso automatizado

de aspectos estructurales, y de comportamiento dinámico, de los patrones de análisis. Se utiliza la lógica de descripción temporal ALC_{LTL} (una combinación de la lógica descriptiva ALC y la lógica temporal LTL) para representar formalmente la estructura estática y el comportamiento dinámico del patrón de análisis.

Para la investigación reportada en esta tesis, con la metodología MECOTPA y con ayuda del editor de ontologías Protegé, se construyó la Ontología Temporal del Patrón de Análisis para Inventarios (OTPAE del PAI), requerida para el caso de estudio descrito en el *Capítulo VI. Caso de estudio: Combinación de la OTPAE del PAI con la OTPAE del PAER.*

En la Tabla 1, se presenta una comparativa de los trabajos descritos en esta sección. Para un mejor entendimiento, a continuación, se describen las columnas y abreviaturas utilizadas en esta tabla.

Columnas:

- Antecedente (1ra columna): Referencia bibliográfica del trabajo antecedente.
- Ontología (2da columna): Tipo de ontología (estática o dinámica) utilizada para representar el problema de estudio.
- Lenguajes formales (3ra columna): Lenguajes formales para representar, restringir y razonar sobre aspectos estáticos o dinámicos del problema de estudio.
- Herramientas y/o recursos (4ta columna): Herramientas o recursos de apoyo para el desarrollo del trabajo de investigación.
- Comportamiento (5ta columna): Medios para la descripción de aspectos de comportamiento dinámico.
- Combinación de ontologías (6ta columna): Ontologías que se combinan si se da el caso.
- Producto resultado (7ma columna): Describe el producto resultado de los trabajos antecedentes, los cuales son herramientas, métodos o metodologías.

Abreviaturas:

- PA: Patrón de análisis y PAs para el plural: Patrones de Análisis.
- PAC: Patrón de Análisis de Cuenta.
- PAER: Patrón de Análisis de Entidades Reutilizables.
- PAI: Patrón de Análisis para Inventarios

- PAIyER: Patrón de Análisis para Inventarios y Entidades Reutilizables (resultado de la combinación de los PAs PAI y PAER).
- OWL (*Ontology Web Language*): Lenguaje de Ontologías Web, que es un lenguaje de marcado (mediante etiquetas), construido sobre el Marco de Descripción de Recursos (RDF, *Resource Description Framework*) y codificado en el Lenguaje de Marcas Extensible (XML, *eXtensible Markup Language*).
- OWL-DL (*Ontology Web Language – Description Logic*): Lenguaje de Ontologías Web y Lógica Descriptiva, incluye todas las construcciones del lenguaje OWL, además de incluir restricciones expresadas en lógica descriptiva.
- SWRL (*Semantic Web Rule Language*): Lenguaje de Reglas para la Web Semántica para expresar reglas, así como también lógica mediante la combinación de OWL-DL y un subconjunto del Lenguaje de Marcado de Reglas (RuleML, *Rule Markup Language*).
- TDL ALC_{LTL}: Lógica de Descripción Temporal que combina la DL ALC y la TL LTL en un formalismo para la representación y razonamiento de conceptos dependientes del tiempo (p. ej., comportamientos dinámicos), basado en la integración de la DL ALC y la TL LTL.
- TL LTL (*Linear Temporal Logic*): Lógica Temporal Lineal, un formalismo utilizado para especificar propiedades (p. ej., comportamientos dinámicos) a verificar en trayectorias de estado de un modelo (p. ej., una estructura Kripke) por herramientas de verificación de modelos (MC, p. ej., nuXmv). Para modelar el tiempo, tiene un conjunto numerable de instantes, organizados linealmente, pero no puede representar el tiempo transcurrido (p. ej., tiempo de duración exacta de un evento).
- GRL (*Goal-oriented Requirement Language*): Lenguaje de requerimientos orientado a metas, soporta el razonamiento sobre sobre requerimientos funcionales y requerimientos no-funcionales (atributos de calidad).
- UCM (*Use Case Maps*): Mapas de Casos de Uso, es una notación cuya meta es enlazar el comportamiento y la estructura en una manera visual y explícita. A través de trayectorias que representan escenarios, describe relaciones causales entre responsabilidades.

Tabla 1. Comparativa de trabajos antecedentes.

Antecedente	Ontología	Lenguajes formales	Recursos y/o herramientas	Comportamiento	Combinación de ontologías	Resultado
(López-Ramírez, 2015)	Estática: Describe la semántica de los PAs.	OWL	Protégé	Textual y gráfica mediante diagramas GRL y UCM	Ninguna	<ul style="list-style-type: none"> • Especificación y diseño de alto nivel del sistema AUPAs • Metodología MECOPA • OPAE del PAER
(Juárez Hernández, 2016)	Estática: Describe la semántica de los PAs de manera consistente.	OWL-DL	Protégé y DLs		Ninguna	<ul style="list-style-type: none"> • Metodología MEVASE integrado en MECOPA. • OPAE válida y consistente del PA PAC.
(Zavaleta, Guadalupe Candia;Garcia, 2017)	Estática: Almacena descripciones de PAs que incluyen su semántica.	OWL, OWL-DL	NCBO BioPortal		Ninguna	<ul style="list-style-type: none"> • Almacen de descripciones de PAs y su semántica. • Jerarquía de ontologías que ayuda el refinamiento de PAs. • PAs PAC y PAER almacenados.
(Piedrola Domínguez N., 2019)	Dinámica: Describe de manera consistente la semántica y comportamiento dinámico de los PAs.	OWL-DL, Reglas SWRL, TDL ALC _{LTL} y TL LTL	Protégé, plugin Chronos, razonador Pellet, nuXmv, Ontología OWL-Time	Especificaciones formales de comportamiento satisficibles en trayectorias de estado	Ninguna	<ul style="list-style-type: none"> • Metodología MECOTPA • OTPAE del PAC • OTPAE del PAER
Esta investigación	Dinámica: Describe de manera consistente la semántica y comportamiento dinámico de PAs compuestos.	OWL-DL, Reglas SWRL, TDL ALC _{LTL} y TL LTL	Protégé, plugin Chronos, razonador Pellet, nuXmv, Ontología OWL-Time	Especificaciones formales de comportamiento satisficibles en trayectorias de estado	Combinación de la OTPAE del PAI con la OTPAE del PAER	<ul style="list-style-type: none"> • Método COMOTPA • OTPAE del PAI. • OTPAE del PA compuesto PAIyER.

Los trabajos descritos en la sección 3.1. *Antecedentes*, son antecedentes directos de este trabajo de investigación que muestran el trabajo que se ha venido realizando en los últimos años con respecto al tema de patrones de análisis. Como se puede observar en la Tabla 1, las investigaciones (Juárez Hernández, 2016; López-Ramírez, 2015; Piedrola Domínguez N., 2019; Zavaleta, Guadalupe Candia;Garcia, 2017) se ocuparon en considerar aspectos semánticos en las descripciones de los patrones de análisis, tal es el caso que en (López-Ramírez, 2015) se propuso una metodología para construir ontologías válidas y consistentes, así como de un repositorio web para facilitar el almacenamiento y recuperación de estas ontologías. Además, se ha trabajado en la descripción del conocimiento dinámico, como en la investigación de (Piedrola Domínguez N., 2019) en el que se propuso una metodología para construir ontologías temporales de patrones de análisis (OTPAEs).

Con los trabajos descritos, hasta ahora, es posible construir ontologías temporales de patrones de análisis que permiten representar el conocimiento estático y dinámico. Sin embargo, es común que se requiera la combinación de dos o más patrones de análisis para representar la problemática de una empresa o de un problema más grande (Jiang & Fernandez, 2009). Ante este problema, surgió el trabajo de investigación realizado en esta tesis, pues proponemos un método para combinar OTPAEs creadas con la metodología MECOTPA propuesta en (Piedrola Domínguez N., 2019), con la finalidad de obtener patrones de análisis compuestos, representados mediante ontologías temporales válidas y consistentes.

3.2. Trabajos relacionados

En esta sección, se describen brevemente los trabajos realizados por otros grupos de investigación en el campo de patrones de análisis y ontologías, y que están relacionados con la presente investigación. La finalidad de esta sección, es conocer qué es lo que se ha hecho en esta línea de investigación y de esta manera posicionar la investigación que se reporta en este documento.

Los trabajos relacionados presentados a continuación, se dividen en dos categorías: 1) trabajos relacionados al tema de patrones de análisis, que tienen que ver con combinación, gestión y manejo de patrones de análisis y; 2) trabajos relacionados al tema de ontologías, que tienen que ver con algoritmos y herramientas para combinar y/o gestionar ontologías.

3.2.1. Trabajos relacionados a patrones de análisis

3.2.1.1 DC2AP Metadata Editor: A Metadata Editor for an Analysis Pattern Reuse Infrastructure (Peixoto, Da Matta Vegi, & Lisboa-Filho, 2013)

En este trabajo, con el fin de estandarizar la documentación de los patrones de análisis y aumentar su reuso, los patrones de análisis se documentan mediante la Infraestructura para el Reuso de Patrones de Análisis (APRI, *Analysis Patterns Reuse Infrastructure*), usando metadatos.

Los autores proponen un editor de metadatos basado en *Dublin Core Application Profile for Analysis Patterns* (DC2AP) para el Portal de patrones de la infraestructura APRI. Con este editor la documentación de los patrones de análisis se valida (de acuerdo con las reglas del perfil DC2AP) y estandarizada en los repositorios distribuidos de metadatos consiguiendo mejorar el reuso de manera automatizada, debido a que la información puede recuperarse como conocimiento en un formato RDF a través de internet como un Servicio Web. Aunque se almacena una plantilla para describir cada patrón de análisis, no se representa la semántica de los elementos estructurales y/o de comportamiento de los PAs, mediante ontologías.

Para la investigación reportada en esta tesis, este trabajo proporciona una forma de aumentar el reuso de los patrones de análisis, mediante una buena documentación de un patrón de análisis, aunque no se representa la semántica de los PAs (mediante la estructura de una ontología, los elementos del PA mediante clases, propiedades de objetos y propiedades de datos). Debido a que no se describe la semántica de los PAs, no se realizan combinaciones de los elementos estructurales ni de comportamiento de los PAs.

3.2.1.2 Patterns for Business-to-Consumer e-Commerce Applications (Yuan & Fernandez, 2011)

En este trabajo, al igual que en (Jiang & Fernandez 2009), se usan los patrones de análisis semánticos (SAPs, *Semantics Analysis Patterns*), pero esta vez para construir Modelos Independientes de Computación (CIMs, *Computation Independent Model*) a partir de los cuales, pueden generarse aplicaciones específicas usando el enfoque MDA.

Para demostrar la construcción de un CIM, los autores generan un modelo de dominio para las aplicaciones de comercio electrónico de negocio-cliente, a partir de la identificación de cinco SAPs relacionados (Catálogo, Carrito de compras, Orden y envío, Factura y Administrador de Existencias) que representan subsistemas que se combinan considerando la inclusión de clases repetidas en un solo subsistema, así como las relaciones entre éstos.

Es importante considerar aspectos de seguridad en los patrones. Por lo tanto, el patrón de Control de Acceso Basado en Roles puede agregarse al patrón de análisis de Carro de

compras, con el fin de que los usuarios sólo tengan acceso a las tareas o actividades que les asignaron.

Este trabajo, presenta un caso de combinación de PAs semánticos para obtener un PA compuesto, aunque utiliza un enfoque diferente, es un ejemplo de lo que se puede lograr a través de la combinación de PAs. Para la investigación reportada en esta tesis, se analizó el Patrón de Análisis de Administrador de Existencias (o Patrón de Análisis para Inventarios) y se construyó la ontología temporal que lo representa (ver *Capítulo V. Creación de la Ontología Temporal del Patrón de Análisis para Inventarios*).

3.2.1.3 Composing analysis patterns to build complex models: Flight reservation. (Jiang & Fernandez, 2009)

En este trabajo, se muestra el uso de los patrones de análisis semánticos (SAP, por sus siglas en inglés), para construir patrones de análisis complejos a partir de la combinación de patrones más simples. Los SAP son mini aplicaciones que describen algunos casos de uso y que juntos describen una aplicación genérica básica.

Se presenta un caso de estudio enfocado a una aplicación de reservación de vuelos en el cual se construye un patrón de análisis compuesto (o complejo) mediante una combinación sistemática de patrones más simples (o atómicos). Los patrones atómicos o componentes corresponden a funciones específicas del sistema y pueden ser tanto nuevos patrones o modelos existentes, quizás especializados para la aplicación. Debido a la forma en que se construye toda la aplicación, el modelo resultante es también flexible y reutilizable.

De manera general, la interacción entre los patrones que conforman esta aplicación se representa mediante un lenguaje de patrones, como se muestra en la Figura 1. De este modo, cada patrón atómico puede representar un patrón nuevo o existente con una función específica bien definida. La composición de todos los patrones es la base para un marco (framework).

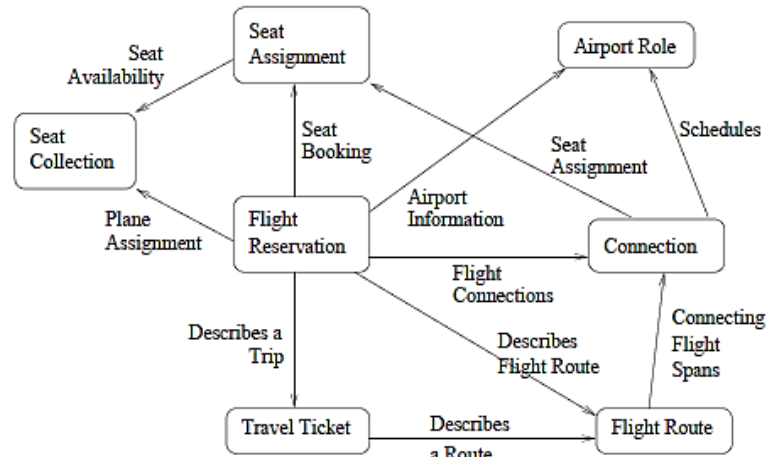


Figura 1. Lenguaje de patrones para la reservación de vuelos.

Para propósitos de la investigación reportada en esta tesis, este trabajo, presenta un ejemplo claro de combinación de patrones de análisis simples para obtener un patrón de análisis más complejo. Aunque en este trabajo no se utiliza un lenguaje formal para representar los PAs combinados, es una idea de lo que se puede lograr con la combinación de PAs.

3.2.1.4 An Analysis Pattern for Invoice Processing (Fernandez & Yuan, 2009)

En este trabajo, se presenta un patrón de análisis para el procesamiento de facturas. El patrón describe eventos como la creación y validación de una factura (garantizar que la información del cliente sea correcta y de acuerdo con los requisitos y regulaciones de cada sistema), seguido del proceso de pago.

Este patrón se compone de dos patrones más simples que describen la creación y el pago de la factura, respectivamente. Es un ejemplo de un patrón de análisis semántico (SAP), ya que expresa aspectos semánticos del modelo.

El patrón compuesto representa una aplicación mínima que puede aplicarse a una variedad de situaciones y puede combinarse con otros patrones relacionados, para describir aplicaciones más complejas. Los patrones componentes tienen valor propio y se pueden usar de forma independiente.

En este trabajo, a diferencia de la investigación reportada en esta tesis, las descripciones de los PAs se realizan a través de diagrama de clases y de actividades, es decir, no utilizan un lenguaje formal para representar la estructura estática y comportamiento dinámico de los PAs. Sin embargo, en este trabajo, se presenta un ejemplo claro de un caso de combinación de patrones de análisis para describir aplicaciones más complejas.

3.2.1.5 Handbook of Knowledge Representation (Harmelen, Frank Van, Vladimir Lifschitz, 2008)

Este libro describe los métodos usados en inteligencia artificial para representar el conocimiento y razonamiento.

La primera parte del libro describe la lógica clásica y los nuevos enfoques que mejoran la trazabilidad computacional, un ejemplo de estos son las lógicas de descripción. Por otro lado, en la segunda parte se explican los retos especiales de la representación y razonamiento con algunos dominios centrales de conocimiento que incluyen tiempo, espacio, causa y acción usando lógicas temporales.

Para propósitos de este trabajo de tesis, este libro fue de gran utilidad, debido a que proporciona información acerca de aspectos básicos y aplicativos de las lógicas de descripción y lógicas temporales en cualquier dominio (p. ej., para representar y razonar sobre el conocimiento estático y dinámico en el dominio de un patrón de análisis en específico). Con la ayuda de este libro, se pudo comprender mejor cómo se utilizan las lógicas temporales, aspectos muy importantes para lograr el objetivo propuesto en esta investigación.

3.2.2. Trabajos relacionados a ontologías

3.2.2.1 Ontology Mapping and Merging Aspects in Semantic Web (Narula GS, Wason R, Jain V, 2018)

En este trabajo, se presenta un análisis y comparación de las principales herramientas disponibles actualmente para el mapeo y fusión de ontologías. El proceso de mapeo implica dos tareas: 1) Descubrir correspondencias semánticas independientes entre los componentes de la ontología (clases, propiedades e instancias) y 2) Definir esas correspondencias descubiertas para que otras ontologías puedan usarlas. La fusión e integración de ontologías se define como la creación de una nueva ontología a partir de múltiples ontologías que consisten de sub ontologías superpuestas. La alineación de ontologías (correspondencias entre conceptos en ontologías) hace que las ontologías de origen sean consistentes y coherentes al unirlos.

Las herramientas presentadas automatizan la integración, unificación, alineación y fusión de diversas ontologías relacionadas. De las herramientas presentadas en este trabajo se encuentran: Chimaera, PROMPT, FCA, MOMIS, GLUE, LILY y ASMOV.

Para propósitos de esta investigación, este trabajo proporciona conocimiento de las principales herramientas utilizadas en la combinación de ontologías, y con base en esto se podrá analizar si es necesario utilizar alguna de éstas como herramienta de apoyo para realizar la combinación de ontologías temporales de patrones de análisis en este trabajo de tesis. Las

herramientas presentadas pueden facilitar la alineación y combinación de ontologías. Sin embargo, con estas herramientas, solo es posible combinar el conocimiento estático de una ontología, es decir, no consideran el comportamiento dinámico, que a diferencia de la investigación reportada en esta tesis, se pretende combinar ontologías temporales de patrones de análisis que incluyan en su formalización el conocimiento estático y el comportamiento dinámico para obtener PAs compuestos representados mediante ontologías.

3.2.2.2 A benchmark for ontologies merging assessment (Mahfoudh, Forestier, & Hassenforder, 2016)

En este trabajo, se presenta un benchmark para la fusión de ontologías que cubre diferentes tipos tales como taxonomías, ontologías ligeras, ontologías pesadas y ontologías multilingües. También, se presenta la herramienta GROM (Reescritura de gráficos para fusión de ontologías) para fusionar ontologías automáticamente.

Gracias al formalismo gramatical del grafo, en particular el NAC (condiciones de aplicación negativa), GROM conserva la consistencia ontológica y elimina las redundancias. Para evaluar el proceso de fusión, se utilizó la cobertura, la redundancia y las métricas de coherencia. La evaluación demostró que GROM obtuvo buenos resultados en términos de redundancia y coherencia.

Para propósitos de la investigación reportada en esta tesis, este trabajo presenta un algoritmo de fusión de ontologías, que aunque no considera el comportamiento dinámico en la ontología, puede ser utilizado como apoyo en el método COMOTPA propuesto en esta investigación, para combinar las ontologías temporales de patrones de análisis. La diferencia que tiene este trabajo con lo que se realiza en esta investigación, es que el método COMOTPA, considera el conocimiento estático y comportamiento dinámico representado en los PAs, por lo que la descripción de los PAs compuestos, es más completo.

3.2.2.3 Research of Ontology Merging Based on Concept Similarity (Zhen-Xing & Xing-Yan, 2015)

En este trabajo, se describe una solución detallada de la fusión de ontologías basada en la similitud semántica entre conceptos y proporciona una nueva idea de la fusión de ontologías para el método manual tradicional.

En primer lugar, se analiza la ontología en un contexto formal y luego se calcula las relaciones semánticas y estructurales entre los conceptos según la ontología de WordNet. Posteriormente, se calcula la similitud semántica con la estructura gráfica compuesta de relaciones conceptuales de acuerdo con la distancia, densidad y profundidad entre conceptos para reflejar las relaciones semánticas entre conceptos.

La distancia más cercana, la profundidad más profunda y la mayor distancia entre los conceptos representan los conceptos más similares. En las mismas circunstancias, los conceptos más alejados de la raíz son semántica y estructuralmente más similares que los más cercanos a la raíz. Con la reducción y la construcción de la red conceptual se obtiene la nueva ontología objetivo que corresponde al resultado de la fusión de las dos ontologías.

Este método proporciona una nueva idea para resolver el problema tradicional de fusión manual. Sin embargo, no es perfecto y aún debe mejorarse. Para propósitos de esta investigación, con este trabajo se obtiene una idea de cómo se podrá realizar el método para combinar las ontologías temporales de los patrones de análisis. La diferencia de este trabajo con la investigación reportada en esta tesis, es que este trabajo solo se centra en la parte estructural de una ontología, es decir, el conocimiento estático. Sin embargo, no considera el comportamiento dinámico, a diferencia de esta investigación, que se considera tanto el conocimiento estático como el comportamiento dinámico.

3.2.2.4 Domain specific ontology merging using semantics (Vidyarthi, Sharma, Sharma, & Soni, 2014)

En este trabajo, se propone el algoritmo de fusión de ontologías semánticas (SOM) como nuevo enfoque de fusión de ontología basado en el análisis semántico que se ocupa de las ontologías específicas del mismo dominio.

El algoritmo propuesto es un algoritmo de fusión de ontología totalmente automatizado que utiliza la semántica de clases y la semántica de propiedades mediante Wordnet. SOM se divide en dos partes, primero fusiona las clases y luego fusiona las propiedades de las dos ontologías. En este algoritmo, el trabajo del usuario es únicamente proporcionar los dos archivos .owl como entrada y el archivo .owl combinado se generará como salida.

De forma general, el algoritmo para la fusión de la ontología semántica sigue estos pasos, primero, carga dos ontologías del mismo dominio y luego realiza el pre procesamiento utilizando la derivación y detiene la eliminación de palabras, el siguiente paso es el análisis semántico utilizando Wordnet en el que encuentra los sinónimos de las clases diferentes de ontología.

Para propósitos de la investigación reportada en esta tesis, este trabajo proporciona elementos a tomar en cuenta a la hora de combinar ontologías y que pueden ser utilizados para combinar ontologías temporales de patrones de análisis. La diferencia de este trabajo con la investigación reportada en esta tesis, es que este trabajo solo se centra en la parte estructural de una ontología, es decir, el conocimiento estático. Sin embargo, no considera el comportamiento dinámico, a diferencia de esta investigación, que se considera tanto el

conocimiento estático como el comportamiento dinámico. Además, aunque el algoritmo de fusión de ontologías es totalmente automatizado, este no asegura su consistencia.

3.2.2.5 Chronos Ed: A tool for handling temporal ontologies in Protégé (Preventis, Petrakis & Batsakis 2012)

En este trabajo de investigación se presenta un complemento para el editor de ontologías Protégé que facilita la gestión de información que evoluciona en el tiempo en una ontología temporal.

Representar la información que evoluciona en el tiempo en ontologías es un problema difícil de tratar. Las relaciones temporales son, de hecho, ternarias (es decir, las propiedades de los objetos que cambian en el tiempo implican también un valor temporal además del objeto y el sujeto) y OWL no puede manejarlas directamente. La solución estándar a este implica la aplicación de un modelo temporal (relaciones n-arias) en donde interviene el tiempo. Sin embargo, las ontologías se vuelven complicadas y difíciles de manejar por los editores de ontologías debido a la introducción de nuevas clases, lo cual implica que el usuario se familiarice con las peculiaridades de la representación temporal.

Por lo tanto, la herramienta propuesta en este trabajo (Chronos), se basa en el modelo temporal de relaciones n-arias y habilita la gestión de ontologías temporales en Protégé como si de ontologías estáticas se tratase.

Para propósitos de esta investigación, se utiliza Chronos como herramienta de apoyo en la gestión de la información de ontologías temporales (p. ej., ontologías temporales de patrones de análisis). Las características y funcionalidad de esta herramienta se describen en (Piedrola Domínguez N., 2019, pp 317).

3.2.2.6 An Approach of Refining the Merged Ontology (Zhang & Lv, 2012)

Al fusionar ontologías, existe una gran cantidad de información redundante o inconsistencias, por lo que la semántica y estructura de la ontología puede no ser la correcta, lo que conduciría a conclusiones falsas y una reducción del rendimiento.

Para resolver este problema, en este trabajo se propone un método para refinar las ontologías fusionadas, a través de la eliminación de la redundancia de la ontología resultante (

Figura 2) mejor conocida como “poda de la ontología”. Este método consiste en eliminar dos tipos de redundancia: la redundancia de instancias y la redundancia de subsunción. Posteriormente, se utiliza el algoritmo tableaux para comprobar la consistencia de la ontología resultante.

La redundancia de instancias se elimina mediante un algoritmo de detección de redundancia de instanciación y la redundancia de subsunción (que consiste en saber si un

concepto es más general que otro) se elimina mediante un algoritmo de detección de redundancia de subsunción. Para comprobar la consistencia de la ontología resultante se utiliza el algoritmo tableaux, un algoritmo de inferencia lógica que se aplica a la ontología fusionada mediante la visualización de axiomas como reglas de inferencia.

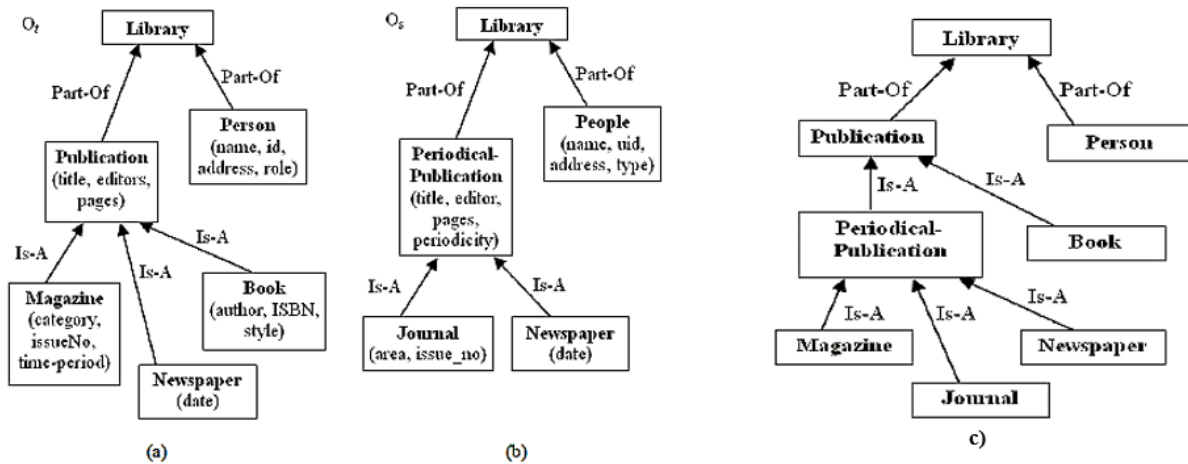


Figura 2. a) Ontología de ejemplo1 O_t , b) Ontología de ejemplo2 O_s y c) Ontología resultado de la fusión de O_t y O_s .

Para propósitos de esta investigación, este trabajo presenta elementos importantes a tomar en cuenta a la hora de combinar las ontologías temporales de patrones de análisis para que no presenten inconsistencias. La diferencia de este trabajo con la investigación reportada en esta tesis, es que este trabajo solo se centra en la parte estructural de una ontología, es decir, el conocimiento estático. Sin embargo, no considera el comportamiento dinámico, a diferencia de esta investigación, que se considera tanto el conocimiento estático como el comportamiento dinámico.

3.2.2.7 Searching for a Time Ontology for Semantic Web Applications (Fernández-López & Gómez-Pérez, 2004)

En este trabajo, los autores presentan un estudio de diferentes ontologías de tiempo, candidatas para modelar el tiempo en una aplicación web semántica del proyecto Esperonto EU llamada Fund Finder. Se analizó un conjunto de ontologías de tiempo implementadas en diferentes lenguajes legibles por máquina, considerando una serie de características típicas de modelos de tiempo, como son diferentes granularidades o diferentes zonas horarias.

De entre las ontologías de tiempo estudiadas y dadas las necesidades del modelado del tiempo en la aplicación del proyecto Esperonto, se seleccionó la ontología DAML-Time (ahora OWL-Time), debido a que provee un modelado de tiempo detallado y bien

documentado. Incluso, la documentación presenta sus definiciones en la lógica de primer orden, la cual es una formalización conocida por la mayoría de los expertos en ciencias de la computación.

Con base en este estudio, para la investigación reportada en esta tesis, se utiliza la ontología de tiempo OWL-Time para habilitar la representación de aspectos temporales en una ontología estática. Esta ontología de tiempo se utilizará para obtener una ontología temporal compuesta.

3.2.2.8 Ontology-based Reasoning in Requirements Elicitation (Dzung, 2009)

En este trabajo, se propone un método de razonamiento basado en ontologías para la obtención de requerimientos. Primero, se comienza con una estructura ontológica que contiene el conocimiento de los requerimientos funcionales y las relaciones entre ellos. Luego, utilizando el método propuesto se obtienen los requerimientos mediante la ontología de requerimientos.

El método de razonamiento basado en ontologías para la obtención de requerimientos se resume de la siguiente manera: primero se realiza un análisis de los requerimientos iniciales, los requerimientos iniciales incluyen una lista de oraciones. Posteriormente, se asigna cada oración a un nodo en la ontología de requerimientos. Después, se usan reglas predefinidas y relaciones entre nodos para razonar errores y posibles requerimientos. Finalmente, se generan preguntas para revisar los requerimientos potenciales que se agregarán al conjunto de requerimientos.

Para implementar el método de razonamiento, en este trabajo se presenta también una herramienta de verificación de requerimientos llamada REO. Esta herramienta toma como entrada, los requerimientos iniciales, la ontología de requerimientos y un conjunto de reglas. Luego aplica el razonamiento para encontrar errores en los requisitos iniciales y mostrar los resultados a los analistas. Con estos resultados, los analistas pueden construir preguntas para los clientes y de esta manera obtener un conjunto de requerimientos más completo.

El método y la herramienta propuesta en este trabajo, pueden ayudar a los analistas a inferir los requisitos obligatorios de los requerimientos existentes. Por lo tanto, pueden ayudar en la obtención de requerimientos de forma rápida, correcta y eficiente. La diferencia de este trabajo con el trabajo reportado en esta tesis, es que aunque los requerimientos son almacenados en una ontología de requerimientos, no presentan una estructura clara de los elementos que intervienen en cada requerimiento (clases, propiedades de objetos, propiedades de datos, restricciones). Además, no se considera el comportamiento dinámico que interviene en cada requerimiento funcional, a diferencia del trabajo reportado en esta tesis que se presenta una estructura clara, además de considerar el conocimiento estático y comportamiento dinámico especificado en los requerimientos funcionales. Sin embargo, este

trabajo puede ser de gran utilidad como apoyo en la fase de “Refinamiento del patrón de análisis” de la metodología MECOTPA para obtener un conjunto de requerimientos más completo.

3.2.2.9 Using Domain Ontology as Domain Knowledge for Requirements Elicitation (Kaiya & Saeki, 2006)

En este trabajo, se propone el método ORE (*Ontology based Requirements Elicitation*) para obtener requerimientos a través de una ontología de dominio que se utiliza como conocimiento del dominio.

En el método ORE, una ontología principal juega un papel en el dominio semántico que da sentido a las declaraciones de requerimientos mediante el uso de una función semántica. Mediante el uso de reglas de inferencia sobre la ontología y cuatro características de calidad (correctez, completitud, consistencia y ambigüedad) sobre la función semántica, un analista puede saber qué requisitos deben agregarse para mejorar la integridad de la versión actual de los requisitos iniciales y/o qué requisitos deben eliminarse de la versión actual para mantener la coherencia.

El procedimiento del método ORE es el siguiente: 1. El analista obtiene de sus clientes/usuarios una lista de requerimientos iniciales, 2. Para cada elemento de los requerimientos, el analista realiza un mapeo del elemento a conceptos en la ontología, 3. El analista evalúa la calidad de la lista de requerimientos calculando las cuatro características de calidad, los requerimientos se actualizarán de acuerdo al valor de cada medida de calidad y 4. El analista actualiza la lista de requerimientos de acuerdo los resultados de la evaluación de la actividad 3. En este procedimiento, se iteran las actividades 3 y 4.

La diferencia que tiene este trabajo con la investigación reportada en esta tesis, es que el enfoque está dirigido al proceso de elicitación de requerimientos. Es decir, ayudar a un analista de requerimientos a ampliar los requerimientos sistemáticamente tomando en cuenta los aspectos semánticos de un conjunto de requerimientos iniciales. Al igual que en la investigación reportada en esta tesis, se utiliza una ontología para almacenar el conocimiento de un dominio específico. Sin embargo, no se toma en cuenta la representación formal del conocimiento estático y el comportamiento dinámico de los requerimientos. A pesar de ello, este es uno de los principales trabajos que proponen el enfoque de utilizar una ontología para almacenar el conocimiento de un dominio específico, que es lo que se realiza en nuestro trabajo de investigación, en el cual se utiliza una ontología para representar formalmente un PA que después pueden ser combinados con otros PAs para obtener PAs compuestos. Por lo tanto, este trabajo es de gran utilidad para entender el problema en la representación del conocimiento mediante ontologías.

3.2.2.10 Real-time Specification Patterns (Konrad & Cheng, 2005)

En este trabajo, se presenta una plantilla para crear patrones de especificación en tiempo real de sistemas embebidos, para facilitar la especificación de propiedades críticas. También, se presenta una colección de patrones de especificación en tiempo real que pueden usarse para especificar propiedades en tiempo real de sistemas embebidos

La creación de los patrones de especificación en tiempo real se realiza en términos de tres lógicas temporales: Lógica temporal métrica (MTL, *Metric Temporal Logic*), Lógica de árbol computacional temporizada (TCTL, *Timed Computational Tree Logic*) y Lógica de intervalo gráfico en tiempo real (RTGIL, *Real-Time Graphical Interval Logic*). Este enfoque múltiple permite a los especificadores utilizar la herramienta de análisis y el lenguaje de especificación más adecuado para el modelar propiedades de un sistema que pueden ser analizados usando técnicas de verificación de modelos.

Para facilitar el uso de los patrones de especificación creados con la plantilla, se describe una gramática estructurada del inglés que facilita la comprensión del significado de una propiedad lógica temporal, haciendo que el enfoque sea más accesible para los usuarios no expertos en métodos formales. La gramática que se proporciona, puede usarse para crear representaciones de lenguaje natural para propiedades cualitativas y en tiempo real de un sistema embebido.

Este trabajo, está destinado a proporcionar estrategias sobre cómo especificar propiedades en tiempo real en un lenguaje de especificación formal, donde las propiedades son susceptibles de análisis automatizado. La diferencia que tiene este trabajo con la investigación reportada en esta tesis, es que aunque se propone el uso de lenguajes formales para representar el conocimiento, no se enfoca en la representación formal del conocimiento representado en los requerimientos de los PAs. Además, en este trabajo no se utilizan ontologías como medio para almacenar el conocimiento. Sin embargo, al igual que en la investigación reportada en esta tesis, en este trabajo se propone el uso de lenguajes formales para representar el conocimiento, lo que lo convierte en uno de los principales trabajos que proponen el uso de lenguajes formales para representar formalmente la información basada en el tiempo.

En la Tabla 2, se presenta una comparativa de los trabajos relacionados descritos en esta sección. Para un mejor entendimiento, a continuación, se describen las columnas y abreviaturas utilizadas en la tabla comparativa.

Columnas: Trabajo (1ra columna): Referencia bibliográfica del trabajo relacionado.

- PAs (2da columna): Patrones de análisis que se utilizan en el trabajo, en el caso de que no se utilice ningún patrón de análisis se representa con el símbolo ✕.

- ¿Combina PAs? (3ra columna): Valor booleano que indica si se combinan PAs o no en el trabajo analizado, el símbolo ✘ es igual a false y el símbolo ✔ es igual a true.
- Ontologías (4ta columna): Se integra de las sub columnas *Estáticas* y *Temporales* que corresponden a las ontologías estáticas o temporales, respectivamente, utilizadas en el trabajo. En el caso de que no se utilice ninguna ontología (estática o temporal), se representa con el símbolo ✘.
- ¿Combina ontologías? (5ta columna): Se integra de las sub columnas *Estáticas* y *Temporales*, valores booleanos que indican si se combinan ontologías o no en el trabajo analizado, el símbolo ✘ es igual a false y el símbolo ✔ es igual a true.
- Lenguaje utilizado (6ta columna): Se integra de las sub columnas *Semi-formal* y *Formal*, que corresponden a los lenguajes utilizados en el trabajo. En el caso de que no se utilice ningún lenguaje (semi-formal o formal) en el trabajo, se representa con el símbolo ✘.
- Razonamiento (7ma columna): Herramientas utilizadas para el razonamiento y verificación para asegurar la satisfacibilidad de propiedades en estructuras.
- Producto resultado (8va columna): Producto (s) obtenido (s) en el trabajo analizado, tales como herramientas, métodos, algoritmos, entre otros).

Abreviaturas:

- PA: Patrón de análisis y PAs para el plural: Patrones de Análisis.
- PAC: Patrón de Análisis de Cuenta.
- PAER: Patrón de Análisis de Entidades Reutilizables.
- PAI: Patrón de Análisis para Inventarios.
- OPAE: Ontología del Patrón de Análisis Estable.
- OTPAE: Ontología Temporal del Patrón de Análisis Estable.
- NL (*Natural Language*): Lenguaje natural.
- OWL-Time: Ontología de conceptos temporales para expresar hechos sobre roles topológicos (ordenamiento) entre instantes e intervalos, junto con información sobre duraciones y sobre posición temporal, incluida información de fecha y hora.
- Pellet: razonador OWL-DL basado en Java y de código abierto. Es el primer razonador que soporta toda la expresividad de la DL *SHOIN(D)*, la explicación de errores y reglas SWRL.

- nuXmv (*New eXtended Symbolic Model Verifier*): verificador de modelos simbólico para el análisis de sistemas síncronos de estado finito e infinito.
- MC (*Model Checking*): Verificación de modelos, es un método automático de verificación de un sistema formal, en la mayoría de las ocasiones derivado del hardware o del software de un sistema informático.
- SUMO (*Suggested Upper Merged Ontology*): La Ontología Combinada Superior Sugerida SUMO y sus ontologías de dominio forman la ontología pública formal más grande que existe en la actualidad. Se están utilizando para investigación y aplicaciones en búsqueda, lingüística y razonamiento. Es la única ontología formal que se ha asignado a todo el léxico de WordNet.
- OWL (*Ontology Web Language*): Lenguaje de Ontologías Web, tiene como objetivo facilitar un modelo de marcado (mediante etiquetas) construido sobre el Marco de Descripción de Recursos (RDF, *Resource Description Framework*) y codificado en el Lenguaje de Marcas Extensible (XML, *eXtensible Markup Language*).
- OWL-DL (*Ontology Web Language – Description Logic*): Lenguaje de Ontologías Web y Lógica Descriptiva, incluye todas las construcciones del lenguaje OWL, además de incluir restricciones expresadas en lógica descriptiva.
- SWRL (*Semantic Web Rule Language*): Lenguaje de Reglas para la Web Semántica para expresar reglas, así como también lógica mediante la combinación de OWL-DL y un subconjunto del Lenguaje de Marcado de Reglas (RuleML, *Rule Markup Language*).
- TDL ALC_{LTL}: Lógica de Descripción Temporal que combina la DL ALC y la TL LTL en un formalismo para la representación y razonamiento de conceptos dependientes del tiempo (p. ej., comportamientos dinámicos), basado en la integración de la DL ALC y la TL LTL.
- TL LTL (*Linear Temporal Logic*): Lógica Temporal Lineal, un formalismo utilizado para especificar propiedades (p. ej., comportamientos dinámicos) a verificar en trayectorias de estado de un modelo (p. ej., una estructura Kripke) por herramientas MC (p. ej., nuXmv). Para modelar el tiempo, tiene un conjunto numerable de instantes organizados linealmente pero no puede representar el tiempo transcurrido (p. ej., tiempo de duración exacta de un evento).

Tabla 2. Tabla comparativa de trabajos relacionados.

Trabajo	PAs	¿Combina PAs?	Ontologías		¿Combina ontologías?		Lenguaje utilizado		Razonamiento	Producto Resultado
			Estáticas	Temporales	Estáticas	Temporales	Semi-formal	Formal		
(Peixoto et al., 2013)	Urban Street Mesh	✗	✗	✗	✗	✗	NL	RDF	✗	Editor de metadatos para PAs basados en DC2AP
(Yuan & Fernandez, 2011)	Catálogo, Carrito de compras, Orden y envío, Factura, Administrador de existencias	✓	✗	✗	✗	✗	Diagramas UML de clases y de secuencias	✗	✗	SAP compuesto para aplicaciones de comercio electrónico negocio-cliente
(Jiang & Fernandez, 2009)	Boleto de viaje, Asignación de asiento, Colección de asientos, Ruta de vuelo	✓	✗	✗	✗	✗	Diagramas UML de clases, de secuencias y de estados.	✗	✗	SAP compuesto para la reservación de vuelos
(Fernandez & Yuan, 2009)	Creación de factura, Pago de factura	✓	✗	✗	✗	✗	Diagramas UML de clases, de secuencias	✗	✗	PA para el procesamiento de facturas
(Harmelen, Frank Van, Vladimir Lifschitz, 2008)	✗	✗	✗	✗	✗	✗	NL	DLs, TLs, TDLS, OWL	Técnicas MC para TDLS	Representación del conocimiento mediante DLs, TLs y TDLs.
(Narula GS, Wason R, Jain V, 2018)	✗	✗	Tienda de libros	✗	✓	✗	NL	OWL	✗	Comparación de herramientas para mapeo y fusión de ontologías.
(Mahfoudh et al., 2016)	✗	✗	Carros europeos, Conferencias, Google, Web	✗	✓	✗	NL	OWL	Condiciones de Aplicación Negativa (NAC)	- Benchmark para la fusión de ontologías. - La herramienta GROM para la fusión de ontologías.
(Zhen-Xing & Xing-Yan, 2015)	✗	✗	Persona, Animal	✗	✓	✗	NL	OWL	✗	Método para la fusión de ontologías basada en la similitud semántica entre conceptos.
(Vidyarthi et al., 2014)	✗	✗	Curso, Trabajo de investigación	✗	✓	✗	NL	OWL	✗	Algoritmo SOM para la fusión de ontologías semánticas
(Preventis, Petrakis & Batsakis 2012)	✗	✗	✗	OWL-Time	✗	✗	NL	OWL-DL, Reglas SWRL, OWL-Time	Pellet	Plugin Chronos para la creación de ontologías temporales
(Zhang & Lv, 2012)	✗	✗	Librería	✗	✓	✗	NL	OWL	Algoritmo de verificación de consistencia	Método para refinar ontologías fusionadas.
(Fernández-López & Gómez-Pérez, 2004)	✗	✗	✗	OWL-Time, SUMO, AKT, Kestrel	✗	✗	NL	OWL	✗	Estado del arte de diferentes ontologías de tiempo.
(Dzung, 2009)	✗	✗	SMS Librería	✗	✗	✗	NL	OWL	Reglas	-Método de razonamiento basado en ontologías. -Herramienta de verificación de requerimientos REO
(Kaiya & Saeki, 2006)	✗	✗	Librería	✗	✗	✗	NL	OWL, Prolog	Reglas de inferencia	-Método ORE para obtener requerimientos.
(Konrad & Cheng, 2005)	✗	✗	✗	✗	✗	✗	NL	MTL, TCTL, RTGIL	Técnicas MC	-Plantilla para crear patrones de especificación. -Gramática para facilitar el uso de patrones de especificación.
Esta investigación	Inventario, Entidades reutilizables, Cuentas.	✓	OPAE del: PAER, PAI y PAC	OTPAE del: PAER, PAI y PAC	✓	✓	NL	TDL ALC _{LTL} , TL LTL, OWL-DL	Pellet NuXmv	- Método para combinar ontologías temporales de patrones de análisis. - OTPAE del PAI

En los trabajos relacionados (Tabla 2) se observa que ya se han realizado combinaciones de patrones de análisis en trabajos anteriores. Sin embargo, las combinaciones se han realizado utilizando diagramas UML, es decir, no se han utilizado lenguajes formales para la descripción y/o representación de los PAs. Por ejemplo, en (Fernandez & Yuan, 2009; Jiang & Fernandez, 2009; Yuan & Fernandez, 2011) se han combinado patrones de análisis semánticos (SAPs) para construir patrones de análisis semánticos compuestos. Sin embargo, estos PAs carecen de una representación formal debido a que se representan mediante diagramas de clases, de secuencias y de estados.

Por otra parte, en trabajos anteriores como en (Mahfoudh et al., 2016; Vidyarthi et al., 2014; Zhang & Lv, 2012) también se han realizado combinaciones de ontologías y se han propuesto algoritmos y métodos para facilitar la combinación. Sin embargo, las ontologías combinadas son estáticas, es decir, no contienen una especificación formal del comportamiento dinámico (como sí las tienen las ontologías creadas con MECOTPA). A diferencia de estos trabajos, con el método para combinar ontologías temporales de PAs (COMOTPA) propuesto en esta investigación, las ontologías que se combinan, representan formalmente el conocimiento estático utilizando la lógica descriptiva ALC y el comportamiento dinámico utilizando la lógica de descripción temporal ALC_{LTL} . Por lo que la descripción de los PAs es más completa.

También, en (Mahfoudh et al., 2016; Narula GS, Wason R, Jain V, 2018) se han propuesto herramientas que pueden ayudar a combinar ontologías. Sin embargo, en estas herramientas no se toma en cuenta el comportamiento dinámico de las ontologías. Para la investigación reportada en esta tesis, las ontologías representan formalmente el conocimiento estático y el comportamiento dinámico de los PAs. Para propósitos de esta investigación, se utilizará el plugin Chronos ya que será de gran utilidad para gestionar la información temporal durante la combinación de ontologías temporales de patrones de análisis. Además, se utilizará el razonador Pellet para verificar la consistencia de la ontología temporal.

Con base en el estudio y análisis de estos trabajos, se conocieron los avances que se han realizado con respecto al tema de combinación de PAs y ontologías y la importancia de la aportación del trabajo de investigación reportado en esta tesis. Además, se puede concluir que actualmente, no existe un método o procedimiento que diga cómo combinar PAs representados mediante ontologías temporales, que especifiquen formalmente el comportamiento dinámico presentado en dichos PAs.

De los trabajos mencionados, ningún patrón de análisis utiliza un lenguaje formal para representar el conocimiento estático y mucho menos el comportamiento dinámico, en vez de ello, se han utilizado lenguajes semi formales como diagramas de clases, de secuencias, de actividades, entre otros. Aunque ya se han realizado combinaciones de PAs, las descripciones de éstos son aún deficientes, lo cual dificulta su reuso. Por otro lado, la combinación de ontologías es un tema en el que se ha trabajado mucho. Sin embargo, muy poco trabajos se han dedicado en la representación del comportamiento dinámico de las ontologías, por el contrario, la mayoría de los trabajos se han enfocado en la alineación y combinación de ontologías tomando en cuenta solamente la parte estructural, es decir, el conocimiento

estático. Uno de los pocos trabajos que se han preocupado en la representación del comportamiento dinámico, es el realizado en (Preventis et al., 2014) donde se presenta Chronos, una herramienta que permite construir una ontología temporal que almacena aspectos estáticos, así como la integración de aspectos dinámicos mediante secuencias de interpretaciones.

Por lo tanto, estos trabajos sirvieron como referencia para crear el método para la Combinación de Ontologías Temporales de Patrones de Análisis (COMOTPA) (ver *Capítulo IV. Método para la Combinación de Ontologías Temporales de Patrones de Análisis*), el cual con ayuda del plugin Chronos en Protégé y el razonador Pellet permite combinar patrones de análisis representados mediante ontologías temporales consistentes, en donde las especificaciones de comportamiento son satisfacibles, mediante la consideración de fórmulas y trayectorias de la TL LTL para la verificación de satisfacibilidad mediante el verificador de modelos nuXmv.

En este capítulo, se realizó una comparativa de los antecedentes y trabajos relacionados a esta investigación, se presentaron las diferencias y similitudes con el presente trabajo de investigación. Con base en esta información, la aportación y la importancia que tiene esta investigación con respecto a los demás se define con mayor claridad.

Capítulo IV. Método para la Combinación de Ontologías Temporales de Patrones de Análisis

En este capítulo, se describe el método para Combinar Ontologías Temporales de Patrones de Análisis (COMOTPA), un método que permite combinar ontologías temporales de patrones de análisis que son construidos con la metodología MECOTPA desarrollada en (Piedrola Domínguez N., 2019).

El método para la Combinación de Ontologías Temporales de Patrones de Análisis (COMOTPA) se describe con la finalidad de facilitar a los ingenieros de software la combinación de dos o más ontologías temporales de patrones de análisis que incluyen en su definición aspectos estáticos y dinámicos utilizando la lógica descriptiva y la lógica temporal, respectivamente. Este método se compone de un conjunto de actividades orientadas a apoyar en la obtención de una OTPAE compuesta mediante la combinación de dos ontologías temporales de patrones de análisis cualquiera, construidas con la metodología MECOTPA.

Con la finalidad de conocer cómo han evolucionado los trabajos en el CENIDET con respecto al estudio de ontologías de patrones de análisis y mostrar el nivel actual de la presente investigación, en la Tabla 3 se presenta la evolución de estos trabajos hasta llegar al método COMOTPA.

Tabla 3. Evolución del trabajo de investigación actual.

Trabajo	Objetivo
(López-Ramírez, 2015) MECOPA	Construcción de Ontologías de Patrones de Análisis.
(Juárez Hernández, 2016) MECOPA + MEVASE	Formalización de Patrones de Análisis con la Lógica Descriptiva ALC (Estructura estática).
(Piedrola Domínguez N., 2019) MECOPA + MEVASE + FOCOPAT = MECOTPA	Formalización Complementaria de Patrones de Análisis con aspectos de Temporalidad utilizando la Lógica de descripción temporal ALC _{LTL} (Estructura estática y Comportamiento dinámico).
(Esta investigación) COMOTPA	Combinación de Ontologías Temporales de Patrones de Análisis.

Como se muestra en la Tabla 1, primero en (López-Ramírez, 2015) se describió una metodología para construir ontologías de patrones de análisis llamada MECOPA. Posteriormente, en (Juárez Hernández, 2016) se le agregó a MECOPA una nueva fase denominada MEVASE para validar la consistencia semántica de los patrones de análisis, en el cual se formalizan los PAs utilizando la lógica descriptiva ALC. Sin embargo, hasta ese momento con MECOPA solo era posible formalizar la estructura estática. Por tal motivo, en (Piedrola Domínguez N., 2019) se actualizó la metodología MECOPA a MECOTPA en la cual se le agregó una nueva fase llamada FOCOPAT, para realizar la formalización complementaria de los PAs con aspectos de temporalidad, se utiliza la lógica de descripción

temporal ALC_{LTL} para representar formalmente la estructura estática y el comportamiento dinámico de los PAs. El trabajo desarrollado en (Piedrola Domínguez N., 2019) es un antecedente directo y el motivo por el cual surge el método COMOTPA descrito en este trabajo de tesis.

4.1. Requisitos para la implementación del método COMOTPA

Para implementar el método COMOTPA, es necesario contar con algunos elementos de entrada (descritos en la sección 4.1.1. *Requisitos de entrada*), con la finalidad de realizar la combinación de OTPAEs correctamente. Además, también es necesario conocer los elementos de salida (descritos en la sección 4.1.2. *Requisitos de salida*) que se deberán obtener una vez finalizado la implementación del método, con la finalidad de obtener el resultado correcto. A continuación, se presentan los requisitos de entrada y salida del método COMOTPA.

4.1.1. Requisitos de entrada

Para iniciar el proceso de combinación, primero es necesario contar con los siguientes elementos:

1. Dos ontologías temporales (OTPAEs) creadas realizando las actividades de la metodología MECOTPA, que serán los elementos principales de entrada para el método COMOTPA y donde cada OTPAE deberá contar con los siguientes elementos:
 - a. Clases (estáticas y temporales)
 - b. Propiedades de objetos (estáticas y temporales)
 - c. Propiedades de datos (estáticas y temporales)
 - d. Relaciones binarias y n-arias.
 - e. Restricciones existenciales y universales
 - f. Características (funcional e inversa funcional) de las propiedades de objetos.
2. El conjunto de fórmulas ALC_{LTL} correspondientes a la especificación formal del comportamiento de los requerimientos funcionales de las dos OTPAEs de estudio.
3. La ontología de tiempo OWL-Time.

4.1.2. Requisitos de salida

Una vez terminadas las actividades del método COMOTPA, se deben obtener los siguientes elementos:

1. Una OTPAE válida y consistente, resultado de la combinación de las dos ontologías temporales de entrada.
2. Las rutas de computación o trayectorias de estados correspondientes a las secuencias de interpretaciones.

3. El nuevo conjunto de fórmulas ALC_{LTL} satisfacibles de acuerdo con su respectiva ruta de computación, resultado de la combinación de las fórmulas ALC_{LTL} de las dos ontologías temporales de entrada.

4.2. Fases del método COMOTPA

El método COMOTPA es un proceso constituido por ocho fases, que a su vez están constituidas por actividades, que permiten combinar ontologías temporales (OTPAEs) de patrones de análisis. En la Figura 3 se presentan en un nivel alto de abstracción, las ocho fases del método COMOTPA.

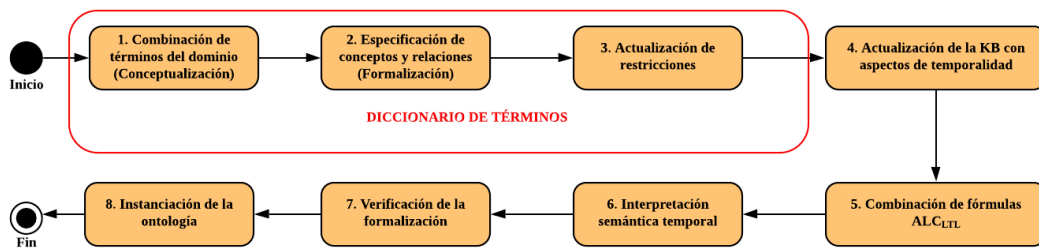


Figura 3. Fases del método COMOTPA.

A continuación, en la Figura 4 se presenta el proceso del método COMOTPA con más detalle, pues se representan gráficamente las entradas y salidas del método y a cada fase y en la Figura 5 se describen a más detalle las fases del método, mostrando las actividades que constituyen a cada fase y sus entradas y salidas.

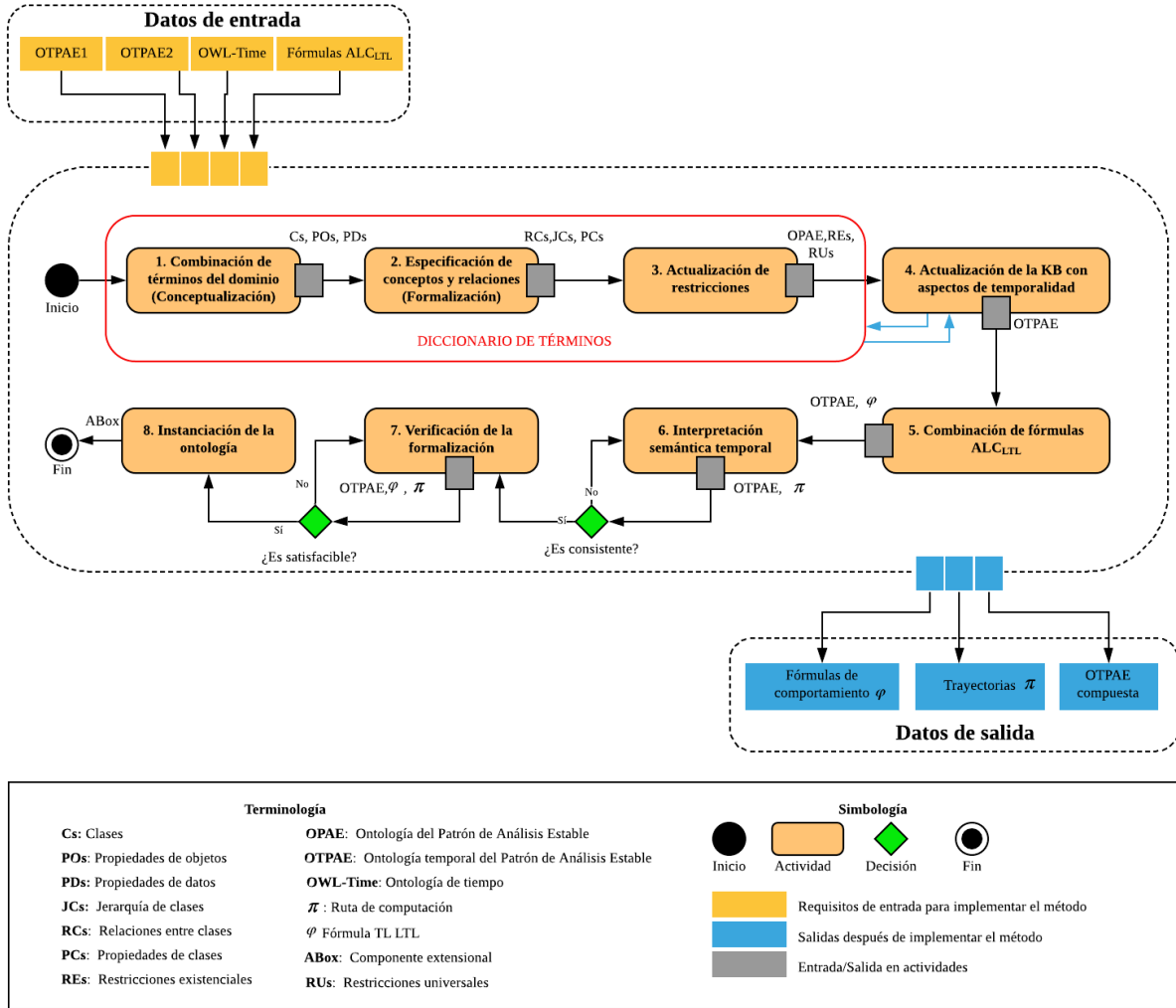


Figura 4. Entradas y salidas del método COMOTPA.

Como se visualiza en la Figura 4, el método COMOTPA necesita como datos de entrada los elementos descritos en la sección 4.1.1. *Requisitos de entrada*. Al finalizar las actividades de cada fase se obtienen productos que son utilizados en las fases posteriores para finalmente obtener los elementos de salida descritos en la sección 4.1.2. *Requisitos de salida*. Para una mejor comprensión, en la Figura 4 se presenta el significado de la terminología utilizada para representar los datos de entrada y de salida. Además, se utilizan tres colores (amarillo, azul y gris) para diferenciar los datos de entrada y salida del método COMOTPA y los datos de entrada y salida en cada fase.

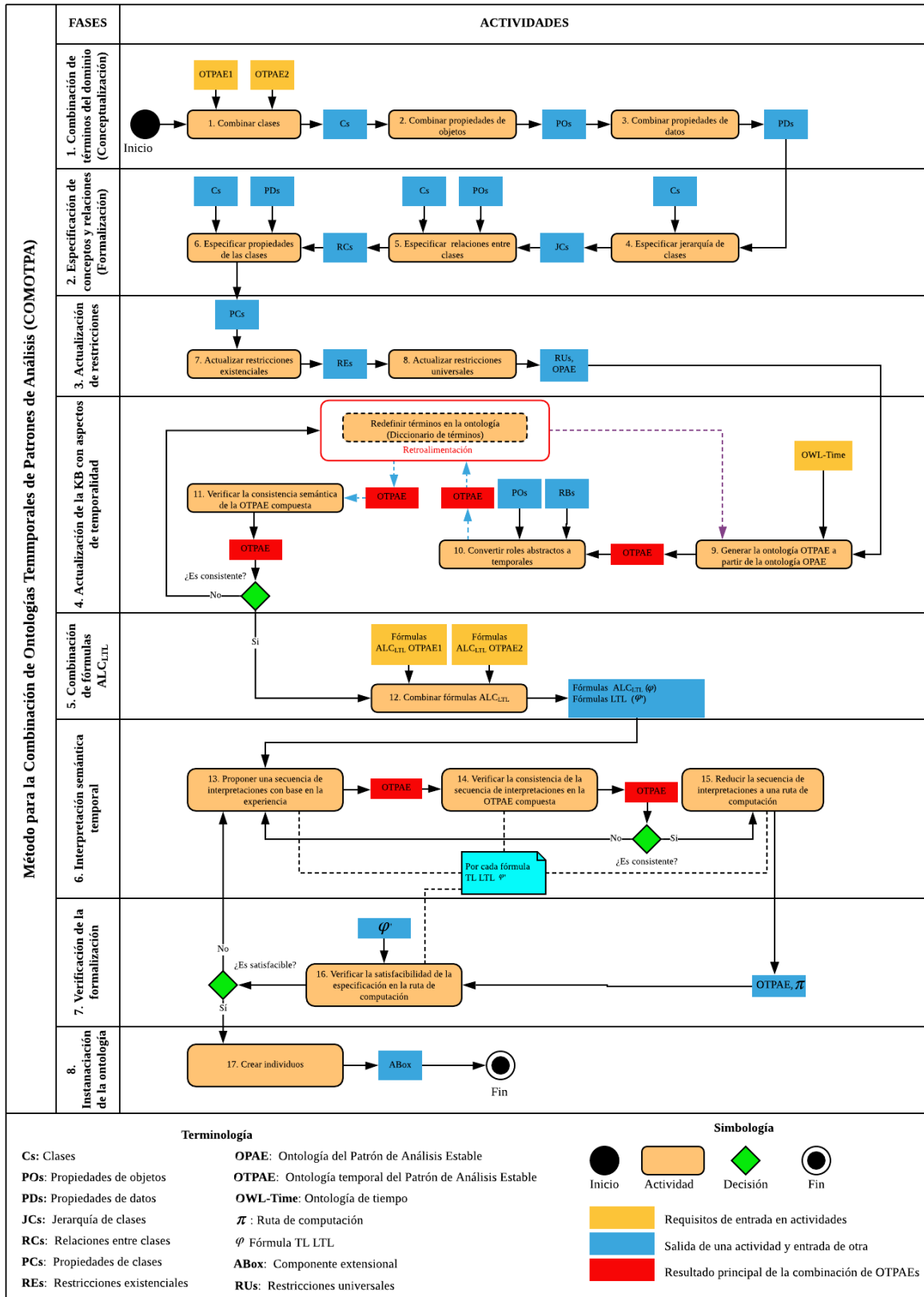


Figura 5. Actividades principales en las fases del método COMOTPA.

Para una mejor comprensión, a continuación, se describen a mayor detalle las fases del método COMOTPA y las actividades que se deben realizar en cada fase.

4.2.1. Combinación de términos del dominio (Conceptualización)

Una OTPAE se compone principalmente de clases, propiedades de objetos y propiedades de datos (estáticos y temporales) (Figura 6) que se utilizan para representar el conocimiento estático y dinámico.

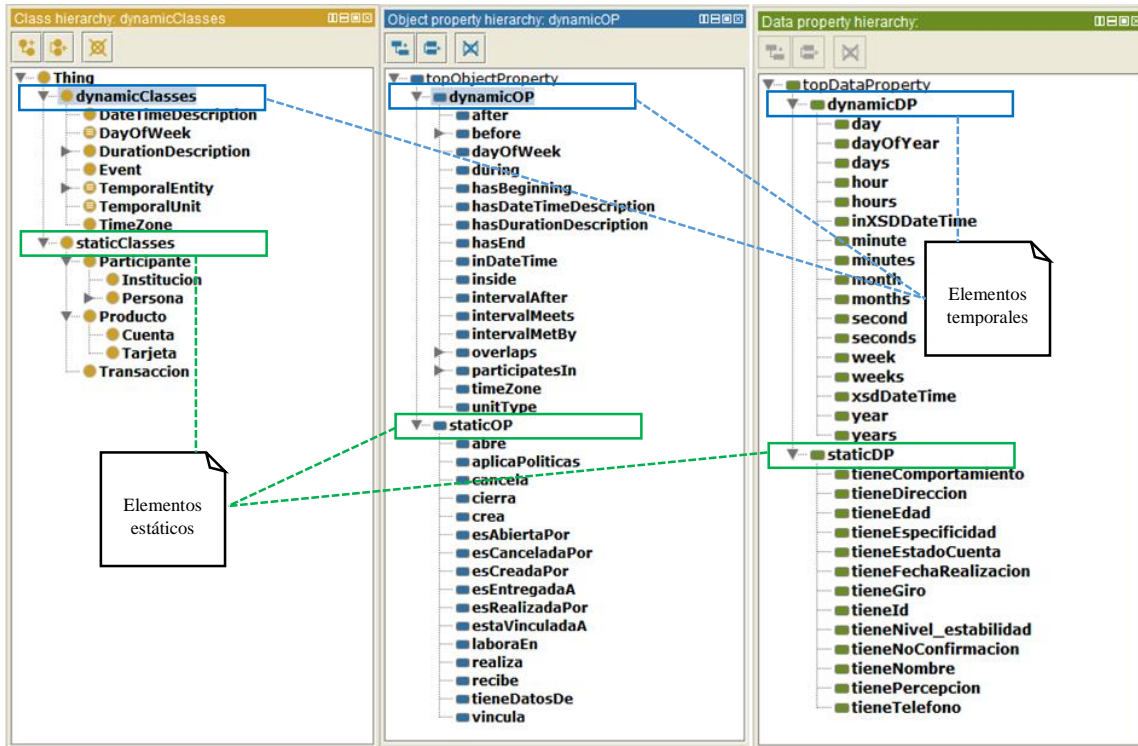


Figura 6. Ejemplo de clasificación de entidades estáticas y dinámicas de una OTPAE.

Para realizar la combinación, en esta fase se requieren solamente los elementos estáticos de las dos ontologías temporales. Es decir, clases estáticas, propiedades de objetos estáticos y propiedades de datos estáticos. La razón de utilizar solamente los elementos estáticos se debe a que son estos los elementos que pueden variar en las dos OTPAEs durante la combinación (p. ej. en el dominio de inventarios se utilizan conceptos como cuenta, artículo. Sin embargo, en el dominio de entidades reutilizables se utilizan conceptos como entidad, reservación, entre otras. Es decir, los elementos estáticos se definen de acuerdo al dominio de aplicación y es por ello que son estos los elementos que pueden cambiar en cada OTPAE). Los elementos temporales (clases, propiedades de objetos y propiedades de datos) son los mismos en las dos OTPAEs debido a que de acuerdo con la metodología MECOTPA (Piedrola, 2019), para convertir una OPAE a una OTPAE, ésta se combina con la ontología OWL-Time, que provee el plugin Chronos en Protegé, con expresividad *ALCROIN(D)*)

(lógica básica ALC extendida con roles transitivos R, nominales O, roles inversos I, restricciones numéricas N y dominios concretos (D)). Así se agregan a la OPAE conceptos temporales (que asocian siempre los mismos conceptos temporales a las clases estáticas, propiedades de objetos estáticas y propiedades de datos estáticas) que permiten representar el conocimiento dinámico convirtiendo a la ontología (OPAE) en una ontología temporal, es decir, una OTPAE.

Con la finalidad de obtener el nuevo conjunto de términos que representarán el nuevo dominio, en esta fase se deben realizar las siguientes actividades: combinar clases, combinar propiedades de objetos y combinar propiedades de datos. Al término de esta fase, se deberá contar con el nuevo conjunto de clases, propiedades de objetos y propiedades de datos que representarán el nuevo dominio de la OTPAE del patrón de análisis compuesto. A continuación, se describe cada una de las actividades de esta fase.

4.2.1.1. Combinar clases

En la jerarquía de clases de una ontología temporal (Figura 6), se tienen dos subclases que pertenecen a la clase Thing: la clase staticClasses que contiene las clases utilizadas para representar el conocimiento estático y; la clase dynamicClasses que contiene las clases utilizadas para representar el conocimiento dinámico. En esta actividad, se combinan las clases contenidas en la clase staticClasses de las dos OTPAEs de entrada.

El proceso de combinación de clases consiste básicamente en comparar todos los nombres las clases estáticas de la primera ontología (OTPAE1), con todos los nombres de las clases estáticas de la segunda ontología (OTPAE2), con la finalidad de identificar aquellas clases que tengan nombres iguales o sinónimos, que puedan combinarse para obtener el conjunto de clases del nuevo dominio. Durante la comparación entre clases nos podemos encontrar con tres situaciones. A continuación, se describen las posibles situaciones y las acciones que se deberán realizar ante cada una.

1. **Los nombres de las dos clases comparadas existen en ambas ontologías (iguales).** Cuando existan dos clases que se repitan en las dos ontologías. Es decir, cuando el nombre de las dos clases es el mismo en las dos ontologías, se creará una sola clase en la ontología combinada.
2. **Los nombres de las dos clases comparadas son sinónimos (diferentes, pero con el mismo significado).** Cuando existan dos clases con nombres sinónimos, se utilizará un solo nombre que se adapte y represente a las dos clases de las dos ontologías, de manera correcta. Se podrá utilizar el nombre de cualquiera de las dos clases sinónimas siempre y cuando muestre claramente lo que se quiere representar.
3. **Los nombres de las dos clases comparadas son diferentes.** Cuando las clases sean diferentes, es decir, cuando el nombre de la clase de la primera ontología no exista en la segunda ontología y/o viceversa, quedarán intactas (no se les cambiará el nombre) y

se tomarán los dos nombres de clases para crearlas en la ontología compuesta. Cabe mencionar, que posteriormente cada clase deberá ubicarse en el nivel correcto de la jerarquía de clases de la ontología compuesta.

A continuación, en la Figura 7 se presenta un ejemplo de la relación que hay entre el conjunto de clases estáticas de la ontología temporal del PAER con el conjunto de clases estáticas de la ontología temporal del PAI. Las flechas continuas significan que son clases diferentes, las flechas segmentadas significan que son clases sinónimas y las flechas segmentadas con puntos significan que son clases iguales, es decir, utilizan el mismo nombre en las dos ontologías.

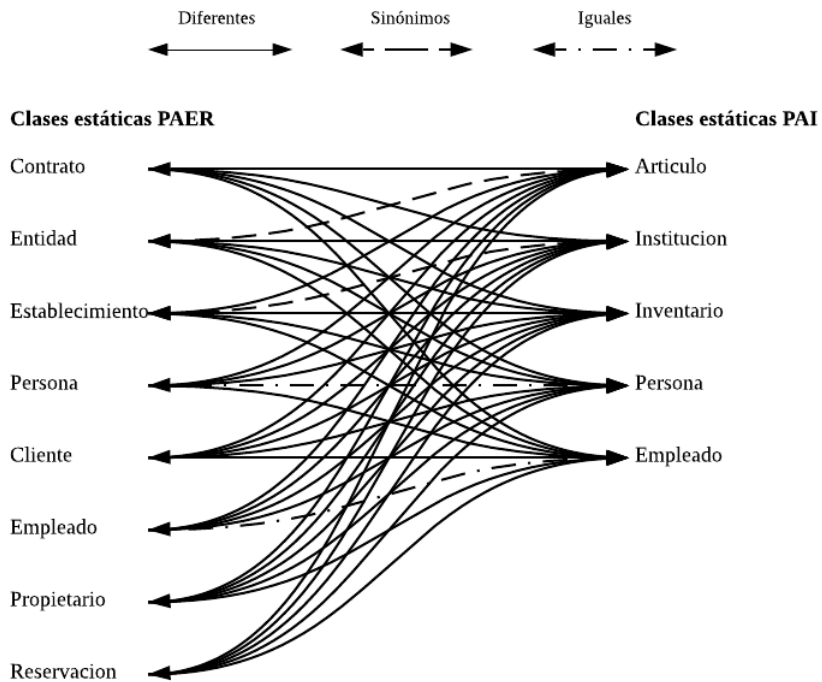


Figura 7. Relación de similitud entre las clases estáticas de la OTPAE del PAER con la OTPAE del PAI.

El criterio para decidir cuál es el conjunto de clases más adecuado, será de acuerdo al criterio del ontólogo o de la persona encargada de realizar la combinación. Por lo tanto, para realizar la combinación correctamente, es importante conocer el contexto de uso de cada una de las clases en cada OTPAE.

Con base en el tipo de relación que existe entre los nombres de las clases de las dos ontologías (diferentes, sinónimos, iguales), se obtendrá el conjunto de clases que representará el nuevo dominio. Como ejemplo, en la Tabla 4 se muestran las clases estáticas que pertenecen a la ontología temporal del PAER, las clases estáticas que pertenecen a la ontología temporal del PAI y las clases que pertenecerán a la ontología temporal del patrón de

análisis compuesto (OTPAE del PAIyER) y que son el resultado de la combinación de las clases estáticas de las dos OTPAEs.

Tabla 4. Conjunto de clases estáticas de las OTPAE de de entrada y salida.

Clases estáticas del PAER	Clases estáticas del PAI	Clases estáticas del PAIyER
Contrato		Contrato
Entidad	Articulo	Entidad
Establecimiento	Institucion	Institucion
Persona	Persona	Persona
Cliente		Cliente
Empleado	Empleado	Empleado
Propietario		Propietario
Reservacion		Reservación
	Inventario	Inventario

Como se puede observar en la Tabla 4, en los nombres de las clases s sinónimas fueron en donde se realizó una verificación para elegir cuál nombre tomar para la nueva OTPAE compuesta. Por ejemplo, para las clases Establecimiento (clase del PAER) e Institucion (clase del PAI), se tomó el nombre de la clase Institucion como nueva clase del patrón de análisis PAIyER. Los nombres de las clases que no sean sinónimas, se tomarán como nuevas clases del patrón de análisis compuesto.

Al finalizar esta actividad, se debe contar con el conjunto de clases estáticas que formarán parte de la nueva OTPAE del patrón de análisis compuesto.

4.2.1.2. Combinar propiedades de objetos

En la jerarquía de propiedades de objetos de la ontología temporal (Figura 6), se visualizan dos subpropiedades que pertenecen a la propiedad topObjectProperty: la subpropiedad staticOP que contiene las propiedades de objetos utilizadas para representar el conocimiento estático y; la subpropiedad dynamicOP que contiene las propiedades de objetos utilizadas para representar el conocimiento dinámico. Para esta actividad, se combinan las propiedades de objetos de la subpropiedad staticOP de las dos OTPAEs de entrada.

El proceso de combinación de propiedades de objetos es similar a la combinación de clases, consiste en comparar los nombres de las propiedades de objetos estáticas de la primera ontología (OTPAE1), con los nombres de las propiedades de objetos estáticas de la segunda ontología (OTPAE2), con la finalidad de identificar aquellas propiedades que puedan combinarse para obtener el conjunto de propiedades de objetos del nuevo dominio.

Durante la comparación entre propiedades de objetos nos podemos encontrar con tres situaciones. A continuación, se describen las posibles situaciones y las acciones que se deben realizar ante cada una.

1. **Los nombres de las dos propiedades de objetos comparadas existen en ambas ontologías (iguales).** Cuando existan dos propiedades de objetos que tengan el mismo nombre en las dos ontologías, se tomará solamente una.
2. **Los nombres de las dos propiedades de objetos comparadas son sinónimas (sus nombres son diferentes, pero con el mismo significado).** Cuando existan dos propiedades de objetos que sean sinónimas, se elegirá un solo nombre que se adapte y represente a las dos propiedades de las dos ontologías de manera correcta. Se podrá utilizar el nombre de cualquiera de las dos propiedades de objetos, procurando que muestre claramente lo que se quiere representar.
3. **Los nombres de las dos propiedades de objetos comparadas son diferentes.** Cuando el nombre de la propiedad de objeto de la primera ontología no exista en la segunda ontología y/o viceversa, no se les cambiará el nombre y se tomarán estas propiedades de objetos para almacenarlas en la ontología compuesta.

A continuación, como ejemplo, en la Tabla 5 se presenta un fragmento de la relación que hay entre el conjunto de propiedades de objetos estáticas de la ontología temporal del PAER y la ontología temporal del PAI. Para mayor claridad, se presentan de color azul las propiedades de objetos que tienen el mismo nombre en ambas ontologías, estas propiedades se integran una sola vez en la OTPAE compuesta. Las propiedades escritas de color negro, representan propiedades de objetos diferentes, es decir, no tienen sinónimos ni se repiten en las dos ontologías temporales.

Tabla 5. Relación entre propiedades de objetos de la OTPAE del PAER con la OTPAE del PAI.

PO estáticas del PAER	PO estáticas del PAI	PO estáticas del PAIyER
actualiza	actualiza	actualiza
laboraEn	laboraEn	laboraEn
tieneDatosDe	tieneDatosDe	tieneDatosDe
seActualizaPor	seActualizaPor	seActualizaPor
esCreadaPor		esCreadaPor
crea		crea
aplicaPolíticas		aplicaPolíticas
adquiere		adquiere
	buscaExistenciasDe	buscaExistenciasDe
	especificaTipoExistenciasDe	especificaTipoExistenciasDe
	seRegistraPor	seRegistraPor
	seSeparaPor	seSeparaPor
...

Al finalizar esta actividad, se debe contar con el conjunto de propiedades de objetos estáticas que formarán parte de la nueva OTPAE del patrón de análisis compuesto.

4.2.1.3. Combinar propiedades de datos

En la jerarquía de propiedades de datos de la ontología temporal (Figura 6), se tienen dos subpropiedades que pertenecen a la propiedad `topDataProperty`: la subpropiedad `staticDP` que contiene las propiedades de datos utilizadas para representar el conocimiento estático; y la subpropiedad `dynamicDP` que contiene las propiedades de datos utilizadas para representar el conocimiento dinámico. Para esta actividad, se combinarán las propiedades de datos de la subpropiedad `staticDP` de las dos OTPAEs de estudio.

El proceso de combinación de propiedades de datos es similar a la combinación de clases, consiste en comparar todas las propiedades de datos estáticas de la primera ontología (OTPAE1) con todas las propiedades de datos estáticas de la segunda ontología (OTPAE2), con la finalidad de identificar aquellas propiedades que puedan combinarse para obtener el conjunto de propiedades de datos del nuevo dominio.

Durante la comparación entre propiedades de datos nos podemos encontrar con tres situaciones. A continuación, se describen las posibles situaciones y las acciones que se deben realizar ante cada una.

1. **Los nombres de las dos propiedades de datos comparadas existen en ambas ontologías (iguales).** Cuando existan dos propiedades de datos que tengan el mismo nombre en las dos ontologías, se tomará solamente una.
2. **Los nombres de las dos propiedades de datos comparadas son sinónimas (sus nombres son diferentes, pero con el mismo significado).** Cuando existan dos propiedades de datos que sean sinónimas, se elegirá un solo nombre que se adapte y represente a las dos propiedades de las dos ontologías de manera correcta. Se podrá utilizar el nombre de cualquiera de las dos propiedades de datos sinónimas, procurando que muestre claramente lo que se quiere representar.
3. **Los nombres de las dos propiedades de datos comparadas son diferentes.** Cuando el nombre de la propiedad de datos de la primera ontología no exista en la segunda ontología y/o viceversa, no se les cambiará el nombre y se tomarán estas propiedades de datos para almacenarlas en la ontología compuesta.

A continuación, como ejemplo, en la Tabla 6 se presenta un fragmento de la relación que hay entre el conjunto de propiedades de datos estáticas de la ontología temporal del PAER y la ontología temporal del PAI. Para mayor claridad, se presentan de color azul las propiedades de datos que tienen el mismo nombre en ambas ontologías, estas propiedades se integran una sola vez en la OTPAE compuesta. Las propiedades escritas de color negro, representan propiedades de datos diferentes, es decir, no tienen sinónimos ni se repiten en las dos ontologías temporales.

Tabla 6. Relación entre propiedades de datos de la OTPAE del PAER con la OTPAE del PAI.

PD estáticas del PAER	PD estáticas del PAI	PD estáticas del PAIyER
tieneComportamiento	tieneComportamiento	TieneComportamiento
tieneNombre	TieneNombre	TieneNombre
tieneEspecificidad	TieneEspecificidad	tieneEspecificidad
tieneGiro	TieneGiro	TieneGiro
tieneId	TieneId	TieneId
tieneNivelEstabilidad	tieneNivelEstabilidad	tieneNivelEstabilidad
tienePercepcion	tienePercepcion	tienePercepcion
tieneTipo	tieneTipo	TieneTipo
tieneDireccion		tieneDireccion
tieneNoConfirmacion		tieneNoConfirmacion
tieneTelefono		TieneTelefono
	tieneEdad	tieneEdad
	tieneUbicacion	tieneUbicacion
	tieneCantidad	tieneCantidad
...

Al finalizar esta actividad, se debe contar con el conjunto de propiedades de datos estáticas que formarán parte de la nueva OTPAE del patrón de análisis compuesto.

4.2.2. Especificación de conceptos y relaciones (Formalización)

Cada ontología temporal cuenta originalmente con la especificación de sus propios conceptos y relaciones (Formalización). Sin embargo, al momento de combinar los términos del dominio en la fase 4.2.1. *Combinación de términos del dominio (Conceptualización)*, puede haber modificaciones en la jerarquía y propiedades de cada elemento, ya que durante el proceso de combinación se pueden cambiar y/o eliminar nombres de conceptos (clases, propiedades de objetos y propiedades de datos).

Por tal motivo, en esta fase se tiene que realizar una especificación explícita de los conceptos y sus relaciones, utilizando una jerarquía. Además, también se deberán especificar las propiedades de cada concepto. A continuación, se describen las actividades a realizar en esta fase.

4.2.2.1. Especificar jerarquía de clases

Para realizar esta actividad, es necesario contar con el conjunto de clases obtenido en la actividad 4.2.1.1. *Combinar clases*. Esta actividad consiste en realizar una especificación del conjunto de clases mediante una jerarquía de clases. Por lo tanto, es necesario verificar que cada clase se encuentre en el nivel correcto, debido a que puede haber clases que sean sinónimas, pero cada una se encuentre originalmente en un nivel diferente en cada ontología. Por tal motivo, es necesario tomar en cuenta el nivel en que se encuentra cada clase, con la

finalidad de ubicarla en el nivel adecuado. Finalmente, se obtendrá una nueva jerarquía de clases que represente el nuevo dominio.

A continuación, en la Tabla 7 se presenta como ejemplo un fragmento de la jerarquía de clases obtenida durante la combinación de la ontología temporal del PAI con la ontología temporal del PAER.

Tabla 7. Especificación de la jerarquía de clases para la OTPAE del PAIyER.

Clases
Contrato
Inventario
Persona
Cliente
Empleado
Propietario
Reservación
Entidad
Institucion

4.2.2.2. Especificar relaciones entre clases

Para realizar esta actividad, es necesario contar con el conjunto de clases y propiedades de objetos obtenidos en la actividad 4.2.1.1. *Combinar clases* y en la actividad 4.2.1.2. *Combinar propiedades de objetos*, respectivamente. Originalmente, en cada ontología temporal se tienen especificadas las relaciones entre clases, es decir, cómo es que las clases se relacionan unas con otras. Sin embargo, durante la combinación de éstas, las relaciones pueden modificarse debido a la eliminación y/o modificación de nombres de clases y/o propiedades. Por tal motivo, se deben actualizar las relaciones entre las clases. Para ello, se debe especificar en una tabla todas las relaciones que existen entre las clases especificando qué clase es la que realiza la acción (dominio), el nombre de la relación (propiedad de objeto), la clase sobre la cual se realiza la acción (rango) y el tipo de característica (funcional o inversa funcional) de la propiedad de objeto.

A continuación, en la Tabla 8, se presenta como ejemplo, un fragmento de la especificación de relaciones entre clases durante la combinación de la OTPAE del PAI con la OTPAE del PAER.

Tabla 8. Fragmento de la especificación de relaciones entre clases en la OTPAE del PAIyER.

Dominio	Relación	Rango	Característica
Contrato	esAdquiridoPor	Cliente	Funcional
Contrato	esExpedidoPor	Empleado	Funcional
Entidad	esDevueltaPor	Cliente	Funcional
Entidad	esUsadaPor	Cliente	Funcional
Entidad	esVerificadaPor	Empleado	Funcional
...

4.2.2.3. Especificar propiedades de las clases

Para realizar esta actividad, es necesario contar con el conjunto de clases y propiedades de datos obtenidos en la actividad 4.2.1.1. *Combinar clases* y en la actividad 4.2.1.3. *Combinar propiedades de datos*, respectivamente. En cada ontología temporal, cada clase tiene originalmente asignado un conjunto de propiedades de datos. Sin embargo, durante la combinación, éstas se pueden eliminar y/o modificar. Por tal motivo se deben actualizar las propiedades de datos del nuevo conjunto de clases.

Es importante tomar en cuenta que, cuando se utilice el nombre de una sola clase cuando tenga el mismo nombre o el nombre sea sinónimo en las dos ontologías (ver actividad 4.2.1.1. *Combinar clases*), las propiedades de datos de la clase a utilizar deberán ser las propiedades de datos de la clase de la primera ontología más las propiedades de datos de la clase de la segunda ontología (sin repetir propiedades de datos). Por ejemplo, sean *C1* y *C2* dos clases sinónimas o iguales, correspondientes a la primera y segunda ontología, respectivamente, y sea una tercera clase *C3* la clase resultado de la combinación de *C1* y *C2*; si las propiedades de datos de *C1* es *p1* y *p2* y las propiedades de datos de *C2* es *p3* y *p4*, entonces las propiedades de datos de la clase *C3* serán *p1*, *p2*, *p3* y *p4*. Esto se realiza con la finalidad de utilizar toda la información y no eliminar conocimiento.

A continuación, en la Tabla 9, se presenta como ejemplo, un fragmento de la especificación de las propiedades de datos de las clases durante la combinación de la ontología temporal del PAI con la ontología temporal del PAER.

Tabla 9. Fragmento de la especificación de las propiedades de las clases de la OTPAE del PAIyER.

Clases	Propiedades de datos
Institucion	tieneId, tieneNombre, tieneGiro, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion
Inventario	tieneId, tieneNombre, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion
Persona	tieneNombre, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion
Cliente	tieneNombre, tieneDireccion, tieneTelefono, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion
Empleado	tieneId, tieneNombre, tieneEdad, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion
...	...

Al finalizar esta fase, se debe contar con la jerarquía de clases, relaciones entre clases y propiedades de datos de todas las clases que conformarán la nueva OTPAE, del patrón de análisis compuesto.

4.2.3. Actualización de restricciones

En esta fase, se realiza una actualización de las restricciones de cuantificación existencial y universal para integrarlas en la ontología temporal compuesta. Cada ontología temporal cuenta con sus propias restricciones. Sin embargo, al momento de combinar las clases, propiedades de objetos y propiedades de datos de dos ontologías temporales puede haber modificaciones debido a que durante el proceso de combinación pueden cambiarse y/o eliminarse nombres de clases y/o propiedades. Por tal motivo, es necesario actualizar las restricciones para la ontología compuesta. A continuación, se describen las dos actividades a realizar en esta fase.

4.2.3.1. Actualizar restricciones existenciales

En esta actividad, se realiza una actualización de las restricciones existenciales y características (funcional e inversa funcional) de las clases y propiedades de objetos de la nueva ontología compuesta, mediante una descripción lógica basada en la DL ALC y su equivalente en la sintaxis funcional de OWL-DL. Para realizar esta actividad, es necesario identificar las restricciones existenciales originales de las dos ontologías OTPAEs de estudio, con la finalidad de actualizar solamente las restricciones existenciales en donde intervienen clases y/o propiedades de objetos (estáticas) que se modifican durante el proceso de combinación.

En el editor Protegé, las restricciones de cuantificación existencial se definen usando la palabra “*some*” (Figura 8) para especificar que un concepto se relaciona con al menos otro. En la Tabla 10 se presenta como ejemplo, un fragmento de las restricciones existenciales de la OTPAE del PAIyER.

Tabla 10. Restricciones existenciales en la OTPAE del PAIyER.

Dominio	Relación	Restricción	Rango
Empleado	separa	some	Entidad
Empleado	registra	some	Entidad
Empleado	actualiza	some	Inventario
Empleado	cancela	some	Reservacion
Empleado	registraUsoDe	some	Entidad
Empleado	verificaDisponibilidadDe	some	Entidad
Cliente	adquiere	some	Entidad
Cliente	daDeBaja	some	Reservacion
Cliente	devuelve	some	Entidad
...

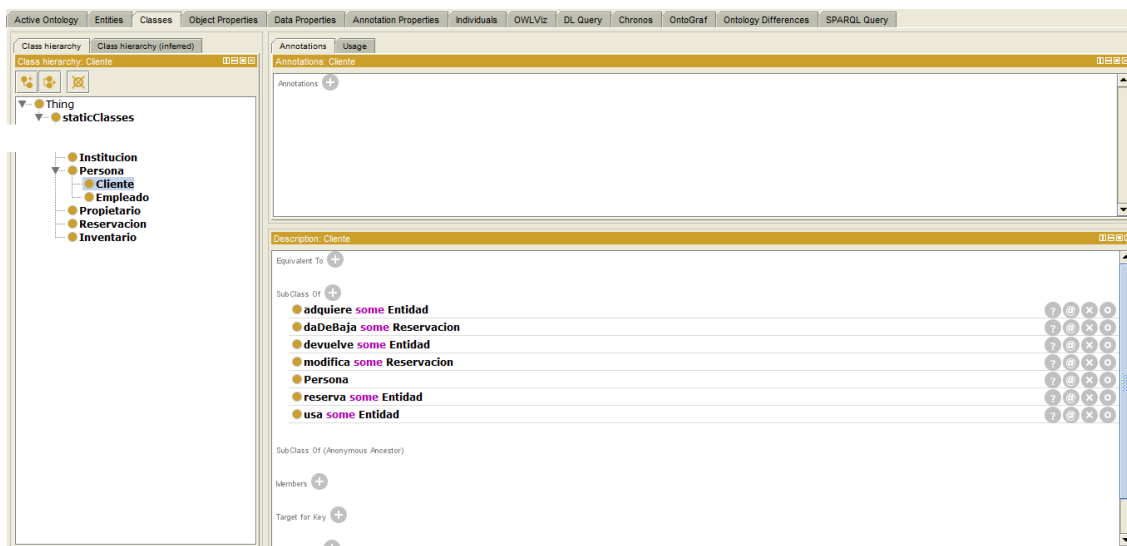


Figura 8. Definición de restricciones existenciales en la OTPAE del PAIyER con Protégé.

4.2.3.2. Actualizar restricciones universales

En esta actividad, se realiza una actualización de las restricciones universales y características (funcional, inversa funcional) de las clases y propiedades de objetos de la nueva ontología compuesta, mediante una descripción lógica basada en la DL ALC y su equivalente en la sintaxis funcional de OWL-DL. Para realizar esta actividad, es necesario identificar las restricciones universales originales de las dos ontologías OTPAEs de estudio, con la finalidad de actualizar solamente las restricciones universales en donde intervienen clases y/o propiedades de objetos (estáticas) que se modificaron durante el proceso de combinación.

En el editor Protégé, las restricciones de cuantificación universal se definen empleando la palabra “only” (Figura 9) para indicar que un concepto se relaciona únicamente con otro. En la Tabla 11 se presenta como ejemplo, un fragmento de las restricciones universales de la OTPAE del PAIyER.

Tabla 11. Restricciones universales en la OTPAE del PAIyER.

Restricciones universales			
Dominio	Relación	Restricción	Rango
Entidad	seSeparaPor	only	Empleado
Entidad	seRegistraPor	only	Empleado
Entidad	esDevueltaPor	only	Cliente
Entidad	esUsadaPor	only	Cliente
Empleado	laboraEn	only	Institucion
Empleado	actualiza	only	Reservacion
Empleado	crea	only	Reservacion
Empleado	expide	only	Contrato
Reservacion	esCreadaPor	only	Empleado
...

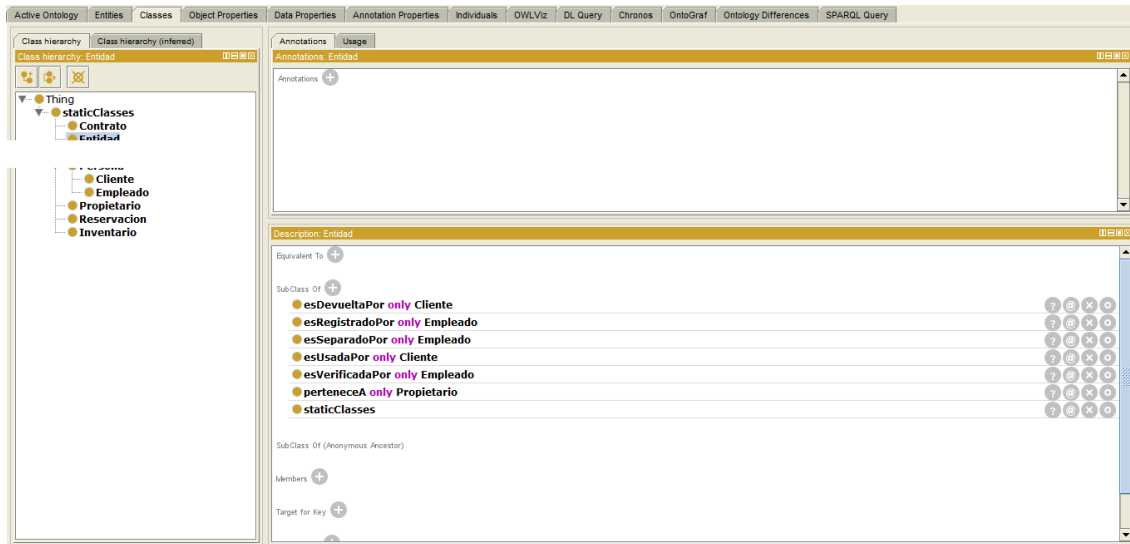


Figura 9. Definición de restricciones universales en la OTPAE del PAIyER con Protégé.

Cuando una restricción incluya algún nombre de clase que fue modificada, se debe modificar la restricción para que el nombre de la clase que interviene en la restricción coincida con el nombre de la clase modificada. Por ejemplo, durante la combinación de clases (ver sección 4.2.1.1. *Combinar clases*), se combinaron las clases *Establecimiento* e *Institucion* (sinónimos) de la OTPAE del PAER y del PAI, respectivamente, eliminándose la clase *Establecimiento* para utilizar la clase *Institucion* en la ontología compuesta. Por lo tanto, si la restricción existencial (utilizando la DL ALC) en la OTPAE del PAER para definir que un *Empleado* (dominio) labora en una *Institucion* (rango) es $Empleado \sqsubseteq \exists laboraEn.Institucion$ y la restricción existencial en la OTPAE del PAI para definir que un *Empleado* (dominio) labora en un *Establecimiento* (rango) es $Empleado \sqsubseteq \exists laboraEn.Establecimiento$, entonces, en la restricción actualizada para la nueva ontología compuesta se tomará la restricción existencial $Empleado \sqsubseteq \exists laboraEn.Institucion$.

Debido a que muchos de los nombres de las clases pueden quedar intactos, es decir, que no presenten ninguna modificación durante la combinación, es evidente que las restricciones en donde intervienen esos nombres de clases no tendrán modificaciones. Por lo tanto, si en las dos ontologías temporales se maneja la misma restricción, se deberá definir solamente una vez, no se definirán en la ontología compuesta restricciones repetidas.

Una vez actualizadas las restricciones existenciales y universales, se utiliza el editor de ontologías Protégé para almacenar la información obtenida hasta esta fase. Para ello, se crea una ontología en Protégé y se integra toda la información obtenida en la fases 4.2.2. *Especificación de conceptos y relaciones (Formalización)* y 4.2.3. *Actualización de restricciones*. De esta manera, se obtiene como resultado una ontología que almacena la estructura estática, es decir, una OPAE del nuevo patrón de análisis compuesto. Para este ejemplo, todos los elementos definidos en la OPAE del PAIyER son estáticos, los elementos

temporales se agregan en la siguiente fase. La ontología OPAE del patrón de análisis PAIyER se describe en el *Anexo C. Ontología OPAE del patrón de análisis PAIyER*.

4.2.4. Actualización de la KB con aspectos de temporalidad

Hasta este punto se cuenta con una ontología estática. Por tal motivo, en esta fase, la ontología estática (OPAE) compuesta será transformada a una ontología temporal (OTPAE) agregando elementos temporales. También, las propiedades de objetos en la ontología OTPAE compuesta se convertirán en propiedades de objetos temporales, en donde se reemplazarán los conceptos de inclusión (CIs, *Concept Inclusions*) de la DL que definen un rol abstracto por otros que definan un rol abstracto temporal (Piedrola Domínguez N., 2019).

Una vez que la ontología OTPAE incluya las propiedades de objeto temporales necesarias, se debe verificar su consistencia para que posteriormente se agreguen como conocimiento (instancias) secuencias de interpretaciones que se definirán con base en la experiencia y de acuerdo a las fórmulas ALC_{LTL} correspondientes.

A continuación, se describe cada una de las actividades a realizar para actualizar la KB con aspectos de temporalidad.

4.2.4.1. Generar la ontología OTPAE a partir de la ontología OPAE

De acuerdo a la metodología MECOTPA desarrollada en (Piedrola Domínguez N., 2019) para convertir una ontología estática a temporal se debe combinar la ontología OPAE con la ontología OWL-Time (proveída por el plugin Chronos en Protégé bajo el poder de expresividad $ALCROIN(D)$), el cual agrega a la nueva ontología elementos temporales que permiten representar el conocimiento dinámico.

Para facilitar la creación y edición de ontologías temporales, en esta actividad se recomienda el uso del editor de ontologías Protégé en su versión 4.3 con el plugin Chronos configurado, el razonador Pellet (el proceso para la configuración del plugin Chronos y el razonador Pellet se describe en (Piedrola Domínguez N., 2019)) para la verificación de consistencia de ontologías temporales así como también contar con la ontología OPAE del patrón de análisis compuesto y la ontología OWL-Time.

Para obtener una ontología OTPAE a partir de una ontología OPAE, solo se tiene que abrir la ontología OPAE en el editor Protégé, dar clic en la pestaña *Chronos* y un cuadro de dialogo solicitará la ubicación del archivo OWL de la ontología OWL-Time como se muestra en la Figura 10. Al terminar el proceso ambas ontologías habrán sido mezcladas poniendo a nuestra disposición las clases, propiedades de objeto y propiedades de datos necesarias para incluir aspectos de temporalidad en la ontología OPAE.

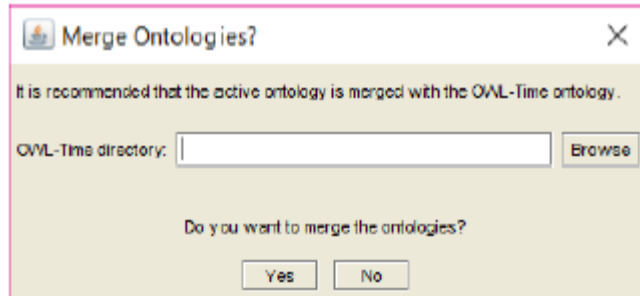


Figura 10. Ventana para combinar la ontología OPAE con la ontología OWL-Time.

Para utilizar la misma estructura que las OTPAEs desarrolladas con MECOTPA (Figura 6), se utiliza la clase *dynamicClasses* para almacenar las clases dinámicas, la propiedad de objetos *dynamicOP* para almacenar las propiedades de objetos dinámicas y la propiedad de datos *dynamicDP* para almacenar las propiedades de datos dinámicas. De este modo, la separación de elementos estáticos y dinámicos facilitará la gestión de la ontología temporal compuesta.

4.2.4.2. Convertir roles abstractos a temporales

Con la interfaz de Chronos (descrita en (Piedrola Domínguez N., 2019)) en Protégé, es posible convertir clases, propiedades de objetos y datos en temporales. Debido a que las OTPAEs a combinar se encuentran desarrolladas utilizando la metodología MECOTPA, nos enfocaremos únicamente en realizar la conversión de propiedades de objeto.

Para la conversión es necesario identificar en cada ontología temporal que se estudia, las propiedades de objetos que tienen un comportamiento temporal, pues aquellas propiedades de objetos que no son temporales se dejarán igual (relación binaria).

Para convertir una relación binaria a temporal, de las propiedades de objetos que tienen un comportamiento temporal, en la interfaz de Chronos, en el panel *Object Property Hierarchy*, seleccionar la propiedad de objeto a convertir en temporal. En el panel *Chronos View>Object Property* dar click en el botón *Convert* y aparecerá el cuadro de dialogo de la Figura 11 indicando los triples afectados por la conversión. Para finalizar, dar clic en el botón *Yes*.

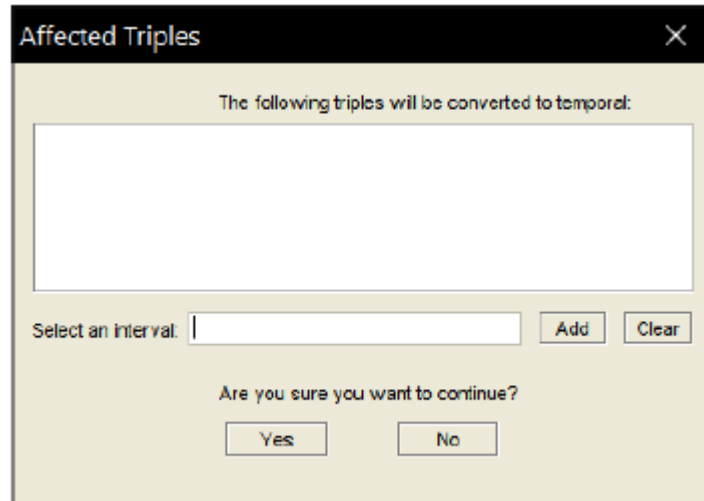


Figura 11. Triples afectados en la conversión de una relación binaria a temporal.

En la conversión de una propiedad de objeto a una propiedad de objeto temporal (un tipo de relación n-aria donde interviene el tiempo), no se perderá la esencia de las relaciones binarias. Las relaciones temporales se diferencian de las relaciones binarias por estar incluidas como subpropiedades de la propiedad de objeto *participatesIn* y por tener un objeto intermediario de la clase *Event* para establecer la relación. A través del evento, se podrá establecer la relación de duración (a través de la clase *during*) así como el momento en que ésta tendrá lugar (clase *TimeInterval*).

En la proposición “*p* = el cliente reserva una entidad”, el verbo reserva representa una relación binaria, en donde el dominio es el cliente y el rango es la entidad como se observa en la Figura 12, la cual generalmente propicia la inclusión de la relación inversa *esReservadaPor*.



Figura 12. Relación binaria a partir de la proposición "el Cliente reserva una Entidad".

La representación formal de la relación puede expresarse mediante un CI básico de la DL ALC:

$$\text{Cliente} \sqsubseteq \exists \text{ reserva. Entidad}$$

Por otra parte, cuando la relación binaria se convierte a temporal (Figura 13), la relación indicará que el cliente solicita reservar una entidad a través de un evento.

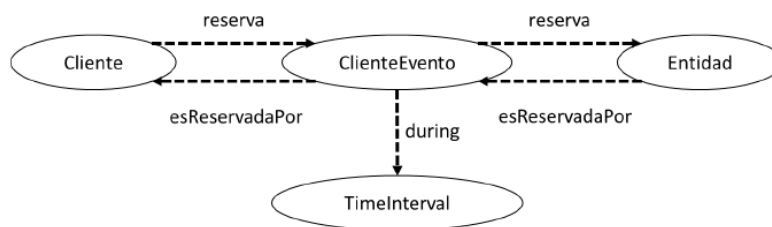


Figura 13. Relación temporal de la proposición "El cliente reserva una Entidad a través de un Evento".

La representación formal de la relación temporal se expresa como un CI temporal de la DL ALC:

$$Cliente \sqsubseteq \exists reserva.(Event \sqcap (\exists reserva.Entidad))$$

Si la relación binaria en la KB cuenta con características tales como Funcional o Inversa funcional, al momento de convertirlas a temporal en el editor Protégé, se generarán las reglas SWRL (Lenguaje de Reglas para la Web Semántica del inglés *Semantic Web Rule Language*) correspondientes. Un ejemplo de las reglas SWRL se describe en la Tabla 12. Un razonador que soporte reglas SWRL podrá inferir nuevo conocimiento sobre conceptos dependientes del tiempo, por ejemplo, *before(intervalo_eventoA, intervalo_eventoB)*, *equals(intervalo_eventoB, intervalo_eventoC)* o *after(intervalo_eventoD, intervalo_eventoC)*.

Tabla 12. Reglas SWRL para características funcional e inversa funcional.

Propiedad de objeto	Característica	Regla SWRL
<i>actualiza</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), actualiza(?e1, ?y1), actualiza(?e2, ?y2), actualiza(?x, ?e1), actualiza(?x, ?e2) -> SameAs(?y1, ?y2), Reservacion(?y1), Reservacion(?y2), Empleado(?x)</i>
<i>seActualizaPor</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), seActualizaPor(?e1, ?y), seActualizaPor(?e2, ?y), seActualizaPor(?x1, ?e1), seActualizaPor(?x2, ?e2) -> SameAs(?x1, ?x2), Empleado(?x1), Empleado(?x2), Reservacion(?y)</i>

P. ej., la regla SWRL para el rol abstracto *actualiza*, establece como antecedente los individuos del concepto Event: *Event(?e1)* y *Event(?e2)*, mediante los cuales se lleva a cabo el rol abstracto temporal *actualiza*: *actualiza(?e1, ?y1)*, *actualiza(?e2, ?y2)*, *actualiza(?x, ?e1)*, *actualiza(?x, ?e2)*. También establece que el rol abstracto temporal solo puede relacionar a un individuo (?x) con otro individuo ?y1 o ?y2 al mismo tiempo *overlaps(?i1, ?i2)*, donde *i1* e *i2* son los intervalos de duración de los eventos: *Event(?e1)* y *Event(?e2)*, es decir, *during(?e1, ?i1)* y *during(?e2, ?i2)*, lo cual implica en el consecuente, que ?x es un individuo del concepto Empleado *Empleado(?x)*, que ?y1 y ?y2 son individuos similares del concepto

Reservacion: *Reservacion(?y1)*, *Reservacion(?y2)*, *SameAs (?y1, ?y2)*. En (Piedrola Domínguez N., 2019, pp 328) se proporciona más información sobre las restricciones de cardinalidad global en la propiedad funcional e inversa funcional y sintaxis en las reglas SWRL.

Siempre que se lleve a cabo el proceso de conversión es recomendable llevar un control de los cambios mediante una retroalimentación al *Diccionario de términos* con el fin de agregar los elementos temporales y mantener la documentación actualizada y verificar la consistencia de la ontología.

4.2.4.3. Verificar la consistencia semántica de la OTPAE compuesta

A pesar de que al utilizar el plugin Chronos de Protégé en la actividad 4.2.4.2. *Convertir roles abstractos a temporales*, no deberían generarse inconsistencias en la ontología OTPAE, es necesario realizar una verificación de consistencia semántica temporal TSCC.

Una vez obtenida la nueva ontología temporal compuesta, utilizando el editor de ontologías Protégé, se utiliza el procedimiento de la 1er verificación TSCC definida en (Piedrola Domínguez N., 2019, pp 339), donde se describe en qué consiste la verificación automática en Protégé utilizando el razonador Pellet, para comprobar que la ontología es consistente. Para iniciar el proceso de razonamiento automatizado, en el editor de ontologías Protégé, se abre la ontología temporal compuesta, después se va al menú *Reasoner* y se selecciona el razonador *Pellet* (se requiere la integración del razonador Pellet en Protégé para hacer uso del razonador, en (Piedrola Domínguez N., 2019, pp 316) se describe el proceso para la configuración del plugin Chronos y el razonador Pellet en Protégé). Para iniciar el razonador, ir nuevamente a *Reasoner* y dar clic en la opción *Start reasoner* y para detener el razonador, ir nuevamente a *Reasoner* y dar clic en la opción *Stop reasoner*.

Al finalizar esta fase, se deberá contar con una OTPAE compuesta consistente, con elementos estáticos y temporales. Además, la ontología deberá tener todas las relaciones temporales correspondientes y por lo tanto las reglas SWRL requeridos. Cabe mencionar que, hasta este punto, la OTPAE se encuentra vacía, es decir, aún no se crean instancias de las clases, esto se realizará más adelante.

4.2.5. Combinación de fórmulas ALC_{LTL}

En esta fase, se realiza la combinación de las fórmulas ALC_{LTL} de las dos ontologías temporales que entran al estudio, con la finalidad de especificar el comportamiento dinámico de la OTPAE compuesta. Cada ontología temporal que se crea a partir de la metodología MECOTPA, cuenta con un conjunto de fórmulas ALC_{LTL} que describen formalmente la especificación de comportamiento que cumple con los requerimientos funcionales (RFs) del patrón de análisis correspondiente. Debido a que cada OTPAE que entra al estudio cuenta con la descripción formal de sus propios requerimientos, es necesario combinar dichas fórmulas

para especificar el comportamiento dinámico de la OTPAE del patrón de análisis compuesto. Como requisito de entrada para esta actividad, se requiere del conjunto de fórmulas ALC_{LTL} correspondientes a las dos OTPAEs a combinar.

4.2.5.1. Combinar fórmulas ALC_{LTL}

Una fórmula ALC_{LTL} , es una especificación formal del comportamiento especificado en un requerimiento funcional (RF). Para combinar las fórmulas ALC_{LTL} de las dos OTPAEs que entran al estudio, primero, es necesario actualizar los nombres de los conceptos y roles que participan en dichas fórmulas. El motivo de esto se debe a que al realizar las actividades de la fase 4.2.1. *Combinación de términos del dominio (Conceptualización)*, pueden eliminarse y/o modificarse los nombres de los conceptos y/o roles (clases y propiedades) que intervienen en cada fórmula ALC_{LTL} . Una vez actualizados los conceptos que intervienen en las fórmulas, el siguiente paso será, verificar si existe alguna relación entre cada especificación formal (requerimiento funcional representado mediante una fórmula ALC_{LTL}), identificando si:

1. **Un requerimiento funcional cubre el mismo requerimiento funcional en la otra ontología.** En el caso de que un requerimiento funcional de la primera ontología cubra el mismo requerimiento funcional en la segunda ontología (o viceversa), pero con diferente orden en los conceptos de inclusión (CI), se deberá obtener una fórmula ALC_{LTL} que satisfaga correctamente el requerimiento funcional de la OTPAE compuesta. Si las fórmulas ALC_{LTL} en los dos PAs son iguales, entonces se deberán unificar para utilizar solamente una.
2. **Un requerimiento funcional amplía/extiende un requerimiento funcional de la otra ontología.** En el caso de que un requerimiento funcional de la primera ontología amplíe o extienda la descripción de un requerimiento funcional de la segunda ontología (o viceversa), se deberán estructurar adecuadamente los conceptos de inclusión (CI) para especificar formalmente el requerimiento funcional base (requerimiento funcional que se está ampliando).

Por ejemplo, sea R1 un RF de la primera ontología que corresponde a “Buscar artículo en el inventario” en donde la búsqueda se realiza por nombre del artículo y R2 un RF de la segunda ontología que corresponde a “Buscar artículo en el inventario” y en donde la búsqueda se realiza por código de barras. En este caso, R1 amplía la descripción de R2 debido a que, para buscar un artículo en el inventario, ahora la búsqueda podrá ser por nombre o por código de barras.

3. **Un requerimiento funcional es parte de un requerimiento funcional más general de la otra ontología.** En el caso de que un requerimiento funcional de la primera ontología sea parte de un requerimiento funcional más general de la segunda ontología

(o viceversa), la fórmula ALC_{LTL} no se modificará, esto se debe a que cada fórmula satisface un requerimiento funcional. Si la ejecución de un requerimiento funcional no interfiere en otro, entonces no es necesario modificar dicha fórmula.

Por ejemplo, el PAER describe cinco RFs (1. Hacer una reservación, 2. Usar una entidad reservada, 3. Modificar una reservación, 4. Cancelar una reservación y 5. Crear un contrato) y el patrón de análisis de cuentas (PAC), descrito en (Piedrola Domínguez N., 2019) describe tres RFs (1. Abrir una cuenta, 2. Realizar una Transacción y 3. Cerrar una cuenta). Si se quisiera combinar estos dos PAs, entonces, los cinco RFs del PAER se podrían tomar como requerimientos específicos del RF “Realizar una transacción” del PAC, debido a que cada RF del PAER podría ser considerado como una transacción. De este modo, los cinco RFs del PAER serían parte de un RF más general, es decir, del RF “Realizar una transacción”.

4. **La ejecución de un requerimiento funcional afecta el comportamiento de la otra ontología.** En el caso de que la ejecución de un requerimiento funcional de la primera ontología afecte en el comportamiento de un requerimiento funcional en la segunda ontología (o viceversa), se deberá complementar la fórmula ALC_{LTL} del requerimiento funcional afectado, con los conceptos de inclusión (CIs) necesarios para cubrir el comportamiento correcto.

Por ejemplo, la fórmula ALC_{LTL} del RF “Hacer una reservación” (Tabla 13), originalmente perteneciente a la ontología temporal del PAER, contiene los conceptos de inclusión necesarios para hacer una reservación. Sin embargo, al combinar el PAER con el PAI es necesario agregar a esta fórmula, el concepto de inclusión $Empleado \sqsubseteq \exists actualiza.Entidad$ (originalmente correspondiente a la OTPAE del PAI) debido a que cuando se crea una reservación, la cantidad de la entidad reservada tiene que ser actualizada en el inventario y esta actualización es originalmente realizada en la ontología temporal del PAI. De esta manera, la fórmula ALC_{LTL} de la Tabla 13, quedará como la fórmula ALC_{LTL} de la Tabla 14.

Tabla 13. Fórmula $ALC_{LTL}(\varphi)$ del RF. Hacer una reservación del PAER.

$$\begin{aligned} & \mathcal{G}(\\ & \quad \mathcal{X}(\text{Cliente} \sqsubseteq \exists \text{reserva.Entidad}) \rightarrow (\\ & \quad \quad \mathcal{XX}(\text{Empleado} \sqsubseteq \exists \text{verificaDisponibilidadDe.Entidad}) \wedge \mathcal{F}(\\ & \quad \quad \quad \neg \mathcal{XXX}(\text{Empleado} \sqsubseteq \forall \text{crea.Reservacion}) \\ & \quad \quad \quad \neg \mathcal{XXXX}(\text{Empleado} \sqsubseteq \forall \text{expide.Contrato}) \\ & \quad \quad) \\ & \quad) \\ &) \end{aligned}$$

Una vez obtenido el conjunto de fórmulas ALC_{LTL} correspondiente al nuevo dominio, las fórmulas tienen que reducirse a fórmulas LTL (φ') debido a la complejidad que implica en la verificación de satisfacibilidad en fórmulas ALC_{LTL} . La reducción consiste en reemplazar los CIs básicos de las fórmulas ALC_{LTL} por proposiciones atómicas de la lógica temporal LTL (Piedrola Domínguez N., 2019). A continuación, en la Tabla 15 se presenta un ejemplo en la reducción de la fórmula ALC_{LTL} (φ) del requerimiento RF. Hacer una reservación (Tabla 14) a una fórmula LTL (φ') correspondiente a la OTPAE del PAIyER.

Tabla 14. Fórmula ALC_{LTL} (φ) del RF. Hacer una reservación del PAIyER.

$\mathcal{G}(\mathcal{X}(\text{Cliente} \sqsubseteq \exists \text{reserva. Entidad}) \rightarrow (\mathcal{XX}(\text{Empleado} \sqsubseteq \exists \text{verificaDisponibilidadDe. Entidad}) \wedge \mathcal{F}(\neg \mathcal{XXX}(\text{Empleado} \sqsubseteq \forall \text{crea. Reservacion} \wedge \text{Empleado} \sqsubseteq \exists \text{actualiza. Entidad}) \wedge \neg \mathcal{XXXX}(\text{Empleado} \sqsubseteq \forall \text{expide. Contrato})))$
--

Tabla 15. Reducción de la fórmula ALC_{LTL} (φ) del RF. Hacer una reservación a una fórmula LTL (φ').

Fórmula LTL(φ')	Proposiciones
RF. Hacer una reservación	
$\mathcal{G}(\mathcal{X}(p) \rightarrow (\mathcal{XX}(q) \wedge \mathcal{F}(\neg \mathcal{XXX}(r \wedge s) \wedge \neg \mathcal{XXXX}(t))))$	$p = \text{Cliente} \sqsubseteq \exists \text{reserva. Entidad}$ $q = \text{Empleado} \sqsubseteq \exists \text{verificaDisponibilidadDe. Entidad}$ $r = \text{Empleado} \sqsubseteq \forall \text{crea. Reservacion}$ $s = \text{Empleado} \sqsubseteq \forall \text{actualiza. Entidad}$ $t = \text{Empleado} \sqsubseteq \forall \text{expide. Contrato}$

Como se puede observar en la Tabla 15, por cada concepto de inclusión (p. ej. $\text{Cliente} \sqsubseteq \exists \text{reserva. Entidad}$) utilizado en la fórmula ALC_{LTL} , se utiliza una proposición (p. ej. p) para obtener finalmente una fórmula LTL.

Al final de esta fase, se debe contar con el conjunto de fórmulas ALC_{LTL} y LTL correspondientes a la especificación formal de los requerimientos funcionales de la OTPAE compuesta. Todas las fórmulas temporales de la ontología temporal del PAIyER se presentan en la sección 1.2. *Componente extensional del Anexo D. Ontología OTPAE del patrón de análisis PAIyER.*

4.2.6. Interpretación semántica temporal

A continuación, se describen las actividades a realizar en esta fase.

4.2.6.1. Proponer una secuencia de interpretaciones con base en la experiencia

En esta actividad se debe proponer secuencias de interpretaciones (SoIs, para casos de éxito y casos de éxito y fracaso) basadas en la experiencia sobre el comportamiento dinámico de acuerdo a lo establecido en las fórmulas ALC_{LTL} . Cada afirmación en determinado momento, debe representarse como conocimiento en la ontología OTPAE mediante una interpretación, estableciendo los axiomas de aserción de propiedades de objetos correspondientes (p. ej. ver sección 1.2. *Componente extensional del Anexo D. Ontología OTPAE del patrón de análisis PAIyER*).

Debido a que incluir secuencias de interpretaciones en la ontología temporal puede ocasionar inconsistencias en la KB, ésta debe verificarse. Si la ontología OTPAE es consistente, la secuencia de interpretaciones podrá reducirse a una ruta de computación, un componente generalmente usado en la verificación de satisfacibilidad de fórmulas de la TL LTL.

Una secuencia de interpretaciones describe cómo han ocurrido los hechos con respecto a una especificación de comportamiento. Generalmente, las interpretaciones incluyen individuos concretos de las clases implicadas. Debido a las limitaciones de la TDL ALC_{LTL} , la secuencia debe iniciar a partir de un momento presente y extenderse infinitamente hacia el futuro, algo muy parecido a una línea de tiempo.

La descripción de cada momento en la secuencia deberá representarse formalmente en la ontología OTPAE estableciendo los axiomas de aserción de propiedades de objetos temporales correspondientes (Tabla 20).

Las propiedades de objeto temporales (*pot*), relacionan individuos del dominio de interpretación, mediante la intervención de un evento (*pot(clase_dominio, evento), pot(event, Clase_Rango)*), cuya duración está dada por un intervalo de tiempo definido por instantes de inicio y fin (*during(evento, intervalo), hasBeginning(intervalo, instante_inicio), hasEnd(intervalo, instante_fin)*). Un intervalo puede ser antecedido, igual o precedido por otro intervalo. Sin embargo, para representar las secuencias de momentos solo usaremos la precedencia de intervalos (*intervalAfter(intervalo2, intervalo1)*).

Para definir los individuos de eventos (Tabla 16), intervalos (Tabla 17), instantes de inicio (Tabla 18) y de fin (Tabla 19) en la ontología OTPAE compuesta, se recomienda utilizar la nomenclatura mostrada a continuación, propuesta por (Piedrola Domínguez N., 2019).

Tabla 16. Nomenclatura propuesta para la clase Event.

<i>Req<número-requerimiento>-S<número-estado>-Ev<número-evento></i>

Tabla 17. Nomenclatura propuesta para la clase DateTimeInterval.

<i>Req<número-requerimiento>-S<número-estado>-Ev<número-evento>-DTI</i>

Tabla 18. Nomenclatura propuesta para la clase Instant para representar el instante de inicio.

<i>Req<número-requerimiento>-S<número-estado>-Ev<número-evento>-DTI-SIns</i>
--

Tabla 19. Nomenclatura propuesta para la clase Instant para representar el instante de fin.

<i>Req<número-requerimiento>-S<número-estado>-Ev<número-evento>-DTI-EIns</i>
--

Tabla 20. Ejemplo de secuencias de interpretaciones para casos de éxito en el requerimiento funcional RF. Hacer una reservación.

Momento	¿Qué sucede o sucederá?	Axiomas de aserción de propiedades de objetos
1	El cliente Heidi reserva la entidad Libro1 en el evento Req1-S0-Ev01, que se llevará a cabo durante el intervalo de tiempo Req1-S0-Ev01-DTI cuyos intervalos de inicio y fin son Req1-S0-Ev01-DTI-SIns y Req1-S0-Ev01-DTI-EIns, respectivamente.	reserva(Heidi, Req1-S0-Ev01) reserva(Req1-S0-Ev01, Libro1) during(Req1-S0-Ev01, Req1-S0-Ev01-DTI) hasBegining(Req1-S0-Ev01-DTI, Req1-S0-Ev01-DTI-SIns) hasEnd(Req1-S0-Ev01-DTI, Req1-S0-Ev01-DTI-EIns)
2	El empleado Valentino verifica la disponibilidad de la entidad Libro1 en el evento Req1-S1-Ev02, que se llevará a cabo durante el intervalo de tiempo Req1-S1-Ev02-DTI cuyos intervalos de inicio y fin son Req1-S1-Ev02-DTI-SIns y Req1-S1-Ev02-DTI-EIns, respectivamente, después del intervalo Req1-S0-Ev01-DTI.	verificaDisponibilidadDe(Valentino, Req1-S1-Ev02) verificaDisponibilidadDe(Req1-S1-Ev02, Libro1) during(Req1-S1-Ev02, Req1-S1-Ev02-DTI) hasBegining(Req1-S1-Ev02-DTI, Req1-S1-Ev02-DTI-SIns) hasEnd(Req1-S1-Ev02-DTI, Req1-S0-Ev02-DTI-EIns) intervalAfter(Req1-S1-Ev02-DTI, Req1-S0-Ev01-DTI)
3	El empleado Valentino crea la reservación Reservacion_prestamo_Heidi en el evento Req1-S2-Ev3_1, que se llevará a cabo durante el intervalo de tiempo Req1-S2-Ev03_1-DTI cuyos intervalos de inicio y fin son Req1-S2-Ev03_1-DTI-SIns y Req1-S2-Ev3_1-DTI-EIns,	crea(Valentino, Req1-S2-Ev03_1) crea(Req1-S2-Ev03_1, Reservacion_prestamo_Heidi) during(Req1-S2-Ev03_1, Req1-S2-Ev03_1-DTI) hasBegining(Req1-S2-Ev03_1-DTI, Req1-S2-Ev03_1-DTI-SIns) hasEnd(Req1-S2-Ev03_1-DTI, Req1-S2-Ev03_1-DTI-EIns) intervalAfter(Req1-S2-Ev03_1-DTI, Req1-S1-Ev02-DTI)

	respectivamente, después del intervalo Req1-S1-Ev02-DTI.	
4	El empleado Valentino actualiza la entidad Libro1 en el evento Req1-S2-Ev3_2, que se llevará a cabo durante el intervalo de tiempo Req1-S2-Ev03_2-DTI cuyos intervalos de inicio y fin son Req1-S2-Ev03_2-DTI-SIns y Req1-S2-Ev3_2-DTI-EIns, respectivamente, después del intervalo Req1-S1-Ev02-DTI.	actualiza(Valentino, Req1-S2-Ev03_2) actualiza(Req1-S2-Ev03_2,Libro1) during(Req1-S2-Ev03_2, Req1-S2-Ev03_2-DTI) hasBeginning(Req1-S2-Ev03_2-DTI, Req1-S2-Ev03_2-DTI-SIns) hasEnd(Req1-S2-Ev03_2-DTI, Req1-S2-Ev03_2-DTI-EIns) intervalAfter(Req1-S2-Ev03_2-DTI, Req1-S1-Ev02-DTI)
5	El empleado Valentino expide el contrato Contrato_Heidi en el evento Req1-S3-Ev4, que se llevará a cabo durante el intervalo de tiempo Req1-S3-Ev4-DTI cuyos intervalos de inicio y fin son Req1-S3-Ev4-DTI-SIns y Req1-S3-Ev4-DTI-EIns, respectivamente, después del intervalo Req1-S2-Ev3_1-DTI.	expide(Valentino, Req1-S3-Ev04) expide(Req1-S3-Ev04,Contrato_Heidi) during(Req1-S3-Ev04, Req1-S3-Ev04-DTI) hasBeginning(Req1-S3-Ev04-DTI, Req1-S3-Ev04-DTI-SIns) hasEnd(Req1-S3-Ev04-DTI, Req1-S3-Ev04-DTI-EIns) intervalAfter(Req1-S3-Ev04-DTI, Req1-S2-Ev3_1-DTI)
6
7
...

Mediante el asistente del plugin Chronos, se pueden crear de manera sencilla los axiomas de propiedades de objeto. Para ello se debe contar con los individuos a relacionar, así como la propiedad de objeto temporal a utilizar. Para crear las aserciones de propiedades de objetos, desde la interfaz de Chronos se selecciona el individuo a relacionar, por ejemplo, en el panel *Individuals by type*, se selecciona el individuo “Heidi”. En el panel *Individuals>Property Assertions >Object property assertions*, se da clic en el botón *Add* y aparecerá un cuadro de dialogo desplegado por el asistente de Chronos para indicar las propiedades de la aserción. En la Figura 14, aparece la pantalla del paso 2 del asistente, en donde se indica la relación “reserva” mediante la cual relacionaremos al individuo “Heidi” con la entidad “Libro1”, es decir, el cliente Heidi reserva la entidad Libro1.

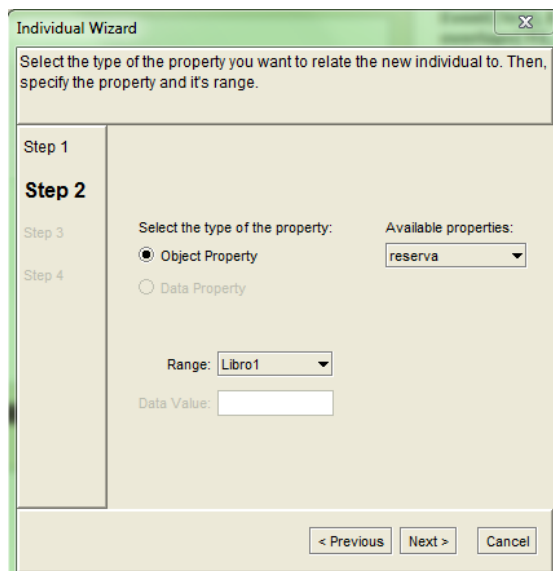


Figura 14. Establecer relación.

Al dar clic en el botón *Next*, aparecerá la siguiente pantalla del asistente mostrada en la Figura 15, en donde indicaremos el intervalo en el que se llevará a cabo el momento de reservar, o si el intervalo sucede después de otro momento en específico.

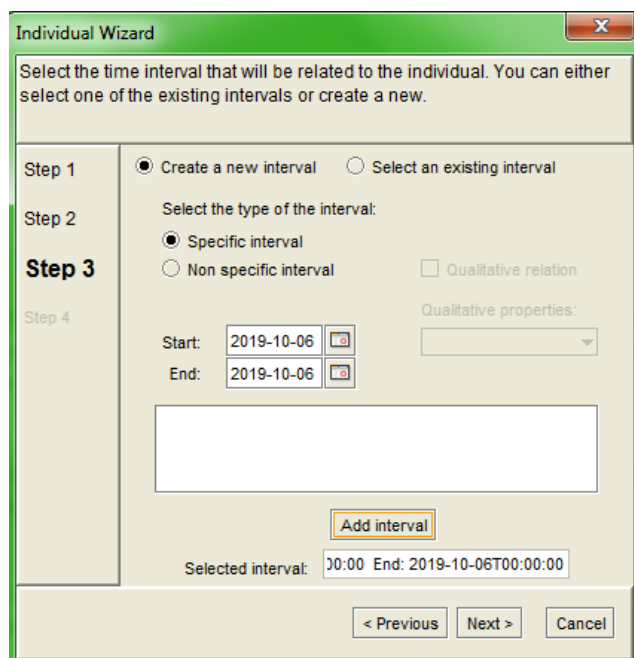


Figura 15. Selección del intervalo en el que se realizará la relación.

Al dar clic en el botón *Next*, el asistente mostrará la pantalla de la Figura 16 que mostrará el resumen de la aserción de propiedad de objeto.



Figura 16. Resumen de la aserción de la propiedad de objeto.

En el panel *Individuals > Property Assertions > Object property assertions* de la Figura 17, aparecerá la aserción de propiedad que se creó.

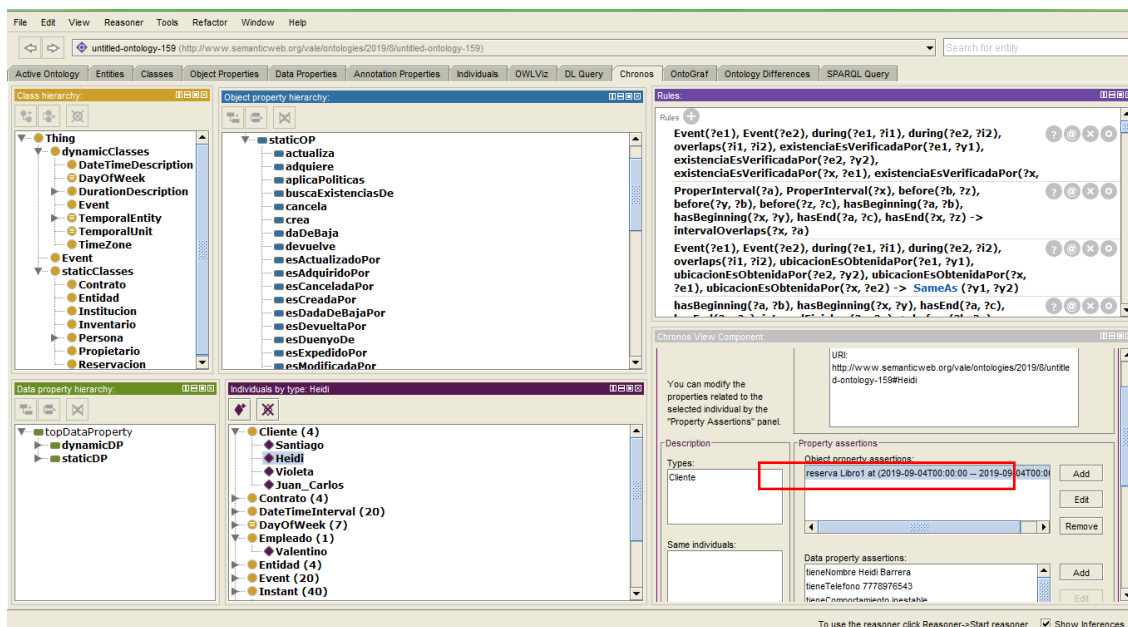


Figura 17. Aserción de la propiedad de objeto reflejada en la interfaz de Chronos.

4.2.6.2. Verificar la consistencia de la secuencia de interpretaciones en la OTPAE compuesta

Una vez agregadas las secuencias de interpretaciones (SoIs), que satisfacen a un requerimiento funcional, en la ontología temporal, es necesario verificar la consistencia de la OTPAE, con la finalidad de verificar que la integración de las secuencias de interpretaciones no genere alguna inconsistencia en la ontología temporal compuesta.

En esta actividad se verificará la consistencia semántica de las secuencias de interpretaciones desde el editor Protégé mediante el razonador Pellet, quien solicitará la implementación del razonamiento SOWL (*Spatio-Temporal Ontology Web Language*) en la ontología, para trabajar con reglas SWRL y comprobar que no existan contradicciones en la secuencia de interpretaciones.

Debido a que hasta el momento el razonador Pellet no soporta del todo el diagnóstico y resolución de errores y únicamente es capaz de indicar la existencia de inconsistencias, será necesario revisar la secuencia de interpretaciones y verificar que los instantes de inicio y fin de propiedad de objeto temporal estén indicados adecuadamente en la secuencia representada y realizar las actualizaciones necesarias hasta obtener una versión consistente y estable de la ontología OTPAE.

Por lo tanto, en esta actividad, se utiliza el procedimiento de la 2da verificación de consistencia semántica temporal propuesta en (Piedrola Domínguez N., 2019, pp 341). Es importante llevar un registro de la verificación por lo que se utilizará el conjunto de documentos descritos en el *Anexo E. Documentación TSCC* y que fueron propuestos en (Piedrola Domínguez N., 2019) para documentar la Verificación de Consistencia Semántica Temporal (TSCC).

Si la ontología OTPAE es consistente, la SoIs ALC puede reducirse a una trayectoria de estado, un componente necesario para llevar a cabo la actividad 4.2.6.3. *Reducir la secuencia de interpretaciones a una ruta de computación*.

4.2.6.3. Reducir la secuencia de interpretaciones a una ruta de computación

Una vez obtenida la ontología OTPAE consistente, en esta actividad se hará una reducción de las secuencias interpretaciones temporales a una trayectoria de estados como se especifica en (Piedrola Domínguez N., 2019, pp 78), esta trayectoria de estados o ruta de computación es un componente generalmente usado en la verificación de satisfacibilidad de fórmulas de la TL LTL.

La trayectoria de estados contiene una secuencia basada en las proposiciones de cada ontología temporal, estas proposiciones se obtienen durante la fase de combinación de

fórmulas ALC_{LTL} cuyos valores de verdad dependerán de su interpretación temporal en la ontología OTPAE compuesta.

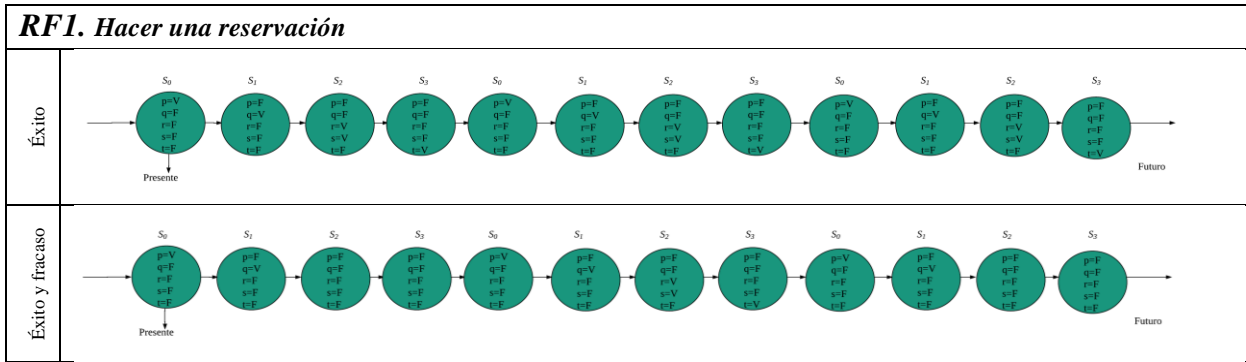
Por ejemplo, en el caso de la proposición “ p = El cliente reserva una entidad”, en la ontología compuesta tenemos la aserción de propiedad de objeto que establece que “El cliente Heidi reserva la entidad Libro1”, en este caso tendremos que verificar en la ontología OTPAE que efectivamente Heidi es un individuo de la clase Cliente y que Libro1 es un individuo de la clase Entidad. De ser así, asignaremos el valor de verdad “Verdadero” a la proposición porque existe un ejemplar de ella en la secuencia de interpretaciones, en caso contrario asignaremos el valor de verdad “Falso”. Cuando se dé el caso de que la proposición no coincide en la interpretación temporal, entonces también se le asignará el valor “Falso”, sin descartar su presencia en la trayectoria de estados. Los valores de verdad para las proposiciones del RF *Hacer una reservación* del patrón de análisis compuesto, se observan en la Tabla 21, que se utilizan para generar la trayectoria de estados mostrada en la Tabla 22.

Tabla 21. Valores de verdad de las proposiciones de acuerdo a la secuencia de interpretaciones para el RF. Hacer una reservación.

Momento	Proposición	¿Qué sucede o sucederá?	Valor de verdad
1	p	El cliente <i>Heidi</i> reserva la entidad <i>Libro1</i>	Verdadero
2	q	El empleado <i>Valentino</i> verificará la disponibilidad de la entidad <i>Libro1</i>	Verdadero
3	r	El empleado <i>Valentino</i> (no) creará la <i>Reservación_prestamo_Heidi</i> .	Verdadero/Falso
3	s	El empleado <i>Valentino</i> (no) actualizará la entidad <i>Libro1</i> .	Verdadero/Falso
4	t	El empleado <i>Valentino</i> (no) expedirá un <i>Contrato_Heidi</i> .	Verdadero/Falso
5	p	El cliente <i>Juan_Carlos</i> reserva la entidad <i>Libro2</i>	Verdadero
6	q	El empleado <i>Valentino</i> verificará la disponibilidad de la entidad <i>Libro2</i>	Verdadero
6	r	El empleado <i>Valentino</i> (no) creará la <i>Reservación_prestamo_JuanCarlos</i> .	Verdadero/Falso
7	s	El empleado <i>Valentino</i> (no) actualizará la entidad <i>Libro2</i> .	Verdadero/Falso
8	t	El empleado <i>Valentino</i> (no) expedirá un <i>Contrato_JuanCarlos</i> .	Verdadero/Falso
...	p
...	q
...	r
...	s
...	t

P. ej., para el caso de éxito, las interpretaciones 1, 2, 3 y 4 (momento 1, 2, 3 y 4) se llevan a cabo en los estados S_0 , S_1 , S_2 y S_3 en donde el valor de verdad de las proposiciones atómicas dependen del estado en el que suceden: p ocurre en el estado S_0 , q ocurre en el estado S_1 , r y s ocurren en el estado S_2 y t ocurre en el estado S_3 .

Tabla 22. Representación gráfica de la trayectoria de estados en la secuencia de interpretaciones para el RF. Hacer una reservación.



4.2.7. Verificación de la formalización

En esta fase, se realiza la verificación de la satisfacibilidad de las fórmulas LTL de las especificaciones formales de la actividad 4.2.5. *Combinación de fórmulas ALCLTL*.

4.2.7.1. Verificar la satisfacibilidad de la especificación en la ruta de computación

En esta actividad, la fórmula LTL deberá evaluarse en la ruta de computación considerada para verificar su satisfacibilidad. La verificación de satisfacibilidad de fórmulas de la LTL es una tarea que implica altas capacidades de razonamiento inductivo y deductivo. Sin embargo, en (Piedrola Domínguez N., 2019) se propone el uso de una herramienta para la verificación automática de modelos (MC, por sus siglas en inglés, *Model Checking*) denominada nuXmv en su versión 1.1.1, con la cual se puede agilizar esta tarea y obtener buenos resultados en la evaluación de fórmulas, sobre rutas de computación, como autómatas deterministas de estado finito.

nuXmv es un verificador de modelos simbólico bien estructurado, abierto (distribuido bajo licencia OpenSource) y flexible con plataforma documentada sobre técnicas de verificación de modelos basadas en la satisfacibilidad proposicional. Este verificador extiende la funcionalidad original de NuSMV. Permite la representación de sistemas de estado finito síncronos o asíncronos y el análisis de especificaciones expresadas en las lógicas temporales CTL y LTL. La interacción con el usuario es a través de una interface textual, así como también en modo batch, nuXmv está disponible para las plataformas Linux: 32-bit x86, 64-bit x86, Mac OS X:64-bit x86 y Windows: Windows x86-32, Windows x86-64. Para mayor

información se sugiere consultar <https://nuxmv.fbk.eu/>, donde se encuentra disponible la documentación completa de nuXmv, así como la herramienta para descargar.

Para llevar a cabo la verificación de la formalización, nuXmv requiere de un archivo con extensión .smv, el cual contiene como modelo un sistema de transición. Es por ello que se modela la ruta de computación como un autómata determinista sobre el cual se podrá evaluar la fórmula LTL.

Por cada trayectoria (éxito, éxito-fracaso), se debe modelar el autómata y crear el respectivo programa SMV que lo represente. Por ejemplo, en el RF Hacer una reservación, los autómatas para las trayectorias de éxito y éxito y fracaso (π' y π'') se muestran en la Figura 18 y en la Figura 19, respectivamente. Además, en la Tabla 23 se muestra el respectivo programa SMV que representa la trayectoria π' y en la Tabla 24 el programa SMV que representa la trayectoria π'' .

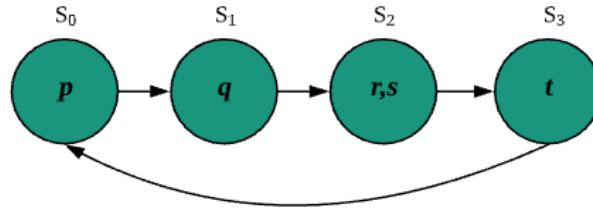


Figura 18. Autómata para la trayectoria de estados π' , de éxito.

Tabla 23. Código fuente del programa Req1-otpaePAIyER-E.smv

<pre>MODULE main</pre>	<p>Sistema de transición en el lenguaje SMV.</p>
<pre>VAR state : {s0,s1,s2,s3}; cliente : {reservaEntidad, none}; empleado_1a : {verificaDisponibilidadDeEntidad, creaReservacion, expideContrato, none}; empleado_1b : {actualizaEntidad, none};</pre>	<p>Párrafo de declaración de variables de estado.</p>
<pre>ASSIGN init(state) := s0; next(state) := case (state = s0) & (cliente = reservaEntidad) & (empleado_1a = none) & (empleado_1b = none): s1; (state = s0) : s1; (state = s1) & (cliente = none) & (empleado_1a = verificaDisponibilidadDeEntidad) & (empleado_1b = none) : s2; (state = s1) : s2; (state = s2) & (cliente = none) & (empleado_1a = creaReservacion) & (empleado_1b = actualizaEntidad) : s3; (state = s2) : s3; (state = s3) & (cliente = none) & (empleado_1a =</pre>	<p>Párrafo de asignación de valores iniciales (init) a variables de estado y definición de relaciones de transición (next) con base en expresiones condicionales dentro de casos (case/esac).</p>

<pre> expideContrato) & (empleado_1b = none) : s0; (state = s3) : s0; TRUE : state; esac; </pre>	
<pre> init(cliente) := reservaEntidad; next(cliente) := case (state = s0) : reservaEntidad; (state = s1) : none; (state = s2) : none; (state = s3) : none; TRUE : cliente; esac; init(empleado_1a) := none; next(empleado_1a) := case (state = s0) : none; (state = s1) : verificaDisponibilidadDeEntidad; (state = s2) : creaReservacion; (state = s3) : expideContrato; TRUE : empleado_1a; esac; init(empleado_1b) := none; next(empleado_1b) := case (state = s0) : none; (state = s1) : none; (state = s2) : actualizaEntidad; (state = s3) : none; TRUE : empleado_1b; esac; </pre>	
<pre> LTLSPEC G (X cliente = reservaEntidad -> (X X empleado_1a = verificaDisponibilidadDeEntidad & F (!X X X (empleado_1a = creaReservacion & empleado_1b = actualizaEntidad) & !X X X X empleado_1a = expideContrato))); </pre>	<p>Especificación lógica basada en la TL LTL.</p>

Para una mejor comprensión del código fuente del programa SMV, se recomienda revisar la sección 2. *Verificación de satisfacibilidad de especificaciones de comportamiento del Anexo F. Herramientas utilizadas* que se encuentra en (Piedrola Domínguez N., 2019, pp 344).

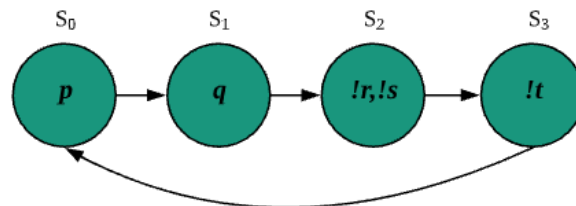


Figura 19. Autómata para la trayectoria de estados π'' , de fracaso.

Tabla 24. Código fuente del programa Req1-otpaEPAIyER-EF.smv

<pre>MODULE main</pre>	<p>Sistema de transición en el lenguaje SMV.</p>
<pre>VAR state : {s0,s1,s2,s3,s0_1,s1_1,s2_1,s3_1,s0_2,s1_2,s2_2,s3_2,s0_3,s1_3,s2_3,s3_3}; cliente : {reservaEntidad, none}; empleado_1a : {verificaDisponibilidadDeEntidad, creaReservacion, expideContrato, none}; empleado_1b : {actualizaEntidad, none};</pre>	<p>Párrafo de declaración de variables de estado.</p>
<pre>ASSIGN init(state) := s0; next(state) := case -- caso de fracaso 1 (state = s0) & (cliente = reservaEntidad) & (empleado_1a = none) & (empleado_1b = none): s1; (state = s0) : s1; (state = s1) & (cliente = none) & (empleado_1a = verificaDisponibilidadDeEntidad) & (empleado_1b = none): s2; (state = s1) : s2; (state = s2) & (cliente = none) & (empleado_1a = creaReservacion) & (empleado_1b = actualizaEntidad) : s3; (state = s2) : s3; (state = s3) & (empleado_1a = expideContrato) : s0; (state = s3) : s0_1; -- caso de éxito 1 (state = s0_1) & (cliente = reservaEntidad) & (empleado_1a = none) & (empleado_1b = none) : s1_1; (state = s0_1) : s1_1; (state = s1_1) & (cliente = none) & (empleado_1a = verificaDisponibilidadDeEntidad) & (empleado_1b = none) : s2_1; (state = s1_1) : s2_1; (state = s2_1) & (cliente = none) & (empleado_1a = creaReservacion) & (empleado_1b = actualizaEntidad) : s3_1; (state = s2_1) : s3_1; (state = s3_1) & (empleado_1a = expideContrato) : s0_2; (state = s3_1) : s0_2; -- caso de fracaso 2 (state = s0_2) & (cliente = reservaEntidad) & (empleado_1a = none) & (empleado_1b = none) : s1_2; (state = s0_2) : s1_2; (state = s1_2) & (cliente = none) & (empleado_1a = verificaDisponibilidadDeEntidad) & (empleado_1b = none) : s2_2; (state = s1_2) : s2_2; (state = s2_2) & (cliente = none) & (empleado_1a = creaReservacion) & (empleado_1b = actualizaEntidad): s3_2; (state = s3_2) & (empleado_1a = expideContrato) : s0_3; (state = s3_2) : s0_3; -- caso de éxito 2 (state = s0_3) & (cliente = reservaEntidad) & (empleado_1a = none) & (empleado_1b = none) : s1_3; (state = s0_3) : s1_3; (state = s1_3) & (cliente = none) & (empleado_1a =</pre>	<p>Párrafo de asignación de valores iniciales (init) a variables de estado y definición de relaciones de transición (next) con base en expresiones condicionales dentro de casos (case/esac).</p>

<pre> verificaDisponibilidadDeEntidad) & (empleado_1b = none) : s2_3; (state = s2_3) & (cliente = none) & (empleado_1a = creaReservacion) & (empleado_1b = actualizaEntidad): s3_3; (state = s3_3) & (empleado_1a = expideContrato) : s0; (state = s3_3) : s0; TRUE : state; esac; </pre>	
<pre> init(cliente) := reservaEntidad; next(cliente) := case -- caso de fracaso 1 (state = s0) : reservaEntidad; (state = s1) : none; (state = s2) : none; (state = s3) : none; -- caso de éxito 1 (state = s0_1) : reservaEntidad; (state = s1_1) : none; (state = s2_1) : none; (state = s3_1) : none; -- caso de fracaso 2 (state = s0_2) : reservaEntidad; (state = s1_2) : none; (state = s2_2) : none; (state = s3_2) : none; -- caso de éxito 2 (state = s0_3) : reservaEntidad; (state = s1_3) : none; (state = s2_3) : none; (state = s3_3) : none; TRUE : cliente; esac; init(empleado_1a) := none; next(empleado_1a) := case -- caso de fracaso 1 (state = s0) : none; (state = s1) : verificaDisponibilidadDe; (state = s2) : none; (state = s3) : none; -- caso de éxito 1 (state = s0_1) : none; (state = s1_1) : verificaDisponibilidadDe; (state = s2_1) : creaReservacion; (state = s3_1) : expideContrato; -- caso de fracaso 2 (state = s0_2) : none; (state = s1_2) : verificaDisponibilidadDe; (state = s2_2) : none; (state = s3_2) : none; -- caso de éxito 2 (state = s0_3) : none; (state = s1_3) : verificaDisponibilidadDe; </pre>	

<pre> (state = s2_3) : creaReservacion; (state = s3_3) : expideContrato; TRUE : empleado_1a; esac; init(empleado_1b) := none; next(empleado_1b) := case -- caso de fracaso 1 (state = s0) : none; (state = s1) : none; (state = s2) : none; (state = s3) : none; -- caso de éxito 1 (state = s0_1) : none; (state = s1_1) : none; (state = s2_1) : actualizaEntidad; (state = s3_1) : none; -- caso de fracaso 2 (state = s0_2) : none; (state = s1_2) : none; (state = s2_2) : none; (state = s3_2) : none; -- caso de éxito 2 (state = s0_3) : none; (state = s1_3) : none; (state = s2_3) : actualizaEntidad; (state = s3_3) : none; TRUE : empleado_1b; esac; </pre>	
<pre> LTLSPEC G (X cliente = reservaEntidad -> (X X empleado_1a = verificaDisponibilidadDeEntidad & F (!X X X (empleado_1a = creaReservacion & empleado_1b = actualizaEntidad) & !X X X X empleado_1a = expideContrato))); </pre>	<p>Especificación lógica basada en la TL LTL.</p>

Para una mejor comprensión del código fuente del programa SMV, se recomienda revisar la sección 2. *Verificación de satisfacibilidad de especificaciones de comportamiento* del Anexo F. *Herramientas* utilizadas que se encuentra en (Piedrola Domínguez N., 2019, pp 344).

Cada programa en lenguaje SMV se debe leer y ejecutar desde la interfaz textual de nuXmv mediante los comandos descritos en la Tabla 25. Posteriormente, se debe ejecutar el comando de la Tabla 26 para evaluar la satisfacibilidad de la fórmula LTL φ en cada autómeta.

Tabla 25. Comandos para la ejecución del autómata.

No.	Comando	Descripción
1	<i>nuXmv -int</i>	Activa el modo interactivo
2	<i>nuXmv > read_model -i Req1-otpaePAIyER-E.smv</i>	Dentro de la carpeta de nuXmv, leer un modelo (o autómata) a partir del archivo indicado.
3	<i>nuXmv > flatten_hierarchy</i>	Alisa la jerarquía de módulos. Instancia módulos y procesos sustituyendo los parámetros actuales por parámetros formales.
4	<i>nuXmv > encode_variables</i>	Construye las variables necesarias para compilar el modelo en un modelo basado en el desarrollo dirigido por comportamiento (BDD, Behavior Driven Development)
5	<i>nuXmv > build_model</i>	Convierte la jerarquía de módulos alisada en un modelo BDD (estados iniciales, invariantes y relaciones de transición).
6	<i>nuXmv > pick_state -i</i>	Toma como estado actual el estado seleccionado por el usuario de acuerdo con al conjunto inicial de estados.
7	<i>nuXmv > simulate -p</i>	Realiza la simulación a partir del estado seleccionado como actual.

Tabla 26. Comando para realizar la verificación SC de una fórmula LTL.

No.	Comando	Descripción
1	<i>nuXmv > check_ltlspec -p "G (X cliente = reservaEntidad -> (X X empleado_1a = verificaDisponibilidadDeEntidad & F(!X X X (empleado_1a = creaReservacion & empleado_1b = actualizaEntidad) & !X X X X empleado_1a = expideContrato)))"</i>	Realiza la verificación de la fórmula LTL en el modelo.

Para este caso en particular, el resultado de la ejecución del comando de la Tabla 26 en ambos autómatas es verdadero (true), lo que indica que la fórmula es satisfacible en los modelos que representan el comportamiento basado en el RF1. Hacer una reservación ($\pi' \models \varphi$ y $\pi'' \models \varphi$, respectivamente) y no se generan contraejemplos. Si sucediera lo contrario, significa que la fórmula LTL no se satisface en la trayectoria considerada ($\pi' \not\models \varphi$ o $\pi'' \not\models \varphi$, respectivamente) y es necesario volver a generar los componentes necesarios sobre los cuales se evaluará nuevamente la satisfacibilidad de la fórmula LTL φ . Para llevar un registro de la verificación, se utiliza el conjunto de documentos descritos en el Anexo F. Documentación SC, estos documentos fueron propuestos en (Piedrola Domínguez N., 2019) para documentar la Verificación de Satisfacibilidad (SC).

4.2.8. Instanciación de la ontología

Una vez definidas las especificaciones de comportamiento que deben satisfacerse en la ontología OTPAE compuesta, en esta fase se crean los individuos (aserciones de clases y aserciones de propiedades de objetos temporales) de la ontología necesarios para representar trayectorias de estado. Un ejemplo de las aserciones de clases y de propiedades de objetos se

presenta en la Tabla 27, cuya tabla muestra la equivalencia entre la sintaxis DL y la sintaxis funcional de OWL-DL.

Tabla 27. Ejemplo de aserciones de clases y de propiedades de objetos en el RF Hacer una reservación del PAIyER.

Sintaxis DL	Sintaxis Funcional
-- Aserciones de clase --	
<i>Cliente(Heidi)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Heidi)</i>
<i>Event(Req1-Ev1)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req1-Ev1)</i>
-- Aserciones de propiedades de objeto --	
<i>reserva(Heidi,Req1-Ev1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:reserva domain-PAIyER:Heidi domain-PAIyER:Req1-Ev1)</i>
<i>reserva(Req1-Ev1,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:reserva domain-PAIyER:Req1-Ev1 domain-PAIyER:Libro1)</i>

P. ej., la aserción de clase *Cliente(Heidi)*, indica que Heidi es un individuo de la clase Cliente y la aserción de clase *Event(Req1-Ev1)*, indica que Req1-Ev1 es un individuo de la clase Event. De la misma manera, la aserciones de propiedad de objeto *reserva(Heidi,Req1-Ev1)* y *reserva(Req1-Ev1,Libro1)*, indican que el individuo Heidi reserva el individuo Libro1 a través del individuo Req1-Ev1.

Los individuos en cada trayectoria de comportamiento en el patrón de análisis PAIyER, se describen en la sección 1.2. *Componente extensional del Anexo D. Ontología OTPAE del patrón de análisis PAIyER*, cuyas tablas muestran la equivalencia entre la sintaxis DL y la sintaxis funcional de OWL-DL.

En este capítulo, se presentó el método COMOTPA para combinar ontologías temporales de patrones de análisis, se describieron las fases y actividades a realizar para obtener una ontología temporal de un patrón de análisis compuesto. En el *Capítulo VI. Caso de estudio: Combinación de la OTPAE del PAI con la OTPAE del PAER*, se presenta un caso de estudio, en donde se realiza la combinación de la OTPAE del PAI (creada en el *Capítulo V. Creación de la Ontología Temporal del Patrón de Análisis para Inventarios*) con la OTPAE del PAER, utilizando el método COMOTPA, descrito en este capítulo.

Capítulo V. Creación de la Ontología Temporal del Patrón de Análisis para Inventarios

En este capítulo, se presentan los resultados obtenidos en la creación de la Ontología Temporal del Patrón de Análisis para Inventarios (OTPAE del PAI). Se utiliza la metodología MECOTPA desarrollada en (Piedrola Domínguez N., 2019) para crear la ontología temporal del PAI.

5.1. Introducción

Actualmente, solamente existen dos ontologías temporales documentadas de patrones de análisis, creadas con la metodología MECOTPA: La Ontología Temporal del Patrón de Análisis de Entidades Reutilizables (OTPAE del PAER) y la Ontología Temporal del Patrón de Análisis de Cuentas (OTPAE del PAC). Sin embargo, para realizar el caso de estudio descrito en el *Capítulo VI. Caso de estudio: Combinación de la OTPAE del PAI con la OTPAE del PAER*, se requirió de otra ontología temporal, además de las ya existentes.

Por tal motivo, se complementó la descripción del patrón de análisis para inventarios (PAI) con aspectos temporales, mediante una ontología temporal que representa su comportamiento, para combinarla con alguno de los dos patrones de análisis que se tienen descritos, hasta ahora, en nuestro grupo de investigación.

El PAI es un modelo genérico a partir de la abstracción de un inventario real y puede extenderse para complementar requerimientos detallados, mantiene un seguimiento de las cantidades y ubicaciones de artículos en un inventario, e incluye su actualización, de acuerdo a las diferentes etapas de manufactura o producción. Fue descrito por primera vez en (Fernandez, 1999) y en (Piedrola, 2019) se utilizó para identificar el tipo de razonamiento temporal descrito en el comportamiento.

5.2. Metodología para la Construcción de Ontologías Temporales de Patrones de Análisis (MECOTPA)

La metodología MECOTPA es una actualización de la metodología MECOPA desarrollada por (López-Ramírez, 2015) y (Juárez Hernández, 2016) que facilita a los ingenieros de software la inclusión de comportamiento dinámico en las ontologías de patrones de análisis mediante aspectos de temporalidad. MECOTPA se compone de diez fases, y cada una contiene un conjunto de actividades, que permite obtener al final una ontología temporal válida y consistente, de algún patrón de análisis en específico.

En la Figura 20 se presentan las diez fases que componen la metodología MECOTPA y que realizándolas, para describir al Patrón de Análisis para Inventarios (PAI), se obtendrá su ontología temporal (OTPAE).

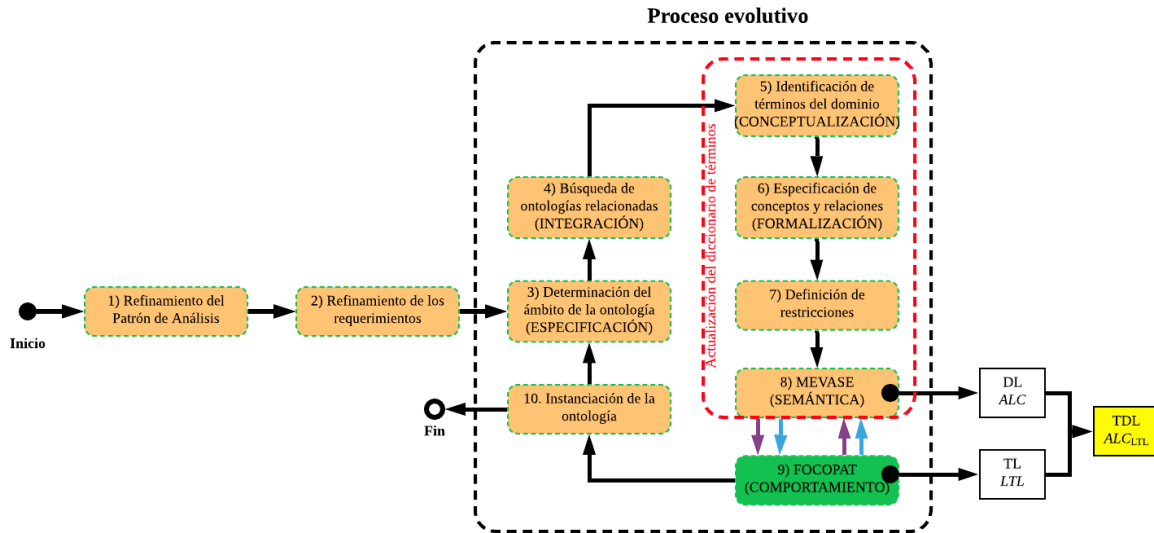


Figura 20. Fases de la metodología MECOTPA.

A continuación, se presentan los resultados obtenidos en cada una de las fases de la metodología MECOTPA durante la construcción de la Ontología Temporal del Patrón de Análisis para Inventarios (OTPAE del PAI).

5.2.1. Refinamiento del patrón de análisis

Con base en la metodología MECOTPA, en esta fase se realizó una descripción del patrón de análisis PAI usando la plantilla PPAAE (García, Jiménez, Brito, & Ramírez, 2015) y se creó un nuevo individuo de la ontología OPPAAE (García et al., 2015). Este individuo de la OPPAAE es una descripción del PAI, usando el lenguaje OWL-DL. En la Figura 21, se presenta como ejemplo los valores de propiedades para la Clase “Problema” correspondiente al punto 5 de la plantilla PPAAE.

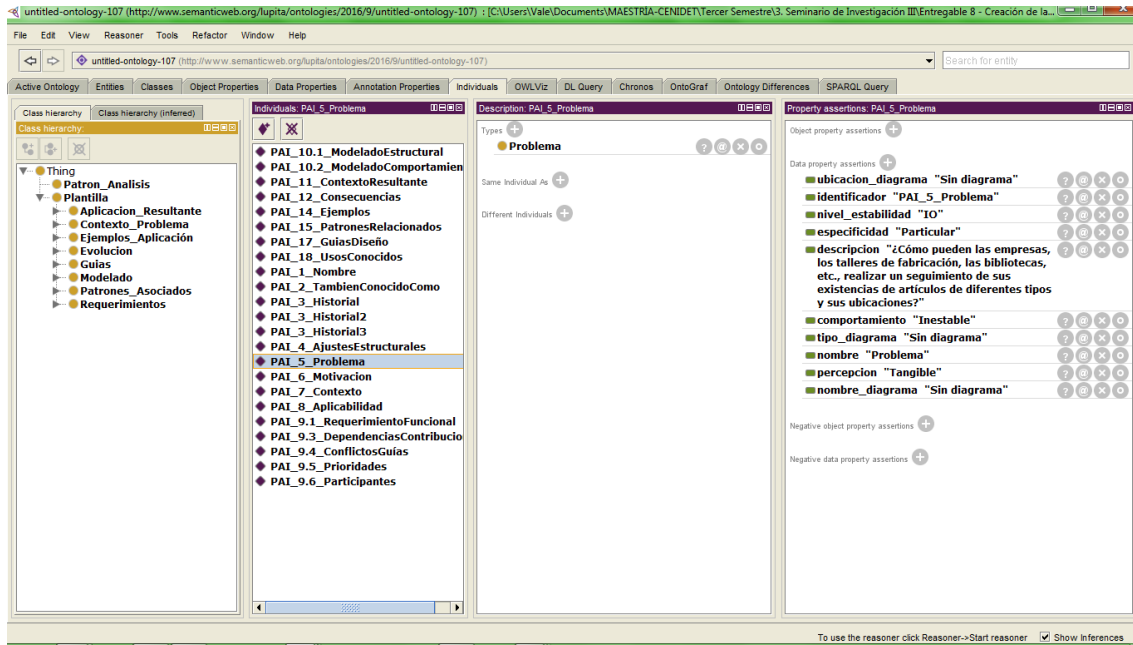


Figura 21. Valores de propiedades para la Clase “Problema” correspondiente al punto 5 de la plantilla PPAE.

Posteriormente, tomando como referencia la descripción del PAI con la plantilla PPAE, y el individuo de la ontología OPPAE que describe el patrón de análisis PAI, usando el lenguaje OWL-DL, se elaboró el documento de razonamiento temporal (DTR), un documento de apoyo para la identificación del tipo de razonamiento temporal empleado en las especificaciones de comportamiento del patrón de análisis PAI. Se siguió el proceso propuesto en (Piedrola Domínguez N., 2019) para construir la tabla de razonamiento temporal.

En la Tabla 28, se describe un fragmento de la tabla de razonamiento temporal obtenida a partir de la plantilla PPAE, basado en el RF Realizar un seguimiento de las ubicaciones de los artículos. Una versión completa del documento DRT del PAI viene adjunto en el directorio *Entregable 8/DRT/*, almacenado en el disco que acompaña a esta tesis (para mayor referencia consultar el *Anexo G. Contenido del disco*).

Tabla 28. Fragmento de la tabla de razonamiento temporal en el patrón de análisis PAI (sección PPAE).

Punto Plantilla PPAE	Especificación	Clarificación	Razonamiento				Proposiciones atómicas	Fórmula LTL
			Cualitativo		Cuantitativo			
			Puntos	Intervalos	Puntos	Intervalos		
Requerimientos Funcionales	RF 3 Realizar un seguimiento de las ubicaciones de los artículos. El inventario debe registrar la distribución de los artículos en ubicaciones específicas.	Realizar un seguimiento de las ubicaciones de los artículos. El inventario debe registrar la distribución de los artículos en ubicaciones específicas.	✓	✗	✗	✗	$p =$ Realizar un seguimiento de las ubicaciones de los artículos. $q =$ El inventario debe registrar la distribución de los artículos en ubicaciones específicas. $\mathcal{G}(p \rightarrow q)$	

Los resultados en la columna *Razonamiento* de la tabla de razonamiento temporal marcados con ✓, indican que en la especificación de comportamiento dinámico existe un tipo de razonamiento temporal cualitativo basado en puntos.

A continuación, se ejemplifica cómo debe leerse la fórmula LTL propuesta en la especificación clarificada de la Tabla 28.

$$\mathcal{G}(p \rightarrow q)$$

Siempre que (\mathcal{G}) p implica que (\rightarrow) q.

En la Tabla 29, se muestra un resumen de las fórmulas LTL propuestas para las descripciones de comportamiento dinámico en los requerimientos funcionales restantes, que también presentaron un tipo de razonamiento temporal cualitativo.

Tabla 29. Fórmulas LTL propuestas en los RFs 1 y 2 del PAI.

Proposiciones atómicas	Fórmula LTL
1. Separar diferentes tipos de existencias	
<i>p = Separar diferentes tipos de existencias</i>	$\mathcal{F}(p)$
2. Realizar seguimiento de las cantidades de artículos en existencias.	
<p><i>p = Realizar un seguimiento de las cantidades de cada artículo en existencias</i> <i>q = Buscar cantidad enMano</i> <i>r = Buscar cantidad enOrden</i> <i>s = Buscar cantidades reservadas para órdenes</i> <i>t = Buscar Botras cantidades</i></p>	$\mathcal{G}(p \rightarrow (q \vee r \vee s \vee \mathcal{F}t))$

5.2.2. Refinamiento de los requerimientos

En esta fase, se elaboraron los diagramas GRL y UCM correspondientes a los requerimientos funcionales del patrón de análisis PAI y se complementó el documento DRT generado en la sección 5.2.1. *Refinamiento del patrón de análisis*. Se incluyó una tabla (Tabla 30) que describe la sección de la metodología MECOTPA que contiene los diagramas GRL y UCM, de donde se extraen especificaciones de comportamiento y una clarificación de las mismas, a partir de las cuales se identificó el tipo de razonamiento temporal utilizado y las proposiciones atómicas necesarias para formalizar las especificaciones clarificadas.

En la Tabla 30, se describe un fragmento de la tabla de razonamiento temporal que complementa el documento DRT, basado en el RF Separar diferentes tipos de existencias. Una versión completa de esta documentación se encuentra en el directorio *Entregable 8/DRT/*, almacenado en el disco que acompaña a esta tesis (para mayor referencia consultar el *Anexo G. Contenido del disco*).

Tabla 30. Fragmento de la tabla de razonamiento temporal del patrón de análisis PAI (sección MECOTPA).

MECOTPA (Piedrola Domínguez N., 2019)	Especificación	Clarificación	Razonamiento				Proposiciones atómicas	Fórmula LTL
			Cualitativo		Cuantitativo			LTL
			P	I	P	I		
4.2.1.2. Refinamiento de los requerimientos	Diagrama GRL (ver Figura 22)	<p>1. Separar diferentes tipos de existencias</p> <p><i>1.1. Seleccionar artículo.</i> Actividad mediante la cual el empleado selecciona el artículo a inventariar.</p> <p><i>1.2. Identificar tipo de artículo.</i> Actividad mediante la cual el empleado identifica si el artículo es un producto o un componente. Un producto está compuesto de dos o más componentes.</p> <p><i>1.3. Registrar información.</i> Actividad realizada por el empleado para registrar la información correspondiente al artículo. Se registran los tipos siguientes de información:</p> <p><i>1.3.1. Especificar datos del artículo.</i> Actividad mediante la cual el empleado proporciona los datos del artículo a inventariar. Cuando el artículo ya se encuentra registrado solamente se deberá actualizar el número de existencias.</p> <p><i>1.3.2. Especificar ubicación del artículo.</i> Actividad mediante la cual el empleado proporciona la ubicación del artículo a inventariar, es decir, el local de distribución en donde se almacenará el artículo. Cuando el artículo ya se encuentra registrado solamente se deberá actualizar el número de existencias.</p> <p><i>1.3.3. Especificar cantidad.</i> Actividad mediante la cual el empleado especifica la cantidad de artículos a inventariar.</p> <p><i>1.3. Actualizar existencias.</i> Actividad que realiza el empleado para actualizar la cantidad de existencias del artículo seleccionado.</p>	✓	✗	✗	✗	<p>p = El empleado separa diferentes tipos de existencias.</p> <p>q = Seleccionar artículo</p> <p>r = El empleado identifica tipo de artículo</p> <p>s = El empleado registra información</p> <p>t = El empleado especifica datos del artículo</p> <p>u = El empleado especifica ubicación del artículo</p> <p>v = El empleado especifica cantidad de artículos</p> <p>w = El empleado actualiza existencias.</p>	<p>(1) $\mathcal{G}(p \rightarrow (q \wedge r \wedge (s \rightarrow (t \wedge u \wedge v) \wedge w)))$</p> <p>(2) $\mathcal{G}(s \rightarrow (t \wedge u \wedge v))$</p>
	Diagrama UCM (ver Figura 23)	<p>1. Separar diferentes tipos de existencias</p> <p>Siempre que un empleado necesite separar tipos de existencias primeramente deberá seleccionar el artículo, posteriormente deberá verificar si existe el artículo en el inventario. Si el artículo se encuentra registrado entonces deberá actualizar las existencias, si el artículo no se encuentra registrado deberá proporcionar los datos correspondientes y deberá guardar la información</p>	✓	✗	✗	✗	<p>p = Empleado separa diferentes tipos de existencias.</p> <p>q = Empleado selecciona el artículo.</p> <p>r = El artículo se encuentra registrado en el inventario.</p> <p>s = El empleado actualiza existencias artículo.</p> <p>t = El empleado proporciona datos del artículo.</p> <p>u = El empleado guarda la información.</p>	<p>(1) $\mathcal{G}(p \rightarrow (q \wedge r \wedge (r \rightarrow s) \wedge u))$</p> <p>(2) $\mathcal{G}(p \rightarrow (q \wedge r \wedge (\neg r \rightarrow t) \wedge u))$</p>

P = Puntos, I = Intervalos

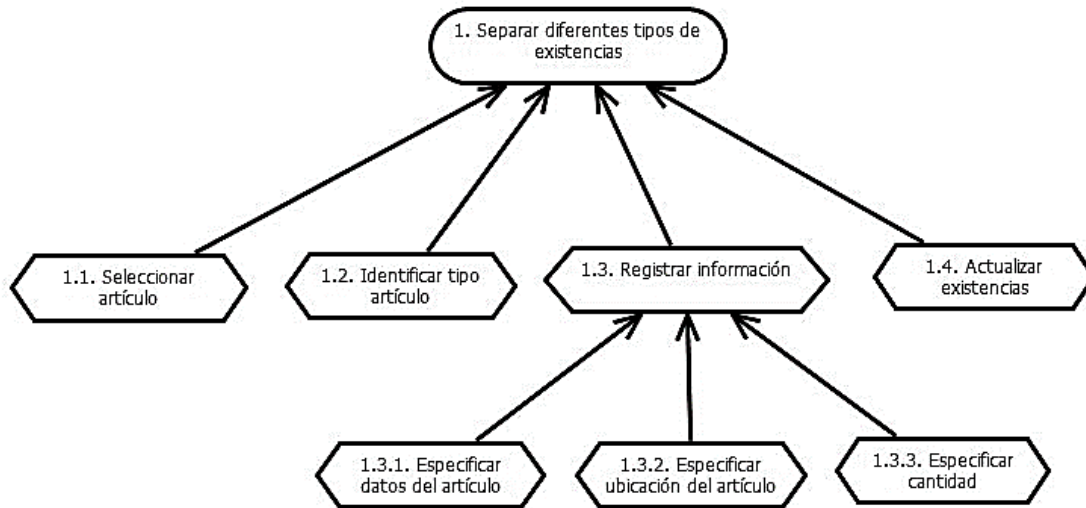


Figura 22. Diagrama GRL del RF. Separar diferentes tipos de existencias.

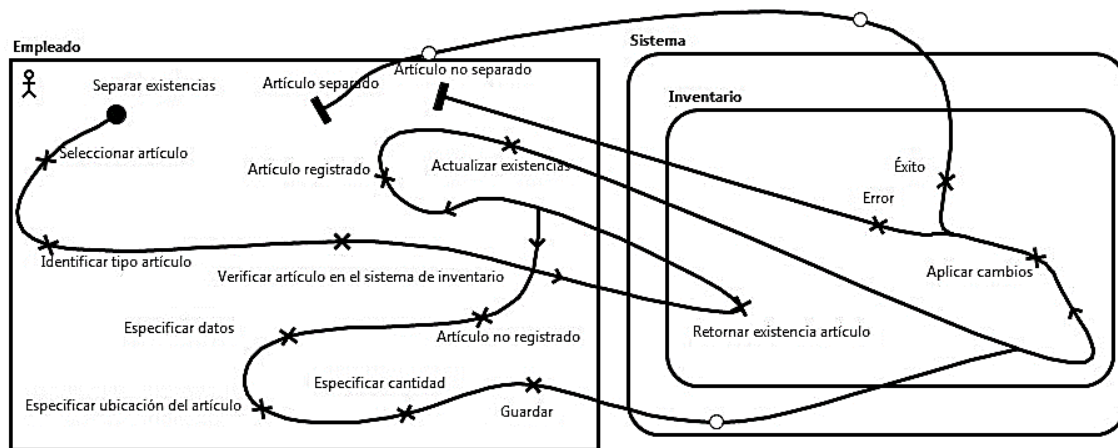


Figura 23. Diagrama UCM del RF. Separar diferentes tipos de existencias.

Al igual que en el ejemplo de la fase anterior (5.2.1. *Refinamiento del patrón de análisis*), los resultados en la columna *Razonamiento* de la tabla de razonamiento temporal marcados con ✓, indican que en las especificaciones de comportamiento dinámico existe un tipo de razonamiento temporal cualitativo basado en puntos. A continuación, se ejemplifica cómo debe leerse la fórmula LTL número (2) propuesta en la especificación clarificada del diagrama GRL, descrito en la Tabla 30:

$$(2) \mathcal{G}(s \rightarrow (t \wedge u \wedge v))$$

Siempre que (\mathcal{G}) s implica que (\rightarrow) t y (\wedge) u y (\wedge) v .

En la Tabla 31 y Tabla 32, se muestra un resumen de las fórmulas LTL propuestas para las descripciones de comportamiento dinámico en los diagramas GRL y UCM de los requerimientos funcionales restantes, en las cuales también se utiliza el razonamiento temporal cualitativo basado en puntos.

Tabla 31. Fórmulas LTL propuestas para las especificaciones de los diagramas GRL de los RFs 2 y 3 del PAI.

Propósiciones atómicas	Fórmula LTL
2. Realizar el seguimiento de las cantidades de artículos en existencias	
<p><i>p</i> = El empleado realiza un seguimiento de las cantidades de cada artículo en existencias. <i>q</i> = El empleado selecciona artículo <i>r</i> = El empleado verifica existencias <i>s</i> = El empleado verifica cantidad de existencias enMano. <i>t</i> = El empleado verifica cantidad de existencias enOrden. <i>u</i> = El empleado verifica cantidad de existencias reservadas.</p>	$\mathcal{G}(p \rightarrow (q \wedge (r \rightarrow (s \vee t \vee u))))$
3. Realizar el seguimiento de las ubicaciones de los artículos.	
<p><i>p</i> = El empleado realiza un seguimiento de las ubicaciones de los artículos. <i>q</i> = El empleado selecciona artículo <i>r</i> = El empleado obtiene la ubicación específica del artículo <i>s</i> = El empleado proporciona datos del artículo.</p>	$\mathcal{G}(p \rightarrow (q \wedge (r \rightarrow s)))$

Tabla 32. Fórmulas LTL propuestas para las especificaciones de los diagramas UCM de los RFs 2 y 3 del PAI.

Propósiciones atómicas	Fórmula LTL
2. Realizar el seguimiento de las cantidades de artículos en existencias	
<p><i>p</i>= el empleado realiza seguimiento de las cantidades de un artículo. <i>q</i>=el empleado selecciona el artículo. <i>r</i>=el empleado especifica el tipo de cantidad de existencias. <i>s</i>=el inventario retorna la cantidad de existencias del artículo. <i>t</i>=el empleado obtiene la cantidad de existencias del artículo.</p>	$\mathcal{G}(p \rightarrow (q \wedge r \wedge s \wedge t))$
3. Realizar el seguimiento de las ubicaciones de los artículos.	
<p><i>p</i>=empleado realiza seguimiento de la ubicación de un artículo. <i>q</i>=el empleado selecciona el artículo. <i>r</i>=empleado ejecuta búsqueda. <i>s</i>=el inventario retorna la ubicación del artículo. <i>t</i>= el empleado obtiene la ubicación del artículo</p>	$\mathcal{G}(p \rightarrow (q \wedge r \wedge s \wedge t))$

5.2.3. Determinación del ámbito de la ontología (Especificación)

Este apartado corresponde a la tercera fase de MECOTPA. Aquí se definen los usuarios, el dominio, el propósito y el tipo de ontología, así como el cuestionario de competencias:

Usuarios

Los usuarios finales previstos para la OPAI son:

- Responsables de análisis en el dominio de la administración de inventarios.
- Desarrolladores de aplicaciones para patrones de análisis.
- Investigadores y expertos en el dominio, para la retroalimentación y mejora continua.

Dominio

La OPAI se desarrolló en el dominio de Administración de inventarios. En una compañía o institución, los artículos pueden ser materiales usados para fabricación o productos terminados. Existe la posibilidad de pérdida o destrucción de existencias. La institución debe ser capaz de realizar un seguimiento del número actual de artículos en existencia.

Propósito

El propósito fundamental de la OPAI es representar, organizar, formalizar y estandarizar el conocimiento relacionado con el dominio del patrón de análisis para la administración de inventarios.

Tipo de ontología

La OPAI es una ontología de dominio.

Cuestionario de competencias

1. ¿Quiénes pueden ser empleados? **R:** Aquéllas personas que estén encargados de administrar el inventario.
2. ¿Qué actores están involucrados en el seguimiento en la cantidad de existencias de un artículo? **R:** empleado y administrador
3. ¿Qué datos se necesitan para dar seguimiento en la cantidad de existencias de un artículo? **R:** nombre artículo, identificador artículo, tipo cantidad (enMano, enOrden, reservados).
4. ¿Dónde se consulta la cantidad de existencias de un artículo? **R:** En el sistema de inventario.
5. ¿Dónde se consulta la ubicación de un artículo? **R:** En el sistema de inventario.
6. ¿Qué datos se necesitan para dar seguimiento a la ubicación de un artículo? **R:** identificador artículo, nombre artículo.
7. ¿Qué datos se necesitan para separar existencias? **R:** identificador artículo, tipo de artículo, cantidad, ubicación
8. ¿Cuál es la diferencia entre seguimiento de las existencias de un artículo y seguimiento de la ubicación de un artículo? **R:** El seguimiento de existencias consiste en verificar la cantidad de existencias de un artículo, estas cantidades pueden ser de tres tipos: existencias enMano, existencias enOrden y existencias reservadas. El seguimiento de la ubicación de un artículo consiste en verificar en dónde se encuentra ubicado cada artículo (ejemplo: al lado de la impresora, en el librero de color negro, entre otras).

9. ¿Cuál es la diferencia entre seguimiento de cantidades de existencias y separación de existencias? **R:** El seguimiento de cantidades de existencias consiste en verificar los diferentes tipos de cantidades de existencias de un artículo y la separación de existencias consiste en separar artículos dependiendo del tipo, es decir, si son productos finales o componentes.
10. ¿Qué se requiere para dar seguimiento en las cantidades de existencias de un artículo? **R:** Que el artículo se encuentre registrado en el sistema de inventario.
11. 8. ¿Qué se requiere para dar seguimiento en la ubicación de un artículo? **R:** Que el artículo se encuentre registrado en el sistema de inventario.

5.2.4. Búsqueda de ontologías relacionadas

A continuación, se lista un conjunto representativo de ontologías de tiempo, enfocadas en los criterios de representación temporal mencionados en la fase *Determinación del ámbito de la ontología (Especificación)* (Fernández López y Gómez Pérez, 2004):

- Upper Cyc
- Ontología de tiempo no restringida
- Ontología de tiempo simple
- Ontología de tiempo Kestrel
- Ontología de combinación superior sugerida (SUMO, *Suggested Upper Merged Ontology*)
- Ontología de tiempo OWL-Time (formalmente DAML-Time).

No se encontraron ontologías relacionadas con los sistemas de inventarios.

5.2.5. Identificación de términos del dominio (Conceptualización)

A continuación, en la Tabla 33 se presentan los conceptos y roles concretos identificados en el patrón de análisis PAI y en la Tabla 34 se presentan los roles abstractos que intervienen en cada requerimiento funcional del patrón de análisis PAI.

Tabla 33. Conceptos y roles concretos en el PAI.

Concepto	Roles concretos (atributos)
Persona	tieneId, tieneNombre, tienEdad, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion
Empleado	tienId, tieneNombre, tienEdad, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion
Institucion	tieneId, tieneNombre, tieneGiro, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion
Inventario	tieneId, tieneNombre, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion
Artículo	tieneId, tieneNombre, tieneTipo, tieneCantidadEnOrden, tieneCantidadEnMano,

	tieneCantidadReservados, tieneUbicacion, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion
--	--

Tabla 34. Roles abstractos seleccionados por RF del patrón de análisis PAI.

Concepto del dominio	Rol abstracto	Concepto del rango	Rol abstracto inverso
1. Separar diferentes tipos de existencias			
<i>Empleado</i>	<i>separa</i>	<i>Articulo</i>	<i>seSeparaPor</i>
<i>Empleado</i>	<i>verificaExistenciaDe</i>	<i>Articulo</i>	<i>existenciaSeVerificaPor</i>
<i>Empleado</i>	<i>actualiza</i>	<i>Articulo</i>	<i>seActualizaPor</i>
<i>Empleado</i>	<i>registra</i>	<i>Articulo</i>	<i>seRegistraPor</i>
<i>Empleado</i>	<i>actualiza</i>	<i>Inventario</i>	<i>seActualizaPor</i>
2. Realizar un seguimiento de las cantidades de cada artículo en existencias			
<i>Empleado</i>	<i>buscaExistenciasDe</i>	<i>Articulo</i>	<i>existenciasSeBuscaPor</i>
<i>Empleado</i>	<i>verificaExistenciaDe</i>	<i>Articulo</i>	<i>existenciaSeVerificaPor</i>
<i>Empleado</i>	<i>especificaTipoExistenciasDe</i>	<i>Articulo</i>	<i>tipoExistenciasSeEspecificaPor</i>
<i>Empleado</i>	<i>obtieneExistenciasDe</i>	<i>Articulo</i>	<i>existenciasSeObtienePor</i>
3. Realizar un seguimiento de las ubicaciones de los artículos			
<i>Empleado</i>	<i>ubica</i>	<i>Articulo</i>	<i>seUbicaPor</i>
<i>Empleado</i>	<i>verificaExistenciaDe</i>	<i>Articulo</i>	<i>existenciaSeVerificaPor</i>
<i>Empleado</i>	<i>solicitaUbicacionDe</i>	<i>Articulo</i>	<i>ubicacionSeSolicitaPor</i>
<i>Empleado</i>	<i>obtieneUbicacionDe</i>	<i>Articulo</i>	<i>ubicacionSeObtienePor</i>

5.2.6. Especificación de conceptos y relaciones (Formalización)

En esta fase, se realizó una especificación explícita de los conceptos (p. ej. Empleado, Inventario, entre otros) y sus relaciones (p. ej. separa, actualiza, entre otras) utilizando una jerarquía, además, se especificaron las propiedades de cada concepto (p. ej. Id, Nombre, entre otras). Una vez especificados en borrador, se utilizó el editor de ontologías Protégé para almacenar la información correspondiente y de esta manera obtener una OPAE del PAI. A continuación, se presentan los conceptos, relaciones (Tabla 35) (p. ej. el concepto Empleado se relaciona con el concepto Artículo por medio del rol abstracto separa, para especificar que un empleado separa un artículo) y propiedades (Tabla 36) (p. ej. el concepto Empleado tiene como propiedades Id, Nombre, Edad, entre otras) del patrón de análisis PAI.

Conceptos

Persona

a. Empleado

Inventario

Artículo

Institucion

Tabla 35. Relaciones entre conceptos del patrón de análisis PAI.

Dominio	Relación	Rango	Propiedad
Empleado	separa	Articulo	Inversa funcional
Articulo	seSeparaPor	Empleado	Funcional
Empleado	registra	Articulo	Inversa funcional
Articulo	seRegistraPor	Empleado	Funcional
Empleado	actualiza	Inventario	Funcional
Inventario	seActualizaPor	Empleado	Inversa funcional
Empleado	buscaExistenciasDe	Articulo	Inversa funcional
Articulo	existenciasSeBuscaPor	Empleado	Funcional
Empleado	verificaExistenciaDe	Articulo	Inversa funcional
Articulo	existenciaSeVerificaPor	Empleado	Funcional
Empleado	especificaTipoExistenciasDe	Articulo	Inversa funcional
Articulo	tipoExistenciasSeEspecificaPor	Empleado	Funcional
Empleado	obtieneExistenciasDe	Articulo	Inversa funcional
Articulo	existenciasSeObtienePor	Empleado	Funcional
Empleado	ubica	Articulo	Inversa funcional
Articulo	seUbicaPor	Empleado	Funcional
Empleado	solicitaUbicacionDe	Articulo	Funcional
Articulo	ubicacionSeSolicitaPor	Empleado	Inversa funcional
Empleado	obtieneUbicacionDe	Articulo	Inversa funcional
Articulo	ubicacionSeObtienePor	Empleado	Funcional
Empleado	laboraEn	Institucion	Funcional
Institucion	tieneDatosDe	Empleado	N/A

P. ej., en la primera fila de la Tabla 35 el rol abstracto separa permite describir a todo empleado que separa un artículo. De manera inversa, el rol abstracto seSeparaPor permite describir a todo artículo que se separa por un empleado.

Tabla 36. Propiedades de los conceptos del patrón de análisis PAI.

Concepto	Propiedades
Persona	tieneId, tieneNombre, tieneEdad, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion
Empleado	tieneId, tieneNombre, tieneEdad, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion
Institucion	tieneId, tieneNombre, tieneGiro, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion

Inventario	tieneId, tieneNombre, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion
Artículo	tieneId, tieneNombre, tienTipo, tieneCantidad, tieneUbicacion, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion

P. ej., el concepto Empleado cuenta con las propiedades tieneId, tieneNombre, tieneEdad, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad y tienePercepcion, estas propiedades describen las características que tiene cada concepto.

5.2.7. Definición de restricciones

Una vez obtenida la ontología OPAE del PAI se definieron las restricciones existenciales y universales para los conceptos y relaciones del patrón de análisis PAI. Cabe mencionar que estas restricciones fueron especificadas en la OPAE del PAI con el editor de ontologías Protégé utilizando la palabra “*some*” y “*only*” para las restricciones existenciales y universales, respectivamente. A continuación, se presentan las restricciones existenciales y universales definidos para el patrón de análisis PAI.

Restricciones existenciales

- Empleado separa *some* Artículo
- Empleado registra *some* Artículo
- Empleado actualiza *some* Inventario
- Empleado buscaExistenciasDe *some* Artículo
- Empleado verificaExistenciaDe *some* Artículo
- Empleado obtieneUbicacionDe *some* Artículo
- Empleado especificaTipoExistenciasDe *some* Artículo
- Empleado obtieneExistenciasDe *some* Artículo
- Empleado ubica *some* Artículo
- Empleado solicitaUbicacionDe *some* Artículo
- Institucion tieneDatosDe *some* Empleado

P. ej., la restricción existencial (Empleado separa *some* Artículo), establece que un Empleado es aquel quien separa al menos un Artículo.

Restricciones universales

Articulo esSeparado *only* Empleado

Articulo esRegistrado *only* Empleado

Empleado laboraEn *only* Institucion

P. ej., la restricción universal (Articulo esSeparado *only* Empleado), establece que un Artículo solo puede ser separado por un Empleado.

5.2.8. Verificación y validación de la base de conocimiento (Semántica)

En esta fase se realizaron las siguientes actividades: Descripción básica del lenguaje; Definición terminológica; Componente intencional: Descripción del universo; Componente extensional; Definición del formalismo básico; Interpretación semántica. A continuación, se describe en qué consistió cada fase y los resultados obtenidos en cada una de estas actividades.

5.2.8.1. Descripción básica del lenguaje

Esta actividad, la descripción se realizó bajo la expresividad de la DL *ALCIF(D)*, que extienden la DL básica (*ALC*) con roles inversos y propiedades funcionales (*IF*) y la expresión (*D*) : Dominios Concretos (*Concrete Domains*), que integra a las DLs con conjuntos concretos tales como los números reales, enteros o cadenas, así como también predicados concretos definidos en estos conjuntos, tales como comparaciones numéricas (p. ej., \geq), comparaciones de cadena (p. ej., *isPrefixOf*), o comparaciones con constantes (p. ej., ≥ 17).

Esta descripción se almacenó en formato OWL-DL. La representación del conocimiento en el patrón de análisis PAI está basada en los símbolos que se describen a continuación:

- a) Símbolos Dependientes de la Aplicación (**AS**, *Application Dependent Symbols*), constituidos por $\mathbf{AS} = \mathbf{AC} \sqcup \mathbf{AR} \sqcup \mathbf{CR} \sqcup \mathbf{AI} \sqcup \mathbf{CI}$
 1. Conceptos Atómicos (**AC**, *Atomic Concepts*)
 $\mathbf{AC} = \{\text{Articulo, Institucion, Inventario, Persona, Empleado}\}$
 2. Roles Abstractos (**AR**, *Abstract Roles*)
 $\mathbf{AR} = \{\text{actualiza, existenciasSeObtienePor, existenciasSeBuscaPor, seActualizaPor, especificaTipoExistenciasDe, seRegistraPor, seSeparaPor, seUbicaPor, existenciaSeVerificaPor, laboraEn, obtieneExistenciasDe, obtieneUbicacionDe, registra, separa, solicitaUbicacionDe, tieneDatosDe, tipoExistenciasSeEspecificaPor, ubica, ubicacionSeObtienePor, ubicacionSeSolicitaPor, buscaExistenciasDe, verificaExistenciaDe}\}$

3. Roles Concretos (**CR**, *Concrete Roles*)

CR = {tieneCantidad, tieneComportamiento, tieneEdad, tieneEspecificidad, tieneGiro, tieneId, tieneNivelEstabilidad, tieneNombre, tienePercepcion, tieneTipo, tieneUbicacion}

4. Individuos Abstractos (**AI**, *Abstract Individuals*)

AI = {Valentino, Libro1, CENIDET, inventario_CENIDET}

5. Individuos Concretos (**CI**, *Concrete Individuals* también conocidos como tipos de datos)

CI = {string, Boolean, integer, double, dateTime, long}

b) Símbolos Independientes de la Aplicación (**A(In)S**, *Application Independent Symbols*)

$$A(In)S = \{ALCIF(D)\}$$

{

- A (concepto atómico)
- T (concepto universal)
- \perp (concepto vacío)
- \neg (negación o complemento)
- \sqcap (conjunción o intersección)
- \sqcup (disyunción o unión)
- \forall (cuantificador universal o restricción de valor)
- \exists (Cuantificador existencial limitada)
- = (Cardinalidad exacta)
- r^- (Roles inversos)
- $Func(r)$ (Funcionalidad de roles)
- Datatypes (Tipos de datos)

}

5.2.8.2. Definición terminológica

En esta actividad se definió el *componente intencional*, también conocido como *TBox*.

5.2.8.2.1. Componente intencional

La *TBox* del patrón de análisis PAI, se integra de los tipos de axiomas referenciados en la Tabla 37. La Tabla A.1 de la sección 1.1. *Componente intencional* en el Anexo A. *Ontología*

OPAE del patrón de análisis PAI, incluye las equivalencias entre la sintaxis DL y la sintaxis OWL-DL.

Tabla 37. Axiomas en el componente intencional del patrón de análisis PAI.

Tipo de axiomas	Sección en la Tabla A.1
Terminológicos (implicaciones y equivalencias conceptuales) (TA, Terminological Axioms)	--Clases--
Conceptos disjuntos (DC, Disjoint Concepts)	--Clases Disjuntas--
Restricciones existenciales y universales	-- Restricciones existenciales y universales --
Restricciones de roles abstractos (ARC, Abstract Role Constraints)	-- Propiedades de objeto – -- Restricciones y características de propiedades de objeto
Restricciones y características de roles concretos (CRC, Concrete Role Constraints)	-- Propiedades de datos – -- Restricciones y características de propiedades de datos

Por ejemplo los TA, se describen en la sección --Clases—de la Tabla A.1.

El diagrama de Venn de la Figura 24, representa los TA mediante conjuntos. En la Figura 25 se presenta un grafo acíclico que representa el modelado conceptual basado en las implicaciones conceptuales.

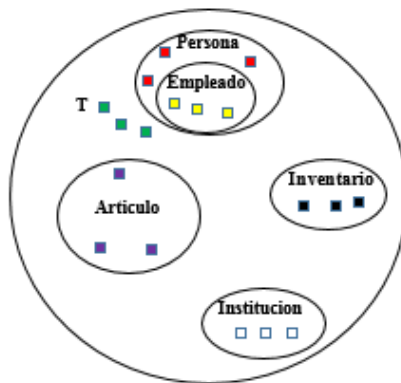


Figura 24 . Representación gráfica de los TA del PAI.

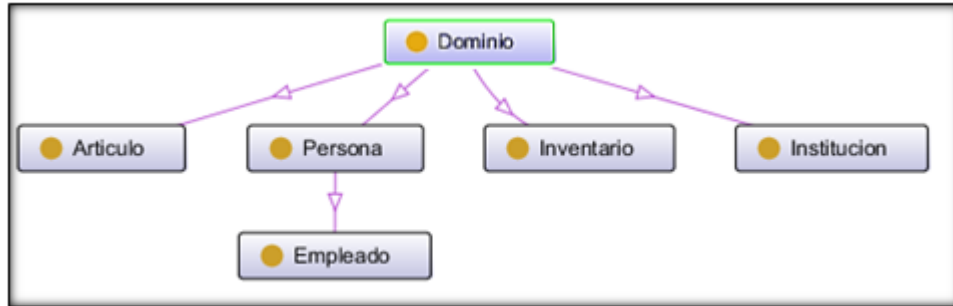


Figura 25 . TBox para el dominio de estudio.

5.2.8.3. Descripción del universo

En esta actividad se definió el *componente extensional*, también conocido como *ABox*, tomando como referencia los símbolos *AS* (*Application Dependent Symbols*) y *A(In)S* (*Application Independent Symbols*).

5.2.8.3.1. Componente extensional

La *ABox* del patrón de análisis PAI, se integra de los axiomas referenciados en la Tabla 38. La Tabla A.2 de la sección 1.2. *Componente extensional* en el Anexo A. *Ontología OPAE del patrón de análisis PAI*, incluye las equivalencias entre la sintaxis DL y la sintaxis OWL-DL.

Tabla 38 . Axiomas en el componente extensional del patrón de análisis PAI.

Tipo de axiomas	Sección en la Tabla A.2
<i>Aserción de conceptos</i> (CA, <i>Concept Assertions</i>)	-- <i>Aserciones de clase</i> --
<i>Aserción de roles abstractos</i> (ARA, <i>Abstract Role Assertions</i>).	-- <i>Aserciones de propiedades de objeto</i> --
<i>Aserciones de roles concretos</i> (CRA, <i>Concrete Role Assertions</i>)	-- <i>Aserciones de propiedades de datos</i> --

Por ejemplo, los axiomas CA, se describen en la sección *Aserciones de clase* de la Tabla A.2. Un ejemplo de una aserción de clase es el siguiente:

$$Articulo(Libro1) \Rightarrow ClassAssertion(domain-pai:Articulo \ domain-pai:Libro1)$$

La definición del lado izquierdo está representada con la sintaxis DL y significa que Libro1 es un individuo de la clase Articulo, del lado derecho se encuentra su equivalencia representado en la sintaxis OWL-DL.

5.2.8.4. Definición del formalismo básico

En esta actividad se obtuvo la base de conocimiento KB para el patrón de análisis PAI integrando el componente intencional y el componente extensional y se generó el modelado de datos mostrado en la Figura 26, que permite validar las reglas definidas en la KB a través de la evidencia lógica basada en la experiencia.

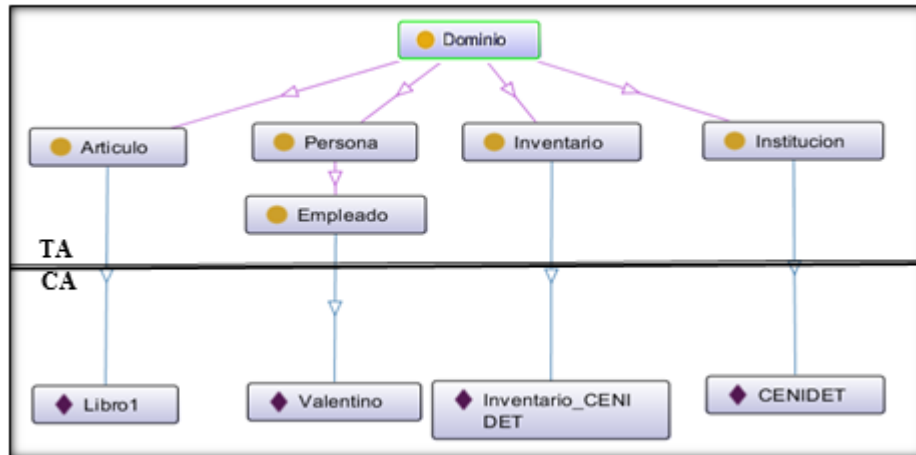


Figura 26 . Modelado de datos del PAI que incluye los axiomas TA y CA

5.2.8.5. Interpretación semántica

En esta actividad se realizó una correspondencia entre los objetos y relaciones del mundo real con los objetos definidos en la KB con base en contextos aplicables para el modelo de estudio mediante una interpretación de reglas. La Figura 28 muestra la estructuración semántica del modelo de estudio de acuerdo a las reglas determinadas en las fases anteriores.

La descripción de conceptos DL, definidos en la KB son interpretados con respecto a una posibilidad infinita de un conjunto de objetos del dominio de interpretación y una asociación de conceptos atómicos, roles atómicos, e individuos para objetos del dominio de interpretación. La representación del conocimiento en el patrón de análisis PAI está basada en los símbolos que se describen a continuación:

Dados los conjuntos:

$AC = \{ \text{Articulo, Institucion, Inventario, Persona, Empleado} \}$

$AR = \{ \text{actualiza, existenciasSeObtienePor, existenciasSeBuscaPor, seActualizaPor, especificaTipoExistenciasDe, seRegistraPor, seSeparaPor, seUbicaPor, existenciaSeVerificaPor, laboraEn, obtieneExistenciasDe, obtieneUbicacionDe, registra, separa, solicitaUbicacionDe, tieneDatosDe, tipoExistenciasSeEspecificaPor, ubica, ubicacionSeObtienePor, ubicacionSeSolicitaPor, buscaExistenciasDe, verificaExistenciaDe} \}$

$CR = \{ \text{tieneCantidad, tieneComportamiento, tieneEdad, tieneEspecificidad, tieneGiro, tieneId, tieneNivelEstabilidad, tieneNombre, tienePercepcion, tieneTipo, tieneUbicacion} \}$

$IA = \{ \text{Valentino, Libro1, CENIDET, inventario_CENIDET} \}$

$IC = \{ \text{string, Boolean, integer, double, dateTime, long} \}$

Sea $I = (\Delta^I, \cdot^I)$ donde

Δ^I	=	{ Valentino, Libro1, CENIDET, inventario_CENIDET }
Articulo ^I	=	{ Libro1 }
Institucion ^I	=	{ CENIDET }
Inventario ^I	=	{ inventario_CENIDET }
Empleado ^I	=	{ Valentino }
actualiza ^I	=	{ (Valentino, inventario_CENIDET) }
seActualizaPor ^I	=	{ (inventario_CENIDET, Valentino) }
obtieneExistenciasDe ^I	=	{ (Valentino, Libro1) }
existenciasSeObtienePor ^I	=	{ (Libro1, Valentino) }
buscaExistenciasDe ^I	=	{ (Valentino, Libro1) }
existenciasSeBuscaPor ^I	=	{ (Libro1, Valentino) }
especificaTipoExistenciasDe ^I	=	{ (Valentino, Libro1) }
tipoExistenciasSeEspecificaPor ^I	=	{ (Libro1, Valentino) }
registra ^I	=	{ (Valentino, Libro1) }
seRegistraPor ^I	=	{ (Libro1, Valentino) }
separa ^I	=	{ (Valentino, Libro1) }
seSeparaPor ^I	=	{ (Libro1, Valentino) }
ubica ^I	=	{ (Valentino, Libro1) }
seUbicaPor ^I	=	{ (Libro1, Valentino) }
verificaExistenciaDe ^I	=	{ (Valentino, Libro1) }
existenciaSeVerificaPor ^I	=	{ (Libro1, Valentino) }
laboraEn ^I	=	{ (Valentino, CENIDET) }
obtieneUbicacionDe ^I	=	{ (Valentino, Libro1) }
ubicacionSeObtienePor ^I	=	{ (Libro1, Valentino) }
solicitaUbicacionDe ^I	=	{ (Valentino, Libro1) }
ubicacionSeSolicitaPor ^I	=	{ (Libro1, Valentino) }
tieneDatosDe ^I	=	{ (CENIDET, Valentino) }
emp ^I	=	Valentino
com ^I	=	Libro1
ins ^I	=	CENIDET
inv ^I	=	inventario_CENIDET

y sea $D = (\Delta^D, \cdot^D)$ donde

Δ^D	= { cantidad, comportamiento, edad, especificidad, giro, id, nivelEstabilidad, nombre, percepcion, tipoMaterial, ubicacion }
tieneCantidad ^D	= {(Libro1, 2)}
tieneComportamiento ^D	= {(Libro1,"estable-adaptable"), (Valentino,"Inestable"), (Inventario_CENIDET,"estable-adaptable"), (CENIDET,"Inestable ")}
tienePercepcion ^D	= {(Libro1,"Semi-tangible"), (Valentino,"Tangible"), (Inventario_CENIDET,"Semi-tangible"), (CENIDET,"Tangible")}
tieneEspecificidad ^D	= {(Libro1,"General"), (Valentino,"Particular"), (Inventario_CENIDET,"General"), (CENIDET,"Particular ")}
tieneNivelEstabilidad ^D	= {(Libro1,"BO"), (Valentino,"IO"), (Inventario_CENIDET,"BO"), (CENIDET,"IO")}
tieneEdad ^D	= {(Valentino, 26)}
tieneGiro ^D	= {(CENIDET,"Control de inventario")}
tieneId ^D	= {(Libro1,"C000001"), (Valentino,"E000001"), (Inventario_CENIDET,"I000001"), (CENIDET,"000001 ")}
tieneNombre ^D	= {(Libro1,"El llano en llamas"), (Valentino,"Valentino Velázquez Olivera"), (Inventario_CENIDET,"Sistema de control de inventario CENIDET"), (CENIDET,"Centro Nacional de Investigación y Desarrollo Tecnológico")}
tieneTipo ^D	= {(Libro1, "Componente")}
tieneUbicacion ^D	= {(Libro1, " En el pasillo #2, en el librero #4 en el stand #3")}
cantidad ^D	= integer
comportamiento ^D	= string
edad ^D	= integer
especificidad ^D	= String
id ^D	= string

giro^D	=	string
$\text{nivelEstabilidad}^D$	=	string
nombre^D	=	string
percepcion^D	=	string
tipo^D	=	string
ubicacion^D	=	string

Entonces I es una (posible) interpretación DL con respecto a la KB. La Figura 27 muestra el dominio y función de interpretación semántica para la KB del patrón de análisis PAI que considera roles concretos. Con esto se comprueba que la KB es consistente porque existe un modelo de interpretación que satisface los axiomas definidos junto a una teoría de datos.

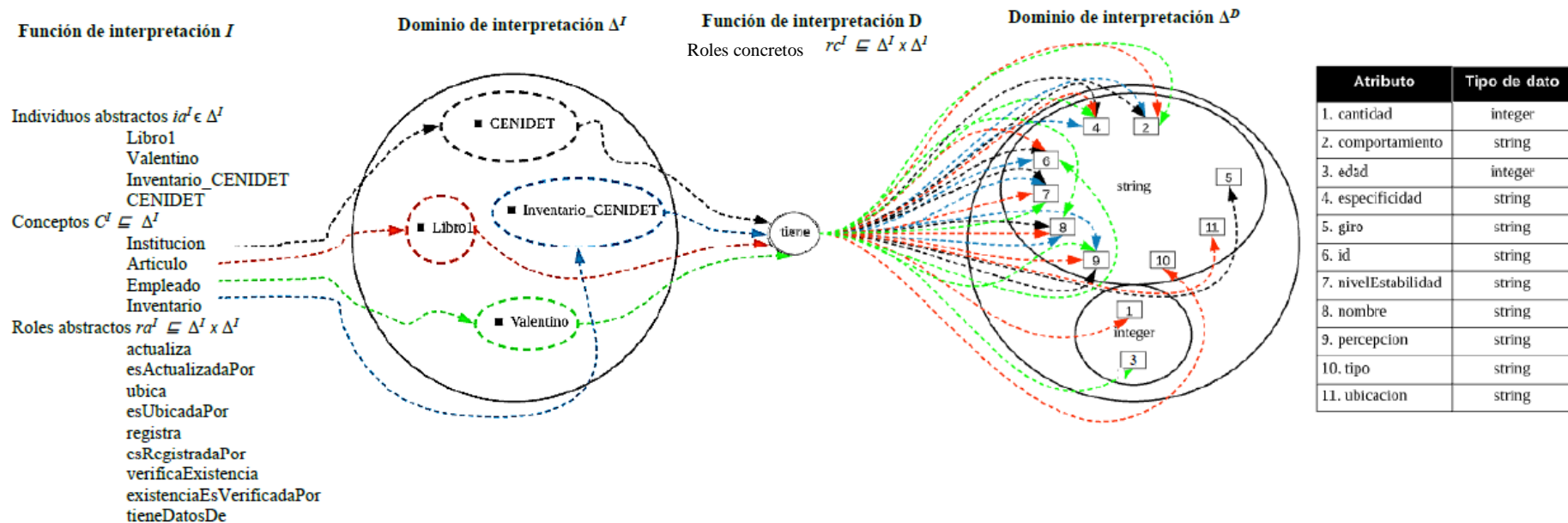


Figura 27. Dominio y función de interpretación semántica del patrón de análisis PAI.

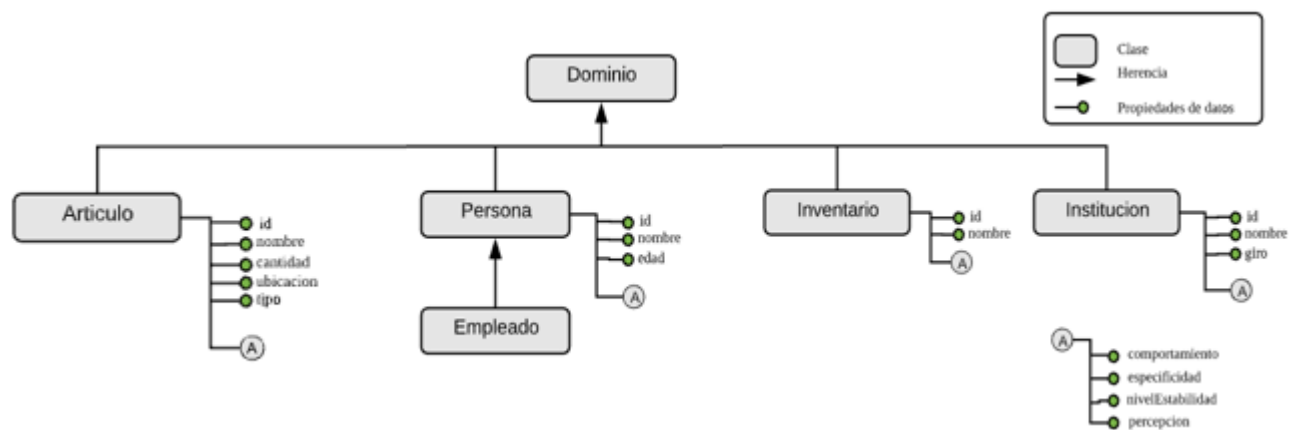


Figura 28. Estructura del modelo de dominio del patrón de análisis PAI.

5.2.9. Formalización complementaria mediante aspectos de temporalidad

Para realizar la formalización complementaria mediante aspectos de temporalidad de la ontología estática (OPAE) del patrón de análisis PAI, se utilizó el método FOCOPAT desarrollado en (Piedrola Domínguez N., 2019). Las actividades que se realizaron en esta fase fueron las siguientes: Procesamiento de especificación; Análisis preliminar; Adaptación de la especificación; Clarificar oraciones; Unir oraciones clarificadas; Generar EC traducible en términos de la TL; Asignación de precedencia; Mapeo; Actualización de la KB con aspectos de temporalidad; Especificación formal de comportamiento; Interpretación semántica temporal y; Verificación de la formalización. Una descripción detallada de cada una de estas actividades se encuentra en (Piedrola Domínguez N., 2019). A continuación, se describen los resultados obtenidos en cada actividad.

5.2.9.1. Procesamiento de especificación

Esta actividad tiene como objetivo obtener una especificación de comportamiento adaptada con orden de precedencia (ECAP) apegada a la TL LTL para formalizar las especificaciones de comportamiento (ECs) descritas en algún requerimiento funcional (RF) del patrón de análisis en estudio. A continuación, se presentan los resultados obtenidos en cada una de las actividades que se realizaron para llevar a cabo el procesamiento de los RFs como EC del patrón de análisis en estudio.

5.2.9.1.1. Análisis preliminar

A continuación, en la Tabla 39 se describen los tres requerimientos funcionales correspondientes al patrón de análisis PAI, a partir de esta descripción, se generaron las ECs presentadas en la Tabla 40, que consiste en un conjunto de oraciones identificadas en cada RF a partir del caso de éxito principal.

Tabla 39 . Requerimientos funcionales (RFs) del patrón de análisis PAI.

<p>RF1 Separar diferentes tipos de existencias. Cuando un empleado decida separar existencias del inventario. El empleado primero debe identificar el tipo de artículo, es decir, identificar si el artículo es un componente o un producto (un producto está compuesto de componentes). Posteriormente, el empleado debe verificar si existe el artículo en el inventario. Si el artículo ya existe, entonces se actualiza la cantidad de existencias del artículo. Si el artículo no existe, entonces se registra el artículo proporcionando los datos del artículo (características), se especifica la ubicación y la cantidad del artículo a registrar. Finalmente, el empleado actualiza el inventario.</p>
<p>RF2 Realizar seguimiento de las cantidades de artículos en existencias. Cuando el empleado decida buscar la cantidad de existencias de un artículo que se encuentra registrado en el inventario. El empleado primero debe proporcionar los datos del artículo correspondiente. Posteriormente, el empleado verifica existencia del artículo en el inventario. Si el artículo existe en el inventario, el empleado especifica el tipo de cantidad de existencias (enMano, enOrden, reservadas) del artículo a dar seguimiento. El inventario retorna la cantidad de existencias de artículos del tipo solicitado por el empleado y finalmente, el empleado obtiene la cantidad de existencias del artículo solicitado.</p>
<p>RF3 Realizar seguimiento de las ubicaciones de los artículos. Cuando el empleado decida ubicar un artículo que se encuentra registrado en el inventario. El empleado primero debe proporcionar los datos del artículo a</p>

ubicar (clave, nombre, etc). Posteriormente, el empleado debe verificar la existencia del artículo en el inventario. Si el artículo existe en el inventario, el empleado solicita la ubicación del artículo. El inventario retorna la ubicación del artículo especificado por el empleado y finalmente, el empleado obtiene la ubicación del artículo especificado.

Tabla 40 . Especificaciones EC de los RFs del patrón de análisis PAI.

RF1 Separar diferentes tipos de existencias
<ol style="list-style-type: none"> 1. El empleado decide separar existencias del inventario. 2. El empleado identifica el tipo de artículo, es decir, identifica si el artículo es un componente o un producto (un producto está compuesto de componentes). 3. El empleado verifica si existe el artículo en el inventario. 4. Si el artículo ya existe, entonces se actualiza la cantidad de existencias del artículo. 5. Si el artículo no existe, entonces se registra el artículo proporcionando los datos del artículo (características), se especifica la ubicación y la cantidad del artículo a registrar. 6. El empleado actualiza el inventario.
RF2 Realizar seguimiento de las cantidades de artículos en existencias
<ol style="list-style-type: none"> 1. El empleado decide buscar la cantidad de existencias de un artículo que se encuentra registrado en el inventario. 2. El empleado proporciona datos del artículo correspondiente. 3. El empleado verifica existencia del artículo en el inventario. 4. Si el artículo existe en el inventario, el empleado especifica el tipo de cantidad de existencias (enMano, enOrden, reservadas) del artículo a dar seguimiento. 5. El inventario retorna la cantidad de existencias de artículos del tipo solicitado por el empleado. 6. El empleado obtiene la cantidad de existencias del artículo solicitado.
RF3 Realizar seguimiento de las ubicaciones de los artículos
<ol style="list-style-type: none"> 1. El empleado decide ubicar un artículo que se encuentra registrado en el inventario. 2. El empleado proporciona datos del artículo a ubicar (clave, nombre, entre otras). 3. El empleado verifica existencia del artículo en el inventario. 4. Si el artículo existe en el inventario, el empleado solicita la ubicación del artículo. 5. El inventario retorna la ubicación del artículo especificado por el empleado. 6. El empleado obtiene la ubicación del artículo especificado.

P. ej., del primer RF (Separar diferentes tipos de existencias) de la Tabla 39, se encontraron las seis oraciones presentadas en el primer renglón de la Tabla 40, que especifican el comportamiento del RF.

5.2.9.1.2. Adaptación de la especificación

A continuación, se presentan los cambios realizados a las especificaciones EC obtenidas en la actividad anterior. Por cada especificación EC, se generaron OCs (oraciones clarificadas) que fueron unidas (Tabla 42) para darles sentido y tener una EC clarificada (Tabla 43).

5.2.9.1.2.1 Clarificar oraciones

Como parte del proceso de clarificación de oraciones de la EC, se consideraron las restricciones especificadas en la metodología MECOTPA. A continuación, en la Tabla 41 se presenta el resultado obtenido durante clarificación de las oraciones de la EC de los RFs del patrón de análisis PAI.

Tabla 41 . Clarificación de oraciones de la EC de los RFs del patrón de análisis PAI.

No.	Oración inicial	Oración clarificada
RF1 Separar diferentes tipos de existencias		
1	<i>El empleado decide separar existencias del inventario.</i>	<i>El empleado separa un artículo.</i>
2	<i>El empleado identifica el tipo de artículo, es decir, identifica si el artículo es un Componente o un producto (un producto está compuesto de componentes).</i>	<i>El empleado identifica el tipo de artículo.</i>
3	<i>El empleado verifica si existe el artículo en el inventario.</i>	<i>El empleado verifica la existencia del artículo.</i>
4	<i>Si el artículo ya existe entonces se actualiza la cantidad de existencias del artículo.</i>	<i>El empleado actualiza el artículo.</i>
5	<i>Si el artículo no existe entonces se registra el artículo proporcionando los datos del artículo (características), se especifica la ubicación y la cantidad del artículo a registrar.</i>	<i>El empleado registra el artículo</i>
6	<i>El empleado actualiza el inventario.</i>	<i>El empleado actualiza el inventario.</i>
RF2 Realizar seguimiento de las cantidades de artículos en existencias		
1	<i>El empleado decide buscar la cantidad de existencias de un artículo que se encuentra registrado en el inventario.</i>	<i>El empleado busca la cantidad de existencias de un artículo.</i>
2	<i>El empleado proporciona datos del artículo correspondiente.</i>	<i>El empleado proporciona datos del artículo.</i>
3	<i>El empleado verifica existencia del artículo en el inventario.</i>	<i>El empleado verifica la existencia del artículo</i>
4	<i>Si el artículo existe en el inventario, el empleado especifica el tipo de cantidad de existencias (enMano, enOrden, reservadas) del artículo a dar seguimiento.</i>	<i>El empleado especifica el tipo de cantidad de existencias del artículo.</i>
5	<i>El inventario retorna la cantidad de existencias de artículos del tipo solicitado por el empleado.</i>	<i>El inventario retorna la cantidad de existencias del artículo.</i>
6	<i>El empleado obtiene la cantidad de existencias del</i>	<i>El empleado obtiene existencias del artículo</i>

	<i>artículo solicitado.</i>	<i>solicitado.</i>
RF3 Realizar seguimiento de las ubicaciones de los artículos		
1	<i>El empleado decide ubicar un artículo que se encuentra registrado en el inventario.</i>	<i>El empleado ubica el artículo.</i>
2	<i>El empleado proporciona datos del artículo correspondiente.</i>	<i>El empleado proporciona datos del artículo.</i>
3	<i>El empleado verifica existencia del artículo en el inventario.</i>	<i>El empleado verifica la existencia del artículo.</i>
4	<i>Si el artículo existe en el inventario, el empleado solicita la ubicación del artículo.</i>	<i>El empleado solicita la ubicación del artículo.</i>
5	<i>El inventario retorna la ubicación del artículo especificado por el empleado.</i>	<i>El inventario retorna la ubicación del artículo.</i>
6	<i>El empleado obtiene la ubicación del artículo especificado.</i>	<i>El empleado obtiene la ubicación del artículo.</i>

5.2.9.1.2.2. Unir oraciones clarificadas

En esta tarea, se unieron las OCs obtenidas en la tarea 5.2.9.1.2.1 *Clarificar oraciones* agregando los signos de puntuación y expresiones necesarias que den sentido lógico a la especificación y que la hagan lo menos ambigua posible y un poco más precisa. A continuación en la Tabla 42 se muestra el resultado de la unión de las oraciones clarificadas de los tres RFs del patrón de análisis PAI y en la Tabla 43 se muestran las especificaciones de comportamiento clarificadas (ECC).

Tabla 42 . Especificaciones EC de los RFs del patrón de análisis PAI a partir de la unión de oraciones clarificadas.

RF1 Separar diferentes tipos de existencias. <i>El empleado separa un artículo. El empleado verifica la existencia del artículo. El empleado actualiza el artículo. El empleado registra el artículo. El empleado actualiza el inventario.</i>
RF2 Realizar seguimiento de las cantidades de artículos en existencias. <i>El empleado busca la cantidad de existencias de un artículo. El empleado verifica existencia del artículo. El empleado especifica el tipo de cantidad de existencias del artículo. El empleado obtiene la cantidad de existencias del artículo solicitado.</i>
RF3 Realizar seguimiento de las ubicaciones de los artículos. <i>El empleado ubica un artículo. El empleado verifica la existencia del artículo. El empleado solicita la ubicación del artículo. El empleado obtiene la ubicación del artículo.</i>

Tabla 43 . Especificaciones ECC de los RFs del patrón de análisis PAI.

RF1 Separar diferentes tipos de existencias. <i>Si el empleado separa un artículo, entonces el empleado verifica la</i>
--

<i>existencia del artículo. El empleado actualiza el artículo. El empleado registra el artículo y el empleado actualiza el inventario.</i>
RF2 Realizar seguimiento de las cantidades de artículos en existencias. <i>Si el empleado busca la cantidad de existencias de un artículo, entonces el empleado verifica existencia del artículo. El empleado especifica el tipo de cantidad de existencias del artículo. El empleado obtiene la cantidad de existencias del artículo solicitado.</i>
RF3 Realizar seguimiento de las ubicaciones de los artículos. <i>Si el empleado ubica un artículo, entonces el empleado verifica la existencia del artículo. El empleado solicita la ubicación del artículo. El empleado obtiene la ubicación del artículo.</i>

5.2.9.1.2.3. Generar EC traducible en términos de la TL

En esta tarea se integraron en las ECC las expresiones necesarias para facilitar la conversión a una fórmula de la TL LTL para formalizar el comportamiento, dando lugar a una EC adaptada (ECA). A continuación, en la Tabla 44 se muestra el resultado de llevar a cabo el proceso de adaptación de las especificaciones ECC.

Tabla 44 . Especificaciones ECA de los RFs del patrón de análisis PAI.

RF1 Separar diferentes tipos de existencias. <i>Siempre que un empleado separe un artículo implica que posteriormente el empleado verifique la existencia del artículo. Frecuentemente, el empleado (no) actualizará el artículo o el empleado (no) registrará el artículo. Finalmente, el empleado (no) actualizará el inventario.</i>
RF2 Realizar seguimiento de las cantidades de artículos en existencias. <i>Siempre que un empleado busque la cantidad de existencias de un artículo, implica que posteriormente el empleado verifique la existencia del artículo. Frecuentemente, el empleado (no) especificará el tipo de cantidad de existencias del artículo y después el empleado (no) obtendrá la cantidad de existencias del artículo solicitado.</i>
RF3 Realizar seguimiento de las ubicaciones de los artículos. <i>Siempre que un empleado ubique un artículo, implica que posteriormente el empleado verifique la existencia del artículo. Frecuentemente el empleado (no) solicitará la ubicación del artículo y después el empleado (no) obtendrá la ubicación del artículo.</i>

La conversión de una ECC a una ECA consistió en agregar expresiones cercanas a los operadores de la TL LTL, en donde la expresión temporal “Siempre” indica que la especificación ECA seguramente ocurrirá en el futuro. También indica de manera explícita qué ocurre en un primer momento y qué ocurre en un segundo momento, la ocurrencia eventual de hechos posteriores mediante las expresiones temporales “frecuentemente” y “posteriormente”, así como expresiones lógicas (“y”, “no” e “implica que”) que le da sentido lógico a la descripción.

5.2.9.1.3. Asignación de precedencia

En esta actividad, se identificaron las expresiones y enunciados de las especificaciones ECA obtenidas en la actividad anterior para crear las especificaciones ECA con orden de

precedencia (ECAP). El proceso para obtener una ECAP consiste en establecer un orden de precedencia entre expresiones y la ocurrencia de lo que especifican los enunciados. Las especificaciones ECAP de los RFs del patrón de análisis PAI se describen a continuación en la Tabla 45.

Tabla 45 . Especificaciones ECAP de los RFs del patrón de análisis PAI.

<p>RF1 Separar diferentes tipos de existencias.</p> <p><i>Siempre que</i> <i>un empleado separe un artículo</i> <i>implica que</i> <i>posteriormente</i> <i>el empleado verifique la existencia del artículo.</i> <i>y frecuentemente</i> <i>el empleado (no) actualizará el artículo</i> <i>o</i> <i>el empleado (no) registrará el artículo.</i> <i>y después</i> <i>el empleado (no) actualizará el inventario.</i> <i>)</i></p>
<p>RF2 Realizar seguimiento de las cantidades de artículos en existencias.</p> <p><i>Siempre que</i> <i>un empleado busque la cantidad de existencias de un artículo</i> <i>implica que</i> <i>posteriormente</i> <i>el empleado verifique la existencia del artículo.</i> <i>y frecuentemente</i> <i>el empleado (no) especificará el tipo de cantidad de existencias del artículo</i> <i>y después</i> <i>el empleado (no) obtendrá la cantidad de existencias del artículo solicitado.</i> <i>)</i></p>
<p>RF3 Realizar seguimiento de las ubicaciones de los artículos.</p> <p><i>Siempre que</i> <i>un empleado ubique un artículo</i> <i>implica que</i> <i>posteriormente</i> <i>el empleado verifique la existencia del artículo.</i> <i>y frecuentemente</i> <i>el empleado (no) solicitará la ubicación del artículo</i> <i>y después</i> <i>el empleado (no) obtendrá la ubicación del artículo.</i> <i>)</i></p>

5.2.9.2. Formalización del comportamiento

5.2.9.2.1. Mapeo

En esta actividad se identificaron componentes tales como: proposiciones atómicas (Pas), conceptos (Cs) y roles abstractos (ARs) a partir de las especificaciones ECAP obtenidas en la sección 5.2.9.1.3. *Asignación de precedencia*. A continuación, en la Tabla 46 se presentan los componentes identificados en los RFs del patrón de análisis PAI.

Tabla 46 . Pas, Cs y ARs en la especificación ECAP de los RFs del patrón de análisis PAI.

Pas	Cs	ARs		
		Dominio		Rango
RF1 Separar diferentes tipos de existencias				
<p><i>p= el empleado separa un artículo.</i> <i>q= el empleado verifica la existencia del artículo.</i> <i>r= el empleado actualiza el artículo.</i> <i>s= el empleado registra el artículo.</i> <i>t= el empleado actualiza el inventario.</i></p>	<p>Empleado Articulo Inventario</p>	Empleado	<i>separa</i>	Articulo
		Empleado	<i>verificaExistenciaDe</i>	Articulo
		Empleado	<i>actualiza</i>	Articulo
		Empleado	<i>registra</i>	Articulo
		Empleado	<i>actualiza</i>	Inventario
RF2 Realizar seguimiento de las cantidades de artículos en existencias				
<p><i>p= el empleado busca la cantidad de existencias de un artículo.</i> <i>q= el empleado verifica la existencia del artículo.</i> <i>r= el empleado especifica el tipo de cantidad de existencias del artículo.</i> <i>s= el empleado obtiene cantidad de existencias del artículo solicitado.</i></p>	<p>Empleado Articulo Inventario</p>	Empleado	<i>buscaExistenciasDe</i>	Articulo
		Empleado	<i>verificaExistenciaDe</i>	Articulo
		Empleado	<i>especificaTipoExistenciasDe</i>	Articulo
		Empleado	<i>obtieneExistenciasDe</i>	Articulo
RF3 Realizar seguimiento de las ubicaciones de los artículos				
<p><i>p= el empleado ubica un artículo.</i> <i>q= el empleado verifica la existencia del artículo.</i> <i>r= el empleado solicita la ubicación del artículo.</i> <i>s= el empleado obtiene la ubicación del artículo.</i></p>	<p>Empleado Articulo Inventario</p>	Empleado	<i>ubica</i>	Articulo
		Empleado	<i>verificaExistenciaDe</i>	Articulo
		Empleado	<i>solicitaUbicacionDe</i>	Articulo
		Empleado	<i>obtieneUbicacionDe</i>	Articulo

P. ej., para el primer RF (Separar diferentes tipos de existencias) del PAI descrito en la Tabla 46, se obtuvieron cinco *Pas* (primera columna), en donde cada *Pa* representa una

especificación de comportamiento. De estos *Pas*, se obtuvieron los conceptos (segunda columna) y roles concretos (cuarta columna) que participan en cada especificación. Para cada rol concreto, se especificó el dominio (tercera columna) y rango (quinta columna), p. ej., los conceptos *Empleado* y *Artículo*, forman parte del dominio y el rango del rol abstracto *separa*, que especifica que un Empleado separa un Artículo.

5.2.9.2.2. Actualización de la KB con aspectos de temporalidad

En esta actividad se llevó a cabo lo siguiente:

- ✓ La construcción de la ontología OTPAE consistente del patrón de análisis PAI (descrita en la sección 1.1. *Componente intencional del Anexo B. Ontología OTPAE del patrón de análisis PAI*,
- ✓ La conversión de los roles abstractos identificadas en cada especificación ECAP en roles abstractos temporales (descritas en la sección -- *Propiedades de objeto temporales* -- de la Tabla B.4 del *Anexo B. Ontología OTPAE del patrón de análisis PAI*,
- ✓ La representación formal de los roles abstractos (inversos) y temporales (inversos) (obtenidas a partir de los componentes identificados en la sección 5.2.9.2.1. *Mapeo*) mediante CIs básicos y temporales de la DL ALC,
- ✓ Las reglas SWRL de características funcional e inversa funcional en los roles abstractos (descritas en la Tabla B.3 del *Anexo B. Ontología OTPAE del patrón de análisis PAI*
- ✓ y se obtuvieron los Conceptos de Inclusión (CIs) básicos y temporales (descritos en la sección -- *Restricciones universales y existenciales* -- de la Tabla A.1 y la Tabla B.4 del *Anexo A. Ontología OP AE del patrón de análisis PAI* y del *Anexo B. Ontología OTPAE del patrón de análisis PAI*, respectivamente).

A continuación, en la Tabla 47 se presentan los conceptos de inclusión (CIs) de roles abstractos y su versión temporal en los RFs del patrón de análisis PAI.

Tabla 47 . CIs de roles abstractos y su versión temporal en los RFs del patrón de análisis PAI.

Id	Pas	CI básico	CI temporal
RF1 Separar diferentes tipos de existencias			
<i>p</i>	<i>El empleado separa un artículo.</i>	$\text{Empleado} \sqsubseteq \exists \text{separa.Articulo}$	$\text{Empleado} \sqsubseteq \exists \text{separa.}(\text{Event} \sqcap (\exists \text{separa.Articulo}))$
<i>q</i>	<i>El empleado verifica la existencia del artículo.</i>	$\text{Empleado} \sqsubseteq \exists \text{verificaExistenciaDe.Articulo}$	$\text{Empleado} \sqsubseteq \exists \text{verificaExistenciaDe.}(\text{Event} \sqcap (\exists \text{verificaExistenciaDe.Articulo}))$
<i>r</i>	<i>El empleado actualiza el artículo.</i>	$\text{Empleado} \sqsubseteq \exists \text{actualiza.Articulo}$	$\text{Empleado} \sqsubseteq \exists \text{actualiza.}(\text{Event} \sqcap (\exists \text{actualiza.Articulo}))$
<i>s</i>	<i>El empleado registra el</i>	$\text{Empleado} \sqsubseteq \exists \text{registra.Articulo}$	$\text{Empleado} \sqsubseteq \exists \text{registra.}(\text{Event} \sqcap$

	<i>artículo.</i>		$(\exists \text{registra}.\text{Articulo}))$
<i>t</i>	<i>El empleado actualiza el inventario.</i>	$\text{Empleado} \sqsubseteq \forall \text{actualiza}.\text{Inventario}$	$\text{Empleado} \sqsubseteq \forall \text{actualiza}.\text{(Event} \sqcap (\forall \text{actualiza}.\text{Inventario}))$
RF2 Realizar seguimiento de las cantidades de artículos en existencias			
<i>p</i>	<i>El empleado busca la cantidad de existencias de un artículo.</i>	$\text{Empleado} \sqsubseteq \exists \text{buscaExistenciasDe}.\text{Articulo}$	$\text{Empleado} \sqsubseteq \exists \text{buscaExistenciasDe}.\text{(Event} \sqcap (\exists \text{buscaExistenciasDe}.\text{Articulo}))$
<i>q</i>	<i>El empleado verifica la existencia del artículo.</i>	$\text{Empleado} \sqsubseteq \exists \text{verificaExistenciaDe}.\text{Articulo}$	$\text{Empleado} \sqsubseteq \exists \text{verificaExistenciaDe}.\text{(Event} \sqcap (\exists \text{verificaExistenciaDe}.\text{Articulo}))$
<i>r</i>	<i>El empleado especifica el tipo de cantidad de existencias del artículo.</i>	$\text{Empleado} \sqsubseteq \exists \text{especificaTipoExistenciasDe}.\text{Articulo}$	$\text{Empleado} \sqsubseteq \exists \text{especificaTipoExistenciasDe}.\text{(Event} \sqcap (\exists \text{especificaTipoExistenciasDe}.\text{Articulo}))$
<i>s</i>	<i>El empleado obtiene existencias del artículo solicitado.</i>	$\text{Empleado} \sqsubseteq \exists \text{obtieneExistenciasDe}.\text{Articulo}$	$\text{Empleado} \sqsubseteq \exists \text{obtieneExistenciasDe}.\text{(Event} \sqcap (\exists \text{obtieneExistenciasDe}.\text{Articulo}))$
RF3 Realizar seguimiento de las ubicaciones de los artículos			
<i>p</i>	<i>El empleado ubica un artículo.</i>	$\text{Empleado} \sqsubseteq \exists \text{ubica}.\text{Articulo}$	$\text{Empleado} \sqsubseteq \exists \text{ubica}.\text{(Event} \sqcap (\exists \text{ubica}.\text{Articulo}))$
<i>q</i>	<i>El empleado verifica la existencia del artículo.</i>	$\text{Empleado} \sqsubseteq \exists \text{verificaExistenciaDe}.\text{Articulo}$	$\text{Empleado} \sqsubseteq \exists \text{verificaExistenciaDe}.\text{(Event} \sqcap (\exists \text{verificaExistenciaDe}.\text{Articulo}))$
<i>r</i>	<i>El empleado solicita la ubicación del artículo.</i>	$\text{Empleado} \sqsubseteq \exists \text{solicitaUbicacionDe}.\text{Articulo}$	$\text{Empleado} \sqsubseteq \exists \text{solicitaUbicacionDe}.\text{(Event} \sqcap (\exists \text{solicitaUbicacionDe}.\text{Articulo}))$
<i>s</i>	<i>El empleado obtiene la ubicación del artículo solicitado.</i>	$\text{Empleado} \sqsubseteq \exists \text{obtieneUbicacionDe}.\text{Articulo}$	$\text{Empleado} \sqsubseteq \exists \text{obtieneUbicacionDe}.\text{(Event} \sqcap (\exists \text{obtieneUbicacionDe}.\text{Articulo}))$

P. ej., el CI básico para el rol abstracto separa de la *Pa p*, del RF Separar diferentes tipos de existencias, describe que un empleado es todo aquel que separa un artículo, mientras que su versión temporal (CI temporal) expresa que un empleado es todo aquel que separa un artículo a través de un evento.

5.2.9.2.3. Especificación formal de comportamiento

En esta actividad se obtuvieron las fórmulas ALC_{LTL} que representan el comportamiento dinámico en los RFs del patrón de análisis PAI. Además, con la finalidad de verificar la satisfacibilidad de las fórmulas ALC_{LTL} fue necesario reducir las especificaciones formales (tal y como se especifica en (Piedrola Domínguez N., 2019)) a fórmulas de la TL LTL .

Para generar las fórmulas $ALCLTL(\varphi)$ de la Tabla 48, se utilizaron los CIs básicos obtenidos en la sección 5.2.9.2.2. *Actualización de la KB con aspectos de temporalidad*. Estas fórmulas representan las especificaciones ECAP obtenidas en la sección 5.2.9.1. *Procesamiento de especificación*.

Tabla 48 . Especificaciones formales a partir de las especificaciones ECAP de los RFs del patrón de análisis PAI.

Fórmula $ALCLTL(\varphi)$
RF1 Separar diferentes tipos de existencias
$\mathcal{G}(\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{separa.Articulo}) \rightarrow (\mathcal{XX}(\text{Empleado} \sqsubseteq \exists \text{verificaExistenciaDe.Articulo}) \wedge \mathcal{F}(\neg \mathcal{XX}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{actualiza.Articulo} \vee (\text{Empleado} \sqsubseteq \exists \text{registra.Articulo})) \wedge \neg \mathcal{XXX}(\text{Empleado} \sqsubseteq \exists \text{actualiza.Inventario})))$
RF2 Realizar seguimiento de las cantidades de artículos en existencias
$\mathcal{G}(\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{buscaExistenciasDe.Articulo}) \rightarrow (\mathcal{XX}(\text{Empleado} \sqsubseteq \exists \text{verificaExistenciaDe.Articulo}) \wedge \mathcal{F}(\neg \mathcal{XX}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{especificaTipoExistencias.Articulo}) \wedge \neg \mathcal{XXX}(\text{Empleado} \sqsubseteq \exists \text{obtieneExistenciasDe.Articulo})))$
RF3 Realizar seguimiento de las ubicaciones de los artículos
$\mathcal{G}(\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{ubica.Articulo}) \rightarrow (\mathcal{XX}(\text{Empleado} \sqsubseteq \exists \text{verificaExistenciaDe.Articulo}) \wedge \mathcal{F}(\neg \mathcal{XX}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{solicitaUbicacionDe.Articulo}) \wedge \neg \mathcal{XXX}(\text{Empleado} \sqsubseteq \exists \text{obtieneUbicacionDe.Articulo})))$

P. ej., la especificación del RF Separar diferentes tipos de existencias especifica que, “Siempre (\mathcal{G}) ocurre que, en un primer momento (\mathcal{X}), el empleado decide separar un artículo ($\text{Empleado} \sqsubseteq \exists \text{separa.Articulo}$), lo cual implica (\rightarrow) que posteriormente (\mathcal{XX}) el empleado verifique la existencia del artículo ($\text{Empleado} \sqsubseteq \exists \text{verificaExistenciaDe.Articulo}$). Eventualmente (\mathcal{F}) en otro momento ($\mathcal{XX}\mathcal{X}$) el empleado (no \neg) actualiza el artículo ($\text{Empleado} \sqsubseteq \exists \text{actualiza.Articulo}$) o el empleado (no \neg) registra el artículo ($\text{Empleado} \sqsubseteq \exists \text{registra.Articulo}$) y posteriormente (\mathcal{XXX}) el empleado (no \neg) actualiza el ($\text{Empleado} \sqsubseteq \exists \text{actualiza.Inventario}$ ”.

Las fórmulas ALC_{LTL} de las especificaciones ECAP de la Tabla 48 se redujeron a las fórmulas LTL (φ') mostradas en la Tabla 49. La reducción consistió en reemplazar los CIs básicos de las fórmulas ALC_{LTL} por proposiciones atómicas de la lógica temporal LTL.

Tabla 49 . Reducción de especificaciones formales a fórmulas de la LTL.

Fórmula LTL (φ')	Proposiciones
RF1 Separar diferentes tipos de existencias	
$\mathcal{G}(\mathcal{X}(p) \rightarrow (\mathcal{X}\mathcal{X}(q) \wedge \mathcal{F}(\neg\mathcal{X}\mathcal{X}\mathcal{X}(r \vee s) \wedge \neg\mathcal{X}\mathcal{X}\mathcal{X}(t))))$	<p>$p = Empleado \sqsubseteq \exists separa.Articulo$ $q = Empleado \sqsubseteq \exists verificaExistenciaDe.Articulo$ $r = Empleado \sqsubseteq \exists actualiza.Articulo$ $s = Empleado \sqsubseteq \exists registra.Articulo$ $t = Empleado \sqsubseteq \exists actualiza.Inventario$</p>
RF2 Realizar seguimiento de las cantidades de artículos en existencias	
$\mathcal{G}(\mathcal{X}(p) \rightarrow (\mathcal{X}\mathcal{X}(q) \wedge \mathcal{F}(\neg\mathcal{X}\mathcal{X}\mathcal{X}(r) \wedge \neg\mathcal{X}\mathcal{X}\mathcal{X}(s))))$	<p>$p = Empleado \sqsubseteq \exists buscaExistenciasDe.Articulo$ $q = Empleado \sqsubseteq \exists verificaExistenciaDe.Articulo$ $r = Empleado \sqsubseteq \exists especificaTipoExistencias.Articulo$ $s = Empleado \sqsubseteq \exists obtieneExistenciasDe.Articulo$</p>
RF3 Realizar seguimiento de las ubicaciones de los artículos	
$\mathcal{G}(\mathcal{X}(p) \rightarrow (\mathcal{X}\mathcal{X}(q) \wedge \mathcal{F}(\neg\mathcal{X}\mathcal{X}\mathcal{X}(r) \wedge \neg\mathcal{X}\mathcal{X}\mathcal{X}(s))))$	<p>$p = Empleado \sqsubseteq \exists ubica.Articulo$ $q = Empleado \sqsubseteq \exists verificaExistenciaDe.Articulo$ $r = Empleado \sqsubseteq \exists solicitaUbicacionDe.Articulo$ $s = Empleado \sqsubseteq \exists obtieneUbicacionDe.Articulo$</p>

P. ej., para reducir la fórmula ALC_{LTL} del RF Separar diferentes tipos de existencias, los CI básicos $Empleado \sqsubseteq \exists separa.Articulo$, $Empleado \sqsubseteq \exists verificaExistenciaDe.Articulo$, $Empleado \sqsubseteq \exists actualiza.Articulo$, $Empleado \sqsubseteq \exists registra.Articulo$, y $Empleado \sqsubseteq \exists actualiza.Inventario$, se reemplazaron por las proposiciones atómicas p , q , r , s y t , respectivamente.

5.2.9.2.4. Interpretación semántica temporal

En esta actividad, se propusieron e integraron secuencias de interpretaciones a la ontología OTPAE del patrón de análisis PAI, mediante el plugin Chronos en el editor de ontologías Protegé. Las secuencias de interpretaciones para este patrón de análisis se presenta en la sección 1.2 *Componente extensional* del Anexo B. *Ontología OTPAE del patrón de análisis PAI*. A continuación, en la Tabla 50 se presentan los resultados obtenidos durante la verificación de la consistencia semántica temporal (TSCC) de la OTPAE del patrón de análisis PAI.

Tabla 50. Resultados de la TSCC de la OTPAE del PAI.

Requerimiento	Caso de Verificación	Resultado de la Verificación
1. Separar diferentes tipos de existencias.	CV-01	Consistente
	CV-02	Consistente con conceptos insatisfacibles
2. Realizar el seguimiento de cantidades de artículos en existencias.	CV-03	Consistente
	CV-04	Consistente con conceptos insatisfacibles
3. Realizar el seguimiento de las ubicaciones de los artículos.	CV-05	Consistente
	CV-06	Consistente con conceptos insatisfacibles

Los requerimientos funcionales para el caso de éxito resultaron ser consistentes, es decir, no existen inconsistencias en la KB. Sin embargo, para los casos de éxito y fracaso la KB resultó ser consistente, pero con conceptos insatisfacibles debido a las restricciones establecidas por las reglas SWRL de aserciones negativas. Las reglas SWRL permiten agregar restricciones temporales en la ontología y son generadas por el editor de ontologías Protegé al momento de convertir propiedades de objetos estáticas a temporales y al momento de crear aserciones de propiedades de objetos negativas, estas reglas se pueden consultar en la sección *Componente extensional* del Anexo B. *Ontología OTPAE del patrón de análisis PAI*.

A partir de las secuencias de interpretaciones presentadas en la sección 1.2 *Componente extensional* del Anexo B. *Ontología OTPAE del patrón de análisis PAI* se realizó una reducción a las trayectoria de estados π' y π'' . En la Figura 29 y Figura 30 se muestran de manera gráfica las trayectorias obtenidas a partir de la reducción de la secuencia de interpretaciones, con base en el comportamiento dinámico especificado en el RF “Realizar seguimiento de las cantidades de artículos en existencias”.

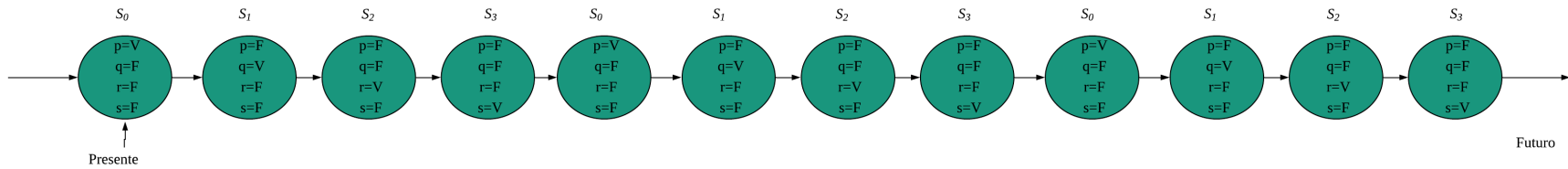


Figura 29. Representación gráfica de la trayectoria de estados π' de casos de éxito del RF Realizar el seguimiento de cantidades de artículos en existencias.

P. ej., para el *Caso de éxito 1*, las interpretaciones 1, 2, 3 y 4 representados por proposiciones atómicas ($p, q, r, y s$), se llevan a cabo en los estados $s_0, s_1, s_2, y s_3$ en donde el valor de verdad (T=true, F=false) de las proposiciones atómicas depende del estado en el que suceden: p (Empleado busca existencias de Artículo) ocurre en el estado s_0 , q (Empleado verifica existencia de Artículo) ocurre en el estado s_1 , r (Empleado especifica tipo existencias de Artículo) ocurre en el estado s_2 y s (Empleado obtiene existencias de Artículo) ocurre en el estado s_3 .

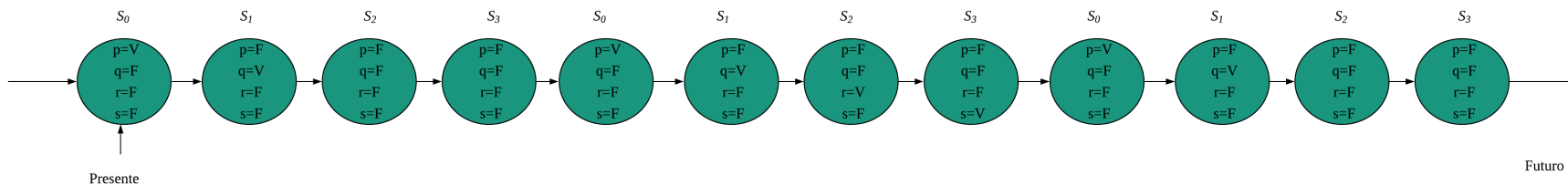


Figura 30 . Representación gráfica de la trayectoria de estados π'' de casos de éxito y fracaso del RF Realizar el seguimiento de cantidades de artículos en existencias.

P. ej., para el *Caso de fracaso 1*, las interpretaciones 1, 2, 3 y 4 representados por proposiciones atómicas ($p, q, r, y s$), se llevan a cabo en los estados $s_0, s_1, s_2, y s_3$ en donde el valor de verdad (T=true, F=false) de las proposiciones atómicas depende del estado en el que suceden: p (Empleado busca existencias de Artículo) ocurre en el estado s_0 y q (Empleado verifica existencia de Artículo) ocurre en el estado s_1 . Sin embargo, r (Empleado especifica tipo existencias de Artículo) no ocurre en el estado s_2 ni s (Empleado obtiene existencias de Artículo) no ocurre en el estado s_3 , por lo que su valor de verdad se establece como F.

Para documentar la verificación TSCC, se utilizó la documentación descrita en *Anexo E. Documentación TSCC* (propuesta en (Piedrola Domínguez N., 2019)). Todos los documentos realizados para la verificación TSCC se encuentran en el directorio *Entregable 9/TSCC/*, almacenado en el disco que acompaña a esta tesis (para mayor referencia consultar el *Anexo G. Contenido del disco*).

5.2.9.2.5. Verificación de la formalización

En esta actividad se verificó la satisfacibilidad de las fórmulas LTL de las especificaciones ECAP obtenidas en la *sección 5.2.9.2.3. Especificación formal de comportamiento*, en las trayectorias obtenidas en la *sección 5.2.9.2.4. Interpretación semántica temporal*, mediante la herramienta nuXmv.

Por cada trayectoria, se modelaron dos autómatas (una para caso de éxito y una para caso de éxito y fracaso) y se generó el respectivo programa SMV (*Symbolic Model Verifier*) que lo representa. Por ejemplo, para el RF1 “Separar diferentes tipos de existencias”, los autómatas para las trayectorias π' y π'' se muestran en la Figura 31 y Figura 32. El código fuente del programa SMV para cada autómata se describe en la Tabla 51 y en la Tabla 52, respectivamente.

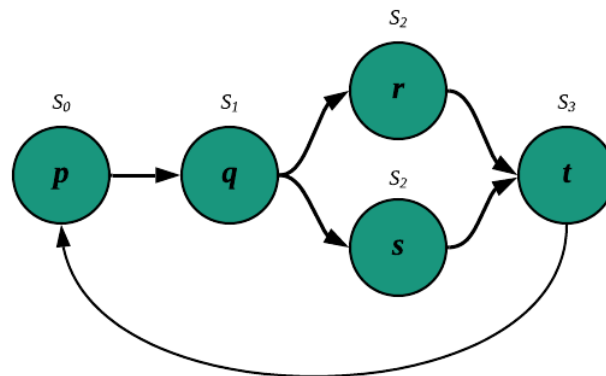


Figura 31. Autómata para la trayectoria de estados π' .

Tabla 51 . Código fuente del programa Req1-otpaEPAI-E.smv

```

MODULE main

VAR
    state : {s0,s1,s2,s3};
    empleado_1a : {separaArticulo, verificaExistenciaDeArticulo, actualizaArticulo,
actualizaInventario,none};
    empleado_1b : { registraArticulo, none};

ASSIGN
    init(state) := s0;
    next(state) := case
        (state = s0) & (empleado_1a = separaArticulo) & (empleado_1b = none) : s1;
        (state = s0) : s1;
        (state = s1) & (empleado_1a = verificaExistenciaDeArticulo) : s2;
        (state = s1) : s2;
        (state = s2) & ((empleado_1a = actualizaArticulo) | (empleado_1b =
registraArticulo)) : s3;
        (state = s2) : s3;
        (state = s3) & (empleado_1a = actualizaInventario) : s0;
        (state = s3) : s0;
        TRUE : state;
    esac;

    init(empleado_1a) := separaArticulo;
    next(empleado_1a) := case
        (state = s0) : separaArticulo;
        (state = s1) : verificaExistenciaDeArticulo;
        (state = s2) : actualizaArticulo;
        (state = s3) : actualizaInventario;
        TRUE : empleado_1a;
    esac;

    init(empleado_1b) := none;
    next(empleado_1b) := case
        (state = s0) : none;
        (state = s1) : none;
        (state = s2) : registraArticulo;
        (state = s3) : none;
        TRUE : empleado_1b;
    esac;

    LTLSPEC G(X (empleado_1a = separaArticulo) ->(
        X X(empleado_1a = verificaExistenciaDeArticulo) & F(
            !X X X (empleado_1a = actualizaArticulo | empleado_1b = registraArticulo) & !X X X
            X empleado_1a = actualizaInventario
        )
    )
);

```

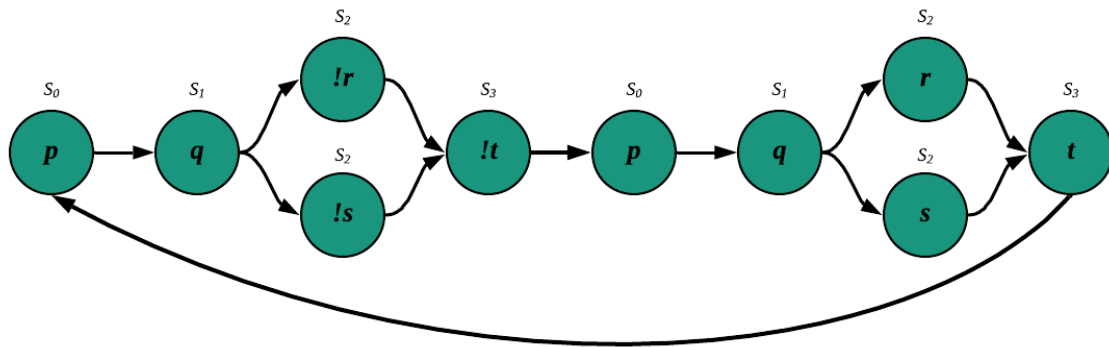


Figura 32 . Autómata para la trayectoria de estados π'' .

Tabla 52 . Código fuente del programa Req1-otpaEPAI-EF.smv.

```

MODULE main
VAR
    state : {s0,s1,s2,s3,s0_1,s1_1,s2_1,s3_1,s0_2,s1_2,s2_2,s3_2,s0_3,s1_3,s2_3,s3_3};
    empleado_1a : {separaArticulo, verificaExistenciaDeArticulo, actualizaArticulo,
actualizaInventario,none};
    empleado_1b : { registraArticulo, none};
ASSIGN
    init(state) := s0;
    next(state) := case
        -- caso de fracaso 1
        (state = s0) & (empleado_1a = separaArticulo) & (empleado_1b = none) : s1;
        (state = s0) : s1;
        (state = s1) & (empleado_1a = verificaExistenciaDeArticulo) : s2;
        (state = s1) : s2;
        (state = s2) & ((empleado_1a = actualizaArticulo) | (empleado_1b = registraArticulo)) :
s3;
        (state = s2) : s3;
        (state = s3) & (empleado_1a = actualizaInventario) : s0;
        (state = s3) : s0_1;
        -- caso de éxito 1
        (state = s0_1) & (empleado_1a = separaArticulo) & (empleado_1b = none) : s1_1;
        (state = s0_1) : s1_1;
        (state = s1_1) & (empleado_1a = verificaExistenciaDeArticulo) : s2_1;
        (state = s1_1) : s2_1;
        (state = s2_1) & ((empleado_1a = actualizaArticulo) | (empleado_1b =
registraArticulo)) : s3_1;
        (state = s2_1) : s3_1;
        (state = s3_1) & (empleado_1a = actualizaInventario) : s0_2;
        (state = s3_1) : s0_2;
        -- caso de fracaso 2
        (state = s0_2) & (empleado_1a = separaArticulo) & (empleado_1b = none) : s1_2;
        (state = s0_2) : s1_2;
        (state = s1_2) & (empleado_1a = verificaExistenciaDeArticulo) : s2_2;
        (state = s1_2) : s2_2;
        (state = s2_2) & ((empleado_1a = actualizaArticulo) | (empleado_1b =

```

```

registraArticulo)) : s3_2;
    (state = s2_2) : s3_2;
    (state = s3_2) & (empleado_1a = actualizaInventario) : s0_3;
    (state = s3_2) : s0_3;
    -- caso de éxito 2
    (state = s0_3) & (empleado_1a = separaArticulo) & (empleado_1b = none) : s1_3;
    (state = s0_3) : s1_3;
    (state = s1_3) & (empleado_1a = verificaExistenciaDeArticulo) : s2_3;
    (state = s1_3) : s2_3;
    (state = s2_3) & ((empleado_1a = actualizaArticulo) | (empleado_1b =
registraArticulo)) : s3_3;
    (state = s2_3) : s3_3;
    (state = s3_3) & (empleado_1a = actualizaInventario) : s0;
    (state = s3_3) : s0;
    TRUE : state;
    esac;

init(empleado_1a) := separaArticulo;
next(empleado_1a) := case
-- caso de fracaso 1
    (state = s0) : separaArticulo;
    (state = s1) : verificaExistenciaDeArticulo;
    (state = s2) : none;
    (state = s3) : none;
-- caso de éxito 1
    (state = s0_1) : separaArticulo;
    (state = s1_1) : verificaExistenciaDeArticulo;
    (state = s2_1) : actualizaArticulo;
    (state = s3_1) : actualizaInventario;
-- caso de fracaso 2
    (state = s0_2) : separaArticulo;
    (state = s1_2) : verificaExistenciaDeArticulo;
    (state = s2_2) : none;
    (state = s3_2) : none;
-- caso de éxito 2
    (state = s0_3) : separaArticulo;
    (state = s1_3) : verificaExistenciaDeArticulo;
    (state = s2_3) : actualizaArticulo;
    (state = s3_3) : actualizaInventario;
    TRUE : empleado_1a;
    esac;

init(empleado_1b) := none;
next(empleado_1b) := case
-- caso de fracaso 1
    (state = s0) : none;
    (state = s1) : none;
    (state = s2) : none;
    (state = s3) : none;
-- caso de éxito 1
    (state = s0_1) : none;
    (state = s1_1) : none;
    (state = s2_1) : registraArticulo;
    (state = s3_1) : none;
-- caso de fracaso 2
    (state = s0_2) : none;
    (state = s1_2) : none;
    (state = s2_2) : none;
    (state = s3_2) : none;
-- caso de éxito 2

```



```

(state = s0_3) : none;
(state = s1_3) : none;
(state = s2_3) : registraArticulo;
(state = s3_3) : none;
TRUE : empleado_1b;
esac;

LTLSPEC G(X (empleado_1a = separaArticulo) ->(
  X X(empleado_1a = verificaExistenciaDeArticulo) & F(
    !X X X (empleado_1a = actualizaArticulo | empleado_1b = registraArticulo) & !X X X
    X empleado_1a = actualizaInventario
  )
)
);

```

Para una mejor comprensión del código fuente del programa SMV, se recomienda revisar la *sección 2 Verificación de satisfacibilidad de especificaciones de comportamiento del Anexo F Herramientas utilizadas* ubicada en (Piedrola Domínguez N., 2019, pp. 344).

Como se puede observar en el código fuente de la Tabla 51 y de la Tabla 52, se han definido y utilizado fórmulas válidas de tipo enumeración en lugar de las proposiciones atómicas p, q, r y s (p. ej., la variable de tipo enumeración para la proposición atómica p es “empleado_1a = separaArticulo”), para mejorar la comprensión de los autómatas dentro del contexto del RF Realizar el seguimiento de cantidades de artículos en existencias.

Los resultados de la verificación SC demostraron que las fórmulas LTL son satisfacibles en las trayectorias de estados consideradas al no generar contraejemplos. Los resultados de la verificación se muestran en la Tabla 53.

Tabla 53 . Resultados de la verificación de satisfacibilidad de las fórmulas LTL.

Requerimiento	Caso de Verificación	Resultado de la Verificación
1. Separar diferentes tipos de existencias.	01 - Éxito	Satisfacible
	02 - Éxito y Fracaso	Satisfacible
2. Realizar el seguimiento de cantidades de artículos en existencias.	03 - Éxito	Satisfacible
	04 - Éxito y Fracaso	Satisfacible
3. Realizar el seguimiento de las ubicaciones de los artículos.	05 - Éxito	Satisfacible
	06 - Éxito y Fracaso	Satisfacible

P. ej., para el RF Separar diferentes tipos de existencias, se realizaron dos verificaciones, uno para casos de éxito y otro para casos de éxito y fracaso. Para ambos autómatas, el resultado en la verificación SC en nuXmv fue verdadero (true), lo que indica que la fórmula es satisfacible en los modelos que representan el comportamiento basado en el RF Separar diferentes tipos de existencias ($\pi' \models \varphi$ y $\pi'' \models \varphi$, respectivamente) y no se generan

contraejemplos. Si sucediera lo contrario, significa que la fórmula LTL no se satisface en la trayectoria considerada ($\pi' \not\models \varphi$ o $\pi'' \not\models \varphi$, respectivamente).

Para documentar la verificación SC, se utilizó la documentación descrita en el *Anexo F. Documentación SC*. Todos los documentos realizados para la verificación SC de las fórmulas del patrón de análisis PAI (que incluyen los autómatas y programas SMV de los RFs restantes) se encuentran en el directorio *Entregable 9/SC/*, almacenado en el disco que acompaña a esta tesis (para mayor referencia consultar el *Anexo G. Contenido del disco*).

5.2.10. Instanciación de la ontología

Una vez definidas las especificaciones de comportamiento que deben satisfacerse en la ontología OTPAE, en esta fase se crearon los individuos (aserciones de clases y aserciones de propiedades de objetos temporales) de la ontología necesarios para representar trayectorias de estados. Un ejemplo de las aserciones de clases y de propiedades de objetos se presenta en la Tabla 54, cuya tabla muestra la equivalencia entre la sintaxis DL y la sintaxis funcional de OWL-DL.

Tabla 54. Ejemplo de aserciones de clases y de propiedades de objetos en el RF Separar diferentes tipos de existencias.

Sintaxis DL	Sintaxis Funcional
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Cliente domain-pai:Valentino)</i>
<i>Event(Req1-Ev1)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req1-Ev1)</i>
-- Aserciones de propiedades de objeto --	
<i>separa(Valentino,Req1-Ev1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:separa domain-pai:Valentino domain-pai:Req1-Ev1)</i>
<i>separa(Req1-Ev1,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:separa domain-pai:Req1-Ev1 domain-pai:Libro1)</i>

P. ej., la aserción de clase *Empleado(Valentino)*, indica que Valentino es un individuo de la clase Empleado y la aserción de clase *Event(Req1-Ev1)*, indica que Req1-Ev1 es un individuo de la clase Event. De la misma manera, la aserciones de propiedad de objeto *separa(Valentino,Req1-Ev1)* y *separa(Req1-Ev1,Libro1)*, indican que el individuo Valentino separa el individuo Libro1 a través del individuo Req1-Ev1.

Los individuos en cada trayectoria de comportamiento en el patrón de análisis PAI, se describen en la sección 1.2. *Componente extensional* del *Anexo B. Ontología OTPAE del patrón de análisis PAI*, cuyas tablas muestran la equivalencia entre la sintaxis DL y la sintaxis funcional de OWL-DL.

En este capítulo, se presentaron los resultados y la documentación generada en el proceso de creación de la ontología temporal (OTPAE) del patrón de análisis para inventarios (PAI), tal y como lo establece la metodología MECOTPA. La OTPAE del PAI, es actualmente, la tercera ontología temporal que se describe en nuestro grupo de investigación y se utilizó en este trabajo de investigación para realizar el caso de estudio descrito en el *Capítulo VI. Caso de estudio: Combinación de la OTPAE del PAI con la OTPAE del PAER* para obtener un patrón de análisis compuesto utilizando el método descrito en el *Capítulo IV. Método para la Combinación de Ontologías Temporales de Patrones de Análisis*.

Capítulo VI. Caso de estudio: Combinación de la OTPAE del PAI con la OTPAE del PAER

En este capítulo, se presenta la descripción de la documentación y las actividades en el seguimiento del método para la Combinación de Ontologías Temporales de Patrones de Análisis (COMOTPA), para combinar la Ontología Temporal del Patrón de Análisis para Inventarios (OTPAE del PAI) con la Ontología Temporal del Patrón de Análisis de Entidades Reutilizables (OTPAE del PAER). La ontología temporal del PAER fue construida en (Piedrola Domínguez N., 2019) y la ontología temporal del PAI, construida en este trabajo de tesis, se describe en el *Capítulo V. Creación de la Ontología Temporal del Patrón de Análisis para Inventarios*.

6.1. Combinación de la Ontología Temporal del PAI con la Ontología Temporal del PAER

En esta sección, se describe el proceso realizado para combinar la Ontología Temporal (OTPAE) del Patrón de Análisis para Inventarios (PAI) con la Ontología Temporal (OTPAE) del Patrón de Análisis de Entidades Reutilizables (PAER) como caso de estudio, utilizando el método COMOTPA descrito en el *Capítulo IV. Método para la Combinación de Ontologías Temporales de Patrones de Análisis*. Para una mejor comprensión del contenido de este capítulo, se presenta nuevamente la Figura 4 que describe el proceso del método COMOTPA, en donde se representan gráficamente las entradas y salidas del método, en la Figura 5 se describen a más detalle las actividades a realizar en cada fase del método.

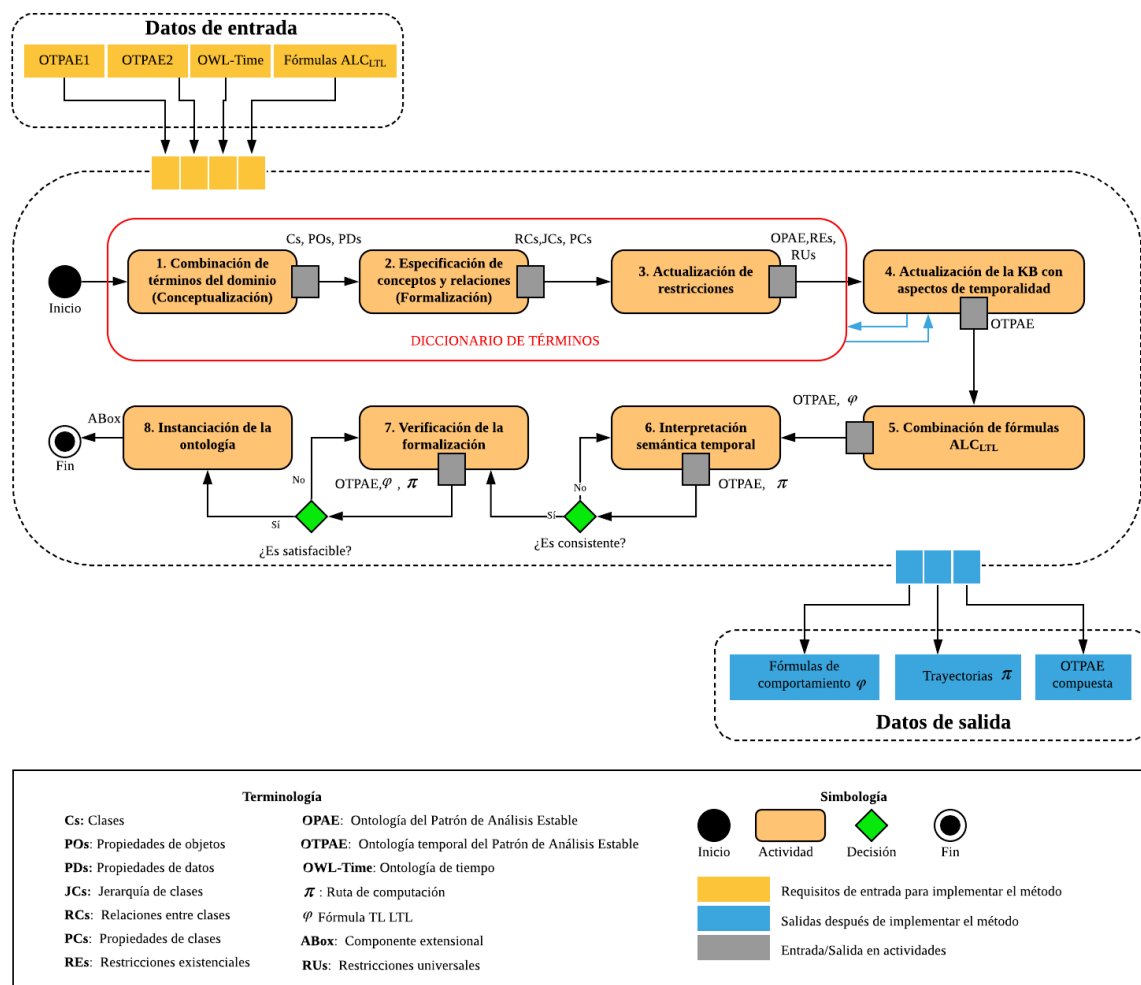


Figura 4. Fases del método COMOTPA, con sus entradas y salidas.

En este capítulo, se presentan los resultados obtenidos en cada una de las actividades de las ocho fases del método COMOTPA durante la combinación de la OTPAE del PAER con la OTPAE del PAI. Los resultados presentados, corresponden a las fases: Combinación de términos del dominio (Conceptualización), Especificación de conceptos y relaciones (Formalización), Actualización de restricciones, Actualización de la KB con aspectos de temporalidad, Combinación de fórmulas ALC_{LTL} , Interpretación semántica temporal, Verificación de la formalización e, Instanciación de la ontología.

6.1.1 Combinación de términos del dominio (Conceptualización)

En esta fase, se realizaron las tres actividades definidas en el método COMOTPA: Combinar clases, Combinar propiedades de objetos y Combinar propiedades de datos.

6.1.1.1. Combinar clases

En esta actividad, se combinaron las clases de las dos OTPAEs, para esta actividad se tomaron en cuenta solamente las clases estáticas de las dos ontologías tal y como se establece en el método COMOTPA. El proceso consistió básicamente en comparar todas las clases estáticas de la primera ontología (OTPAE del PAER) con todas las clases estáticas de la segunda ontología (OTPAE del PAI) con la finalidad de identificar aquellas clases (conceptos) que puedan combinarse para obtener el conjunto de clases del nuevo dominio.

A continuación, en la Figura 33 se presenta el tipo de relación identificada entre el conjunto de clases estáticas de la ontología temporal del PAER con el conjunto de clases estáticas de la ontología temporal del PAI, las flechas de color negro significan que son clases diferentes, las flechas de color verde significan que son clases sinónimas y las de color azul significan que son clases iguales, es decir, utilizan el mismo nombre en las dos ontologías.

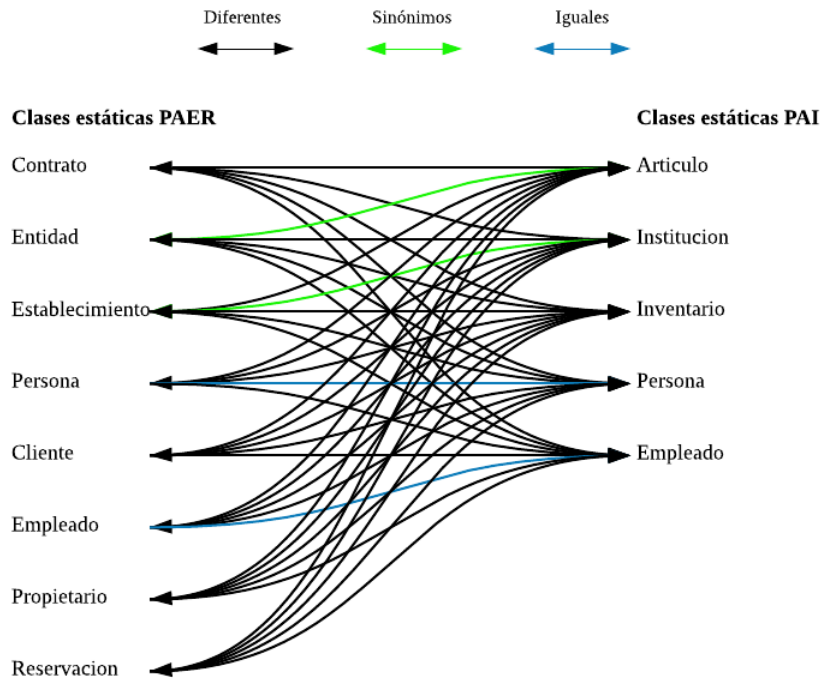


Figura 33. Relación de similitud entre las clases estáticas de la OTPAE del PAER y la OTPAE del PAI.

Con base en el tipo de relación que existe entre las diferentes clases de las dos ontologías se obtuvo el nuevo conjunto de clases que representarán el nuevo dominio. En las clases con el tipo de relación “sinónimos”, se tomaron solamente una de ellas por cada par y se incluyeron al nuevo conjunto de clases, en las clases con el tipo de relación “iguales”, se incluyeron, sin repetir, al nuevo conjunto de clases y en las clases con el tipo de relación “diferentes”, se incluyeron automáticamente al nuevo conjunto de clases. En la Tabla 55 se muestran las clases estáticas que pertenecen a la ontología temporal del PAER, las clases estáticas que pertenecen a la ontología temporal del PAI y las clases que pertenecerán a la ontología temporal compuesta (de aquí en adelante como OTPAE del PAIyER) y que son el resultado de la combinación de clases estáticas de las dos OTPAEs.

Tabla 55. Conjunto de clases estáticas correspondientes a las OTPAEs de estudio.

Clases estáticas del PAER	Clases estáticas del PAI	Clases estáticas del PAIyER
Contrato		Contrato
Entidad	Artículo	Entidad
Establecimiento	Institucion	Institucion
Persona	Persona	Persona
Cliente		Cliente
Empleado	Empleado	Empleado
Propietario		Propietario
Reservacion		Reservacion
	Inventario	Inventario

Como se observa en la Tabla 55, las clases sinónimas fueron las únicas que cambiaron, por ejemplo, para las clases Entidad y Artículo, se tomó la clase Entidad como nueva clase del patrón de análisis PAIyER, de igual manera para las clases Establecimiento e Institucion se tomó le clase Institucion como nueva clase del patrón de análisis PAIyER. Los nombres de las demás clases formarán parte del nuevo patrón de análisis sin ninguna modificación.

Como resultado de esta actividad, se obtuvo el conjunto de clases estáticas que formarán parte de la nueva OTPAE del patrón de análisis PAIyER.

6.1.1.2. Combinar propiedades de objetos

En esta actividad, se combinaron las propiedades de objetos (POs) de las dos OTPAEs, se tomaron en cuenta solamente las POs estáticas de las dos ontologías tal y como se establece en el método COMOTPA. El proceso consistió básicamente en comparar todas las propiedades de objetos estáticas de la primera ontología (OTPAE del PAER) con todas las propiedades de objetos estáticas de la segunda ontología (OTPAE del PAI) con la finalidad de identificar aquellas propiedades que puedan combinarse para obtener el conjunto de propiedades de objetos del nuevo dominio.

En las propiedades de objetos con el tipo de relación “iguales”, se incluyeron, sin repetir, al nuevo conjunto y las POs con el tipo de relación “diferentes”, se incluyeron automáticamente al nuevo conjunto de POs. En este caso de estudio, no se encontraron propiedades de objetos con el tipo de relación “sinónimos”.

A continuación, en la Tabla 56 se presenta la relación que hay entre el conjunto de propiedades de objetos estáticas de la ontología temporal del PAER y la ontología temporal del PAI. Para mayor claridad, se presentan de color azul las propiedades de objetos que se repiten en ambas ontologías. Es decir, propiedades de objetos que son iguales en las dos ontologías. Las propiedades de color negro, representan propiedades de objetos diferentes, no tienen sinónimos ni se repiten en las dos ontologías temporales.

Tabla 56. Relación de similitud entre propiedades de objetos de la OTPAE del PAER con la OTPAE del PAI.

POs estáticas del PAER	POs estáticas del PAI	POs estáticas del PAIyER
actualiza	actualiza	actualiza
laboraEn	laboraEn	laboraEn
tieneDatosDe	tieneDatosDe	tieneDatosDe
seActualizaPor	seActualizaPor	seActualizaPor
esCreadaPor		esCreadaPor
crea		crea
aplicaPolíticas		aplicaPolíticas
adquiere		adquiere
esAdquiridoPor		esAdquiridoPor
devuelve		devuelve

esCanceladaPor		esCanceladaPor
daDeBaja		daDeBaja
esDadaDeBajaPor		esDadaDeBajaPor
esDevueltaPor		esDevueltaPor
esDuenyoDe		esDuenyoDe
esExpedidoPor		esExpedidoPor
esModificadaPor		esModificadaPor
esReservadaPor		esReservadaPor
esUsadaPor		esUsadaPor
esVerificadaPor		esVerificadaPor
expide		expide
cancela		cancela
modifica		modifica
perteneceA		perteneceA
registraUsoDe		registraUsoDe
reserva		reserva
suUsoEsRegistradaPor		suUsoEsRegistradaPor
usa		usa
verificaDisponibilidad		verificaDisponibilidad
	buscaExistenciasDe	buscaExistenciasDe
	especificaTipoExistenciasDe	especificaTipoExistenciasDe
	seRegistraPor	seRegistraPor
	seSeparaPor	seSeparaPor
	seUbicaPor	seUbicaPor
	existenciaSeVerificaPor	existenciaSeVerificaPor
	existenciasSeBuscaPor	existenciasSeBuscaPor
	existenciasSeObtienePor	existenciasSeObtienePor
	obtieneExistenciasDe	obtieneExistenciasDe
	obtieneUbicacionDe	obtieneUbicacionDe
	registra	registra
	separa	separa
	solicitaUbicacionDe	solicitaUbicacionDe
	tipoExistenciasSeEspecificaPor	tipoExistenciasSeEspecificaPor
	ubica	ubica
	ubicacionSeObtienePor	ubicacionSeObtienePor
	ubicacionSeSolicitaPor	ubicacionSeSolicitaPor
	verificaExistenciaDe	verificaExistenciaDe

Como resultado de esta actividad, se obtuvo el conjunto de propiedades de objetos estáticas que formarán parte de la nueva OTPAE del patrón de análisis PAIyER.

6.1.1.3. Combinar propiedades de datos

En esta actividad, se combinaron las propiedades de datos (PDs) de las dos OTPAEs, se tomó en cuenta solamente las propiedades de datos estáticas de las dos ontologías tal y como se establece en el método COMOTPA. El proceso consistió básicamente en comparar todas las propiedades de datos estáticas de la primera ontología (OTPAE del PAER) con todas las

propiedades de objetos estáticas de la segunda ontología (OTPAE del PAI) con la finalidad de identificar aquellas propiedades que puedan combinarse para obtener el conjunto de propiedades de datos del nuevo dominio.

En las propiedades de datos con el tipo de relación “iguales”, se incluyeron, sin repetir, al nuevo conjunto y las PDs con el tipo de relación “diferentes”, se incluyeron automáticamente al nuevo conjunto de PDs. En este caso de estudio, no se encontraron propiedades de datos con el tipo de relación “sinónimos”.

A continuación, en la Tabla 57 se presenta la relación que hay entre el conjunto de propiedades de datos estáticas de la ontología temporal del PAER y la ontología temporal del PAI. Para mayor claridad, se presentan de color azul las propiedades de datos que se repiten en ambas ontologías. Las propiedades presentadas de color negro, representan propiedades diferentes, es decir, no tienen sinónimos ni se repiten en las dos ontologías temporales.

Tabla 57. Relación de similitud entre propiedades de datos de la OTPAE del PAER con la OTPAE del PAI.

PD estáticas del PAER	PD estáticas del PAI	PD estáticas del PAIyER
tieneComportamiento	tieneComportamiento	tieneComportamiento
tieneNombre	tieneNombre	tieneNombre
tieneEspecificidad	tieneEspecificidad	tieneEspecificidad
tieneGiro	tieneGiro	tieneGiro
tieneId	tieneId	tieneId
tieneNivelEstabilidad	tieneNivelEstabilidad	tieneNivelEstabilidad
tienePercepcion	tienePercepcion	tienePercepcion
tieneTipo	tieneTipo	tieneTipo
tieneDireccion		tieneDireccion
tieneNoConfirmacion		tieneNoConfirmacion
tieneTelefono		tieneTelefono
tieneDisponibilidad		tieneDisponibilidad
tieneFechaFin		tieneFechaFin
tieneFechaInicio		tieneFechaInicio
tieneFolio		tieneFolio
tieneNoReservacion		tieneNoReservacion
tienePrecio		tienePrecio
tieneRfc		tieneRfc
tieneTipoPago		tieneTipoPago
	tieneEdad	tieneEdad
	tieneUbicacion	tieneUbicacion
	tieneCantidad	tieneCantidad

Como resultado de esta actividad, se obtuvo el conjunto de propiedades de datos estáticas que formarán parte de la nueva OTPAE del patrón de análisis PAIyER.

6.1.2. Actualización de la especificación de conceptos y relaciones (Formalización)

En esta fase, se realizó una actualización en la especificación de las clases y sus relaciones utilizando una jerarquía. Además, se especificaron las propiedades de cada clase. A continuación, se presentan los resultados obtenidos en cada actividad.

6.1.2.1. Especificar jerarquía de clases

En esta actividad, se realizó una especificación explícita del conjunto de clases mediante una jerarquía de clases. Para definir la jerarquía fue necesario verificar que cada clase se encontrara en el nivel correcto jerárquicamente (tomando en cuenta el significado de los términos clase y herencia en el contexto de la programación orientada a objetos), para ello fue necesario verificar el nivel jerárquico en que se encontraba cada clase antes de combinarla, con la finalidad de ubicarla en el nivel adecuado. Finalmente, se obtuvo una nueva jerarquía de clases que representa el nuevo dominio. A continuación, en la Tabla 58 se presenta la jerarquía de clases para el patrón de análisis PAIyER.

Tabla 58. Especificación de la jerarquía de clases para el PAIyER.

Clases
Contrato
Entidad
Institucion
Inventario
Persona
Cliente
Empleado
Propietario
Reservación

6.1.2.2. Especificar relaciones entre clases

Debido a que, durante la combinación, las relaciones pueden modificarse y/o eliminarse. En esta actividad, se actualizaron las relaciones entre clases. Para ello, se especificaron en una tabla (Tabla 59) todas las relaciones que existen entre las clases, especificando qué clase es la que realiza la acción (dominio), el nombre de la relación (propiedad de objeto), la clase sobre la cual se realiza la acción (rango) y el tipo de característica (funcional o inversa funcional) de la propiedad de objeto. Parar realizar esta actividad, fue necesario conocer las relaciones existentes entre las clases de las dos OTPAEs de estudio antes de la combinación. A partir de esto, se actualizaron las relaciones entre las clases del nuevo conjunto de clases, tomando en cuenta todos los elementos (clases, POs, PDs) que fueron modificados durante el proceso de combinación. A continuación, en la Tabla 59 se presentan las relaciones entre las clases del patrón de análisis PAIyER.

Tabla 59. Relaciones entre clases del patrón de análisis PAIyER.

Dominio	Relación	Rango	Característica
Contrato	esAdquiridoPor	Cliente	Funcional
Contrato	esExpedidoPor	Empleado	Funcional
Contrato	tieneDatosDe	Empleado Entidad Institucion Propietario Reservacion	N/A
Entidad	esDevueltaPor	Cliente	Funcional
Entidad	esUsadaPor	Cliente	Funcional
Entidad	esVerificadaPor	Empleado	Funcional
Entidad	perteneceA	Propietario	Funcional
Entidad	seRegistraPor	Empleado	Funcional
Entidad	seSeparaPor	Empleado	Funcional
Entidad	existenciasSeBuscaPor	Empleado	Funcional
Entidad	existenciaSeVerificaPor	Empleado	Funcional
Entidad	tipoExistenciasSeEspecificaPor	Empleado	Funcional
Entidad	existenciasSeObtienePor	Empleado	Funcional
Entidad	seUbicaPor	Empleado	Funcional
Entidad	ubicacionSeSolicitaPor	Empleado	Inversa funcional
Entidad	ubicacionSeObtienePor	Empleado	Funcional
Cliente	adquiere	Contrato	Inversa funcional
Cliente	daDeBaja	Reservacion	Inversa funcional
Cliente	devuelve	Entidad	Inversa funcional
Cliente	modifica	Reservacion	Inversa funcional
Cliente	reserva	Entidad	Inversa funcional
Cliente	usa	Entidad	Inversa funcional
Empleado	actualiza	Entidad Reservacion Inventario	Inversa funcional
Empleado	aplicaPoliticlas	Cliente Reservacion	N/A
Empleado	cancela	Reservacion	Inversa funcional
Empleado	crea	Reservacion	Inversa funcional
Empleado	expide	Contrato	Inversa funcional
Empleado	registraUsoDe	Entidad	Inversa funcional
Empleado	verificaDisponibilidadDe	Entidad	Inversa funcional
Empleado	buscaExistenciasDe	Entidad	Inversa funcional
Empleado	buscaExistenciasDe	Entidad	Inversa funcional
Empleado	especificaTipoExistenciasDe	Entidad	Inversa funcional
Empleado	laboraEn	Institucion	Funcional
Empleado	obtieneExistenciasDe	Entidad	Inversa funcional
Empleado	obtieneUbicacionDe	Entidad	Inversa funcional
Empleado	registra	Entidad	Inversa funcional
Empleado	separa	Entidad	Inversa funcional
Empleado	solicitaUbicacionDe	Entidad	Funcional
Empleado	ubica	Entidad	Inversa funcional
Empleado	verificaExistenciaDe	Entidad	Inversa funcional

Propietario	esDuenyoDe	Entidad	N/A
Reservacion	esCreadaPor	Empleado	Funcional
Institucion	tieneDatosDe	Empleado	N/A
Inventario	seActualizaPor	Empleado	Funcional

P. ej., en la primera fila de la Tabla 59, hay una relación entre las clases Contrato y Cliente por medio de la propiedad de objeto esAdquiridoPor, que indica que un Contrato es adquirido por un Cliente, en donde Contrato y Cliente corresponden al dominio y rango de la propiedad de objeto esAdquiridoPor. Cada propiedad de objeto puede tener una propiedad funcional o inversa funcional que corresponden a restricciones de cardinalidad global (Piedrola Domínguez N., 2019, pp. 328).

6.1.2.3. Especificar propiedades de las clases

Cada clase de las OTPAEs de estudio tienen originalmente asignado un conjunto de propiedades de datos. Sin embargo, durante la combinación, éstas se pueden eliminar y/o modificar. Por tal motivo, en esta actividad se actualizaron las propiedades de datos del nuevo conjunto de clases. A continuación, en la Tabla 60 se presentan las propiedades de cada clase del patrón de análisis PAIyER.

Tabla 60. Propiedades de las clases del patrón de análisis PAIyER

Concepto	Propiedades
Contrato	tieneFolio, tieneTipo, tieneNoReservacion, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion
Entidad	tieneId, tieneNombre, tieneTipo, tieneCantidad, tieneUbicacion, tienePrecio, tieneDisponibilidad, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion
Institucion	tieneId, tieneNombre, tieneGiro, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion
Inventario	tieneId, tieneNombre, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion
Persona	tieneNombre, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion
Cliente	tieneNombre, tieneDireccion, tieneTelefono, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion
Empleado	tieneId, tieneNombre, tieneEdad, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion
Propietario	tieneRfc, tieneNombre, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion
Reservación	tieneFechaInicio, tieneFechaFin, tieneNoConfirmacion, tieneTipoPago, tieneComportamiento, tieneEspecificidad, tieneNivelEstabilidad, tienePercepcion

Como resultado de esta actividad, se obtuvo el conjunto de clases con sus respectivas propiedades.

6.1.3. Actualización de restricciones

En esta fase, se realizó una actualización de las restricciones de cuantificación existencial y universal para la OPAE del PAIyER. A continuación, se describen los resultados obtenidos en cada actividad.

6.1.3.1. Actualizar restricciones existenciales

En esta actividad, se realizó una actualización de las restricciones existenciales y características (funcional, inversa funcional) de las clases y propiedades de objetos de la nueva ontología compuesta mediante una descripción lógica basada en la DL ALC y su equivalente en la sintaxis funcional de OWL-DL.

En Protégé, las restricciones de cuantificación existencial se definieron usando la palabra “*some*” (Figura 34) para especificar que un concepto se relaciona con al menos otro. A continuación, en la Tabla 61 se presentan las restricciones existenciales definidas en la OPAE del PAIyER.

Tabla 61. Restricciones existenciales de la OPAE del PAIyER.

Dominio	Relación	Restricción	Rango
Empleado	separa	some	Entidad
Empleado	registra	some	Entidad
Empleado	actualiza	some	Inventario
Empleado	buscaExistenciasDe	some	Entidad
Empleado	verificaExistenciaDe	some	Entidad
Empleado	obtieneUbicacionDe	some	Entidad
Empleado	especificaTipoExistenciasDe	some	Entidad
Empleado	obtieneExistenciasDe	some	Entidad
Empleado	ubica	some	Entidad
Empleado	solicitaUbicacionDe	some	Entidad
Empleado	cancela	some	Reservacion
Empleado	registraUsoDe	some	Entidad
Empleado	verificaDisponibilidadDe	some	Entidad
Cliente	adquiere	some	Entidad
Cliente	daDeBaja	some	Reservacion
Cliente	devuelve	some	Entidad
Cliente	modifica	some	Reservacion
Cliente	reserva	some	Entidad
Cliente	usa	some	Entidad
Institucion	tieneDatosDe	some	Empleado
Propietario	esDuenyoDe	some	Entidad

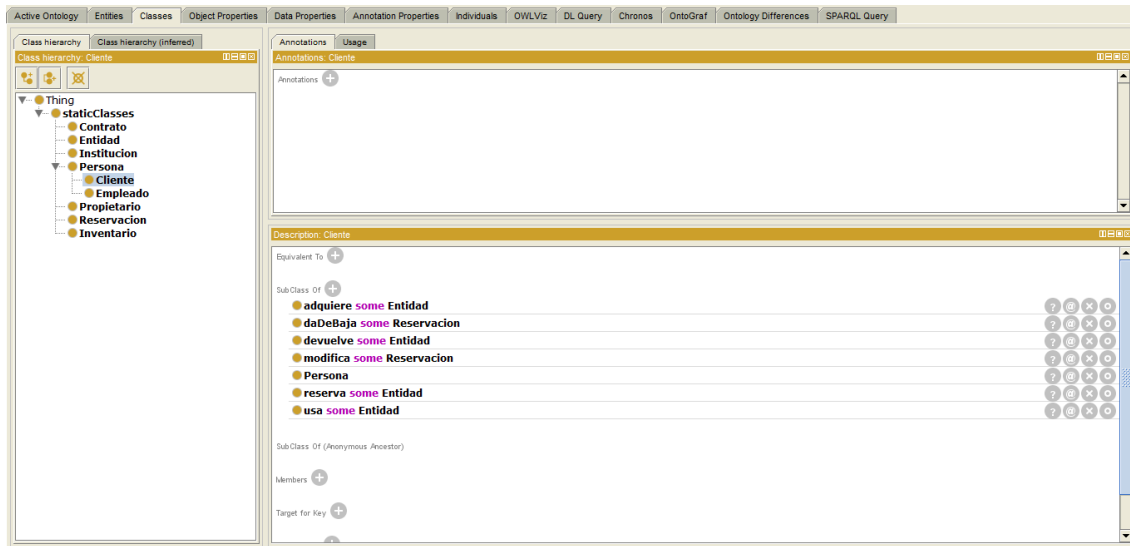


Figura 34. Definición de restricciones existenciales en Protégé.

6.1.3.2. Actualizar restricciones universales

En esta actividad, se realizó una actualización de las restricciones universales y características (funcional, inversa funcional) de las clases y propiedades de objetos de la nueva ontología compuesta mediante una descripción lógica basada en la DL ALC y su equivalente en la sintaxis funcional de OWL-DL.

En Protégé, las restricciones de cuantificación universal se definieron usando la palabra “only” (Figura 35) para especificar que un concepto se relaciona únicamente con otro. A continuación, en la Tabla 62 se presentan las restricciones universales definidas en la OPAE del PAIyER.

Tabla 62. Restricciones universales de la OPAE del PAIyER.

Restricciones universales			
Dominio	Relación	Restricción	Rango
Entidad	seSeparaPor	only	Empleado
Entidad	seRegistraPor	only	Empleado
Entidad	esDevueltaPor	only	Cliente
Entidad	esUsadaPor	only	Cliente
Entidad	esVerificadoPor	only	Empleado
Entidad	perteneceA	only	Propietario
Contrato	esAdquiridoPor	only	Cliente
Contrato	esExpedidoPor	only	Empleado
Contrato	tieneDatosDe	only	Cliente Empleado Entidad Institucion Propietario

			Reservacion
Empleado	laboraEn	only	Institucion
Empleado	actualiza	only	Reservacion
Empleado	aplicaPoliticass	only	Cliente Reservacion
Empleado	crea	only	Reservacion
Empleado	expide	only	Contrato
Reservacion	esCreadaPor	only	Empleado

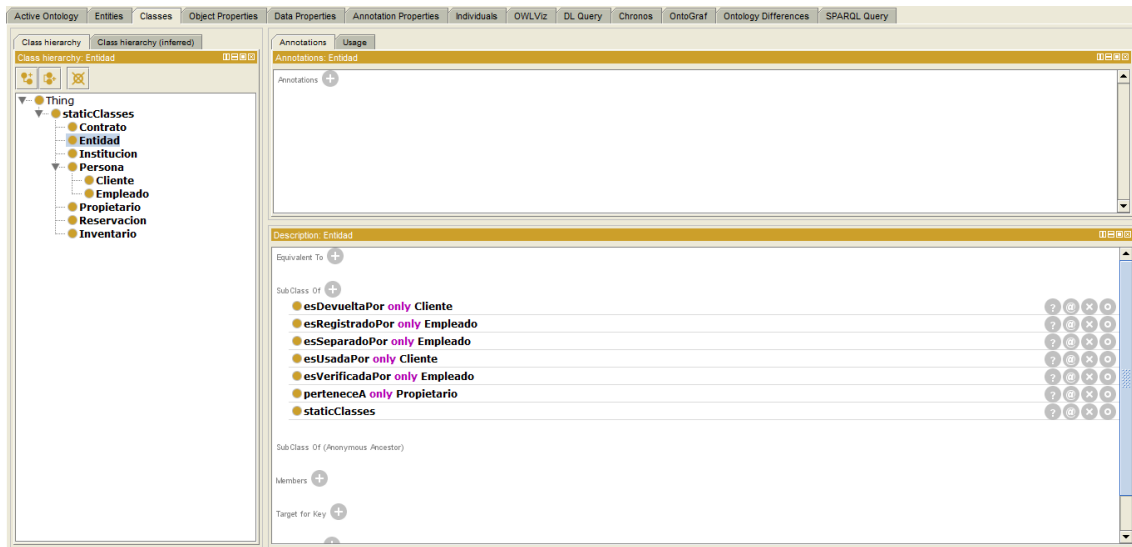


Figura 35. Definición de restricciones universales en Protégé.

Con la ayuda del editor de ontologías Protégé, se creó la ontología OPAE del PAIyER para almacenar la información (Figura 36) obtenida en las fases 6.1.2. *Actualización de la especificación de conceptos y relaciones (Formalización)* y 6.1.3. *Actualización de restricciones*. En el ámbito de las ontologías, ahora, los conceptos representan las clases o subclases, los roles abstractos representan propiedades de objeto y los roles concretos representan propiedades de datos.

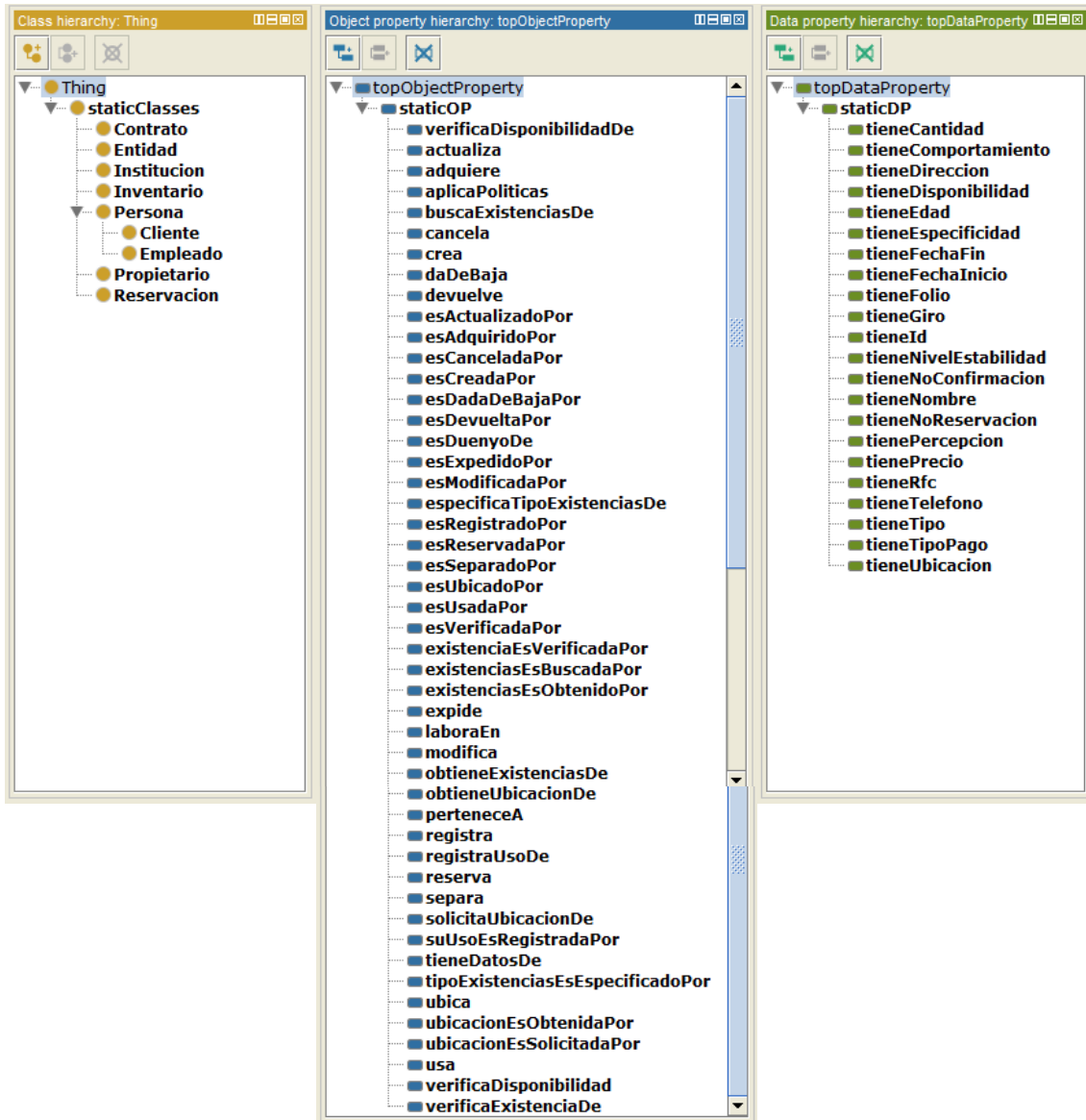


Figura 36. Ontología OPAE del patrón de análisis PAIyER, con clases, POs y PDs.

A continuación, en la Figura 37 y en la Figura 38 se presenta un ejemplo de la especificación de las relaciones y propiedades entre clases en el editor de ontologías Protégé.

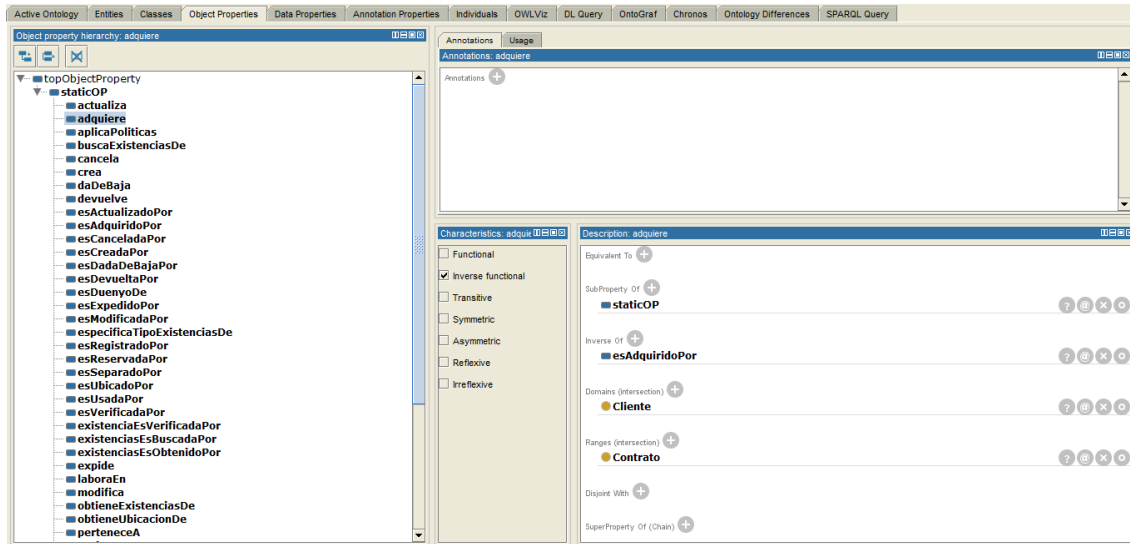


Figura 37. Relación entre clases de la OPAE del PAIyER.

P. ej. en la Figura 37, se especifica que la clase *Cliente* (dominio) se relaciona con la clase *Contrato* (rango) a través de la propiedad de objeto *adquiere*, para indicar que un cliente adquiere un contrato.

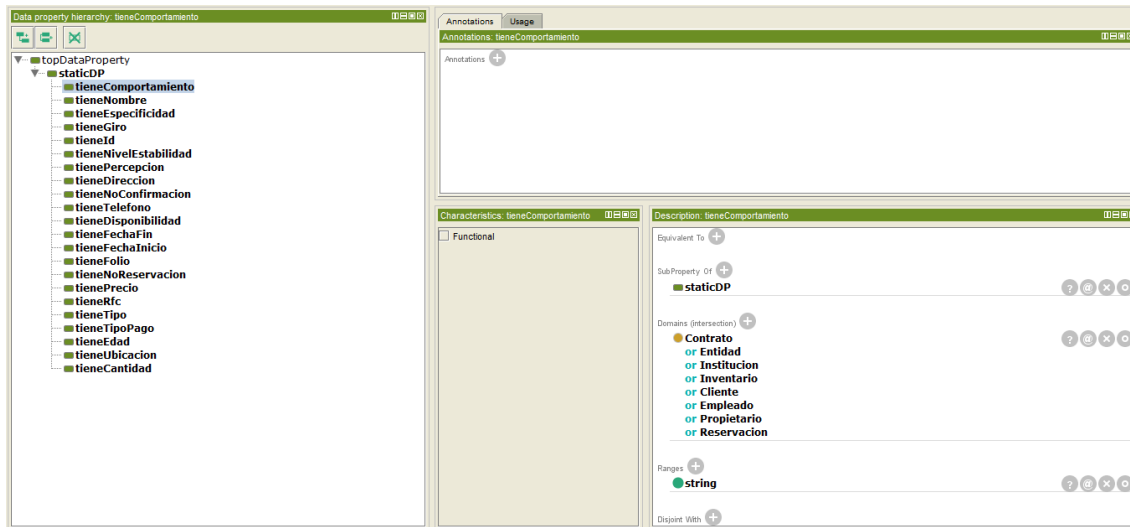


Figura 38. Propiedades de las clases de la OPAE del PAIyER.

P. ej., en la Figura 38, se especifica que las clases *Contrato*, *Entidad*, *Institucion*, *Inventario*, *Cliente*, *Empleado*, *Propietario* y *Reservacion*, tendrán la propiedad de dato *tieneComportamiento* que es de tipo *string*.

Hasta este punto del método COMOTPA, se tiene como resultado una ontología estática del patrón de análisis PAIyER, es decir, una OPAE. Todos los elementos definidos en la OPAE del PAIyER hasta este punto son estáticos, los elementos temporales se le agregarán en la siguiente fase. La ontología estática del patrón de análisis PAIyER se describe en el *Anexo*

C. *Ontología OPAE del patrón de análisis PAIyER*, cuyas tablas muestran la equivalencia entre la sintaxis DL y la sintaxis funcional de OWL-DL.

6.1.4. Actualización de la KB con aspectos de temporalidad

En esta fase, se incluyó en la ontología OPAE conceptos temporales que permiten representar el conocimiento dinámico. A continuación se describen los resultados obtenidos en cada actividad.

6.1.4.1. Generar la ontología OTPAE a partir de la ontología OPAE

En esta actividad se combinó la ontología estática (OPAE del PAIyER) con la ontología OWL-Time (proveída por el plugin Chronos en Protégé bajo el poder de expresividad *ALCROIN(D)*). Para realizar esta actividad, se utilizó el editor de ontologías Protégé en su versión 4.3 con el plugin Chronos configurado y el razonador Pellet para la verificación de consistencia de la nueva ontología temporal compuesta (OTPAE del PAIyER), el proceso para la configuración del plugin Chronos y el razonador Pellet se describe en (Piedrola Domínguez N., 2019) . A continuación, en la Figura 39 se presenta la nueva OTPAE, que incluye tanto elementos estáticos como elementos dinámicos, las clases, propiedades de objetos y propiedades de datos (estáticos y temporales) fueron separados para una mejor gestión. Las clases *dynamicClasses* y *staticClasses*, contienen las clases temporales y estáticas, respectivamente; las POs *dynamicOP* y *staticOP*, contienen las propiedades de objetos temporales y estáticas, respectivamente y; las PDs *dynamicDP* y *staticDP*, contienen las propiedades de datos temporales y estáticas, respectivamente.

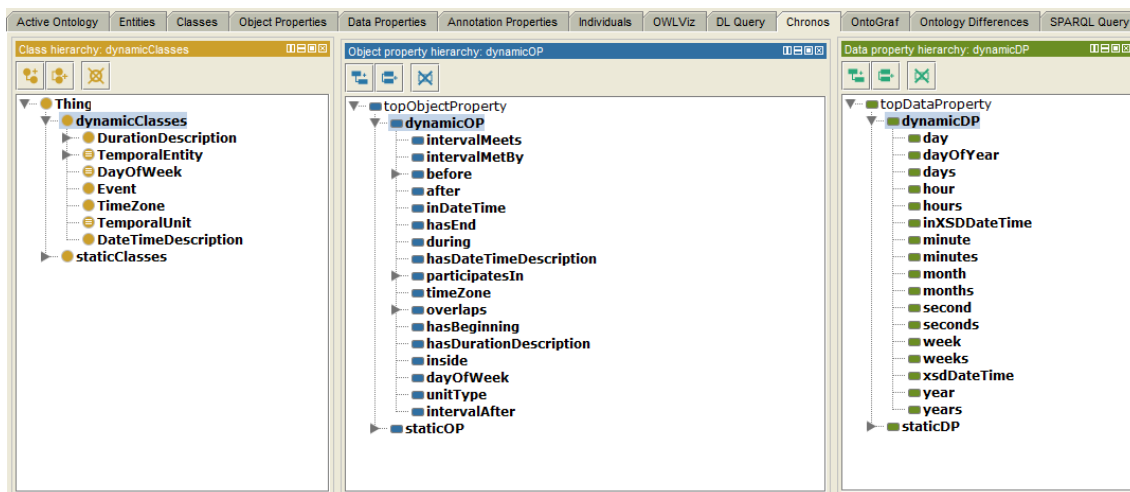


Figura 39. Ontología OTPAE del PAIyER con elementos separados (estáticos y temporales).

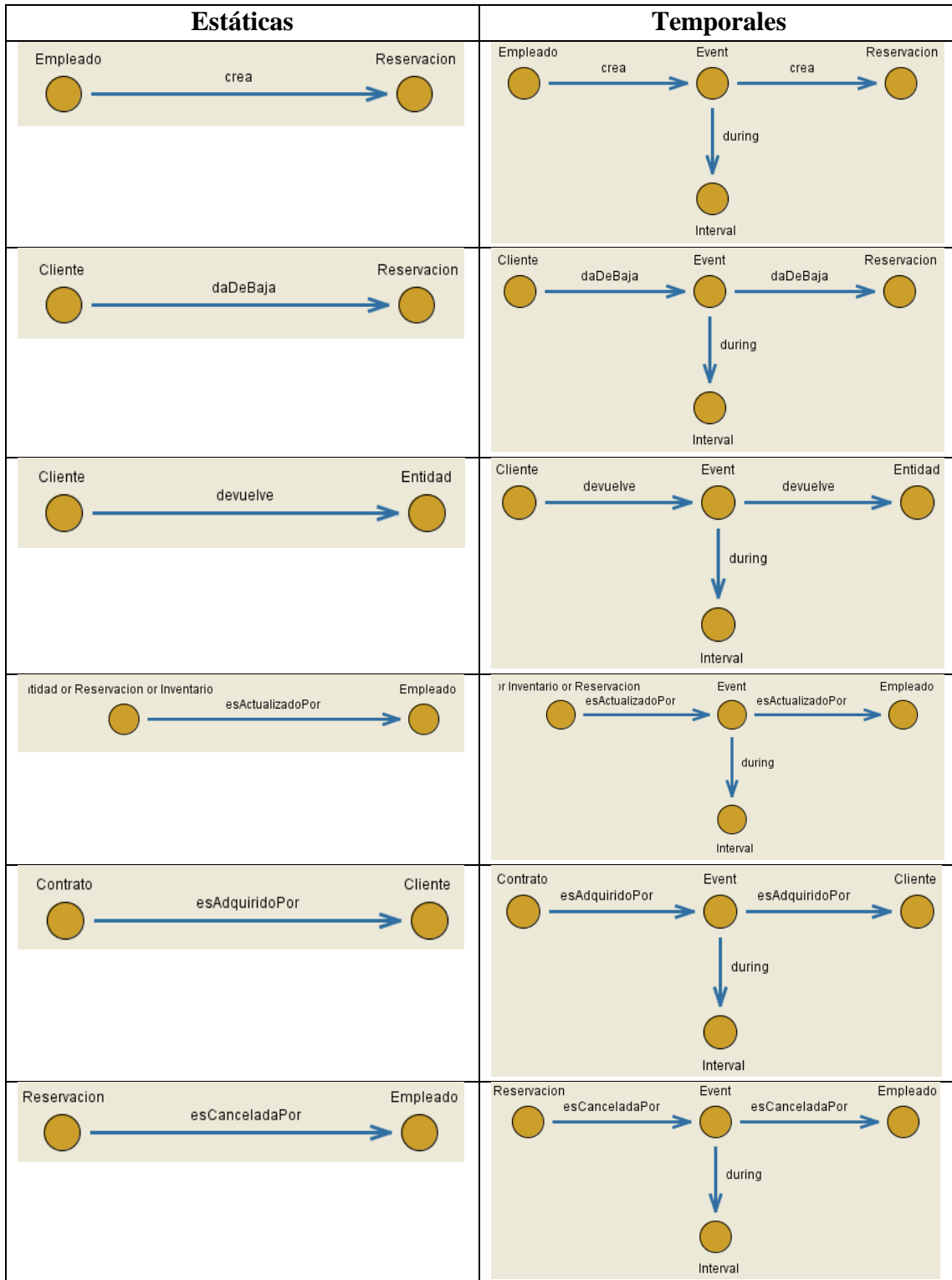
6.1.4.2. Convertir roles abstractos a temporales

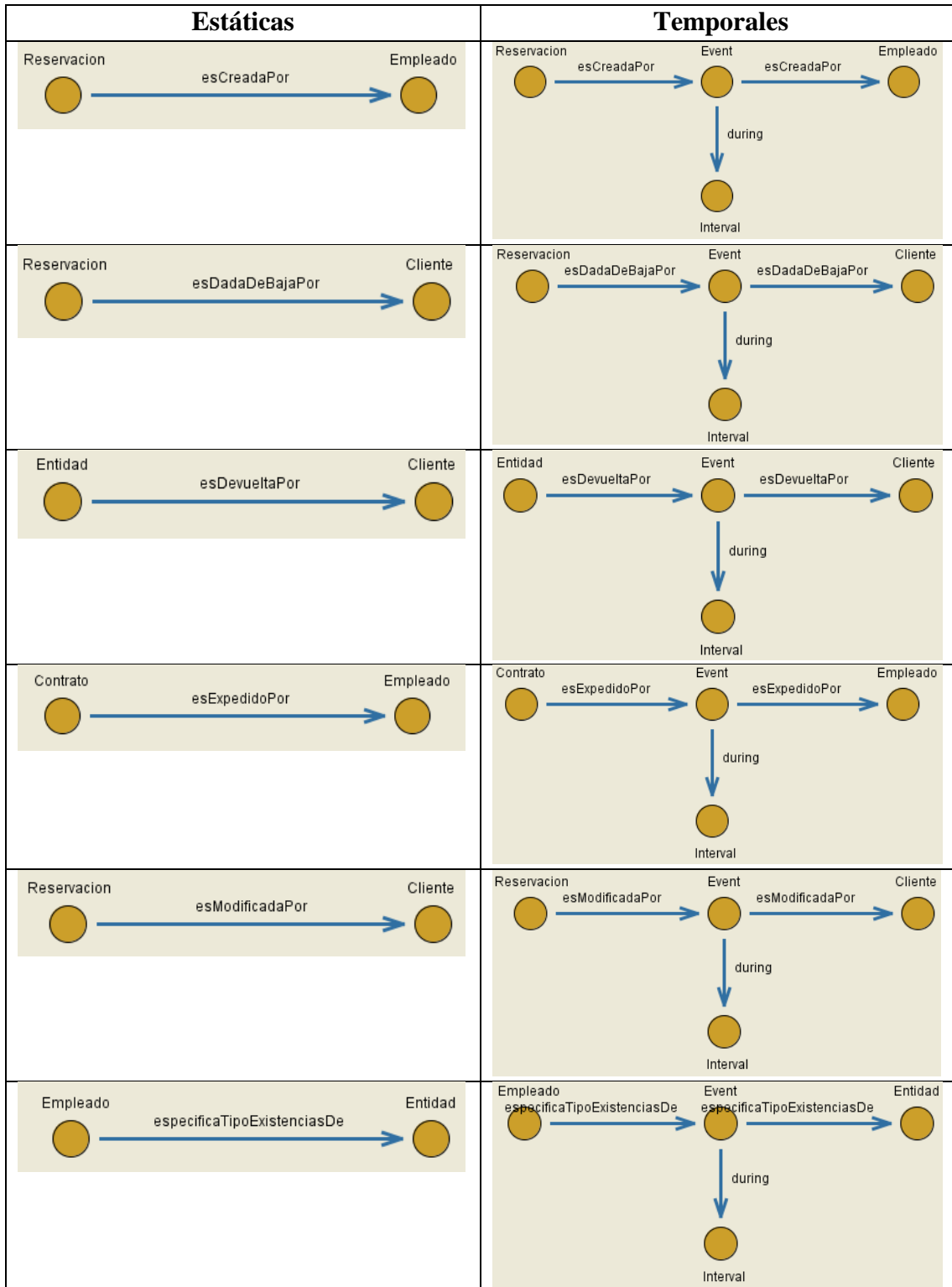
En esta actividad, se realizó la conversión de las propiedades de objetos estáticas a propiedades de objetos temporales (Tabla 63), para ello, se identificaron las propiedades de objetos que tienen un comportamiento temporal y con ayuda de la interfaz de Chronos en Protégé se realizó la conversión.

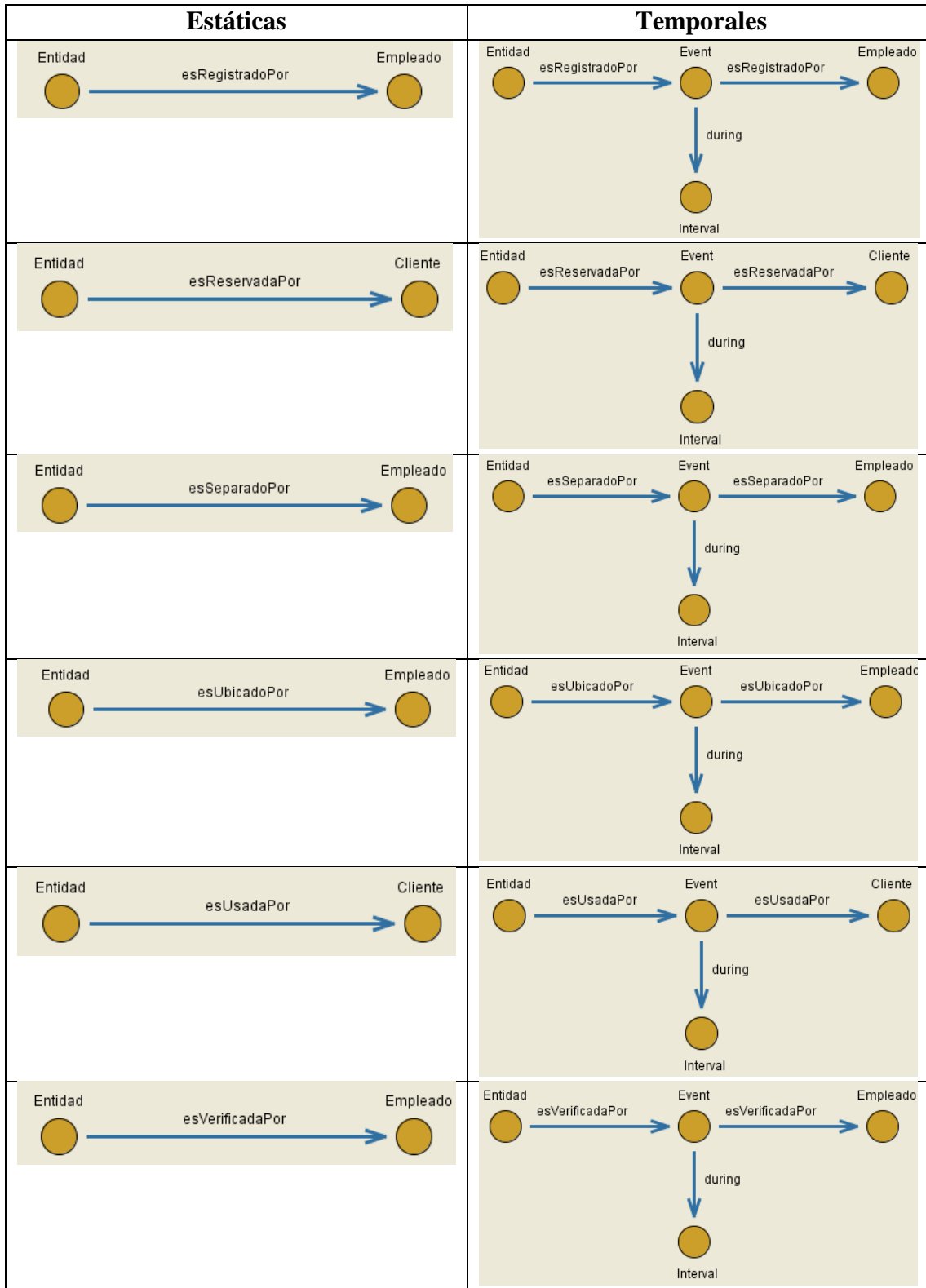
Las relaciones temporales se diferencian de las relaciones binarias por estar incluidas como subpropiedades de la propiedad de objeto temporal *participatesIn* y por tener un objeto intermediario de la clase *Event* para establecer la relación. A través del evento, se establece la relación de duración (*during*) así como el momento en que ésta tendrá lugar (clase *TimeInterval*). A continuación, en la Tabla 63 se presentan las propiedades de objetos estáticas que fueron convertidas a propiedades de objetos temporales.

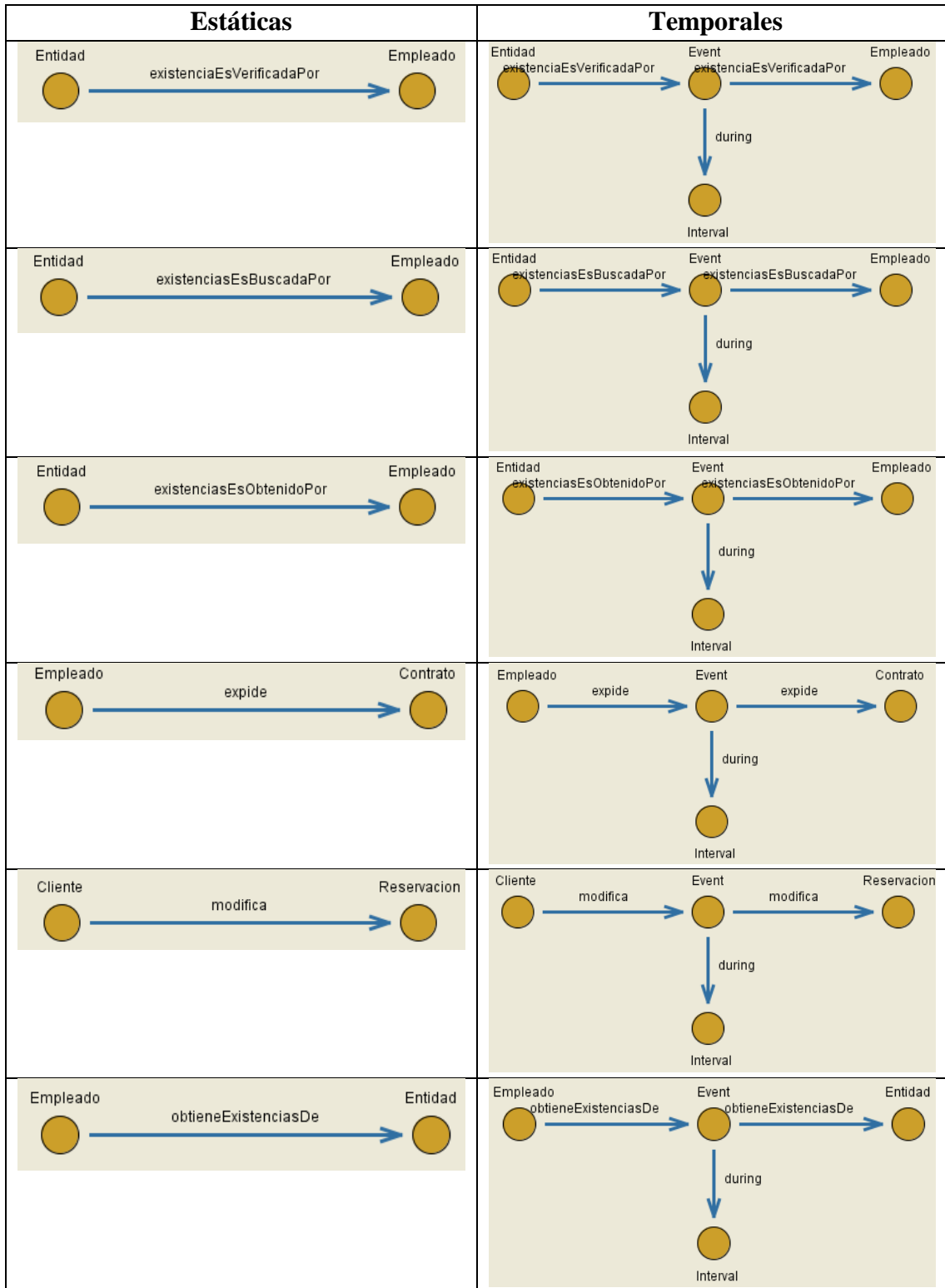
Tabla 63. Conversión de propiedades estáticas a temporales en la OTPAE del PAIyER.

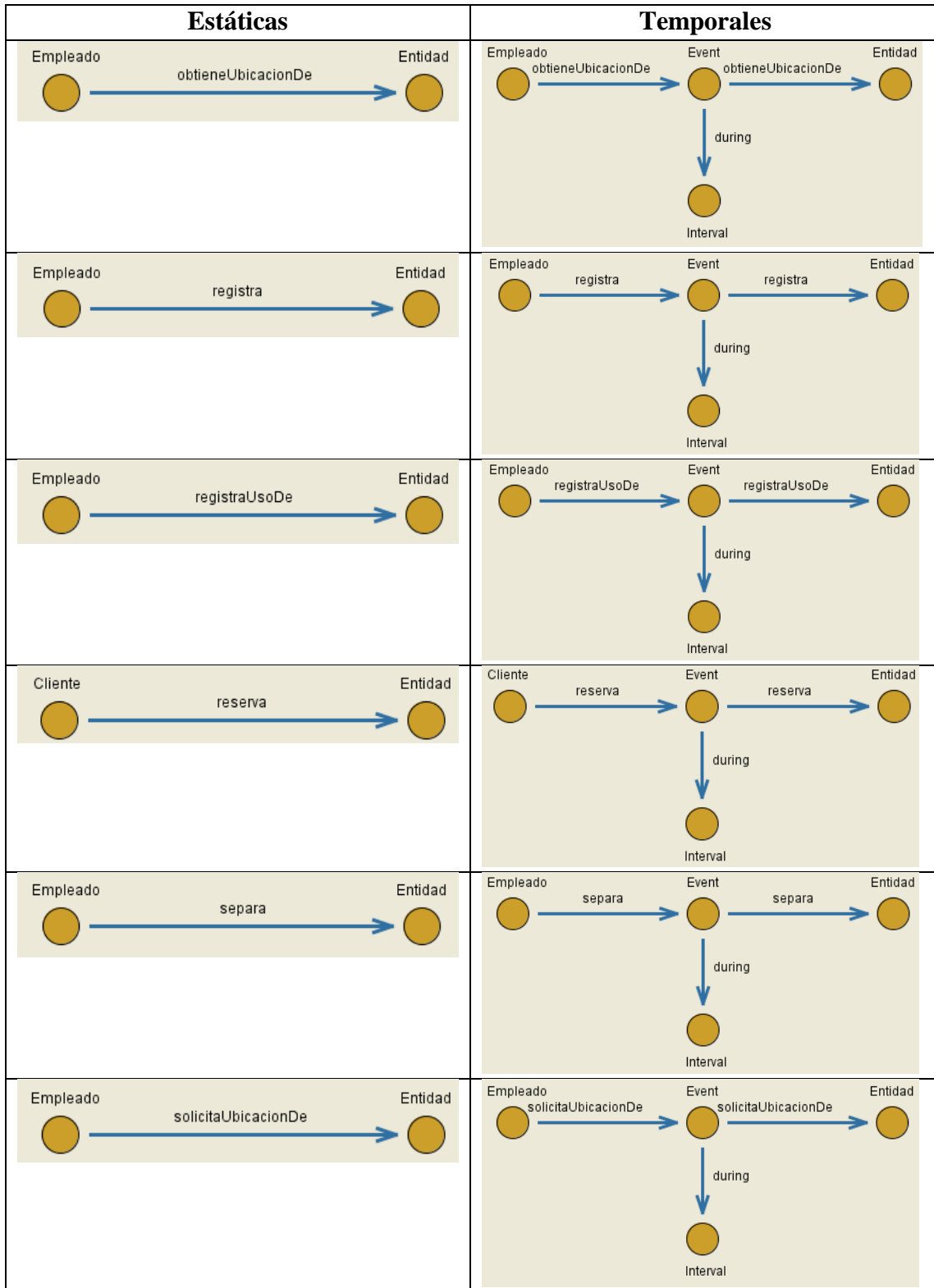
Estáticas	Temporales
<p>Empleado → actualiza → Entidad or Reservacion or Inven</p>	<p>Empleado → actualiza → Event → actualiza → Entidad or Reservacion or Inven Event ↓ during ↓ Interval</p>
<p>Cliente → adquiere → Contrato</p>	<p>Cliente → adquiere → Event → adquiere → Contrato Event ↓ during ↓ Interval</p>
<p>Empleado → buscaExistenciasDe → Entidad</p>	<p>Empleado → buscaExistenciasDe → Event → buscaExistenciasDe → Entidad Event ↓ during ↓ Interval</p>
<p>Empleado → cancela → Reservacion</p>	<p>Empleado → cancela → Event → cancela → Reservacion Event ↓ during ↓ Interval</p>

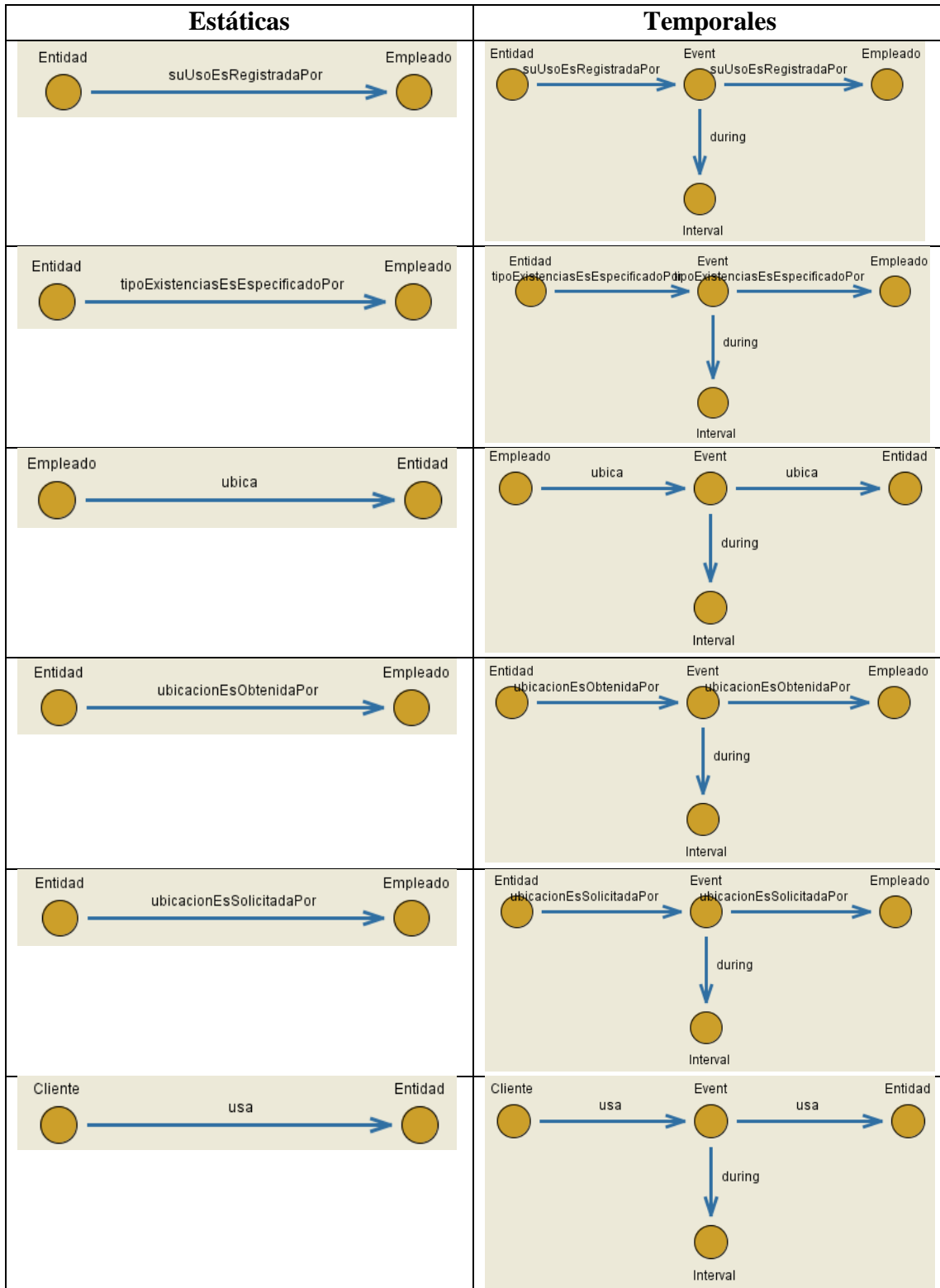


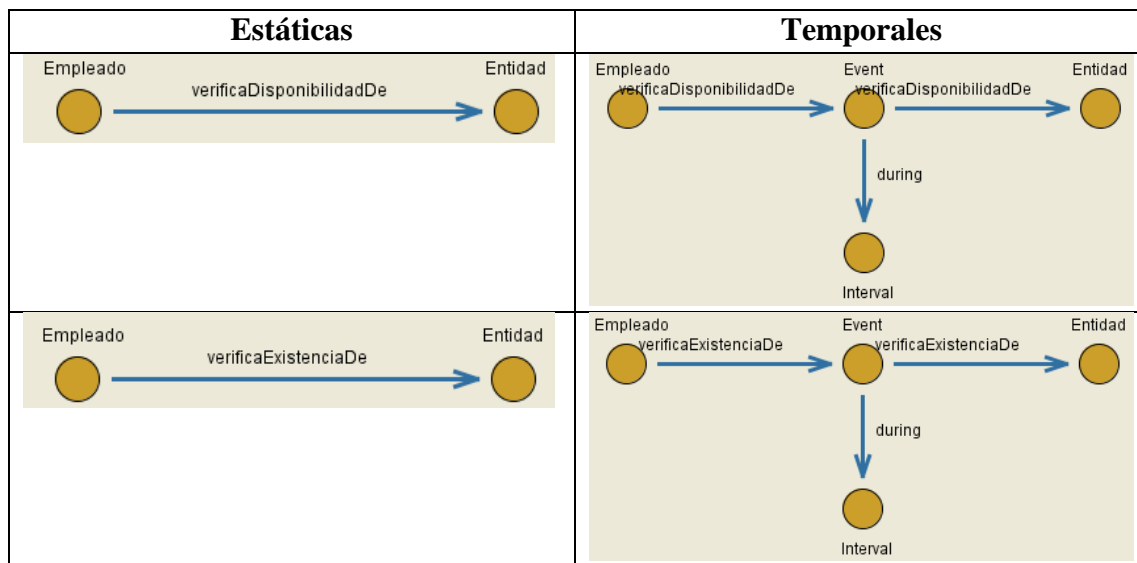












En la primera columna de la segunda fila de la Tabla 63, la propiedad de objeto *adquiere*, representa una relación binaria, en donde el dominio es el *cliente* y el rango es el *contrato*. Por otra parte, cuando la relación binaria se convierte a temporal (segunda columna, segunda fila de la Tabla 63), la relación indica que el *cliente* adquirirá un *contrato* a través de un *evento*.

Las relaciones binarias que contaban con características tales como Funcional o Inversa funcional, al momento de convertirlas a temporales con ayuda de Protegé se generaron las reglas SWRL (Lenguaje de Reglas para la Web Semántica del inglés *Semantic Web Rule Language*) correspondientes que permitirán a un razonador (como Pellet) inferir nuevo conocimiento, las reglas generadas se describen en la Tabla 64.

Tabla 64. Reglas SWRL en características funcional e inversa funcional de propiedades de objetos en la OTPAE del PAIyER.

Propiedad de objeto	Característica	Regla SWRL
actualiza	Funcional	$Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), actualiza(?e1, ?y1), actualiza(?e2, ?y2), actualiza(?x, ?e1), actualiza(?x, ?e2) \rightarrow SameAs(?y1, ?y2)$
buscaExistenciasDe	Inversa funcional	$Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), buscaExistenciasDe(?e1, ?y), buscaExistenciasDe(?e2, ?y), buscaExistenciasDe(?x1, ?e1), buscaExistenciasDe(?x2, ?e2) \rightarrow SameAs(?x1, ?x2)$
seActualizaPor	Inversa funcional	$Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), seActualizaPor(?e1, ?y), seActualizaPor(?e2, ?y), seActualizaPor(?x1, ?e1), seActualizaPor(?x2, ?e2) \rightarrow SameAs(?x1, ?x2)$
especificaTipoExistenciasDe	Inversa funcional	$Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), especificaTipoExistenciasDe(?e1, ?y),$

Propiedad de objeto	Característica	Regla SWRL
		<i>especificaTipoExistenciasDe(?e2, ?y), especificaTipoExistenciasDe(?x1, ?e1), especificaTipoExistenciasDe(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
seRegistraPor	Funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), seRegistraPor(?e1, ?y1), seRegistraPor(?e2, ?y2), seRegistraPor(?x, ?e1), seRegistraPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
seSeparaPor	Funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), seSeparaPor(?e1, ?y1), seSeparaPor(?e2, ?y2), seSeparaPor(?x, ?e1), seSeparaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
seUbicaPor	Funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), seUbicaPor(?e1, ?y1), seUbicaPor(?e2, ?y2), seUbicaPor(?x, ?e1), seUbicaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
existenciaEsVerificadoPor	Funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), existenciaSeVerificaPor(?e1, ?y1), existenciaSeVerificaPor(?e2, ?y2), existenciaSeVerificaPor(?x, ?e1), existenciaSeVerificaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
existenciasSeBuscaPor	Funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), existenciasSeBuscaPor(?e1, ?y1), existenciasSeBuscaPor(?e2, ?y2), existenciasSeBuscaPor(?x, ?e1), existenciasSeBuscaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
existenciasSeObtienePor	Funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), existenciasSeObtienePor(?e1, ?y1), existenciasSeObtienePor(?e2, ?y2), existenciasSeObtienePor(?x, ?e1), existenciasSeObtienePor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
laboraEn	Funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), laboraEn(?e1, ?y1), laboraEn(?e2, ?y2), laboraEn(?x, ?e1), laboraEn(?x, ?e2) -> SameAs (?y1, ?y2)</i>
obtieneExistenciasDe	Inversa funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), obtieneExistenciasDe(?e1, ?y), obtieneExistenciasDe(?e2, ?y), obtieneExistenciasDe(?x1, ?e1), obtieneExistenciasDe(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
obtieneUbicacionDe	Inversa funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), obtieneUbicacionDe(?e1, ?y), obtieneUbicacionDe(?e2, ?y), obtieneUbicacionDe(?x1, ?e1), obtieneUbicacionDe(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
registra	Funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), registra(?e1, ?y), registra(?e2, ?y), registra(?x1, ?e1), registra(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
separa	Funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), separa(?e1, ?y), separa(?e2, ?y),</i>

Propiedad de objeto	Característica	Regla SWRL
		<i>separa(?x1, ?e1), separa(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
solicitaUbicacionDe	Inversa funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), solicitaUbicacionDe(?e1, ?y1), solicitaUbicacionDe(?e2, ?y2), solicitaUbicacionDe(?x, ?e1), solicitaUbicacionDe(?x, ?e2) -> SameAs (?y1, ?y2)</i>
tipoExistenciasSeEspecificaPor	Funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), tipoExistenciasSeEspecificaPor(?e1, ?y1), tipoExistenciasSeEspecificaPor(?e2, ?y2), tipoExistenciasSeEspecificaPor(?x, ?e1), tipoExistenciasSeEspecificaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
ubica	Inversa funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), ubica(?e1, ?y), ubica(?e2, ?y), ubica(?x1, ?e1), ubica(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
ubicacionSeObtienePor	Funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), ubicacionSeObtienePor(?e1, ?y1), ubicacionSeObtienePor(?e2, ?y2), ubicacionSeObtienePor(?x, ?e1), ubicacionSeObtienePor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
ubicacionSeSolicitaPor	Funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), ubicacionSeSolicitaPor(?e1, ?y), ubicacionSeSolicitaPor(?e2, ?y), ubicacionSeSolicitaPor(?x1, ?e1), ubicacionSeSolicitaPor(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
verificaExistenciaDe	Inversa funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), verificaExistenciaDe(?e1, ?y), verificaExistenciaDe(?e2, ?y), verificaExistenciaDe(?x1, ?e1), verificaExistenciaDe(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
esCanceladaPor	Funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), esCanceladaPor(?e1, ?y1), esCanceladaPor(?e2, ?y2), esCanceladaPor(?x, ?e1), esCanceladaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
verificaDisponibilidadDe	Inversa funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), verificaDisponibilidadDe(?e1, ?y), verificaDisponibilidadDe(?e2, ?y), verificaDisponibilidadDe(?x1, ?e1), verificaDisponibilidadDe(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
esAdquiridoPor	funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), esAdquiridoPor(?e1, ?y1), esAdquiridoPor(?e2, ?y2), esAdquiridoPor(?x, ?e1), esAdquiridoPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
esVerificadaPor	funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), esVerificadaPor(?e1, ?y1), esVerificadaPor(?e2, ?y2), esVerificadaPor(?x, ?e1), esVerificadaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
esDevueltaPor	Funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), esDevueltaPor(?e1, ?y1),</i>

Propiedad de objeto	Característica	Regla SWRL
		<i>esDevueltaPor(?e2, ?y2), esDevueltaPor(?x, ?e1), esDevueltaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
devuelve	Inversa funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), devuelve(?e1, ?y), devuelve(?e2, ?y), devuelve(?x1, ?e1), devuelve(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
usa	Inversa funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), usa(?e1, ?y), usa(?e2, ?y), usa(?x1, ?e1), usa(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
daDeBaja	Inversa funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), daDeBaja(?e1, ?y), daDeBaja(?e2, ?y), daDeBaja(?x1, ?e1), daDeBaja(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
crea	Inversa funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), crea(?e1, ?y), crea(?e2, ?y), crea(?x1, ?e1), crea(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
suUsoEsRegistradaPor	Funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), suUsoEsRegistradaPor(?e1, ?y1), suUsoEsRegistradaPor(?e2, ?y2), suUsoEsRegistradaPor(?x, ?e1), suUsoEsRegistradaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
reserva	Inversa funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), reserva(?e1, ?y), reserva(?e2, ?y), reserva(?x1, ?e1), reserva(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
esCreadaPor	Funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), esCreadaPor(?e1, ?y1), esCreadaPor(?e2, ?y2), esCreadaPor(?x, ?e1), esCreadaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
esDadaDeBajaPor	Funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), esDadaDeBajaPor(?e1, ?y1), esDadaDeBajaPor(?e2, ?y2), esDadaDeBajaPor(?x, ?e1), esDadaDeBajaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
registraUsoDe	Inversa funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), registraUsoDe(?e1, ?y), registraUsoDe(?e2, ?y), registraUsoDe(?x1, ?e1), registraUsoDe(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
esExpedidoPor	Funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), esExpedidoPor(?e1, ?y1), esExpedidoPor(?e2, ?y2), esExpedidoPor(?x, ?e1), esExpedidoPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
expide	Inversa funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), expide(?e1, ?y), expide(?e2, ?y), expide(?x1, ?e1), expide(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
cancela	Inversa funcional	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), cancela(?e1, ?y), cancela(?e2, ?y), cancela(?x1, ?e1), cancela(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
modifica	Inversa	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2),</i>

Propiedad de objeto	Característica	Regla SWRL
	funcional	$overlaps(?i1, ?i2), modifica(?e1, ?y), modifica(?e2, ?y), modifica(?x1, ?e1), modifica(?x2, ?e2) \rightarrow SameAs(?x1, ?x2)$
esReservadaPor	Funcional	$Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), esReservadaPor(?e1, ?y1), esReservadaPor(?e2, ?y2), esReservadaPor(?x, ?e1), esReservadaPor(?x, ?e2) \rightarrow SameAs(?y1, ?y2)$
esModificadaPor	Funcional	$Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), esModificadaPor(?e1, ?y1), esModificadaPor(?e2, ?y2), esModificadaPor(?x, ?e1), esModificadaPor(?x, ?e2) \rightarrow SameAs(?y1, ?y2)$
esUsadaPor	Funcional	$Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), esUsadaPor(?e1, ?y1), esUsadaPor(?e2, ?y2), esUsadaPor(?x, ?e1), esUsadaPor(?x, ?e2) \rightarrow SameAs(?y1, ?y2)$
adquiere	Inversa funcional	$Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), adquiere(?e1, ?y), adquiere(?e2, ?y), adquiere(?x1, ?e1), adquiere(?x2, ?e2) \rightarrow SameAs(?x1, ?x2)$

P. ej., la regla SWRL para el rol abstracto *actualiza*, establece como antecedente los individuos del concepto Event: $Event(?e1)$ y $Event(?e2)$, mediante los cuales se lleva a cabo el rol abstracto temporal *actualiza*: $actualiza(?e1, ?y1)$, $actualiza(?e2, ?y2)$, $actualiza(?x, ?e1)$, $actualiza(?x, ?e2)$. También establece que el rol abstracto temporal solo puede relacionar a un individuo ($?x$) con otro individuo $?y1$ o $?y2$ al mismo tiempo $overlaps(?i1, ?i2)$, donde $i1$ e $i2$ son los intervalos de duración de los eventos: $Event(?e1)$ y $Event(?e2)$, es decir, $during(?e1, ?i1)$ y $during(?e2, ?i2)$, lo cual implica en el consecuente, que $?y1$ y $?y2$ son individuos similares $SameAs(?y1, ?y2)$.

6.1.4.3. Verificar la consistencia semántica de la OTPAE compuesta

En esta actividad, se realizó la verificación de la consistencia semántica temporal (TSCC) de la OTPAE de forma automatizada utilizando el razonador Pellet (Figura 40). Los resultados obtenidos por el razonador Pellet demostraron que la ontología temporal compuesta es consistente por el momento. Se dice que es consistente por el momento debido a que hasta este punto del método COMOTPA aún no se han integrado las secuencias de interpretaciones que representen el comportamiento dinámico, estas secuencias de interpretaciones se agregaron más adelante.

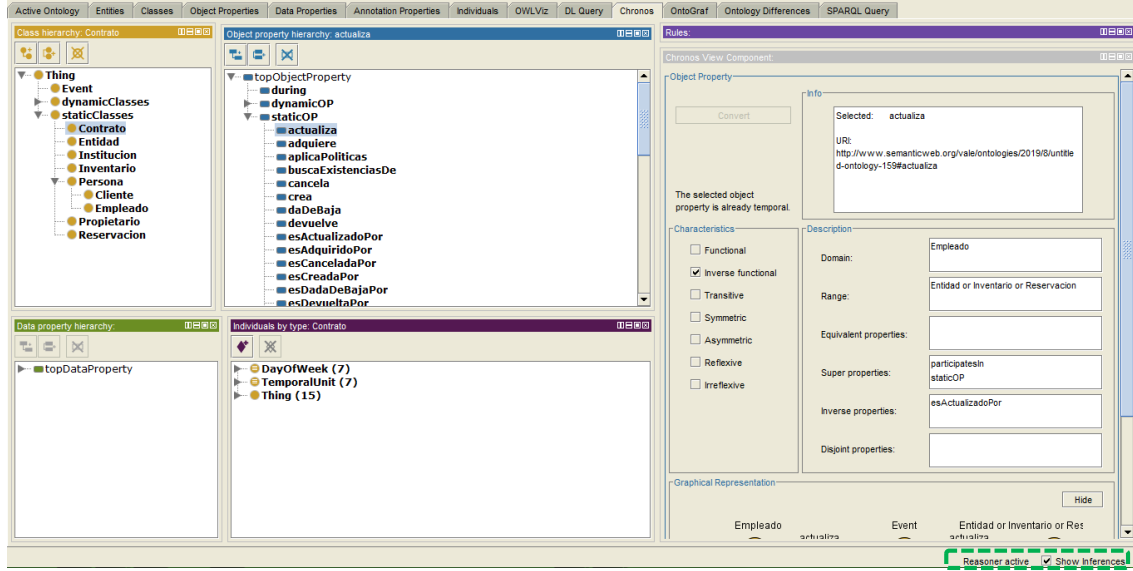


Figura 40. Verificación de la consistencia de la OTPAE del PAIyER utilizando el razonador Pellet.

6.1.5. Combinación de fórmulas ALC_{LTL}

En esta fase, se realizó la combinación de las fórmulas ALC_{LTL} para especificar el comportamiento dinámico de la OTPAE del patrón de análisis PAIyER. Como entrada en esta actividad, se utilizó el conjunto de fórmulas ALC_{LTL} de la ontología temporal del PAI y el conjunto de fórmulas ALC_{LTL} de la ontología temporal del PAER. A continuación, en la Tabla 65 se presenta el conjunto de fórmulas ALC_{LTL} de la OTPAE del PAER y del PAI.

Tabla 65. Elementos de entrada para realizar la combinación de fórmulas ALC_{LTL} .

Fórmulas $ALC_{LTL}(\varphi)$ PAER	Fórmulas $ALC_{LTL}(\varphi)$ PAI
RF1. Hacer una reservación $\mathcal{G}(\mathcal{X}(\text{Cliente} \sqsubseteq \exists \text{reserva. Entidad}) \rightarrow (\mathcal{XX}(\text{Empleado} \sqsubseteq \exists \text{verificaDisponibilidadDe. Entidad}) \wedge \mathcal{F}(\neg \mathcal{XXX}(\text{Empleado} \sqsubseteq \forall \text{crea. Reservacion}) \wedge \neg \mathcal{XXXX}(\text{Empleado} \sqsubseteq \forall \text{expide. Contrato})))$	RF1. Separar diferentes tipos de existencias $\mathcal{G}(\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{separa. Entidad}) \rightarrow (\mathcal{XX}(\text{Empleado} \sqsubseteq \exists \text{verificaExistenciaDe. Entidad}) \wedge \mathcal{F}(\neg \mathcal{XXX}(\text{Empleado} \sqsubseteq \exists \text{actualiza. Entidad} \vee (\text{Empleado} \sqsubseteq \exists \text{registra. Entidad}))) \wedge \neg \mathcal{XXXX}(\text{Empleado} \sqsubseteq \exists \text{actualiza. Inventario})))$
RF2. Usar una entidad reservada $\mathcal{G}(\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{registraUsoDe. Entidad}) \rightarrow \mathcal{F}(\neg \mathcal{XX}(\text{Cliente} \sqsubseteq \exists \text{usa. Entidad}) \wedge \neg \mathcal{XXX}(\text{Cliente} \sqsubseteq \exists \text{devuelve. Entidad})))$	RF2. Realizar seguimiento de las cantidades de entidades en existencias $\mathcal{G}(\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{buscaExistenciasDe. Entidad}) \rightarrow (\mathcal{XX}(\text{Empleado} \sqsubseteq \exists \text{verificaExistenciaDe. Entidad}) \wedge \mathcal{F}(\neg \mathcal{XXX}(\text{Empleado} \sqsubseteq \exists \text{especificaTipoExistencias. Entidad})))$

)	$\neg \mathcal{X}\mathcal{X}\mathcal{X}\mathcal{X}$ (Empleado \sqsubseteq \exists obtieneExistenciasDe.Entidad)
RF3. Modificar una reservación	RF3. Realizar seguimiento de las ubicaciones de las entidades
$\mathcal{G}(\mathcal{X}(\text{Cliente} \sqsubseteq \exists \text{ modifica.Reservacion}) \rightarrow (\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{ verificaDisponibilidadDe.Entidad}) \wedge \mathcal{F}(\neg \mathcal{X}\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{ cancela.Reservacion} \wedge \text{Empleado} \sqsubseteq \forall \text{ actualiza.Reservacion})))$	$\mathcal{G}(\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{ ubica.Entidad}) \rightarrow (\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{ verificaExistenciaDe.Entidad}) \wedge \mathcal{F}(\neg \mathcal{X}\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{ solicitaUbicacionDe.Entidad}) \wedge \neg \mathcal{X}\mathcal{X}\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{ obtieneUbicacionDe.Entidad})))$
RF4. Cancelar una reservación	
$\mathcal{G}(\mathcal{X}(\text{Cliente} \sqsubseteq \exists \text{ daDeBaja.Reservacion}) \rightarrow (\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{ aplicaPolíticas.(Reservacion} \sqcup \text{Cliente})) \wedge \mathcal{F}(\neg \mathcal{X}\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \text{cancela.Reservacion})))$	
RF5. Crear un contrato	
$\mathcal{G}(\mathcal{X}(\text{Cliente} \sqsubseteq \exists \text{ reserva.Entidad}) \rightarrow \mathcal{F}(\neg \mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{ crea.Reservacion}) \wedge \neg \mathcal{X}\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{ expide.Contrato}) \wedge \neg \mathcal{X}\mathcal{X}\mathcal{X}\mathcal{X}(\text{Contrato} \sqsubseteq \exists \text{ esAdquiridoPor.Cliente})))$	

Para combinar las fórmulas, primero se actualizaron los nombres de los conceptos y roles que participan en las fórmulas ALC_{LTL} debido a que durante el proceso de combinación de términos del dominio se realizaron cambios en los nombres de algunos conceptos. Para este caso de estudio, solamente se actualizó el concepto “Artículo” (que es utilizado en las fórmulas ALC_{LTL} de la OTPAE del PAI) por el concepto “Entidad”.

Posteriormente, se analizó cada concepto de inclusión que participa en cada fórmula ALC_{LTL} y se agregaron en algunas fórmulas, nuevos conceptos de inclusión que permiten ampliar la representación del requerimiento funcional correspondiente. Las fórmulas modificadas fueron aquellas en donde los conceptos de inclusión afectan en el comportamiento de la otra ontología. Por ejemplo, en el RF1. Hacer una reservación (originalmente perteneciente a la OTPAE del PAER), se le agregó el concepto de inclusión $\text{Empleado} \sqsubseteq \exists \text{ actualiza.Entidad}$ (originalmente correspondiente a la OTPAE del PAI) debido a que cuando se crea una reservación, la cantidad de la entidad reservada tiene que ser actualizada en el inventario y esta actualización era originalmente realizada en la ontología

temporal del PAI. Finalmente, se obtuvieron las nuevas fórmulas ALC_{LTL} , que permiten representar el conocimiento de la nueva ontología temporal compuesta, el conjunto de fórmulas ALC_{LTL} se presenta en la Tabla 66.

Tabla 66. Conjunto de fórmulas ALC_{LTL} de las especificaciones ECAP del patrón de análisis PAIyER.

Fórmula $ALC_{LTL} (\varphi)$
RF1. Hacer una reservación
$\mathcal{G}(\mathcal{X}(\text{Cliente} \sqsubseteq \exists \text{reserva. Entidad}) \rightarrow (\mathcal{XX}(\text{Empleado} \sqsubseteq \exists \text{verificaDisponibilidadDe. Entidad}) \wedge \mathcal{F}(\neg \mathcal{XXX}(\text{Empleado} \sqsubseteq \forall \text{crea. Reservacion} \wedge \text{Empleado} \sqsubseteq \exists \text{actualiza. Entidad}) \wedge \neg \mathcal{XXX}(\text{Empleado} \sqsubseteq \forall \text{expide. Contrato})))$
RF2. Usar una entidad reservada
$\mathcal{G}(\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{registraUsoDe. Entidad}) \rightarrow \mathcal{F}(\neg \mathcal{XX}(\text{Cliente} \sqsubseteq \exists \text{usa. Entidad}) \wedge \neg \mathcal{XXX}(\text{Cliente} \sqsubseteq \exists \text{devuelve. Entidad})))$
RF3. Modificar una reservación
$\mathcal{G}(\mathcal{X}(\text{Cliente} \sqsubseteq \exists \text{ modifica. Reservacion}) \rightarrow (\mathcal{XX}(\text{Empleado} \sqsubseteq \exists \text{verificaDisponibilidadDe. Entidad}) \wedge \mathcal{F}(\neg \mathcal{XXX}(\text{Empleado} \sqsubseteq \exists \text{cancela. Reservacion} \wedge \text{Empleado} \sqsubseteq \forall \text{actualiza. Reservacion}) \wedge \neg \mathcal{XXX}(\text{Empleado} \sqsubseteq \exists \text{actualiza. Entidad})))$
RF4. Cancelar una reservación
$\mathcal{G}(\mathcal{X}(\text{Cliente} \sqsubseteq \exists \text{daDeBaja. Reservacion}) \rightarrow (\mathcal{XX}(\text{Empleado} \sqsubseteq \exists \text{aplicaPolíticas. (Reservacion} \sqcup \text{Cliente)}) \wedge \mathcal{F}(\neg \mathcal{XXX}(\text{Empleado} \sqsubseteq \text{cancela. Reservacion} \wedge \text{Empleado} \sqsubseteq \exists \text{actualiza. Entidad})))$
RF5. Crear un contrato
$\mathcal{G}(\mathcal{X}(\text{Cliente} \sqsubseteq \exists \text{reserva. Entidad}) \rightarrow \mathcal{F}(\neg \mathcal{XX}(\text{Empleado} \sqsubseteq \exists \text{crea. Reservacion}) \wedge \neg \mathcal{XXX}(\text{Empleado} \sqsubseteq \exists \text{expide. Contrato}) \wedge \neg \mathcal{XXX}(\text{Contrato} \sqsubseteq \exists \text{esAdquiridoPor. Cliente})))$
RF6. Separar diferentes tipos de existencias
$\mathcal{G}(\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{separa. Entidad}) \rightarrow (\mathcal{F}(\neg \mathcal{XX}(\text{Empleado} \sqsubseteq \exists \text{separa. Entidad}))))$

$\begin{aligned} & \mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{verificaExistenciaDe. Entidad}) \wedge \mathcal{F}(\\ & \neg \mathcal{X}\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{actualiza. Entidad} \vee (\text{Empleado} \sqsubseteq \exists \text{registra. Entidad})) \wedge \\ & \neg \mathcal{X}\mathcal{X}\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{actualiza. Inventario}) \\ &) \\ &) \\ &) \end{aligned}$
<p>RF7. Realizar seguimiento de las cantidades de entidades en existencias</p>
$\begin{aligned} & \mathcal{G}(\\ & \mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{buscaExistenciasDe. Entidad}) \rightarrow (\\ & \mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{verificaExistenciaDe. Entidad}) \wedge \mathcal{F}(\\ & \neg \mathcal{X}\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{especificaTipoExistencias. Entidad}) \wedge \\ & \neg \mathcal{X}\mathcal{X}\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{obtieneExistenciasDe. Entidad}) \\ &) \\ &) \\ &) \end{aligned}$
<p>RF8. Realizar seguimiento de las ubicaciones de las entidades</p>
$\begin{aligned} & \mathcal{G}(\\ & \mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{ubica. Entidad}) \rightarrow (\\ & \mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{verificaExistenciaDe. Entidad}) \wedge \mathcal{F}(\\ & \neg \mathcal{X}\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{solicitaUbicacionDe. Entidad}) \wedge \\ & \neg \mathcal{X}\mathcal{X}\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{obtieneUbicacionDe. Entidad}) \\ &) \\ &) \\ &) \end{aligned}$

P. ej., la especificación del RF Hacer una reservación, especifica que, “Siempre (\mathcal{G}) ocurre que, en un primer momento (\mathcal{X}), el cliente solicita reservar una entidad ($\text{Cliente} \sqsubseteq \exists \text{reserva. Entidad}$), lo cual implica que posteriormente ($\mathcal{X}\mathcal{X}$) un empleado verifique la disponibilidad de la entidad ($\text{Empleado} \sqsubseteq \exists \text{verificaDisponibilidadDe. Entidad}$). Eventualmente (\mathcal{F}) en otro momento ($\mathcal{X}\mathcal{X}\mathcal{X}$) el empleado (**no** \neg) crea la reservación ($\text{Empleado} \sqsubseteq \forall \text{crea. Reservacion}$) y (**no** \neg) actualiza la entidad ($\text{Empleado} \sqsubseteq \exists \text{actualiza. Entidad}$), posteriormente ($\mathcal{X}\mathcal{X}\mathcal{X}\mathcal{X}$) (**no** \neg) expide un contrato al cliente ($\text{Empleado} \sqsubseteq \forall \text{expide. Contrato}$)”.

Las fórmulas ALC_{LTL} de las especificaciones ECAP (Tabla 66), fueron reducidas a fórmulas LTL (φ') con la finalidad utilizarlas posteriormente en la fase 6.1.7. *Verificación de la formalización* para verificar la satisfacibilidad de la especificación en la ruta de computación. La reducción consistió en reemplazar los CIs básicos de las fórmulas ALC_{LTL} por proposiciones atómicas de la lógica temporal LTL. A continuación, en la Tabla 67 se presentan las fórmulas LTL, resultado de la reducción de las especificaciones formales (fórmulas ALC_{LTL}) de la OTPAE del PAIyER.

Tabla 67. Reducción de las especificaciones formales de la OTPAE compuesta a fórmulas de la LTL.

Fórmula $ALC_{LTL}(\varphi)$	Proposiciones
RF1. Hacer una reservación	
$\mathcal{G}(\mathcal{X}(p) \rightarrow (\mathcal{X}\mathcal{X}(q) \wedge \mathcal{F}(\neg \mathcal{X}\mathcal{X}\mathcal{X}(r \wedge s) \wedge \neg \mathcal{X}\mathcal{X}\mathcal{X}\mathcal{X}(t))))$	$p = Cliente \sqsubseteq \exists reserva.Entidad$ $q = Empleado \sqsubseteq \exists verificaDisponibilidadDe.Entidad$ $r = Empleado \sqsubseteq \forall crea.Reservacion$ $s = Empleado \sqsubseteq \forall actualiza.Entidad$ $t = Empleado \sqsubseteq \forall expide.Contrato$
RF2. Usar una entidad reservada	
$\mathcal{G}(\mathcal{X}(p) \rightarrow \mathcal{F}(\neg \mathcal{X}\mathcal{X}(q) \wedge \neg \mathcal{X}\mathcal{X}\mathcal{X}(r))))$	$p = Empleado \sqsubseteq \exists registraUsoDe.Entidad$ $q = Cliente \sqsubseteq \exists usa.Entidad$ $r = Cliente \sqsubseteq \exists devuelve.Entidad$
RF3. Modificar una reservación	
$\mathcal{G}(\mathcal{X}(p) \rightarrow (\mathcal{X}\mathcal{X}(q) \wedge \mathcal{F}(\neg \mathcal{X}\mathcal{X}\mathcal{X}(r \wedge s) \wedge \neg \mathcal{X}\mathcal{X}\mathcal{X}\mathcal{X}(t))))$	$p = Cliente \sqsubseteq \exists modifica.Reservacion$ $q = Empleado \sqsubseteq \exists verificaDisponibilidadDe.Entidad$ $r = Empleado \sqsubseteq \exists cancela.Reservacion$ $s = Empleado \sqsubseteq \forall actualiza.Reservacion$ $t = Empleado \sqsubseteq \forall actualiza.Entidad$
RF4. Cancelar una reservación	
$\mathcal{G}(\mathcal{X}(p) \rightarrow (\mathcal{X}\mathcal{X}(q) \wedge \mathcal{F}(\neg \mathcal{X}\mathcal{X}\mathcal{X}(r \wedge s))))$	$p = Cliente \sqsubseteq \exists daDeBaja.Reservacion$ $q = Empleado \sqsubseteq \exists aplicaPolitic.(Reservacion \sqcup Cliente)$ $r = Empleado \sqsubseteq \exists cancela.Reservacion$ $s = Empleado \sqsubseteq \exists actualiza.Entidad$
RF5. Crear un contrato	
$\mathcal{G}(\mathcal{X}(p) \rightarrow \mathcal{F}(\neg \mathcal{X}\mathcal{X}(q) \wedge \neg \mathcal{X}\mathcal{X}\mathcal{X}(r) \wedge \neg \mathcal{X}\mathcal{X}\mathcal{X}\mathcal{X}(s))))$	$p = Cliente \sqsubseteq \exists reserva.Entidad$ $q = Empleado \sqsubseteq \exists crea.Reservacion$ $r = Empleado \sqsubseteq \exists expide.Contrato$ $s = Contrato \sqsubseteq \exists esAdquiridoPor.Cliente$
RF6. Separar diferentes tipos de existencias	
$\mathcal{G}(\mathcal{X}(p) \rightarrow (\mathcal{X}\mathcal{X}(q) \wedge \mathcal{F}(\neg \mathcal{X}\mathcal{X}\mathcal{X}(r \wedge s))))$	$p = Empleado \sqsubseteq \exists separa.Entidad$ $q = Empleado \sqsubseteq \exists verificaExistenciaDe.Entidad$

$\begin{aligned} & \mathcal{X}\mathcal{X}(q) \wedge \mathcal{F}(\\ & \neg \mathcal{X}\mathcal{X}\mathcal{X}(r \vee s) \wedge \\ & \neg \mathcal{X}\mathcal{X}\mathcal{X}\mathcal{X}(t) \\ &) \\ &) \\ &) \end{aligned}$	$\begin{aligned} r &= \text{Empleado} \sqsubseteq \exists \text{actualiza. Entidad} \\ s &= \text{Empleado} \sqsubseteq \exists \text{registra. Entidad} \\ t &= \text{Empleado} \sqsubseteq \exists \text{actualiza. Inventario} \end{aligned}$
RF7. Realizar seguimiento de las cantidades de entidades en existencias	
$\begin{aligned} & \mathcal{G}(\\ & \mathcal{X}(p) \rightarrow (\\ & \mathcal{X}\mathcal{X}(q) \wedge \mathcal{F}(\\ & \neg \mathcal{X}\mathcal{X}\mathcal{X}(r) \wedge \\ & \neg \mathcal{X}\mathcal{X}\mathcal{X}\mathcal{X}(s) \\ &) \\ &) \\ &) \end{aligned}$	$\begin{aligned} p &= \text{Empleado} \sqsubseteq \exists \text{buscaExistenciasDe. Entidad} \\ q &= \text{Empleado} \sqsubseteq \exists \text{verificaExistenciaDe. Entidad} \\ r &= \text{Empleado} \sqsubseteq \exists \text{especificaTipoExistencias. Entidad} \\ s &= \text{Empleado} \sqsubseteq \exists \text{obtieneExistenciasDe. Entidad} \end{aligned}$
RF8. Realizar seguimiento de las ubicaciones de las entidades	
$\begin{aligned} & \mathcal{G}(\\ & \mathcal{X}(p) \rightarrow (\\ & \mathcal{X}\mathcal{X}(q) \wedge \mathcal{F}(\\ & \neg \mathcal{X}\mathcal{X}\mathcal{X}(r) \wedge \\ & \neg \mathcal{X}\mathcal{X}\mathcal{X}\mathcal{X}(s) \\ &) \\ &) \\ &) \end{aligned}$	$\begin{aligned} p &= \text{Empleado} \sqsubseteq \exists \text{ubica. Entidad} \\ q &= \text{Empleado} \sqsubseteq \exists \text{verificaExistenciaDe. Entidad} \\ r &= \text{Empleado} \sqsubseteq \exists \text{solicitaUbicacionDe. Entidad} \\ s &= \text{Empleado} \sqsubseteq \exists \text{obtieneUbicacionDe. Entidad} \end{aligned}$

P. ej., para reducir la fórmula ALC_{LTL} del RF Separar diferentes tipos de existencias, los CI básicos $\text{Empleado} \sqsubseteq \exists \text{separa. Artículo}$, $\text{Empleado} \sqsubseteq \exists \text{verificaExistenciaDe. Artículo}$, $\text{Empleado} \sqsubseteq \exists \text{actualiza. Artículo}$, $\text{Empleado} \sqsubseteq \exists \text{registra. Artículo}$, y $\text{Empleado} \sqsubseteq \exists \text{actualiza. Inventario}$, se reemplazaron por las proposiciones atómicas p , q , r , s y t , respectivamente.

6.1.6. Interpretación semántica temporal

En esta fase, se propusieron secuencias de interpretaciones con base en la experiencia por cada fórmula TDL ALC_{LTL} . Además, se verificó la consistencia semántica de las secuencias de interpretaciones y finalmente, se redujeron las secuencias de interpretaciones a rutas de computación. A continuación, se presenta los resultados obtenidos en cada actividad.

6.1.6.1. Proponer una secuencia de interpretaciones con base en la experiencia

En esta actividad, se propusieron e integraron secuencias de interpretaciones a la ontología OTPAE del patrón de análisis PAIyER basadas en la en la experiencia y de acuerdo a lo que

establece cada fórmula ALC_{LTL} . Las secuencias de interpretaciones de la OTPAE compuesta se describen en la sección 1.2. *Componente extensional del Anexo D. Ontología OTPAE del patrón de análisis PAIyER.*

6.1.6.2. Verificar la consistencia de la secuencia de interpretaciones en la OTPAE compuesta

En esta actividad, se realizó una 2da verificación TSCC (Piedrola Domínguez N., 2019) por cada secuencias de interpretaciones incorporadas en la OTPAE del PAIyER. La verificación TSCC de las secuencias de interpretaciones propuestas en la actividad anterior se realizó automáticamente utilizando el razonador Pellet en Protégé y se utilizó la documentación descrita en *el Anexo E. Documentación TSCC* para documentar la verificación TSCC.

A continuación, en la Tabla 68 se presentan los resultados obtenidos en la verificación TSCC de la OTPAE del patrón de análisis PAIyER.

Tabla 68. Resultados por requerimiento funcional en la verificación TSCC de la OTPAE del PAIyER.

Requerimiento	Caso de Verificación	Resultado de la Verificación
RF1. Hacer una reservación	01 - Éxito	Consistente
	02 - Éxito y Fracaso	Consistente con conceptos insatisfacibles
RF2. Usar una entidad reservada	03 - Éxito	Consistente
	04 - Éxito y Fracaso	Consistente con conceptos insatisfacibles
RF3. Modificar una reservación	05 - Éxito	Consistente
	06 - Éxito y Fracaso	Consistente con conceptos insatisfacibles
RF4. Cancelar una reservación	07 - Éxito	Consistente
	08 - Éxito y Fracaso	Consistente con conceptos insatisfacibles
RF5. Crear un contrato	09 - Éxito	Consistente
	10 - Éxito y Fracaso	Consistente con conceptos insatisfacibles
RF6. Separar diferentes tipos de existencias	11 - Éxito	Consistente
	12 - Éxito y Fracaso	Consistente con conceptos insatisfacibles
RF7. Realizar el seguimiento de cantidades de entidades en existencias	13 - Éxito	Consistente
	14 - Éxito y Fracaso	Consistente con conceptos insatisfacibles
RF8. Realizar el seguimiento de las ubicaciones de las entidades	15 - Éxito	Consistente
	16 - Éxito y Fracaso	Consistente con conceptos insatisfacibles

Razonamiento en la Lógica Descriptiva y en OWL-DL

Consistencia

En la lógica descriptiva, una ABox A (afirmaciones de conceptos y roles) es consistente con respecto a una TBox T (afirmaciones de conceptos), si existe una interpretación I que es modelo tanto de A como de T . En OWL-DL, el razonador Pellet verifica la existencia de contradicciones en la ontología.

Conceptos insatisfacibles

En la lógica descriptiva, un concepto C es satisfacible con respecto a T , si existe una interpretación I tal que $C \neq \emptyset$. En este caso se dice que I es un modelo de A . En OWL-DL, el razonador Pellet, verifica si es posible para una clase tener alguna instancia. Si la clase es insatisfacible, entonces definir una instancia de la clase puede causar que la ontología completa sea inconsistente.

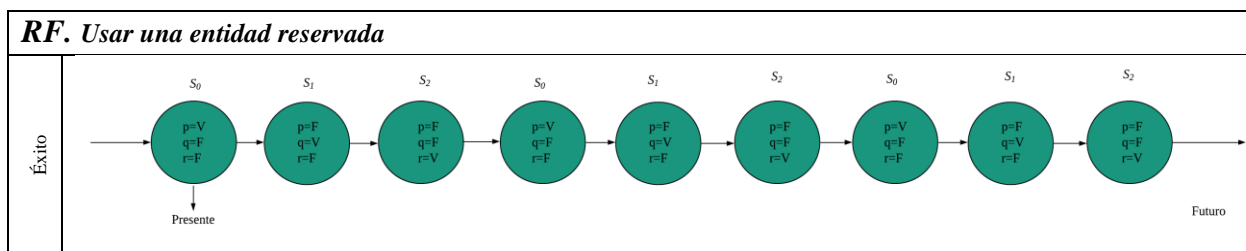
Las secuencias de interpretaciones de acuerdo a los requerimientos funcionales para el caso de éxito resultaron ser consistentes, es decir, no existen inconsistencias en la KB. Sin embargo, para los casos de éxito y fracaso la KB resultaron ser consistentes pero con conceptos insatisfacibles debido a las restricciones establecidas por las reglas SWRL de aserciones negativas.

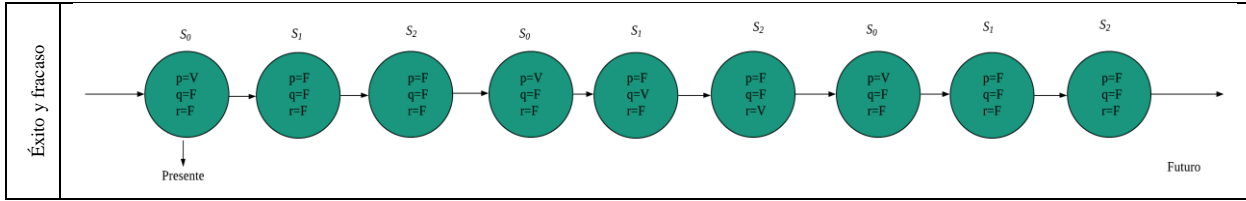
La documentación completa realizada para la verificación TSCC de la OTPAE del patrón de análisis PAIyER se encuentra en el directorio *Entregable 12/TSCC/*, almacenado en el disco que acompaña a esta tesis (para mayor referencia consultar el *Anexo G. Contenido del disco*).

6.1.6.3. Reducir la secuencia de interpretaciones a una ruta de computación

En esta actividad, se redujeron las secuencias de interpretaciones temporales a trayectorias de estados o rutas de computación. Estas trayectorias de estados son un componente generalmente usado en la verificación de satisfacibilidad de fórmulas de la TL LTL. En la Tabla 69 se muestra de manera gráfica, las trayectorias obtenidas a partir de la reducción de las secuencias de interpretaciones basadas en el comportamiento dinámico especificado en el RF Usar una entidad reservada, del patrón de análisis PAIyER.

Tabla 69. Representación gráfica de las trayectorias de estados del RF Usar una entidad reservada.





6.1.7. Verificación de la formalización

En esta fase, se realizó la verificación de la satisfacibilidad de las fórmulas LTL de las especificaciones formales obtenidas en la fase 6.1.5. *Combinación de fórmulas ALCLTL*.

6.1.7.1. Verificar la satisfacibilidad de la especificación en la ruta de computación

En esta actividad, las fórmulas LTL fueron evaluadas en la ruta de computación considerada para verificar su satisfacibilidad. Se utilizó la herramienta para la verificación automática de modelos (MC, por sus siglas en inglés, *Model Checking*) nuXmv en su versión 1.1.1.

Para llevar a cabo la verificación de la formalización, nuXmv requirió de un archivo con extensión .smv, el cual contiene como modelo un sistema de transición de estados. Es por ello que se modeló la ruta de computación (o trayectoria de estados) como un autómata determinista sobre el cual se evaluó la fórmula LTL. Por cada trayectoria de éxito (π') y éxito-fracaso (π'') de los ocho requerimientos del PAIyER, se modeló el autómata y se creó el respectivo programa SMV (*Symbolic Model Verifier*) que lo representa.

P. ej, para el RF Separar diferentes tipos de existencias, los autómatas para las trayectorias π' y π'' se muestran en la Figura 41 y Figura 42. El código fuente del programa SMV para cada autómata se describe en la Tabla 70 y en la Tabla 71, respectivamente. Los programas SMV de los RFs restantes se encuentran en la carpeta *Entregables/Entregable 12/SC*, almacenada en el disco que acompaña a esta tesis (para mayor referencia consultar el *Anexo G. Contenido del disco*).

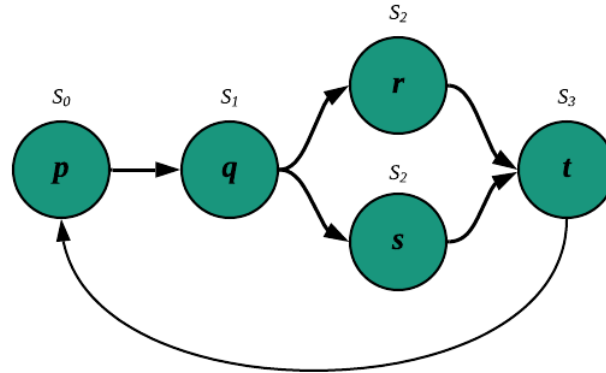


Figura 41. Autómata para la trayectoria de estados π' del RF Separar diferentes tipos de existencias.

Tabla 70 . Código fuente del programa Req6-otpaePAIyER-E.smv

```

MODULE main

VAR
  state: {s0,s1,s2,s3};
  empleado_1a: {separaEntidad, verificaExistenciaDeEntidad, actualizaEntidad,
  actualizaInventario,none};
  empleado_1b: { registraEntidad, none};

ASSIGN
  init(state):= s0;
  next(state):= case
    (state = s0) & (empleado_1a = separaEntidad) & (empleado_1b = none): s1;
    (state = s0): s1;
    (state = s1) & (empleado_1a = verificaExistenciaDeEntidad): s2;
    (state = s1): s2;
    (state = s2) & ((empleado_1a = actualizaEntidad) | (empleado_1b = registraEntidad)):
s3;
    (state = s2): s3;
    (state = s3) & (empleado_1a = actualizaInventario) : s0;
    (state = s3): s0;
    TRUE: state;
  esac;
  init(empleado_1a):= separaEntidad;
  next(empleado_1a):= case
    (state = s0): separaEntidad;
    (state = s1): verificaExistenciaDeEntidad;
    (state = s2): actualizaEntidad;
    (state = s3): actualizaInventario;
    TRUE: empleado_1a;
  esac;
  init(empleado_1b):= none;
  next(empleado_1b):= case
    (state = s0): none;
    (state = s1): none;

```

```

(state = s2): registraEntidad;
(state = s3): none;
TRUE: empleado_1b;
esac;

LTLSPEC G(X (empleado_1a = separaEntidad) ->(
  X X(empleado_1a = verificaExistenciaDeEntidad) & F(
    !X X X (empleado_1a = actualizaEntidad | empleado_1b = registraEntidad) & !X X X X empleado_1a =
      actualizaInventario
    )
  )
);

```

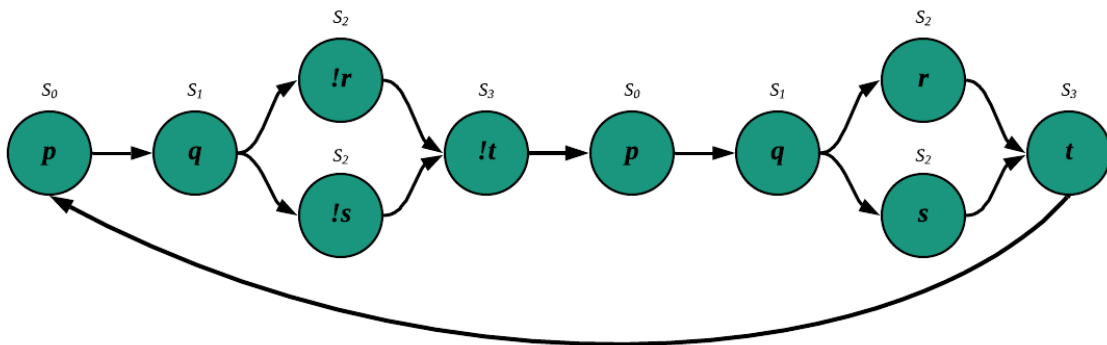


Figura 42 . Autómata para la trayectoria de estados π'' del RF Separara diferentes tipos de existencias.

Tabla 71 . Código fuente del programa Req6-otpaePAIyER-EF.smv.

```

MODULE main

VAR
  state: {s0,s1,s2,s3,s0_1,s1_1,s2_1,s3_1,s0_2,s1_2,s2_2,s3_2,s0_3,s1_3,s2_3,s3_3};
  empleado_1a: {separaEntidad, verificaExistenciaDeEntidad, actualizaEntidad,
actualizaInventario,none};
  empleado_1b: { registraEntidad, none};

ASSIGN
  init(state):= s0;
  next(state):= case
    -- caso de fracaso 1
    (state = s0) & (empleado_1a = separaEntidad) & (empleado_1b = none): s1;
    (state = s0): s1;
    (state = s1) & (empleado_1a = verificaExistenciaDeEntidad): s2;
    (state = s1): s2;
    (state = s2) & ((empleado_1a = actualizaEntidad) | (empleado_1b = registraEntidad)):
s3;

```

```

(state = s2): s3;
(state = s3) & (empleado_1a = actualizaInventario) : s0;
(state = s3): s0_1;
-- caso de éxito 1
(state = s0_1) & (empleado_1a = separaEntidad) & (empleado_1b = none): s1_1;
(state = s0_1): s1_1;
(state = s1_1) & (empleado_1a = verificaExistenciaDeEntidad): s2_1;
(state = s1_1): s2_1;
(state = s2_1) & ((empleado_1a = actualizaEntidad) | (empleado_1b = registraEntidad)):
s3_1;

(state = s2_1): s3_1;
(state = s3_1) & (empleado_1a = actualizaInventario) : s0_2;
(state = s3_1): s0_2;
-- caso de fracaso 2
(state = s0_2) & (empleado_1a = separaEntidad) & (empleado_1b = none): s1_2;
(state = s0_2): s1_2;
(state = s1_2) & (empleado_1a = verificaExistenciaDeEntidad): s2_2;
(state = s1_2): s2_2;
(state = s2_2) & ((empleado_1a = actualizaEntidad) | (empleado_1b = registraEntidad)):
s3_2;

(state = s2_2): s3_2;
(state = s3_2) & (empleado_1a = actualizaInventario) : s0_3;
(state = s3_2): s0_3;
-- caso de éxito 2
(state = s0_3) & (empleado_1a = separaEntidad) & (empleado_1b = none): s1_3;
(state = s0_3): s1_3;
(state = s1_3) & (empleado_1a = verificaExistenciaDeEntidad): s2_3;
(state = s1_3): s2_3;
(state = s2_3) & ((empleado_1a = actualizaEntidad) | (empleado_1b = registraEntidad)):
s3_3;

(state = s2_3): s3_3;
(state = s3_3) & (empleado_1a = actualizaInventario) : s0;
(state = s3_3): s0;
TRUE: state;
esac;

init(empleado_1a):= separaEntidad;
next(empleado_1a):= case
-- caso de fracaso 1
(state = s0): separaEntidad;
(state = s1): verificaExistenciaDeEntidad;
(state = s2): none;
(state = s3): none;
-- caso de éxito 1
(state = s0_1): separaEntidad;
(state = s1_1): verificaExistenciaDeEntidad;
(state = s2_1): actualizaEntidad;
(state = s3_1): actualizaInventario;
-- caso de fracaso 2
(state = s0_2): separaEntidad;
(state = s1_2): verificaExistenciaDeEntidad;
(state = s2_2): none;
(state = s3_2): none;
-- caso de éxito 2
(state = s0_3): separaEntidad;
(state = s1_3): verificaExistenciaDeEntidad;
(state = s2_3): actualizaEntidad;
(state = s3_3): actualizaInventario;
TRUE: empleado_1a;
esac;

```

```

init(empleado_1b):= none;
next(empleado_1b):= case
-- caso de fracaso 1
    (state = s0): none;
    (state = s1): none;
    (state = s2): none;
    (state = s3): none;
-- caso de éxito 1
    (state = s0_1): none;
    (state = s1_1): none;
    (state = s2_1): registraEntidad;
    (state = s3_1): none;
-- caso de fracaso 2
    (state = s0_2): none;
    (state = s1_2): none;
    (state = s2_2): none;
    (state = s3_2): none;
-- caso de éxito 2
    (state = s0_3): none;
    (state = s1_3): none;
    (state = s2_3): registraEntidad;
    (state = s3_3): none;
    TRUE: empleado_1b;
esac;

LTLSPEC G(X (empleado_1a = separaEntidad) ->(
  X X(empleado_1a = verificaExistenciaDeEntidad) & F(
    !X X X (empleado_1a = actualizaEntidad | empleado_1b = registraEntidad) & !X X X X empleado_1a =
      actualizaInventario
    )
  )
);

```

Los resultados de la verificación SC demostraron que las fórmulas LTL son satisfacibles en las trayectorias de estados consideradas al no generar contraejemplos. La verificación SC de las fórmulas LTL de las especificaciones de comportamiento obtenidas en la fase 6.1.5. *Combinación de fórmulas ALCLTL*, se realizó con ayuda de la herramienta nuXmv y se registró mediante el uso de la documentación descrita en el *Anexo F. Documentación SC*.

A continuación, en la Tabla 72 se presentan los resultados obtenidos durante la verificación SC.

Tabla 72 . Resultados de la verificación de satisfacibilidad de las fórmulas LTL.

Requerimiento	Caso de Verificación	Resultado de la Verificación
RF1. Hacer una reservación	01 - Éxito	Satisfacible
	02 - Éxito y Fracaso	Satisfacible
RF2. Usar una entidad reservada	03 - Éxito	Satisfacible
	04 - Éxito y Fracaso	Satisfacible

RF3. Modificar una reservación	05 - Éxito	Satisfacible
	06 – Éxito y Fracaso	Satisfacible
RF4. Cancelar una reservación	07 - Éxito	Satisfacible
	08 – Éxito y Fracaso	Satisfacible
RF5. Crear un contrato	09 - Éxito	Satisfacible
	10 – Éxito y Fracaso	Satisfacible
RF6. Separar diferentes tipos de existencias	11 - Éxito	Satisfacible
	12 – Éxito y Fracaso	Satisfacible
RF7. Realizar el seguimiento de cantidades de entidades en existencias	13 - Éxito	Satisfacible
	14 - Éxito y Fracaso	Satisfacible
RF8. Realizar el seguimiento de las ubicaciones de las entidades	15 - Éxito	Satisfacible
	16 - Éxito y Fracaso	Satisfacible

Los resultados de la verificación SC, demostraron que las fórmulas LTL son satisfacibles en las trayectorias de estado consideradas al no generar contraejemplos. La documentación completa realizada para la verificación SC de las fórmulas LTL se encuentran en el directorio *Entregable 12/SC/*, almacenada en el disco que acompaña a esta tesis (para mayor referencia consultar el *Anexo G. Contenido del disco*).

6.1.8. Instanciación de la ontología

Una vez definidas las especificaciones de comportamiento que deben satisfacerse en la ontología OTPAE compuesta, en esta fase se crearon los individuos (aserciones de conceptos y aserciones de roles abstractos temporales) de la ontología necesarios para representar trayectorias de estado. Un ejemplo de las aserciones de clases y de propiedades de objetos se presenta en la Tabla 73, cuya tabla muestra la equivalencia entre la sintaxis DL y la sintaxis funcional de OWL-DL.

Tabla 73. Ejemplo de aserciones de clases y de propiedades de objetos en el RF Hacer una reservación del PAIyER.

Sintaxis DL	Sintaxis Funcional
-- Aserciones de clase --	
<i>Cliente(Heidi)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Heidi)</i>
<i>Event(Req1-Ev1)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req1-Ev1)</i>
-- Aserciones de propiedades de objeto --	
<i>reserva(Heidi,Req1-Ev1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:reserva domain-PAIyER:Heidi domain-PAIyER:Req1-Ev1)</i>
<i>reserva(Req1-Ev1,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:reserva domain-PAIyER:Req1-Ev1 domain-PAIyER:Libro1)</i>

P. ej., la aserción de clase *Cliente(Heidi)*, indica que Heidi es un individuo de la clase Cliente y la aserción de clase *Event(Req1-Ev1)*, indica que Req1-Ev1 es un individuo de la

clase Event. De la misma manera, la aserciones de propiedad de objeto *reserva(Heidi,Req1-Ev1)* y *reserva(Req1-Ev1,Libro1)*, indican que el individuo Heidi reserva el individuo Libro1 a través del individuo Req1-Ev1.

Los individuos en cada trayectoria de comportamiento en el patrón de análisis PAIyER, se describen en la sección 1.2. *Componente extensional del Anexo D. Ontología OTPAE del patrón de análisis PAIyER* cuyas tablas muestran la equivalencia entre la sintaxis DL y la sintaxis funcional de OWL-DL.

En este capítulo, se presentaron los resultados y la documentación creada durante el proceso de combinación de la ontología temporal del PAI con la ontología temporal del PAER utilizando el método COMOTPA. La documentación presentada permite reforzar el entendimiento del método mediante un ejemplo de su utilización. Para concluir este capítulo, podemos decir que la OTPAE del PAIyER, es actualmente, la primera ontología temporal de patrones de análisis compuestos y el cuarto patrón de análisis documentado, hasta ahora, en nuestro grupo de investigación.

Capítulo VII. Conclusiones

En este capítulo, se presentan las conclusiones, las aportaciones de esta investigación y los trabajos que se pueden realizar a futuro.

7.1. Conclusiones

Con base en el trabajo presentado en esta tesis, se concluye que el objetivo general: “*Describir un método para la combinación de patrones de análisis representados mediante ontologías que incluyan en su formalización la lógica descriptiva y la lógica temporal para aspectos estructurales y de comportamiento, respectivamente*” fue cumplido exitosamente, debido a que se consumaron todos los objetivos específicos listados a continuación:

1. *Seleccionar los métodos a utilizar para realizar la combinación de ontologías que utilizan lógicas descriptiva y temporal en su formalización, cuyo logro permitió conocer los principales puntos a tomar en cuenta en el proceso de combinación de dos ontologías.*
2. *Definir una forma de combinar los patrones de análisis representados mediante ontologías que incluyen la lógica descriptiva y la lógica temporal, cuyo logro permitió obtener un método para combinar Ontologías Temporales de Patrones de Análisis denominado COMOTPA. Con este método, es posible combinar de forma manual dos ontologías temporales de patrones de análisis creadas con la metodología MECOTPA. Además, también es posible verificar la consistencia semántica temporal y la satisfacibilidad de los patrones de análisis compuestos.*
3. *Probar la combinación con una nueva ontología de un PA, utilizando el método propuesto, cuyo logro permitió obtener la ontología temporal del patrón de análisis PAIyER (resultado de la combinación de la ontología temporal del PAI con la ontología temporal del PAER) mediante el seguimiento del método COMOTPA. Además, se realizó la verificación de la consistencia semántica temporal y la verificación de la satisfacibilidad del patrón de análisis PAIyER.*

Los resultados de este trabajo de investigación, benefician a los ontólogos u otras personas interesadas, debido a que se podrán representar problemas más grandes, en la fase de análisis del proceso de desarrollo de software, a través de la combinación de patrones de análisis más pequeños representados mediante ontologías temporales. Aunque la implementación del método propuesto en este trabajo se realiza de forma manual, proporciona el proceso para combinar ontologías temporales de patrones de análisis y puede ser utilizado en un futuro para automatizar dicho método.

7.2. Aportaciones

Las principales aportaciones de este trabajo de investigación son:

- El método para la Combinación de Ontologías Temporales de Patrones de Análisis (COMOTPA) para combinar patrones de análisis representados mediante ontologías temporales.
- La Ontología del Patrón de Análisis para Inventarios (OPAE del PAI).

- La Ontología Temporal del Patrón de Análisis para Inventarios (OTPAE del PAI), cuyo proceso de construcción fue descrito en el artículo que lleva por nombre “Creación de la Ontología Temporal del Patrón de Análisis para Inventarios”.
- La Ontología Temporal del Patrón de Análisis para Inventarios y Entidades Reutilizables (OTPAE del PAIyER), un patrón de análisis compuesto, resultado de la combinación de la ontología temporal del PAI con la ontología temporal del PAER (creada en (Piedrola Domínguez N., 2019)).
- La verificación de consistencia semántica temporal (TSCC) y la verificación de satisfacibilidad (SC) de los patrones de análisis PAI y PAIyER.

7.3. Trabajos futuros

A pesar de haber cumplido con el objetivo general establecido para esta investigación, se han detectado algunos trabajos que se pueden realizar a futuro y que pueden tener un impacto positivo en esta misma línea de investigación. A continuación, se presentan algunos de estos trabajos:

- Combinar más ontologías temporales de patrones de análisis utilizando el método COMOTPA con la finalidad de incrementar el catálogo de patrones de análisis compuestos. Esto a su vez, implica crear más OTPAEs a través de la metodología MECOTPA; mientras más ontologías temporales de patrones de análisis se creen, mayores posibilidades de combinación habrán.
- Desarrollar un sistema para combinar ontologías temporales de patrones de análisis de forma automática o semiautomática.

- Adesina, O. (2017). *Integrating formal methods with model-driven engineering*. University of Ottawa Ottawa, Canada.
- Barchini, G. E., & Álvarez, M. M. (2010). ontología Dimensions and indicators of the ontology quality. *Avances En Sistemas e Informática*, 7(1), 29–38.
- Borst, W. N. (1997). Construction of engineering ontologies for knowledge sharing and reuse.
- Dzung, D. V. (2009). Ontology-based Reasoning in Requirements Elicitation, 263–272. <https://doi.org/10.1109/SEFM.2009.31>
- Fernández-López, M., & Gómez-Pérez, A. (2004). Searching for a Time Ontology for Semantic Web Applications, 114, 331–341.
- Fernandez, E. B. (1998). Building Systems Using Analysis Patterns. *Florida Atlantic University Boca Raton, FL*, (May). <https://doi.org/10.1145/288408.288418>
- Fernandez, E. B. (1999). Stock Manager: An Analysis Pattern for Inventories. *Dept. of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431*.
- Fernandez, E. B., & Yuan, X. (2009). An Analysis Pattern for Invoice Processing. *Pattern Languages of Program*, 1–10. <https://doi.org/10.1145/1943226.1943239>
- Fowler, M. (1996). Analysis patterns: reusable object models. *Concurrent Engineering*. <https://doi.org/doi:10.1177/1063293X9500300302>
- García, M. G., Jiménez, R. C. M., Brito, J. M. V., & Ramírez, R. M. L. (2015). Stable Template for Stable Analysis Patterns (Represented by Ontologies). *IEEE Latin America Transactions*, 13(7), 2430–2440. <https://doi.org/10.1109/TLA.2015.7273809>
- González, Moisés, C., Martínez, R. C. C., Villamil, J. M. C., & López, Ramiro Mar, C. (2015). Stable Template for Stable Analysis Patterns. *IEEE Latin America Transactions*, 13(7), 2430–2440.
- Grier, R., Sam, J., Hennessey, E., State, P., & Valley, G. (n.d.). Analysis Patterns, 12.
- Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications by A Translation Approach to Portable Ontology Specifications, (April).
- Hamza, H., & Fayad, M. (2002). A pattern language for building stable analysis patterns. *Proceedings of PLoP*.
- Harmelen, Frank Van, Vladimir Lifschitz, B. P. (2008). *Handbook of Knowledge Representation*. (B. P. Frank van Harmelen, Vladimir Lifschitz, Ed.) (First Edit).

- IEEE. (2008). *IEEE Standard for Software and System Test Documentation* (Vol. 2008). <https://doi.org/10.1109/IEEESTD.2008.4578383>
- Jiang, Z., & Fernandez, E. (2009). Composing analysis patterns to build complex models: Flight reservation. *Pattern Languages of Program*, 1–27. <https://doi.org/10.1145/1943226.1943240>
- Juárez Hernández, J. D. C. (2016). *Metodología de Validación de Consistencia de Patrones de Análisis*. Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET).
- Kaiya, H., & Saeki, M. (2006). Using Domain Ontology as Domain Knowledge for Requirements Elicitation. *IEE Computer Society*, 10. <https://doi.org/DOI:10.1109/RE.2006.72>
- Konrad, S., & Cheng, B. H. C. (2005). Real-time Specification Patterns*, (May), 10.
- López-Ramírez, R. M. (2015). *Almacenamiento y Uso de Patrones de Análisis Apegados a su Semántica*. Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET).
- López Ramírez, R. M. (2015). *Almacenamiento y uso de patrones de análisis*. Centro Nacional de Investigación y Desarrollo Tecnológico.
- Mahfoudh, M., Forestier, G., & Hassenforder, M. (2016). A benchmark for ontologies merging assessment. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9983 LNAI, 555–566. https://doi.org/10.1007/978-3-319-47650-6_44
- Narula GS, Wason R, Jain V, B. A. (2018). (2018). Ontology Mapping and Merging Aspects in Semantic Web. *International Robotics & Automation Journal*, 4(1). <https://doi.org/10.15406/iratj.2018.04.00087>
- Peixoto, D. A., Da Matta Vegi, L. F., & Lisboa-Filho, J. (2013). DC2AP metadata editor: A metadata editor for an analysis pattern reuse infrastructure. *CEUR Workshop Proceedings*, 998, 138–145.
- Piedrola Domínguez N. (2019). *Thesis for the degree of Master of Science: FOCOPAT: Conversión de ontologías de patrones de análisis estáticas a temporales, para la especificación de comportamientos dinámicos y su validación*. Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET).
- Prentis, A., Petrakis, E. G. M., & Batsakis, S. (2014). CHRONOS Ed: A Tool for Handling Temporal Ontologies in Protégé. *International Journal on Artificial Intelligence Tools*, 23(04), 1460018. <https://doi.org/10.1142/S0218213014600185>
- Studer, R., Benjamins, V. & Fensel, D. (1998). Knowledge engineering: principles and methods. *Data & Knowledge Engineering*, pp.161–197.

-
- Vidyarthi, A., Sharma, A., Sharma, H., & Soni, A. (2014). Domain specific ontology merging using semantics. *Proceedings of the 5th International Conference on Confluence 2014: The Next Generation Information Technology Summit*, 957–961. <https://doi.org/10.1109/CONFLUENCE.2014.6949313>
- Yuan, X., & Fernandez, E. B. (2011). Patterns for Business-to-Consumer e-Commerce Applications. *International Journal of Software Engineering & Applications (IJSEA)*, 2(3), 1–20.
- Zavaleta, Guadalupe Candia;Garcia, M. G. (2017). *Thesis for the degree of Master of Science: Almacenamiento y manipulación de ontologías que describen patrones de análisis y su semántica*. Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET).
- Zhang, J., & Lv, Y. (2012). An approach of refining the merged ontology. *Proceedings - 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2012*, (Fskd), 802–806. <https://doi.org/10.1109/FSKD.2012.6233973>
- Zhen-Xing, W., & Xing-Yan, T. (2015). Research of Ontology Merging Based on Concept Similarity. *Proceedings - 2015 7th International Conference on Measuring Technology and Mechatronics Automation, ICMTMA 2015*, (2), 831–834. <https://doi.org/10.1109/ICMTMA.2015.204>

En esta sección, se presentan los anexos (información relevante) para la comprensión de algunas secciones del presente documento de tesis.

Anexo A. Ontología OPAE del patrón de análisis PAI

En este anexo se presenta la última versión de la ontología OPAE del patrón de análisis para inventarios (PAI), obtenida a través de la metodología MECOTPA antes de convertirse en una ontología temporal.

1. Base de conocimiento

En esta sección se presentan las características de la KB (OPAAE) del patrón de análisis PAI utilizadas en este trabajo de investigación.

La *TBox* (o componente intencional) y la *ABox* (componente extensional) se describen bajo el poder de expresividad de la DL *ALCIF(D)*. En ambos componentes se incluyen las equivalencias correspondientes en la sintaxis funcional de OWL-DL.

Una versión completa de la KB se encuentra en el directorio *Entregables/Entregable 8/Ontologías/OPAAE PAI/*, almacenado en el disco que acompaña esta tesis (para mayor referencia consultar el *Anexo G. Contenido del disco*).

1.1. Componente intencional

Para obtener el componente *TBox* descrito en la Tabla A.1, se utilizaron:

1. Clases (en OWL-DL o nombres de conceptos N_C en las DLs):
{Articulo, Institucion, Inventario, Persona, Empleado}
a partir de las cuales se generó la jerarquía de clases de la Figura A.1, que modela el conocimiento central del patrón de análisis PAI.
2. Propiedades de objeto (en OWL-DL o nombres de roles abstractos N_{RA} en las DLs):
{actualiza, existenciasSeObtienePor, existenciasSeBuscaPor, seActualizaPor, especificaTipoExistenciasDe, seRegistraPor, seSeparaPor, seUbicaPor, existenciaSeVerificaPor, laboraEn, obtieneExistenciasDe, obtieneUbicacionDe, registra, separa, solicitaUbicacionDe, tieneDatosDe, tipoExistenciasSeEspecificaPor, ubica, ubicacionSeObtienePor, ubicacionSeSolicitaPor, buscaExistenciasDe, verificaExistenciaDe}
3. Propiedades de datos (en OWL-DL o nombres de roles concretos N_{RC} en las DLs):
{tieneCantidad, tieneComportamiento, tieneEdad, tieneEspecificidad, tieneGiro, tieneId, tieneNivelEstabilidad, tieneNombre, tienePercepcion, tieneTipo, tieneUbicacion }
4. Espacios de nombres (o prefijos en OWL para abreviar URIs extensas):
 - Prefix(owl:=<http://www.w3.org/2002/07/owl#>)
 - Ontología que describe parcialmente las clases y propiedades integradas que forman la base de la sintaxis RDF / XML de OWL 2.
 - Prefix(rdf:=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
 - Esquema RDF para el vocabulario RDF definido en el espacio de nombres RDF.
 - Prefix(xml:=<http://www.w3.org/XML/1998/namespace>)
 - El espacio de nombres XML proporciona un método simple para calificar los nombres de elementos y atributos utilizados en documentos XML al asociarlos con espacios de nombres identificados por referencias de URI.
 - Prefix(xsd:=<http://www.w3.org/2001/XMLSchema#>)

- XML Schema describe la estructura de un documento XML, también se conoce como definición de esquema XML (XSD).
- Prefix(rdfs:=<http://www.w3.org/2000/01/rdf-schema#>)
- RDF Schema proporciona un vocabulario de modelado de datos para datos RDF. El esquema RDF es una extensión del vocabulario RDF básico.
- Prefix(domain-pai:=<https://www.cenidet.edu.mx/SoftwareEngineering/AnalysisPatterns/StaticOntology/PAI#>)
- Vocabulario en el dominio del patrón de análisis PAI (aspectos estáticos).

Tabla A.1 TBox de la KB del patrón de análisis PAI (cont.)

	Sintaxis DL	Sintaxis Funcional
<i>TBox = {</i>		
<i>-- Clases --</i>		
<i>T</i>		\Rightarrow <i>Declaration(Class(owl:Thing))</i>
<i>Articulo $\sqsubseteq T$</i>		\Rightarrow <i>Declaration(Class(domain-pai:Articulo))</i> <i>SubClassOf(domain-pai:Articulo owl:Thing)</i>
<i>Institucion $\sqsubseteq T$</i>		\Rightarrow <i>Declaration(Class(domain-pai:Institucion))</i> <i>SubClassOf(domain-pai:Institucion owl:Thing)</i>
<i>Inventario $\sqsubseteq T$</i>		\Rightarrow <i>Declaration(Class(domain-pai:Inventario))</i> <i>SubClassOf(domain-pai:Inventario owl:Thing)</i>
<i>Persona $\sqsubseteq T$</i>		\Rightarrow <i>Declaration(Class(domain-pai:Persona))</i> <i>SubClassOf(domain-pai:Persona owl:Thing)</i>
<i>Articulo $\sqsubseteq T$</i>		\Rightarrow <i>Declaration(Class(domain-pai:Articulo))</i> <i>SubClassOf(domain-pai:Articulo owl:Thing)</i>
<i>Empleado $\sqsubseteq Persona$</i>		\Rightarrow <i>Declaration(Class(domain-pai:Empleado))</i> <i>SubClassOf(domain-pai:Empleado domain-pai:Persona)</i>
<i>-- Clases disjuntas --</i>		
<i>Articulo \sqcap Institucion \sqcap Inventario \sqcap Empleado $\sqsubseteq \perp$</i>		\Rightarrow <i>DisjointClasses(domain-pai:Articulo domain-pai:Institucion domain-pai:Inventario domain-pai:Empleado)</i>
<i>-- Restricciones existenciales y universales --</i>		
<i>Empleado $\sqsubseteq \exists$separa.Articulo</i>		\Rightarrow <i>SubClassOf(domain-pai:Empleado</i> <i>ObjectAllValuesFrom(domain-pai:separa</i> <i>Domain-pai:Articulo))</i>
<i>Empleado $\sqsubseteq \exists$registra.Articulo</i>		\Rightarrow <i>SubClassOf(domain-pai:Empleado</i> <i>ObjectAllValuesFrom(domain-pai:registra</i> <i>Domain-pai:Articulo))</i>
<i>Empleado $\sqsubseteq \exists$buscaExistenciasDe.Articulo</i>		\Rightarrow <i>SubClassOf(domain-pai:Empleado</i> <i>ObjectAllValuesFrom(domain-pai:</i> <i>buscaExistenciasDe Domain-pai:Articulo))</i>
<i>Empleado $\sqsubseteq \exists$verificaExistenciaDe.Articulo</i>		\Rightarrow <i>SubClassOf(domain-pai:Empleado</i> <i>ObjectAllValuesFrom(domain-pai:</i> <i>verificaExistenciaDe Domain-pai:Articulo))</i>
<i>Empleado $\sqsubseteq \exists$obtieneUbicacionDe.Articulo</i>		\Rightarrow <i>SubClassOf(domain-pai:Empleado</i> <i>ObjectAllValuesFrom(domain-pai:</i> <i>obtieneUbicacionDe Domain-pai:Articulo))</i>
<i>Empleado $\sqsubseteq \exists$especificaTipoExistenciasDe.Articulo</i>		\Rightarrow <i>SubClassOf(domain-pai:Empleado</i> <i>ObjectAllValuesFrom(domain-pai:</i>

Tabla A.1 TBox de la KB del patrón de análisis PAI (cont.)

Sintaxis DL	Sintaxis Funcional
$Empleado \sqsubseteq \exists \text{obtieneExistenciasDe} . \text{Articulo}$	\Rightarrow <i>especificaTipoExistenciasDe Domain-pai:Articulo</i>) <i>SubClassOf(domain-pai:Empleado ObjectAllValuesFrom(domain-pai:obtieneExistenciasDe Domain-pai:Articulo</i>))
$Empleado \sqsubseteq \exists \text{ubica} . \text{Articulo}$	\Rightarrow <i>SubClassOf(domain-pai:Empleado ObjectAllValuesFrom(domain-pai:ubica Domain-pai:Articulo</i>))
$Empleado \sqsubseteq \exists \text{solicitaUbicacionDe} . \text{Articulo}$	\Rightarrow <i>SubClassOf(domain-pai:Empleado ObjectAllValuesFrom(domain-pai:solicitaUbicacionDe Domain-pai:Articulo</i>))
$Institucion \sqsubseteq \exists \text{tieneDatosDe} . \text{Empleado}$	\Rightarrow <i>SubClassOf(domain-pai:Institucion ObjectAllValuesFrom(domain-pai:tieneDatosDe domain-pai:Empleado))</i>
$\text{Articulo} \sqsubseteq \forall \text{seSeparaPor} . \text{Empleado}$	\Rightarrow <i>SubClassOf(domain-pai:Articulo ObjectAllValuesFrom(domain-pai:seSeparaPor domain-pai:Empleado))</i>
$\text{Articulo} \sqsubseteq \forall \text{seRegistraPor} . \text{Empleado}$	\Rightarrow <i>SubClassOf(domain-pai:Articulo ObjectAllValuesFrom(domain-pai:seRegistraPor domain-pai:Empleado))</i>
$\text{Empleado} \sqsubseteq \forall \text{actualiza} . \text{Inventario}$	\Rightarrow <i>SubClassOf(domain-pai:Empleado ObjectAllValuesFrom(domain-pai:actualiza domain-pai:Inventario))</i>
$\text{Empleado} \sqsubseteq \forall \text{laboraEn} . \text{Institucion}$	\Rightarrow <i>SubClassOf(domain-pai:Empleado ObjectAllValuesFrom(domain-pai:laboraEn domain-pai:Institucion))</i>
-- Propiedades de objeto --	
<i>actualiza</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-pai:actualiza))</i>
<i>seActualizaPor</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-pai:seActualizaPor))</i>
<i>obtieneExistenciasDe</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-pai:obtieneExistenciasDe))</i>
<i>existenciasSeObtienePor</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-pai:existenciasSeObtienePor))</i>
<i>buscaExistenciasDe</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-pai:buscaExistenciasDe))</i>
<i>existenciasSeBuscaPor</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-pai:existenciasSeBuscaPor))</i>
<i>especificaTipoExistenciasDe</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-pai:especificaTipoExistenciasDe))</i>
<i>tipoExistenciasSeEspecificaPor</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-pai:tipoExistenciasSeEspecificaPor))</i>
<i>registra</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-pai:registra))</i>
<i>seRegistraPor</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-pai:seRegistraPor))</i>
<i>separa</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-pai:separa))</i>
<i>seSeparaPor</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-pai:seSeparaPor))</i>
<i>ubica</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-pai:ubica))</i>

Tabla A.1 TBox de la KB del patrón de análisis PAI (cont.)

Sintaxis DL	Sintaxis Funcional
<i>seUbicaPor</i>	\Rightarrow Declaration(ObjectProperty(domain-pai: seUbicaPor))
<i>verificaExistenciaDe</i>	\Rightarrow Declaration(ObjectProperty(domain-pai: verificaExistenciaDe))
<i>existenciaSeVerificaPor</i>	\Rightarrow Declaration(ObjectProperty(domain-pai: existenciaSeVerificaPor))
<i>laboraEn</i>	\Rightarrow Declaration(ObjectProperty(domain-pai: laboraEn))
<i>obtieneUbicacionDe</i>	\Rightarrow Declaration(ObjectProperty(domain-pai: obtieneUbicacionDe))
<i>ubicacionSeObtienePor</i>	\Rightarrow Declaration(ObjectProperty(domain-pai: ubicacionSeObtienePor))
<i>solicitaUbicacionDe</i>	\Rightarrow Declaration(ObjectProperty(domain-pai: solicitaUbicacionDe))
<i>ubicacionSeSolicitaPor</i>	\Rightarrow Declaration(ObjectProperty(domain-pai: ubicacionSeSolicitaPor))
<i>tieneDatosDe</i>	\Rightarrow Declaration(ObjectProperty(domain-pai: tieneDatosDe))
-- Restricciones y características de propiedades de objeto --	
\exists actualiza.T \sqsubseteq Empleado	\Rightarrow ObjectPropertyDomain(domain-pai:actualiza domain-pai:Empleado))
$T \sqsubseteq \forall$ actualiza.Inventario	\Rightarrow ObjectPropertyRange(domain-pai:actualiza domain-pai:Inventario))
$T \sqsubseteq \leq_1$ actualiza	\Rightarrow FunctionalObjectProperty(domain-pai:actualiza)
\exists seActualizaPor.T \sqsubseteq Inventario	\Rightarrow ObjectPropertyDomain(domain-pai:seActualizaPor domain-pai:Inventario))
$T \sqsubseteq \forall$ seActualizaPor. Empleado	\Rightarrow ObjectPropertyRange(domain-pai:seActualizaPor domain-pai:Empleado))
$T \sqsubseteq \leq_1$ seActualizaPor ⁻	\Rightarrow InverseFunctionalObjectProperty(domain-pai: seActualizaPor)
\exists obtieneExistenciasDe.T \sqsubseteq Empleado	\Rightarrow ObjectPropertyDomain(domain-pai: obtieneExistenciasDe domain-pai:Empleado))
$T \sqsubseteq \forall$ obtieneExistenciasDe.Articulo	\Rightarrow ObjectPropertyRange(domain-pai: obtieneExistenciasDe Domain-pai:Articulo)
\exists cantidadExistenciasSeObtienePor.T \sqsubseteq .Articulo	\Rightarrow ObjectPropertyDomain(domain-pai: cantidadExistenciasSeObtienePor Domain-pai:Articulo)
$T \sqsubseteq \forall$ cantidadExistenciasSeObtienePor. Empleado	\Rightarrow ObjectPropertyRange(domain-pai: cantidadExistenciasSeObtienePor domain-pai:Empleado))
\exists buscaExistenciasDe.T \sqsubseteq Empleado	\Rightarrow ObjectPropertyDomain(domain-pai: buscaExistenciasDe domain-pai:Empleado))
$T \sqsubseteq \forall$ buscaExistenciasDe. Articulo	\Rightarrow ObjectPropertyRange(domain-pai: buscaExistenciasDe Domain-pai:Articulo)
\exists existenciasSeBuscaPor.T \sqsubseteq Articulo	\Rightarrow ObjectPropertyDomain(domain-pai: existenciasSeBuscaPor Domain-pai:Articulo)
$T \sqsubseteq \forall$ existenciasSeBuscaPor.Empleado	\Rightarrow ObjectPropertyRange(domain-pai: existenciasSeBuscaPor domain-pai:Empleado))
\exists especificaTipoExistenciasDe.T \sqsubseteq Empleado	\Rightarrow ObjectPropertyDomain(domain-pai: especificaTipoExistenciasDe domain-pai:Empleado))

Tabla A.1 TBox de la KB del patrón de análisis PAI (cont.)

Sintaxis DL	Sintaxis Funcional
$T \sqsubseteq \forall \text{especificaTipoExistenciasDe. Articulo}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: especificaTipoExistenciasDe Domain-pai:Articulo})$
$\exists \text{tipoExistenciasSeEspecificaPor.T} \sqsubseteq \text{Articulo}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: tipoExistenciasSeEspecificaPor Domain-pai:Articulo})$
$T \sqsubseteq \forall \text{tipoExistenciasSeEspecificaPor. Empleado}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: tipoExistenciasSeEspecificaPor domain-pai:Empleado})$
$\exists \text{registra.T} \sqsubseteq \text{Empleado}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: registra domain-pai:Empleado})$
$T \sqsubseteq \forall \text{registra. Articulo}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: registra Domain-pai:Articulo})$
$T \sqsubseteq \leq_1 \text{registra}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-pai: registra})$
$\exists \text{seRegistraPor.T} \sqsubseteq \text{Articulo}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: seRegistraPor Domain-pai:Articulo})$
$T \sqsubseteq \forall \text{seRegistraPor. Empleado}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: seRegistraPor domain-pai:Empleado})$
$T \sqsubseteq \leq_1 \text{seRegistraPor}$	$\Rightarrow \text{FunctionalObjectProperty}(\text{domain-pai: seRegistraPor})$
$\exists \text{separa.T} \sqsubseteq \text{Empleado}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: separa domain-pai:Empleado})$
$T \sqsubseteq \forall \text{separa. Articulo}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: separa Domain-pai:Articulo})$
$T \sqsubseteq \leq_1 \text{separa}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-pai: separa})$
$\exists \text{seSeparaPor.T} \sqsubseteq \text{Articulo}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: seSeparaPor Domain-pai:Articulo})$
$T \sqsubseteq \forall \text{seSeparaPor. Empleado}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: seSeparaPor domain-pai:Empleado})$
$T \sqsubseteq \leq_1 \text{seSeparaPor}$	$\Rightarrow \text{FunctionalObjectProperty}(\text{domain-pai: seSeparaPor})$
$\exists \text{ubica.T} \sqsubseteq \text{Empleado}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: ubica domain-pai:Empleado})$
$T \sqsubseteq \forall \text{ubica. Articulo}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: ubica Domain-pai:Articulo})$
$\exists \text{seUbicaPor.T} \sqsubseteq \text{Articulo}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: seUbicaPor Domain-pai:Articulo})$
$T \sqsubseteq \forall \text{seUbicaPor. Empleado}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: seUbicaPor domain-pai:Empleado})$
$\exists \text{verificaExistenciaDe.T} \sqsubseteq \text{Empleado}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: verificaExistenciaDe domain-pai:Empleado})$
$T \sqsubseteq \forall \text{verificaExistenciaDe. Articulo}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: verificaExistenciaDe Domain-pai:Articulo})$
$\exists \text{existenciaSeVerificaPor.T} \sqsubseteq \text{Articulo}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: existenciaSeVerificaPor Domain-pai:Articulo})$
$T \sqsubseteq \forall \text{existenciaSeVerificaPor. Empleado}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: existenciaSeVerificaPor domain-pai:Empleado})$
$\exists \text{obtieneUbicacionDe.T} \sqsubseteq \text{Empleado}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: obtieneUbicacionDe domain-pai:Empleado})$
$T \sqsubseteq \forall \text{obtieneUbicacionDe. Articulo}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: obtieneUbicacionDe domain-pai:Articulo})$

Tabla A.1 TBox de la KB del patrón de análisis PAI (cont.)

Sintaxis DL	Sintaxis Funcional
$\exists \text{ubicacionSeObtienePor}.T \sqsubseteq \text{Articulo}$	$\Rightarrow \text{obtieneUbicacionDe Domain-pai:Articulo)}$ $\Rightarrow \text{ObjectPropertyDomain(domain-pai:ubicacionSeObtienePor Domain-pai:Articulo)}$
$T \sqsubseteq \forall \text{ubicacionSeObtienePor}. \text{Empleado}$	$\Rightarrow \text{ObjectPropertyRange(domain-pai:ubicacionSeObtienePor domain-pai:Empleado)}$
$\exists \text{solicitaUbicacionDe}.T \sqsubseteq \text{Empleado}$	$\Rightarrow \text{ObjectPropertyDomain(domain-pai:solicitaUbicacionDe domain-pai:Empleado)}$
$T \sqsubseteq \forall \text{solicitaUbicacionDe}. \text{Articulo}$	$\Rightarrow \text{ObjectPropertyRange(domain-pai:solicitaUbicacionDe Domain-pai:Articulo)}$
$\exists \text{ubicacionSeSolicitaPor}.T \sqsubseteq \text{Articulo}$	$\Rightarrow \text{ObjectPropertyDomain(domain-pai:ubicacionSeSolicitaPor Domain-pai:Articulo)}$
$T \sqsubseteq \forall \text{ubicacionSeSolicitaPor}. \text{Empleado}$	$\Rightarrow \text{ObjectPropertyRange(domain-pai:ubicacionSeSolicitaPor domain-pai:Empleado)}$
$\exists \text{tieneDatosDe}.T \sqsubseteq \text{Institucion}$	$\Rightarrow \text{ObjectPropertyDomain(domain-pai:tieneDatosDe domain-pai:Institucion)}$
$T \sqsubseteq \forall \text{tieneDatosDe}. \text{Empleado}$	$\Rightarrow \text{ObjectPropertyRange(domain-pai:tieneDatosDe domain-pai:Empleado)}$
$\exists \text{laboraEn}.T \sqsubseteq \text{Empleado}$	$\Rightarrow \text{ObjectPropertyDomain(domain-pai:laboraEn domain-pai:Empleado)}$
$T \sqsubseteq \forall \text{laboraEn}. \text{Institucion}$	$\Rightarrow \text{ObjectPropertyRange(domain-pai:laboraEn domain-pai:Institucion)}$
$\text{actualiza} \equiv \text{seActualizaPor}^-$	$\Rightarrow \text{InverseObjectProperties(domain-pai:actualiza domain-pai:seActualizaPor)}$
$\text{obtieneExistenciasDe} \equiv \text{existenciasSeObtienePor}^-$	$\Rightarrow \text{InverseObjectProperties(domain-pai:obtieneExistenciasDe domain-pai:existenciasSeObtienePor)}$
$\text{buscaExistenciasDe} \equiv \text{existenciasSeBuscaPor}^-$	$\Rightarrow \text{InverseObjectProperties(domain-pai:buscaExistenciasDe domain-pai:existenciasSeBuscaPor)}$
$\text{especificaTipoExistenciasDe} \equiv \text{tipoExistenciasSeEspecificaPor}^-$	$\Rightarrow \text{InverseObjectProperties(domain-pai:especificaTipoExistenciasDe domain-pai:tipoExistenciasSeEspecificaPor)}$
$\text{registra} \equiv \text{seRegistraPor}^-$	$\Rightarrow \text{InverseObjectProperties(domain-pai:registra domain-pai:seRegistraPor)}$
$\text{separa} \equiv \text{seSeparaPor}^-$	$\Rightarrow \text{InverseObjectProperties(domain-pai:separa domain-pai:seSeparaPor)}$
$\text{ubica} \equiv \text{seUbicaPor}^-$	$\Rightarrow \text{InverseObjectProperties(domain-pai:ubica domain-pai:seUbicaPor)}$
$\text{verificaExistenciaDe} \equiv \text{existenciaSeVerificaPor}^-$	$\Rightarrow \text{InverseObjectProperties(domain-pai:verificaExistenciaDe domain-pai:existenciaSeVerificaPor)}$
$\text{obtieneUbicacionDe} \equiv \text{ubicacionSeObtienePor}^-$	$\Rightarrow \text{InverseObjectProperties(domain-pai:obtieneUbicacionDe domain-pai:ubicacionSeObtienePor)}$
$\text{solicitaUbicacionDe} \equiv \text{ubicacionSeSolicitaPor}^-$	$\Rightarrow \text{InverseObjectProperties(domain-pai:solicitaUbicacionDe domain-pai:ubicacionSeSolicitaPor)}$
-- Propiedades de datos --	
tieneCantidad	$\Rightarrow \text{Declaration(DataProperty(domain-pai:tieneCantidad))}$
$\text{tieneComportamiento}$	$\Rightarrow \text{Declaration(DataProperty(domain-pai:tieneComportamiento))}$

Tabla A.1 TBox de la KB del patrón de análisis PAI (cont.)

Sintaxis DL	Sintaxis Funcional
<i>tienePercepcion</i>	\Rightarrow Declaration(DataProperty(domain-pai: <i>tienePercepcion</i>))
<i>tieneEspecificidad</i>	\Rightarrow Declaration(DataProperty(domain-pai: <i>tieneEspecificidad</i>))
<i>tieneNivelEstabilidad</i>	\Rightarrow Declaration(DataProperty(domain-pai: <i>tieneNivelEstabilidad</i>))
<i>tieneEdad</i>	\Rightarrow Declaration(DataProperty(domain-pai: <i>tieneEdad</i>))
<i>tieneGiro</i>	\Rightarrow Declaration(DataProperty(domain-pai: <i>tieneGiro</i>))
<i>tieneId</i>	\Rightarrow Declaration(DataProperty(domain-pai: <i>tieneId</i>))
<i>tieneNombre</i>	\Rightarrow Declaration(DataProperty(domain-pai: <i>tieneNombre</i>))
<i>tieneTipo</i>	\Rightarrow Declaration(DataProperty(domain-pai: <i>tieneTipo</i>))
<i>tieneUbicacion</i>	\Rightarrow Declaration(DataProperty(domain-pai: <i>tieneUbicacion</i>))
-- Restricciones y características de propiedades de datos --	
≥ 1 <i>tieneCantidad</i> \sqsubseteq <i>Articulo</i>	\Rightarrow DataPropertyDomain(domain-pai: <i>tieneCantidad</i> Domain-pai: <i>Articulo</i>)
$T \sqsubseteq \leq 1$ <i>tieneCantidad</i>	\Rightarrow FunctionalDataProperty(domain-pai: <i>tieneCantidad</i>)
$T \sqsubseteq \forall$ <i>tieneCantidad</i> .integer	\Rightarrow DataPropertyRange(domain-pai: <i>tieneCantidad</i> xsd:integer)
≥ 1 <i>tieneComportamiento</i> \sqsubseteq (<i>Articulo</i> \sqcup <i>Empleado</i> \sqcup <i>Inventario</i> \sqcup <i>Institucion</i>)	\Rightarrow DataPropertyDomain(domain-pai: <i>tieneComportamiento</i> ObjectUnionOf(domain-pai: <i>Articulo</i> domain-pai: <i>Empleado</i> domain-pai: <i>Inventario</i> domain-pai: <i>Institucion</i>))
$T \sqsubseteq \forall$ <i>tieneComportamiento</i> .string	\Rightarrow DataPropertyRange(domain-pai: <i>tieneComportamiento</i> xsd:string)
$T \sqsubseteq \leq$ <i>tieneComportamiento</i>	\Rightarrow FunctionalDataProperty(domain-pai: <i>tieneComportamiento</i>)
≥ 1 <i>tieneEdad</i> \sqsubseteq <i>Empleado</i>	\Rightarrow DataPropertyDomain(domain-pai: <i>tieneEdad</i> domain-pai: <i>Empleado</i>)
$T \sqsubseteq \forall$ <i>tieneEdad</i> .integer	\Rightarrow DataPropertyRange(domain-pai: <i>tieneEdad</i> xsd:integer)
$T \sqsubseteq \leq$ <i>tieneEdad</i>	\Rightarrow FunctionalDataProperty(domain-pai: <i>tieneEdad</i>)
≥ 1 <i>tieneEspecificidad</i> \sqsubseteq (<i>Articulo</i> \sqcup <i>Empleado</i> \sqcup <i>Inventario</i> \sqcup <i>Institucion</i>)	\Rightarrow DataPropertyDomain(domain-pai: <i>tieneEspecificidad</i> ObjectUnionOf(domain-pai: <i>Articulo</i> domain-pai: <i>Empleado</i> domain-pai: <i>Inventario</i> domain-pai: <i>Institucion</i>))
$T \sqsubseteq \forall$ <i>tieneEspecificidad</i> .string	\Rightarrow DataPropertyRange(domain-pai: <i>tieneEspecificidad</i> xsd:string)
$T \sqsubseteq \leq$ <i>tieneEspecificidad</i>	\Rightarrow FunctionalDataProperty(domain-pai: <i>tieneEspecificidad</i>)
≥ 1 <i>tieneGiro</i> \sqsubseteq <i>Institucion</i>	\Rightarrow DataPropertyDomain(domain-pai: <i>tieneGiro</i> domain-pai: <i>Institucion</i>)
$T \sqsubseteq \forall$ <i>tieneGiro</i> .string	\Rightarrow DataPropertyRange(domain-pai: <i>tieneGiro</i> xsd:string)
$T \sqsubseteq \leq$ <i>tieneGiro</i>	\Rightarrow FunctionalDataProperty(domain-pai: <i>tieneGiro</i>)

Tabla A.1 TBox de la KB del patrón de análisis PAI (cont.)

Sintaxis DL	Sintaxis Funcional
$\geq 1 \text{ tieneId} \sqsubseteq (\text{Articulo} \sqcup \text{Empleado} \sqcup \text{Inventario} \sqcup \text{Institucion})$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-pai: tieneId} \text{ ObjectUnionOf}(\text{domain-pai:Articulo} \text{ domain-pai:Empleado} \text{ domain-pai:Inventario} \text{ domain-pai:Institucion}))$
$T \sqsubseteq \forall \text{ tieneId.string}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-pai: tieneId} \text{ xsd:string})$
$T \sqsubseteq \leq \text{ tieneId}$	$\Rightarrow \text{FunctionalDataProperty}(\text{domain-pai: tieneId})$
$\geq 1 \text{ tieneNivelEstabilidad} \sqsubseteq (\text{Articulo} \sqcup \text{Empleado} \sqcup \text{Inventario} \sqcup \text{Institucion})$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-pai: tieneNivelEstabilidad} \text{ ObjectUnionOf}(\text{domain-pai:Articulo} \text{ domain-pai:Empleado} \text{ domain-pai:Inventario} \text{ domain-pai:Institucion}))$
$T \sqsubseteq \forall \text{ tieneNivelEstabilidad.string}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-pai: tieneNivelEstabilidad} \text{ xsd:string})$
$T \sqsubseteq \leq \text{ tieneNivelEstabilidad}$	$\Rightarrow \text{FunctionalDataProperty}(\text{domain-pai: tieneNivelEstabilidad})$
$\geq 1 \text{ tieneNombre} \sqsubseteq (\text{Articulo} \sqcup \text{Empleado} \sqcup \text{Inventario} \sqcup \text{Institucion})$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-pai: tieneNombre} \text{ ObjectUnionOf}(\text{domain-pai:Articulo} \text{ domain-pai:Empleado} \text{ domain-pai:Inventario} \text{ domain-pai:Institucion}))$
$T \sqsubseteq \forall \text{ tieneNombre.string}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-pai: tieneNombre} \text{ xsd:string})$
$T \sqsubseteq \leq \text{ tieneNombre}$	$\Rightarrow \text{FunctionalDataProperty}(\text{domain-pai: tieneNombre})$
$\geq 1 \text{ tienePercepcion} \sqsubseteq (\text{Articulo} \sqcup \text{Empleado} \sqcup \text{Inventario} \sqcup \text{Institucion})$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-pai: tienePercepcion} \text{ ObjectUnionOf}(\text{domain-pai:Articulo} \text{ domain-pai:Empleado} \text{ domain-pai:Inventario} \text{ domain-pai:Institucion}))$
$T \sqsubseteq \forall \text{ tienePercepcion.string}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-pai: tienePercepcion} \text{ xsd:string})$
$T \sqsubseteq \leq \text{ tienePercepcion}$	$\Rightarrow \text{FunctionalDataProperty}(\text{domain-pai: tienePercepcion})$
$\geq 1 \text{ tieneTipo} \sqsubseteq \text{Articulo}$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-pai: tieneTipo} \text{ domain-pai:Articulo})$
$T \sqsubseteq \forall \text{ tieneTipo.string}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-pai: tieneTipo} \text{ xsd:string})$
$T \sqsubseteq \leq \text{ tieneTipo}$	$\Rightarrow \text{FunctionalDataProperty}(\text{domain-pai: tieneTipo})$
$\geq 1 \text{ tieneUbicacion} \sqsubseteq \text{Articulo}$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-pai: tieneUbicacion} \text{ Domain-pai:Articulo})$
$T \sqsubseteq \forall \text{ tieneUbicacion.string}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-pai: tieneUbicacion} \text{ xsd:string})$
$T \sqsubseteq \leq 1 \text{ tieneUbicacion}$	$\Rightarrow \text{FunctionalDataProperty}(\text{domain-pai: tieneUbicacion})$

Para una mejor comprensión de las descripciones lógicas presentadas en la TBox de la Tabla A.1, a continuación se describen algunos ejemplos tomados de las secciones *Clases*, *Clases Disjuntas*, *Restricciones existenciales y universales*, *Propiedades de objeto*, *Restricciones y características de propiedades de objeto*, *Propiedades de datos y Restricciones y características de propiedades de datos* en la misma tabla:

Clases

T : Conjunto de todos los individuos en el dominio.

$Persona \sqsubseteq T$: La clase Persona es un tipo de individuos en el dominio.

$Empleado \sqsubseteq Persona$: La clase Empleado es un tipo de la clase Persona.

Clases Disjuntas

$Articulo \sqcap Institucion \sqcap Inventario \sqcap Empleado \sqsubseteq \perp$: Establece que los individuos que son miembros de una de las clases (Articulo, Institucion, Inventario y Empleado), no son instancias de otra clase.

Restricciones existenciales y universales

$Inventario \sqsubseteq \forall seActualizaPor.Empleado$: Establece que un Inventario es todo aquello que es actualizado por un Empleado.

$Empleado \sqsubseteq \exists actualiza.Inventario$: Establece que un Empleado es quien al menos actualiza un inventario.

Propiedades de objeto

actualiza: representa la declaración de la propiedad de objeto actualiza.

Restricciones y características de propiedades de objeto

$\exists actualiza.T \sqsubseteq Empleado$: Establece a la clase Empleado en el dominio de la propiedad de objeto actualiza.

$T \sqsubseteq \forall actualiza.Inventario$: Establece a la clase Inventario como rango de la propiedad de objeto actualiza.

$T \sqsubseteq \leq I actualiza$: Establece que la propiedad de objeto actualiza es una propiedad de objeto funcional.

$actualiza \equiv seActualizaPor^-$: Establece que la propiedad de objeto actualiza es la propiedad inversa de la propiedad de objeto seActualizaPor.

Propiedades de datos

tieneComportamiento: representa la declaración de la propiedad de datos tieneComportamiento.

Restricciones y características de propiedades de datos

$\geq I tieneComportamiento \sqsubseteq (Articulo \sqcup Inventario \sqcup Institucion \sqcup Empleado)$: Establece a las clases Articulo, Institucion, Inventario y Empleado en el dominio de la propiedad de datos tieneComportamiento.

$T \sqsubseteq \forall tieneComportamiento.string$: Establece el tipo de datos string como el rango de la propiedad de objeto tieneComportamiento.

$T \sqsubseteq \leq I tieneComportamiento$: Establece que la propiedad de datos tieneComportamiento es una propiedad de datos funcional.

Como se puede observar, la definición de algunas propiedades de datos consiste en nombres de conceptos basados en el modelo SSM (Comportamiento (tieneComportamiento), Especificidad (tieneEspecificidad, Nivel de estabilidad (tieneNivel_estabilidad) y Percepción (tienePercepcion)) y permiten relacionar individuos abstractos (individuo *Valentino* de la clase

Empleado) con individuos concretos (valor “*Inestable*” de tipo string), por ejemplo, *tieneComportamiento(Valentino, “Inestable”)*. Por otra parte, existen propiedades de datos que definen atributos como: nombre, giro, edad, entre otros. Cada propiedad de datos se caracteriza por ser funcional (“ $T \sqsubseteq \leq I$ ” en las DLs y “*FunctionalDataProperty*” en OWL-DL) y tener un dominio (“ $\geq I \cup \sqsubseteq C$ ” en las DLs “*DataPropertyDomain*” en OWL-DL) y rango (“ $T \sqsubseteq \forall U.D$ ” en las DLs y “*DataPropertyRange*” en OWL-DL) definidos.

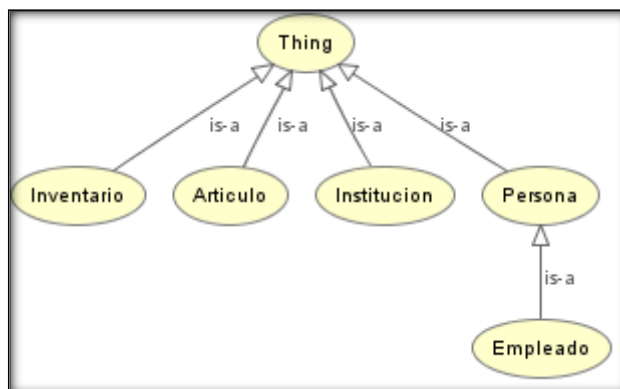


Figura A.1 Jerarquía de clases en la KB del patrón de análisis PAI.

1.2. Componente extensional

Para obtener el componente *ABox*, se utilizaron individuos (nombres de individuos abstractos N_{IA} y nombres de individuos concretos N_{IC} en las DLs) como:

$N_{IA} = \{ \text{Libro1, Valentino, CENIDET, Inventario_CENIDET} \}$

$N_{IC} = \{ \text{string, Boolean, integer, double, dateTime, long} \}$

en donde los individuos abstractos Libro1, Valentino, CENIDET e Inventario_CENIDET son aplicables a los contextos determinados en el Refinamiento de los requerimientos del patrón de análisis PAI: punto 16 descrito en (Juárez Hernández, 2016).

El detalle acerca de las aserciones de propiedades de datos que hacen uso de los individuos concretos en la *ABox*, pueden consultarse en la versión completa de la KB (OPAE) del patrón de análisis PAI que se encuentra en el directorio *Ontologías/OPAE PAI/*, almacenado en el disco que acompaña esta tesis (para mayor referencia consultar el *Anexo G. Contenido del disco*). Al igual que en la ontología OPAE, algunas de estas aserciones contemplan propiedades de datos basadas en el modelo SSM como son *Comportamiento*, *Especificidad*, *Nivel de estabilidad* y *Percepción*, entre otras.

Tabla A.2 ABox de la KB del patrón de análisis PAI (cont.)

<i>Sintaxis DL</i>	<i>Sintaxis Funcional</i>
<i>ABox = {</i>	
<i>-- Aserciones de clase --</i>	
<i>Articulo(Libro1)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro1)</i>
<i>Institucion(CENIDET)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Institucion domain-pai:CENIDET)</i>
<i>Inventario(Inventario_CENIDET)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Inventario domain-pai:Inventario_CENIDET)</i>
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Empleado domain-pai:Valentino)</i>
<i>-- Aserciones de propiedades de objeto --</i>	
<i>actualiza(Valentino, Inventario_CENIDET)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:actualiza domain-pai:Valentino domain-pai:Inventario_CENIDET)</i>
<i>seActualizaPor(Inventario_CENIDET, Valentino)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:seActualizaPor domain-pai:Inventario_CENIDET domain-pai:Valentino)</i>
<i>obtieneExistenciasDe(Valentino, Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:obtieneExistenciasDe domain-pai:Valentino domain-pai:Libro1)</i>
<i>existenciasSeObtienePor(Libro1, Valentino)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:existenciasSeObtienePor domain-pai:Libro1 domain-pai:Valentino)</i>
<i>buscaExistenciasDe(Valentino, Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:buscaExistenciasDe domain-pai:Valentino domain-pai:Libro1)</i>
<i>existenciasSeBuscaPor(Libro1, Valentino)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:existenciasSeBuscaPor domain-pai:Libro1 domain-pai:Valentino)</i>
<i>especificaTipoExistenciasDe(Valentino, Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:especificaTipoExistenciasDe domain-pai:Valentino domain-pai:Libro1)</i>
<i>tipoExistenciasSeEspecificaPor(Libro1, Valentino)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:tipoExistenciasSeEspecificaPor domain-pai:Libro1 domain-pai:Valentino)</i>
<i>registra(Valentino, Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:registra domain-pai:Valentino domain-pai:Libro1)</i>
<i>seRegistraPor(Libro1, Valentino)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:seRegistraPor domain-pai:Libro1 domain-pai:Valentino)</i>
<i>separa(Valentino, Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:separa domain-pai:Valentino domain-pai:Libro1)</i>
<i>seSeparaPor(Libro1, Valentino)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:seSeparaPor domain-pai:Libro1 domain-pai:Valentino)</i>
<i>ubica(Valentino, Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:ubica domain-pai:Valentino domain-pai:Libro1)</i>

Tabla A.2 ABox de la KB del patrón de análisis PAI (cont.)

<i>Sintaxis DL</i>	<i>Sintaxis Funcional</i>
<i>seUbicaPor(Libro1, Valentino)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:seUbicaPor domain-pai:Libro1 domain-pai:Valentino)</i>
<i>verificaExistenciaDe(Valentino, Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:verificaExistenciaDe domain-pai:Valentino domain-pai:Libro1)</i>
<i>existenciaSeVerificaPor(Libro1, Valentino)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:existenciaSeVerificaPor domain-pai:Libro1 domain-pai:Valentino)</i>
<i>laboraEn(Valentino, CENIDET)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:laboraEn domain-pai:Valentino domain-pai:CENIDET)</i>
<i>obtieneUbicacionDe(Valentino, Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:obtieneUbicacionDe domain-pai:Valentino domain-pai:Libro1)</i>
<i>ubicacionSeObtienePor(Libro1, Valentino)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:ubicacionSeObtienePor domain-pai:Libro1 domain-pai:Valentino)</i>
<i>solicitaUbicacionDe(Valentino, Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:solicitaUbicacionDe domain-pai:Valentino domain-pai:Libro1)</i>
<i>ubicacionSeSolicitaPor(Libro1, Valentino)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:ubicacionSeSolicitaPor domain-pai:Libro1 domain-pai:Valentino)</i>
<i>tieneDatosDe(CENIDET, Valentino)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:tieneDatosDe domain-pai:CENIDET domain-pai:Valentino)</i>
-- Aserciones de propiedades de datos --	
<i>tieneCantidad(Libro1, 2)</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai:tieneCantidad domain-pai:Libro1 "2"^^xsd:integer)</i>
<i>tieneComportamiento(Libro1, "estable-adaptable")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai:tieneComportamiento domain-pai:Libro1 "estable-adaptable"^^xsd:string)</i>
<i>tieneComportamiento(Valentino, "inestable")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai:tieneComportamiento domain-pai:Valentino "inestable"^^xsd:string)</i>
<i>tieneComportamiento(Inventario_CENIDET, "estable-adaptable")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai:tieneComportamiento domain-pai:Inventario_CENIDET "estable-adaptable"^^xsd:string)</i>
<i>tieneComportamiento(CENIDET, "inestable")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai:tieneComportamiento domain-pai:CENIDET "inestable"^^xsd:string)</i>
<i>tienePercepcion (Libro1, "semi-tangible")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai:tienePercepcion domain-pai:Libro1 "semi-tangible"^^xsd:string)</i>
<i>tienePercepcion (Valentino, "tangible")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai:tienePercepcion domain-pai:Valentino "tangible"^^xsd:string)</i>
<i>tienePercepcion (Inventario_CENIDET, "semi-tangible")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai:tienePercepcion domain-</i>

Tabla A.2 ABox de la KB del patrón de análisis PAI (cont.)

<i>Sintaxis DL</i>	<i>Sintaxis Funcional</i>
<i>tienePercepcion (CENIDET, "tangible")</i>	\Rightarrow <i>pai:Inventario_CENIDET "semi-tangible"^^xsd:string</i> \Rightarrow <i>DataPropertyAssertion(domain-pai: tienePercepcion domain-pai:CENIDET "tangible"^^xsd:string)</i>
<i>tieneEspecificidad (Libro1, "general")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai: tieneEspecificidad domain-pai:Libro1 "general"^^xsd:string)</i>
<i>tieneEspecificidad (Valentino, "particular")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai: tieneEspecificidad domain-pai:Valentino "particular"^^xsd:string)</i>
<i>tieneEspecificidad (Inventario_CENIDET, "general")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai: tieneEspecificidad domain-pai:Inventario_CENIDET "general"^^xsd:string)</i>
<i>tieneEspecificidad (CENIDET, "particular")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai: tieneEspecificidad domain-pai:CENIDET "particular"^^xsd:string)</i>
<i>tieneNivelEstabilidad (Libro1, "BO")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai: tieneNivelEstabilidad domain-pai:Libro1 "BO"^^xsd:string)</i>
<i>tieneNivelEstabilidad (Valentino, "IO")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai: tieneNivelEstabilidad domain-pai:Valentino "IO"^^xsd:string)</i>
<i>tieneNivelEstabilidad (Inventario_CENIDET, "BO")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai: tieneNivelEstabilidad domain-pai:Inventario_CENIDET "BO"^^xsd:string)</i>
<i>tieneNivelEstabilidad (CENIDET, "IO")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai: tieneNivelEstabilidad domain-pai:CENIDET "IO"^^xsd:string)</i>
<i>tieneEdad(Valentino, 26)</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai: tieneEdad domain-pai:Valentino "26"^^xsd:integer)</i>
<i>tieneGiro(CENIDET, "Control de inventario")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai: tieneGiro domain-pai:CENIDET "Control de inventario"^^xsd:string)</i>
<i>tieneId (Libro1, "C000001")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai: tieneId domain-pai:Libro1 "C000001"^^xsd:string)</i>
<i>tieneId (Valentino, "E000001")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai: tieneId domain-pai:Valentino "E000001"^^xsd:string)</i>
<i>tieneId (Inventario_CENIDET, "I000001")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai: tieneId domain-pai:Inventario_CENIDET "I000001"^^xsd:string)</i>
<i>tieneId (CENIDET, "000001")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai: tieneId domain-pai:CENIDET "000001"^^xsd:string)</i>
<i>tieneNombre (Libro1, "El llano en llamas")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai: tieneNombre domain-pai:Libro1 "El llano en llamas"^^xsd:string)</i>
<i>tieneNombre (Valentino, "Valentino Velázquez Olivera")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai: tieneNombre domain-pai:Valentino "Valentino Velázquez Olivera"^^xsd:string)</i>
<i>tieneNombre (Inventario_CENIDET, "Sistema de control de inventario CENIDET")</i>	\Rightarrow <i>DataPropertyAssertion(domain-pai: tieneNombre domain-pai:Inventario_CENIDET "Sistema de control</i>

Tabla A.2 ABox de la KB del patrón de análisis PAI (cont.)

<i>Sintaxis DL</i>	<i>Sintaxis Funcional</i>
<i>tieneNombre</i> (CENIDET, "Centro Nacional de Investigación y Desarrollo Tecnológico")	\Rightarrow <i>DataPropertyAssertion</i> (domain-pai: <i>tieneNombre</i> domain-pai:CENIDET " Centro Nacional de Investigación y Desarrollo Tecnológico"^^xsd:string)
<i>tieneTipo</i> (Libro1, "Componente")	\Rightarrow <i>DataPropertyAssertion</i> (domain-pai: <i>tieneTipo</i> domain-pai:Libro1 "Componente"^^xsd:string)
<i>tieneUbicacion</i> (Libro1, "En el pasillo #2, en el librero #4 en el stand #3")	\Rightarrow <i>DataPropertyAssertion</i> (domain-pai: <i>tieneUbicacion</i> domain-pai:Libro1 " En el pasillo #2, en el librero #4 en el stand #3"^^xsd:string)

Para una mejor comprensión de las descripciones lógicas presentadas en la ABox de la Tabla A.2, a continuación se describen algunos ejemplos tomados de las secciones *Aserciones de clase*, *Aserciones de propiedades de objeto* y *Aserciones de propiedades de datos* en la misma tabla:

Aserciones de clase

Articulo(Libro1): Afirma que Libro1 es un individuo de la clase Articulo.

Empleado(Valentino): Afirma que Valentino es un individuo de la clase Empleado.

Institucion(CENIDET): Afirma que CENIDET es un individuo de la clase Institucion.

Inventario(Inventario_CENIDET): Afirma que Inventario_CENIDET es un individuo de la clase Inventario.

Aserciones de propiedades de objeto

actualiza(Valentino,Inventario_CENIDET): Afirma que el empleado Valentino actualiza el inventario Inventario_CENIDET.

ubica(Valentino,Libro1): Afirma que el empleado Valentino ubica el articulo Libro1.

Aserciones de propiedades de datos

tieneComportamiento(CENIDET, "inestable"): Afirma que la institución CENIDET tiene un comportamiento inestable en términos del modelo SSM.

tieneComportamiento(Valentino, "Inestable"): Afirma que el empleado Valentino tiene un tipo de comportamiento Inestable en términos del modelo SSM.

Anexo B. Ontología OTPAE del patrón de análisis PAI

En este anexo, se presenta la última versión de la ontología OTPAE del patrón de análisis para inventarios (PAI), obtenida a través de la metodología MECOTPA.

1. Base de conocimiento

En esta sección se presentan algunas de las clases, propiedades de objeto y propiedades de objeto temporales en la KB con aspectos temporales (OTPAE) del patrón de análisis PAI. Esta ontología hereda las características y restricciones de las propiedades de objeto y de las propiedades de datos definidas en la ontología OPAE descrita en el *Anexo A. Ontología OPAE del patrón de análisis PAI* y agrega algunas otras pertenecientes al dominio de la ontología OWL-Time. De la ontología OWL-Time, solo se hace referencia a algunas características y restricciones asociadas a sus propiedades de objeto.

Una versión completa de la KB se encuentra en el directorio *Ontologías/OTPAE PAI/*, almacenado en el disco que acompaña esta tesis (para mayor referencia consultar el *Anexo G. Contenido del disco*). Debido a que en una ontología temporal puede llegar a existir una extensa proliferación de aserciones de clase y de roles (individuos), se elaboraron dos ontologías temporales por cada RF para facilitar su lectura y comprensión. Estas ontologías describen el comportamiento dinámico considerando casos de éxito y una combinación de casos de éxito y fracaso/alternativo, respectivamente.

La *TBox* (o componente intencional) y la *ABox* (componente extensional) de la KB, se describen bajo el poder de expresividad de la DL *ALCOIN(D)*. En ambos componentes se incluyen las equivalencias correspondientes en la sintaxis funcional de OWL-DL.

1.1. Componente intencional

Para obtener el componente *TBox*, fue necesario modificar algunos axiomas de la ontología OPAE del patrón de análisis PAI, para incluir en su definición los conceptos temporales de la ontología OWL-Time. En su construcción se utilizaron:

1. Clases (en OWL-DL o nombres de conceptos N_C en las DLs):
 - *OWL-Time*: {dynamicClasses, DateTimeDescription, DayOfWeak, DurationDescription, Year, Event, TemporalEntit, Instant, Interval, ProperInterval, DateTimeInterval, TemporalUnit, TimeZone}
 - *Dominio del patrón de análisis PAI*: {staticClasses, Artículo, Artículo, Institucion, Inventario, Persona, Empleado}

a partir de las cuales se generó la jerarquía de clases (nombres de concepto N_C en las DLs) de la Figura B.1, que modela el conocimiento central del patrón de análisis PAI.

2. Propiedades de objeto estáticas (en OWL-DL o nombres de roles abstractos N_{RA} en las DLs):
 - *OWL-Time* : { dynamicOP, after, before, intervalBefore, dayOfWeak, during, hasBeginning, hasDateTimeDescription, hasEnd, inDateTime, inside, intervalAfter, intervalMeets, intervalMeets, intervalMetBy, overlaps, equals, intervalContains,

intervalDuring, intervalEquals, intervalFinishes, intervalOverlappedBy, intervalOverlaps, intervalStartedBy intervalStarts, timeZone, unitType }

- *Dominio del patrón de análisis PAI:* { staticOP, actualiza, existenciasSeObtienePor, existenciasSeBuscaPor, seActualizaPor, especificaTipoExistenciasDe, seRegistraPor, seSeparaPor, seUbicaPor, existenciaSeVerificaPor, laboraEn, obtieneExistenciasDe, obtieneUbicacionDe, registra, separa, solicitaUbicacionDe, tieneDatosDe, tipoExistenciasSeEspecificaPor, ubica, ubicacionSeObtienePor, ubicacionSeSolicitaPor, buscaExistenciasDe, verificaExistenciaDe }
3. Propiedades de objeto temporales
 - *OWL-Time y dominio del patrón de análisis PAI:* { dynamicOP, participatesIn, actualiza, existenciasSeObtienePor, existenciasSeBuscaPor, seActualizaPor, especificaTipoExistenciasDe, seRegistraPor, seSeparaPor, seUbicaPor, existenciaSeVerificaPor, obtieneExistenciasDe, obtieneUbicacionDe, registra, separa, solicitaUbicacionDe, tipoExistenciasSeEspecificaPor, ubica, ubicacionSeObtienePor, ubicacionSeSolicitaPor, buscaExistenciasDe, verificaExistenciaDe }
 4. Propiedades de datos estáticas (en OWL-DL o nombres de roles concretos N_{RC} en las DLs):
 - *OWL-Time:* { dynamicDP, inXSDDateTime }
 - *Dominio del patrón de análisis PAI:* { staticDP, tieneCantidad, tieneComportamiento, tieneEdad, tieneEspecificidad, tieneGiro, tieneId, tieneNivelEstabilidad, tieneNombre, tienePercepcion, tieneTipo, tieneUbicacion }
 5. Propiedades de datos temporales
 - Ninguna ($\{\emptyset\}$).
 6. Espacios de nombres (complemento a los prefijos descritos en el *Anexo A. Ontología OPAE del patrón de análisis PAI*):
 - Prefix(swrl:=<http://www.w3.org/2003/11/swrl#>)

Espacio de nombres para el lenguaje de reglas de la web semántica (SWRL).
 - Prefix(time:=<http://www.w3.org/2006/time#>)

Espacio de nombres para los términos de la ontología OWL-Time.
 - Prefix(swrlb:=<http://www.w3.org/2003/11/swrlb#>)

Espacio de nombres de Incorporaciones de SWRL.
 - Prefix(timeZone:=<http://www.w3.org/2006/timezone#>)

Ontología de zona horaria.

- Prefix(domain-pai:=<https://www.cenidet.edu.mx/SoftwareEngineering/AnalysisPatterns/TemporalOntology/PAI#>)
Vocabulario en el dominio del patrón de análisis PAI (aspectos estáticos).
- Prefix(timeEvents:=<http://www.intelligence.tuc.gr/2011/timeEvents#>)
Versión estable de la ontología OWL-Time de conceptos temporales proveída por el plugin Chronos.

Las propiedades de objeto temporales fueron utilizadas en la representación del comportamiento expresado en los requerimientos funcionales del patrón de análisis PAI.

Debido a que cada una de las propiedades de objeto temporales cuenta con características tales como funcional o inversa funcional, en la Tabla B.3 se describen las reglas SWRL que las representan. Estas reglas son el resultado del proceso de conversión de propiedades de objeto a propiedades de objeto temporales realizado por el plugin Chronos en el editor de ontologías Protégé.

Tabla B.3 Reglas SWRL en características funcional e inversa funcional de propiedades de objeto en el patrón de análisis PAI (cont.)

Propiedad de objeto	Característica	Regla SWRL
<i>actualiza</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), actualiza(?e1, ?y1), actualiza(?e2, ?y2), actualiza(?x, ?e1), actualiza(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>buscaExistenciasDe</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), buscaExistenciasDe(?e1, ?y), buscaExistenciasDe(?e2, ?y), buscaExistenciasDe(?x1, ?e1), buscaExistenciasDe(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>seActualizaPor</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), seActualizaPor(?e1, ?y), seActualizaPor(?e2, ?y), seActualizaPor(?x1, ?e1), seActualizaPor(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>especificaTipoExistenciasDe</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), especificaTipoExistenciasDe(?e1, ?y), especificaTipoExistenciasDe(?e2, ?y), especificaTipoExistenciasDe(?x1, ?e1), especificaTipoExistenciasDe(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>seRegistraPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), seRegistraPor(?e1, ?y1), seRegistraPor(?e2, ?y2), seRegistraPor(?x, ?e1), seRegistraPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>seSeparaPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2),</i>

Tabla B.3 Reglas SWRL en características funcional e inversa funcional de propiedades de objeto en el patrón de análisis PAI (cont.)

Propiedad de objeto	Característica	Regla SWRL
		<i>overlaps(?i1, ?i2), seSeparaPor(?e1, ?y1), seSeparaPor(?e2, ?y2), seSeparaPor(?x, ?e1), seSeparaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>seUbicaPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), seUbicaPor(?e1, ?y1), seUbicaPor(?e2, ?y2), seUbicaPor(?x, ?e1), seUbicaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>existenciaEsVerificadoPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), existenciaSeVerificaPor(?e1, ?y1), existenciaSeVerificaPor(?e2, ?y2), existenciaSeVerificaPor(?x, ?e1), existenciaSeVerificaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>existenciasSeBuscaPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), existenciasSeBuscaPor(?e1, ?y1), existenciasSeBuscaPor(?e2, ?y2), existenciasSeBuscaPor(?x, ?e1), existenciasSeBuscaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>existenciasSeObtienePor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), existenciasSeObtienePor(?e1, ?y1), existenciasSeObtienePor(?e2, ?y2), existenciasSeObtienePor(?x, ?e1), existenciasSeObtienePor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>laboraEn</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), laboraEn(?e1, ?y1), laboraEn(?e2, ?y2), laboraEn(?x, ?e1), laboraEn(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>obtieneExistenciasDe</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), obtieneExistenciasDe(?e1, ?y), obtieneExistenciasDe(?e2, ?y), obtieneExistenciasDe(?x1, ?e1), obtieneExistenciasDe(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>obtieneUbicacionDe</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), obtieneUbicacionDe(?e1, ?y), obtieneUbicacionDe(?e2, ?y), obtieneUbicacionDe(?x1, ?e1), obtieneUbicacionDe(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>registra</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), registra(?e1, ?y), registra(?e2, ?y), registra(?x1, ?e1), registra(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>separa</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), separa(?e1, ?y), separa(?e2, ?y), separa(?x1, ?e1), separa(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>solicitaUbicacionDe</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), solicitaUbicacionDe(?e1, ?y1), solicitaUbicacionDe(?e2, ?y2), solicitaUbicacionDe(?x, ?e1), solicitaUbicacionDe(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>tipoExistenciasSeEspecificaPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2),</i>

Tabla B.3 Reglas SWRL en características funcional e inversa funcional de propiedades de objeto en el patrón de análisis PAI (cont.)

Propiedad de objeto	Característica	Regla SWRL
		<i>overlaps(?i1, ?i2), tipoExistenciasSeEspecificaPor(?e1, ?y1), tipoExistenciasSeEspecificaPor(?e2, ?y2), tipoExistenciasSeEspecificaPor(?x, ?e1), tipoExistenciasSeEspecificaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>ubica</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), ubica(?e1, ?y), ubica(?e2, ?y), ubica(?x1, ?e1), ubica(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>ubicacionSeObtienePor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), ubicacionSeObtienePor(?e1, ?y1), ubicacionSeObtienePor(?e2, ?y2), ubicacionSeObtienePor(?x, ?e1), ubicacionSeObtienePor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>ubicacionSeSolicitaPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), ubicacionSeSolicitaPor(?e1, ?y), ubicacionSeSolicitaPor(?e2, ?y), ubicacionSeSolicitaPor(?x1, ?e1), ubicacionSeSolicitaPor(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>verificaExistenciaDe</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), verificaExistenciaDe(?e1, ?y), verificaExistenciaDe(?e2, ?y), verificaExistenciaDe(?x1, ?e1), verificaExistenciaDe(?x2, ?e2) -> SameAs (?x1, ?x2)</i>

El componente *TBox*, resultado de la inclusión de aspectos temporales en la KB de la ontología OPAE del patrón de análisis PAI, se describe en la Tabla B.4.

Tabla B.4 TBox de la KB del patrón de análisis PAI con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
<i>TBox = {</i>	
<i>-- Clases --</i>	
<i>T</i>	\Rightarrow <i>Declaration(Class(owl:Thing))</i>
<i>staticClasses \sqsubseteq T</i>	\Rightarrow <i>Declaration(Class(domain-pai:staticClasses))</i> <i>SubClassOf(domain-pai:staticClasses owl:Thing)</i>
<i>dynamicClasses \sqsubseteq T</i>	\Rightarrow <i>Declaration(Class(domain-pai:dynamicClasses))</i> <i>SubClassOf(domain-pai:dynamicClasses owl:Thing)</i>
<i>Articulo \sqsubseteq staticClasses</i>	\Rightarrow <i>Declaration(Class(domain-pai:Articulo))</i> <i>SubClassOf(domain-pai:Articulo domain-pai:staticClasses)</i>
<i>Institucion \sqsubseteq staticClasses</i>	\Rightarrow <i>Declaration(Class(domain-pai:Institucion))</i> <i>SubClassOf(domain-pai:Institucion domain-pai:staticClasses)</i>
<i>Inventario \sqsubseteq staticClasses</i>	\Rightarrow <i>Declaration(Class(domain-pai:Inventario))</i> <i>SubClassOf(domain-pai:Inventario domain-pai:staticClasses)</i>

Tabla B.4 TBox de la KB del patrón de análisis PAI con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
$Persona \sqsubseteq staticClasses$	$\Rightarrow Declaration(Class(domain-pai:Persona))$ $SubClassOf(domain-pai:Persona domain-pai:staticClasses)$
$Articulo \sqsubseteq staticClasses$	$\Rightarrow Declaration(Class(domain-pai:Articulo))$ $SubClassOf(domain-pai:Articulo domain-pai:staticClasses)$
$Empleado \sqsubseteq Persona$	$\Rightarrow Declaration(Class(domain-pai:Empleado))$ $SubClassOf(domain-pai:Empleado domain-pai:Persona)$
$Event \sqsubseteq dynamicClasses$	$\Rightarrow Declaration(Class(timeEvents:Event))SubClassOf(timeEvents:Event domain-pai:dynamicClasses)$
$TemporalEntity \sqsubseteq dynamicClasses$	$\Rightarrow Declaration(Class(time:TemporalEntity))$ $SubClassOf(time:TemporalEntity domain-pai:dynamicClasses)$
$Instant \sqsubseteq TemporalEntity$	$\Rightarrow Declaration(Class(time:Instant))$ $SubClassOf(time:Instant time:TemporalEntity)$
$Interval \sqsubseteq TemporalEntity$	$\Rightarrow Declaration(Class(time:Interval))$ $SubClassOf(time:Interval time:TemporalEntity)$
$ProperInterval \sqsubseteq Interval$	$\Rightarrow Declaration(Class(time:ProperInterval))$ $SubClassOf(time:ProperInterval time:Interval)$
$DateTimeInterval \sqsubseteq ProperInterval$	$\Rightarrow Declaration(Class(time:DateTimeInterval))$ $SubClassOf(time:DateTimeInterval time:ProperInterval)$
-- Clases disjuntas --	
$ProperInterval \sqcap Instant \sqsubseteq \perp$	$\Rightarrow DisjointClasses(time:Instant time:ProperInterval)$
$Articulo \sqcap Inventario \sqcap Empleado \sqcap Institucion \sqsubseteq \perp$	$\Rightarrow DisjointClasses(domain-pai:Articulo domain-pai:Inventario domain-pai:Empleado domain-pai:Institucion)$
-- Restricciones existenciales y universales --	
$Empleado \sqsubseteq \exists separa.(Event \exists separa.Articulo)$	$\Rightarrow SubClassOf(domain-pai:Empleado$ $ObjectSomeValuesFrom(domain-pai:separa$ $ObjectIntersectionOf(ObjectSomeValuesFrom(domain-pai:separa Domain-pai:Articulo timeEvents:Event)))$
$Empleado \sqsubseteq \exists registra.(Event \exists registra.Articulo)$	$\Rightarrow SubClassOf(domain-pai:Empleado$ $ObjectSomeValuesFrom(domain-pai:registra$ $ObjectIntersectionOf(ObjectSomeValuesFrom(domain-pai:registra Domain-pai:Articulo timeEvents:Event)))$
$Empleado \sqsubseteq \exists buscaExistenciasDe.(Event \exists buscaExistenciasDe.Articulo)$	$\Rightarrow SubClassOf(domain-pai:Empleado$ $ObjectSomeValuesFrom(domain-pai:buscaExistenciasDe$ $ObjectIntersectionOf(ObjectSomeValuesFrom(domain-pai:buscaExistenciasDe Domain-pai:Articulo timeEvents:Event)))$
$Empleado \sqsubseteq \exists verificaExistenciaDe.(Event \exists verificaExistenciaDe.Articulo)$	$\Rightarrow SubClassOf(domain-pai:Empleado$ $ObjectSomeValuesFrom(domain-pai:verificaExistenciaDe$ $ObjectIntersectionOf(ObjectSomeValuesFrom(domain-pai:verificaExistenciaDe Domain-pai:Articulo timeEvents:Event)))$
$Empleado \sqsubseteq \exists obtieneUbicacionDe.(Event$	$\Rightarrow SubClassOf(domain-pai:Empleado$ $ObjectSomeValuesFrom(domain-pai:obtieneUbicacionDe$

Tabla B.4 TBox de la KB del patrón de análisis PAI con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
$\exists \text{obtieneUbicacionDe}.\text{Articulo}$	$\text{ObjectIntersectionOf}(\text{ObjectSomeValuesFrom}(\text{domain-pai: obtieneUbicacionDe Domain-pai:Articulo timeEvents:Event}))$
$\text{Empleado} \sqsubseteq \exists \text{especificaTipoExistenciasDe}.\text{Event}$ $\exists \text{especificaTipoExistenciasDe}.\text{Articulo}$	$\Rightarrow \text{SubClassOf}(\text{domain-pai:Empleado ObjectSomeValuesFrom}(\text{domain-pai: especificaTipoExistenciasDe ObjectIntersectionOf}(\text{ObjectSomeValuesFrom}(\text{domain-pai: especificaTipoExistenciasDe Domain-pai:Articulo timeEvents:Event}))))$
$\text{Empleado} \sqsubseteq \exists \text{obtieneExistenciasDe}.\text{Event}$ $\exists \text{obtieneExistenciasDe}.\text{Articulo}$	$\Rightarrow \text{SubClassOf}(\text{domain-pai:Empleado ObjectSomeValuesFrom}(\text{domain-pai: obtieneExistenciasDe ObjectIntersectionOf}(\text{ObjectSomeValuesFrom}(\text{domain-pai: obtieneExistenciasDe Domain-pai:Articulo timeEvents:Event}))))$
$\text{Empleado} \sqsubseteq \exists \text{ubica}.\text{Event}$ $\exists \text{ubica}.\text{Articulo}$	$\Rightarrow \text{SubClassOf}(\text{domain-pai:Empleado ObjectSomeValuesFrom}(\text{domain-pai: ubica ObjectIntersectionOf}(\text{ObjectSomeValuesFrom}(\text{domain-pai: ubica Domain-pai:Articulo timeEvents:Event}))))$
$\text{Empleado} \sqsubseteq \exists \text{solicitaUbicacionDe}.\text{Event}$ $\exists \text{solicitaUbicacionDe}.\text{Articulo}$	$\Rightarrow \text{SubClassOf}(\text{domain-pai:Empleado ObjectSomeValuesFrom}(\text{domain-pai: solicitaUbicacionDe ObjectIntersectionOf}(\text{ObjectSomeValuesFrom}(\text{domain-pai: solicitaUbicacionDe Domain-pai:Articulo timeEvents:Event}))))$
$\text{Institucion} \sqsubseteq \exists \text{tieneDatosDe}.\text{Event}$ $\exists \text{tieneDatosDe}.\text{Empleado}$	$\Rightarrow \text{SubClassOf}(\text{domain-pai:Institucion ObjectSomeValuesFrom}(\text{domain-pai: tieneDatosDe ObjectIntersectionOf}(\text{ObjectSomeValuesFrom}(\text{domain-pai: tieneDatosDe domain-pai:Empleado timeEvents:Event}))))$
$\text{Articulo} \sqsubseteq \forall \text{seSeparaPor}.\text{Event}$ $\forall \text{seSeparaPor}.\text{Empleado}$	$\Rightarrow \text{SubClassOf}(\text{domain-pai:Articulo ObjectAllValuesFrom}(\text{domain-pai: seSeparaPor ObjectIntersectionOf}(\text{ObjectAllValuesFrom}(\text{domain-pai: seSeparaPor domain-pai:Empleado timeEvents:Event}))))$
$\text{Articulo} \sqsubseteq \forall \text{seRegistraPor}.\text{Event}$ $\forall \text{seRegistraPor}.\text{Empleado}$	$\Rightarrow \text{SubClassOf}(\text{domain-pai:Articulo ObjectSomeValuesFrom}(\text{domain-pai: seRegistraPor ObjectIntersectionOf}(\text{ObjectSomeValuesFrom}(\text{domain-pai: seRegistraPor domain-pai:Empleado timeEvents:Event}))))$
$\text{Empleado} \sqsubseteq \forall \text{actualiza}.\text{Event}$ $\forall \text{actualiza}.\text{Inventario}$	$\Rightarrow \text{SubClassOf}(\text{domain-pai:Empleado ObjectAllValuesFrom}(\text{domain-pai: actualiza ObjectIntersectionOf}(\text{ObjectAllValuesFrom}(\text{domain-pai: actualiza domain-pai:Inventario timeEvents:Event}))))$
$\text{Empleado} \sqsubseteq \forall \text{laboraEn}.\text{Event}$ $\forall \text{laboraEn}.\text{Institucion}$	$\Rightarrow \text{SubClassOf}(\text{domain-pai:Empleado ObjectAllValuesFrom}(\text{domain-pai: laboraEn ObjectIntersectionOf}(\text{ObjectAllValuesFrom}(\text{domain-pai: laboraEn domain-pai:Institucion timeEvents:Event}))))$
-- Propiedades de objeto temporales -- dynamicOP	$\Rightarrow \text{Declaration}(\text{ObjectProperty}(\text{domain-pai:dynamicOP}))$

Tabla B.4 TBox de la KB del patrón de análisis PAI con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
<i>during</i> \sqsubseteq <i>dynamicOP</i>	\Rightarrow <i>Declaration(ObjectProperty(time:during))</i> <i>SubObjectPropertyOf(time:during domain-pai:dynamicOP)</i>
<i>hasBeginning</i> \sqsubseteq <i>dynamicOP</i>	\Rightarrow <i>Declaration(ObjectProperty(time:hasBeginning))</i> <i>SubObjectPropertyOf(time:hasBeginning domain-pai:dynamicOP)</i>
<i>hasEnd</i> \sqsubseteq <i>dynamicOP</i>	\Rightarrow <i>Declaration(ObjectProperty(time:hasEnd))</i> <i>SubObjectPropertyOf(time:hasEnd domain-pai:dynamicOP)</i>
<i>participatesIn</i> \sqsubseteq <i>dynamicOP</i>	\Rightarrow <i>Declaration(ObjectProperty(timeEvents:participatesIn))</i> <i>SubObjectPropertyOf(timeEvents:participatesIn domain-pai:dynamicOP)</i>
<i>actualiza</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-pai:actualiza timeEvents:participatesIn)</i>
<i>buscaExistenciasDe</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-pai: buscaExistenciasDe timeEvents:participatesIn)</i>
<i>seActualizaPor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-pai: seActualizaPor timeEvents:participatesIn)</i>
<i>especificaTipoExistenciasDe</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-pai: especificaTipoExistenciasDe timeEvents:participatesIn)</i>
<i>seRegistraPor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-pai: seRegistraPor timeEvents:participatesIn)</i>
<i>seSeparaPor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-pai: seSeparaPor timeEvents:participatesIn)</i>
<i>seUbicaPor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-pai: seUbicaPor timeEvents:participatesIn)</i>
<i>existenciasEsVerificadaPor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-pai: existenciasEsVerificadaPor timeEvents:participatesIn)</i>
<i>existenciasSeBuscaPor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-pai: existenciasSeBuscaPor timeEvents:participatesIn)</i>
<i>existenciasSeObtienePor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-pai: existenciasSeObtienePor timeEvents:participatesIn)</i>
<i>laboraEn</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-pai: laboraEn timeEvents:participatesIn)</i>
<i>obtieneExistenciasDe</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-pai: obtieneExistenciasDe timeEvents:participatesIn)</i>
<i>obtieneUbicacionDe</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-pai: obtieneUbicacionDe timeEvents:participatesIn)</i>
<i>registra</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-pai: registra timeEvents:participatesIn)</i>
<i>separa</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-pai: separa timeEvents:participatesIn)</i>
<i>solicitaUbicacionDe</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-pai: solicitaUbicacionDe timeEvents:participatesIn)</i>

Tabla B.4 TBox de la KB del patrón de análisis PAI con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
$tipoExistenciasSeEspecificaPor \sqsubseteq$	$timeEvents:participatesIn$
$participatesIn$	$\Rightarrow SubObjectPropertyOf(domain-pai:$
$ubica \sqsubseteq participatesIn$	$tipoExistenciasSeEspecificaPor\ timeEvents:participatesIn)$
	$\Rightarrow SubObjectPropertyOf(domain-pai: ubica$
	$timeEvents:participatesIn)$
$ubicacionSeObtienePor \sqsubseteq participatesIn$	$\Rightarrow SubObjectPropertyOf(domain-pai: ubicacionSeObtienePor$
	$timeEvents:participatesIn)$
$ubicacionSeSolicitaPor \sqsubseteq participatesIn$	$\Rightarrow SubObjectPropertyOf(domain-pai: ubicacionSeSolicitaPor$
	$timeEvents:participatesIn)$
$verificaExistenciaDe \sqsubseteq participatesIn$	$\Rightarrow SubObjectPropertyOf(domain-pai: verificaExistenciaDe$
	$timeEvents:participatesIn)$
-- Restricciones y características de propiedades de objeto temporales--	
$\exists actualiza.T \sqsubseteq (Event \sqcup Empleado)$	$\Rightarrow ObjectPropertyDomain(domain-pai:actualiza$
	$ObjectUnionOf(domain-pai:Empleado\ timeEvents:Event))$
$T \sqsubseteq \forall actualiza.(Event \sqcup Inventario)$	$\Rightarrow ObjectPropertyRange(domain-pai:actualiza$
	$ObjectUnionOf(domain-pai:Inventario\ timeEvents:Event))$
$T \sqsubseteq \leq_1 actualiza$	$\Rightarrow FunctionalObjectProperty(domain-pai:actualiza)$
$\exists seActualizaPor.T \sqsubseteq (Event \sqcup$	$\Rightarrow ObjectPropertyDomain(domain-pai: seActualizaPor$
$Inventario)$	$ObjectUnionOf(domain-pai:Inventario\ timeEvents:Event))$
$T \sqsubseteq \forall seActualizaPor.(Event \sqcup$	$\Rightarrow ObjectPropertyRange(domain-pai: seActualizaPor$
$Empleado)$	$ObjectUnionOf(domain-pai:Empleado\ timeEvents:Event))$
$T \sqsubseteq \leq_1 seActualizaPor^-$	$\Rightarrow InverseFunctionalObjectProperty(domain-$
	$pai:seActualizaPor)$
$\exists obtieneExistenciasDe.T \sqsubseteq (Event \sqcup$	$\Rightarrow ObjectPropertyDomain(domain-pai: obtieneExistenciasDe$
$Empleado)$	$ObjectUnionOf(domain-pai:Empleado\ timeEvents:Event))$
$T \sqsubseteq \forall obtieneExistenciasDe.(Event \sqcup$	$\Rightarrow ObjectPropertyRange(domain-pai: obtieneExistenciasDe$
$Articulo)$	$ObjectUnionOf(Domain-pai:Articulo\ timeEvents:Event))$
$\exists cantidadExistenciasSeObtienePor.T \sqsubseteq$	$\Rightarrow ObjectPropertyDomain(domain-pai:$
$(Event \sqcup Articulo)$	$cantidadExistenciasSeObtienePor\ ObjectUnionOf(Domain-$
	$pai:Articulo\ timeEvents:Event))$
$T \sqsubseteq \forall$	$\Rightarrow ObjectPropertyRange(domain-pai:$
$cantidadExistenciasSeObtienePor.(Event$	$cantidadExistenciasSeObtienePor\ ObjectUnionOf(domain-$
$\sqcup Empleado)$	$pai:Empleado\ timeEvents:Event))$
$\exists buscaExistenciasDe.T \sqsubseteq (Event \sqcup$	$\Rightarrow ObjectPropertyDomain(domain-pai: buscaExistenciasDe$
$Empleado)$	$ObjectUnionOf(domain-pai:Empleado\ timeEvents:Event))$
$T \sqsubseteq \forall buscaExistenciasDe.(Event \sqcup$	$\Rightarrow ObjectPropertyRange(domain-pai: buscaExistenciasDe$
$Articulo)$	$ObjectUnionOf(Domain-pai:Articulo\ timeEvents:Event))$
$\exists existenciasSeBuscaPor.T \sqsubseteq (Event \sqcup$	$\Rightarrow ObjectPropertyDomain(domain-pai: existenciasSeBuscaPor$
$Articulo)$	$ObjectUnionOf(Domain-pai:Articulo\ timeEvents:Event))$
$T \sqsubseteq \forall existenciasSeBuscaPor.(Event \sqcup$	$\Rightarrow ObjectPropertyRange(domain-pai: existenciasSeBuscaPor$
$Empleado)$	$ObjectUnionOf(domain-pai:Empleado\ timeEvents:Event))$
$\exists especificaTipoExistenciasDe.T \sqsubseteq$	$\Rightarrow ObjectPropertyDomain(domain-pai:$
$(Event \sqcup Empleado)$	$especificaTipoExistenciasDe\ ObjectUnionOf(domain-$
	$pai:Empleado\ timeEvents:Event))$
$T \sqsubseteq \forall$	$\Rightarrow ObjectPropertyRange(domain-pai:$

Tabla B.4 TBox de la KB del patrón de análisis PAI con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
$especificaTipoExistenciasDe.(Event \sqcup Artículo)$	$especificaTipoExistenciasDe \text{ ObjectUnionOf}(\text{Domain-pai:Articulo } timeEvents:Event))$
$\exists \text{ tipoExistenciasSeEspecificaPor}.T \sqsubseteq (Event \sqcup Artículo)$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: } \text{tipoExistenciasSeEspecificaPor } \text{ObjectUnionOf}(\text{Domain-pai:Articulo } timeEvents:Event))$
$T \sqsubseteq \forall$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai:}$
$\text{tipoExistenciasSeEspecificaPor}.(Event \sqcup Empleado)$	$\text{tipoExistenciasSeEspecificaPor } \text{ObjectUnionOf}(\text{domain-pai:Empleado } timeEvents:Event))$
$\exists \text{ registra}.T \sqsubseteq (Event \sqcup Empleado)$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: } \text{registra } \text{ObjectUnionOf}(\text{domain-pai:Empleado } timeEvents:Event))$
$T \sqsubseteq \forall \text{ registra}.(Event \sqcup Artículo)$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: } \text{registra } \text{ObjectUnionOf}(\text{domain-pai:Articulo } timeEvents:Event))$
$T \sqsubseteq \leq_1 \text{ registra}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-pai:registra})$
$\exists \text{ seRegistraPor}.T \sqsubseteq (Event \sqcup Artículo)$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: } \text{seRegistraPor } \text{ObjectUnionOf}(\text{domain-pai:Articulo } timeEvents:Event))$
$T \sqsubseteq \forall \text{ seRegistraPor}.(Event \sqcup Empleado)$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: } \text{seRegistraPor } \text{ObjectUnionOf}(\text{domain-pai:Empleado } timeEvents:Event))$
$T \sqsubseteq \leq_1 \text{ seRegistraPor}$	$\Rightarrow \text{FunctionalObjectProperty}(\text{domain-pai:seRegistraPor})$
$\exists \text{ separa}.T \sqsubseteq (Event \sqcup Empleado)$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: } \text{separa } \text{ObjectUnionOf}(\text{domain-pai:Empleado } timeEvents:Event))$
$T \sqsubseteq \forall \text{ separa}.(Event \sqcup Artículo)$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: } \text{separa } \text{ObjectUnionOf}(\text{Domain-pai:Articulo } timeEvents:Event))$
$T \sqsubseteq \leq_1 \text{ separa}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-pai:separa})$
$\exists \text{ seSeparaPor}.T \sqsubseteq (Event \sqcup Artículo)$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: } \text{seSeparaPor } \text{ObjectUnionOf}(\text{domain-pai:Articulo } timeEvents:Event))$
$T \sqsubseteq \forall \text{ seSeparaPor}.(Event \sqcup Empleado)$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: } \text{seSeparaPor } \text{ObjectUnionOf}(\text{domain-pai:Empleado } timeEvents:Event))$
$T \sqsubseteq \leq_1 \text{ seSeparaPor}$	$\Rightarrow \text{FunctionalObjectProperty}(\text{domain-pai:seSeparaPor})$
$\exists \text{ ubica}.T \sqsubseteq (Event \sqcup Empleado)$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: } \text{ubica } \text{ObjectUnionOf}(\text{domain-pai:Empleado } timeEvents:Event))$
$T \sqsubseteq \forall \text{ ubica}.(Event \sqcup Artículo)$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: } \text{ubica } \text{ObjectUnionOf}(\text{Domain-pai:Articulo } timeEvents:Event))$
$\exists \text{ seUbicaPor}.T \sqsubseteq (Event \sqcup Artículo)$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: } \text{seUbicaPor } \text{ObjectUnionOf}(\text{Domain-pai:Articulo } timeEvents:Event))$
$T \sqsubseteq \forall \text{ seUbicaPor}.(Event \sqcup Empleado)$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: } \text{seUbicaPor } \text{ObjectUnionOf}(\text{domain-pai:Empleado } timeEvents:Event))$
$\exists \text{ verificaExistenciaDe}.T \sqsubseteq (Event \sqcup Empleado)$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: } \text{verificaExistenciaDe } \text{ObjectUnionOf}(\text{domain-pai:Empleado } timeEvents:Event))$
$T \sqsubseteq \forall \text{ verificaExistenciaDe}.(Event \sqcup Artículo)$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: } \text{verificaExistenciaDe } \text{ObjectUnionOf}(\text{Domain-pai:Articulo } timeEvents:Event))$
$\exists \text{ existenciaSeVerificaPor}.T \sqsubseteq (Event \sqcup Artículo)$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: } \text{existenciaSeVerificaPor } \text{ObjectUnionOf}(\text{Domain-pai:Articulo } timeEvents:Event))$
$T \sqsubseteq \forall \text{ existenciaSeVerificaPor}.(Event \sqcup Empleado)$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: } \text{existenciaSeVerificaPor } \text{ObjectUnionOf}(\text{domain-pai:Empleado } timeEvents:Event))$

Tabla B.4 TBox de la KB del patrón de análisis PAI con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
$\exists \text{obtieneUbicacionDe}.T \sqsubseteq (\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: obtieneUbicacionDe} \text{ObjectUnionOf}(\text{domain-pai:Empleado} \text{timeEvents:Event}))$
$T \sqsubseteq \forall \text{obtieneUbicacionDe}.(\text{Event} \sqcup \text{Articulo})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: obtieneUbicacionDe} \text{ObjectUnionOf}(\text{Domain-pai:Articulo} \text{timeEvents:Event}))$
$\exists \text{ubicacionSeObtienePor}.T \sqsubseteq (\text{Event} \sqcup \text{Articulo})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: ubicacionSeObtienePor} \text{ObjectUnionOf}(\text{Domain-pai:Articulo} \text{timeEvents:Event}))$
$T \sqsubseteq \forall \text{ubicacionSeObtienePor}.(\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: ubicacionSeObtienePor} \text{ObjectUnionOf}(\text{domain-pai:Empleado} \text{timeEvents:Event}))$
$\exists \text{solicitaUbicacionDe}.T \sqsubseteq (\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: solicitaUbicacionDe} \text{ObjectUnionOf}(\text{domain-pai:Empleado} \text{timeEvents:Event}))$
$T \sqsubseteq \forall \text{solicitaUbicacionDe}.(\text{Event} \sqcup \text{Articulo})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: solicitaUbicacionDe} \text{ObjectUnionOf}(\text{Domain-pai:Articulo} \text{timeEvents:Event}))$
$\exists \text{ubicacionSeSolicitaPor}.T \sqsubseteq (\text{Event} \sqcup \text{Articulo})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: ubicacionSeSolicitaPor} \text{ObjectUnionOf}(\text{Domain-pai:Articulo} \text{timeEvents:Event}))$
$T \sqsubseteq \forall \text{ubicacionSeSolicitaPor}.(\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: ubicacionSeSolicitaPor} \text{ObjectUnionOf}(\text{domain-pai:Empleado} \text{timeEvents:Event}))$
$\exists \text{tieneDatosDe}.T \sqsubseteq (\text{Event} \sqcup \text{Institucion})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: tieneDatosDe} \text{ObjectUnionOf}(\text{domain-pai:Institucion} \text{timeEvents:Event}))$
$T \sqsubseteq \forall \text{tieneDatosDe}.(\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: tieneDatosDe} \text{ObjectUnionOf}(\text{domain-pai:Empleado} \text{timeEvents:Event}))$
$\exists \text{laboraEn}.T \sqsubseteq (\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-pai: laboraEn} \text{ObjectUnionOf}(\text{domain-pai:Empleado} \text{timeEvents:Event}))$
$T \sqsubseteq \forall \text{laboraEn}.(\text{Event} \sqcup \text{Institucion})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-pai: laboraEn} \text{ObjectUnionOf}(\text{domain-pai:Institucion} \text{timeEvents:Event}))$
$\text{actualiza} \equiv \text{seActualizaPor}^-$	$\Rightarrow \text{InverseObjectProperties}(\text{domain-pai:actualiza} \text{domain-pai:seActualizaPor})$
$\text{obtieneExistenciasDe} \equiv \text{existenciasSeObtienePor}^-$	$\Rightarrow \text{InverseObjectProperties}(\text{domain-pai: obtieneExistenciasDe} \text{domain-pai: existenciasSeObtienePor})$
$\text{buscaExistenciasDe} \equiv \text{existenciasSeBuscaPor}^-$	$\Rightarrow \text{InverseObjectProperties}(\text{domain-pai: buscaExistenciasDe} \text{domain-pai: existenciasSeBuscaPor})$
$\text{especificaTipoExistenciasDe} \equiv \text{tipoExistenciasSeEspecificaPor}^-$	$\Rightarrow \text{InverseObjectProperties}(\text{domain-pai: especificaTipoExistenciasDe} \text{domain-pai: tipoExistenciasSeEspecificaPor})$
$\text{registra} \equiv \text{seRegistraPor}^-$	$\Rightarrow \text{InverseObjectProperties}(\text{domain-pai: registra} \text{domain-pai: seRegistraPor})$
$\text{separa} \equiv \text{seSeparaPor}^-$	$\Rightarrow \text{InverseObjectProperties}(\text{domain-pai: separa} \text{domain-pai: seSeparaPor})$
$\text{ubica} \equiv \text{seUbicaPor}^-$	$\Rightarrow \text{InverseObjectProperties}(\text{domain-pai: ubica} \text{domain-pai: seUbicaPor})$
$\text{verificaExistenciaDe} \equiv \text{existenciaSeVerificaPor}^-$	$\Rightarrow \text{InverseObjectProperties}(\text{domain-pai: verificaExistenciaDe} \text{domain-pai: existenciaSeVerificaPor})$
$\text{obtieneUbicacionDe}$	$\Rightarrow \text{InverseObjectProperties}(\text{domain-pai: obtieneUbicacionDe}$

Tabla B.4 TBox de la KB del patrón de análisis PAI con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
$\equiv \text{ubicacionSeObtienePor}^-$	$\text{domain-pai: ubicacionSeObtienePor}$
$\text{solicitaUbicacionDe} \equiv \text{ubicacionSeSolicit}$ aPor^-	$\Rightarrow \text{InverseObjectProperties}(\text{domain-pai: solicitaUbicacionDe}$ $\text{domain-pai: ubicacionSeSolicitaPor})$
}	

Para una mejor comprensión de las descripciones lógicas presentadas en la *TBox* de la Tabla B.4, a continuación se describen algunos ejemplos tomados de las secciones *Clases*, *Clases Disjuntas*, *Restricciones existenciales y universales* y *Restricciones y características de propiedades de objeto temporales* en la misma tabla:

Clases

T : Conjunto de todos los individuos en el dominio.

$\text{staticClasses} \sqsubseteq T$: La clase *staticClasses* es un tipo de individuos en el dominio del patrón de análisis PAI.

$\text{dynamicClasses} \sqsubseteq T$: La clase *dynamicClasses* es un tipo de individuos en el dominio temporal de la ontología OWL-Time.

$\text{Persona} \sqsubseteq \text{staticClasses}$: La clase *Persona* es un tipo de individuos de la clase *staticClasses*.

$\text{Empleado} \sqsubseteq \text{Persona}$: La clase *Empleado* es un tipo de la clase *Persona*.

Clases Disjuntas

$\text{Articulo} \sqcap \text{Empleado} \sqcap \text{Inventario} \sqcap \text{Institucion} \sqsubseteq \perp$: Establece que los individuos que son miembros de una de las clases (*Articulo*, *Empleado*, *Inventario*, *Institucion*), no son instancias de otra clase.

Restricciones existenciales y universales

$\text{Inventario} \sqsubseteq \forall \text{seActualizaPor} . (\text{Event} \forall \text{esActualizado} . \text{Empleado})$: Establece que un *Inventario* es todo aquello que es actualizado por un *Empleado* a través de un *Evento*.

$\text{Empleado} \sqsubseteq \forall \text{actualiza} . (\text{Event} \forall \text{actualiza} . \text{Inventario})$: Establece que un *Empleado* es quien al menos actualiza un *Inventario* a través de un *Evento*.

Restricciones y características de propiedades de objeto temporales

$\exists \text{actualiza} . T \sqsubseteq (\text{Event} \sqcup \text{Empleado})$: Establece a la clase *Empleado* y a la clase *Event* en el dominio de la propiedad de objeto actualiza.

$T \sqsubseteq \forall \text{actualiza} . (\text{Event} \sqcup \text{Inventario})$: Establece a la clase *Inventario* y a la clase *Event* en el rango de la propiedad de objeto actualiza.

$T \sqsubseteq \leq I$ *actualiza*: Establece que la propiedad de objeto temporal *actualiza* es una propiedad de objeto funcional.

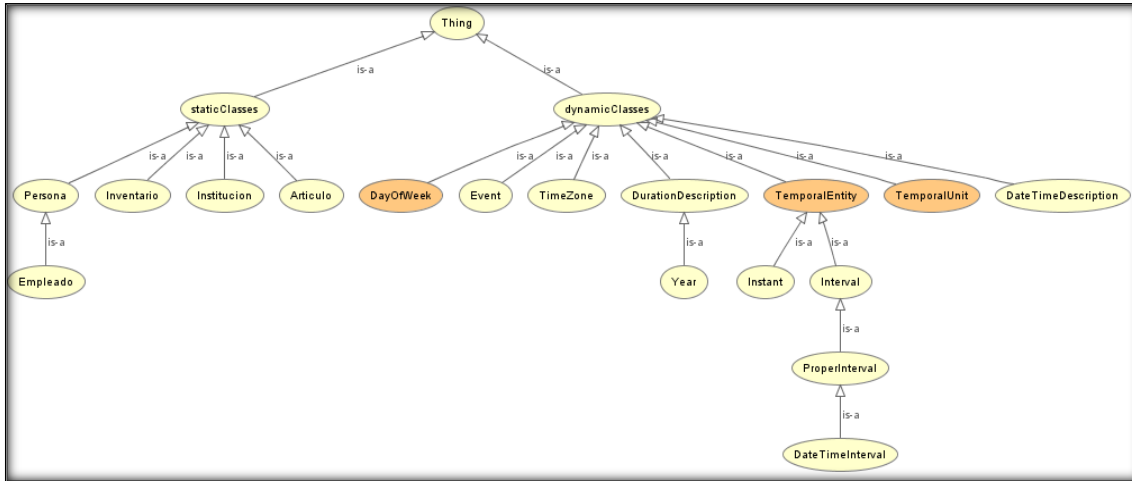


Figura B.1 Jerarquía de clases estáticas y temporales en la KB del patrón de análisis PAI.

1.2. Componente extensional

Para obtener el componente *ABox de casos de éxito* y el componente *ABox de casos de éxito y fracaso*, se utilizaron individuos (nombres de individuos abstractos N_{IA} y nombres de individuos concretos N_{IC} en las DLs) como:

{Valentino, Orlando, Julia, Juan, Inventario_CENIDET, CENIDET, Libro1, Libro2, Libro3, Libro4}

y los valores de datos (dominios concretos en las DLs y valores de datos o datatypes en OWL-DL) como:

{string, Boolean, integer, double, dateTime, long }

Generalmente, una *ABox* se integra de secuencias de interpretaciones (*ABox1*, *ABox2*... *ABoxn*). En las secciones subsecuentes se describe un fragmento de las aserciones más relevantes sobre secuencias de interpretaciones de casos de éxito y casos de éxito y fracaso que representa comportamientos basados en los requerimientos funcionales del patrón de análisis PAI. Se ha omitido el detalle acerca de las aserciones de propiedades de datos que hacen uso de los individuos concretos como string, Boolean, integer, double, dateTime y long en la *ABox*. Estas y otras características pueden consultarse en la versión completa de la KB con aspectos temporales (OTPAE) del patrón de análisis PAI. Al igual que en la ontología OPAE descrita en el *Anexo A. Ontología OPAE del patrón de análisis PAI*, algunas de estas aserciones contemplan propiedades de datos basadas en el modelo SSM como son *Comportamiento*, *Especificidad*, *Nivel de estabilidad* y *Percepción*, entre otras.

1.2.1. RF Separar diferentes tipos de existencias

Las secuencias de interpretaciones descritas en esta sección satisfacen/cumplen con el comportamiento dinámico del RF Separar diferentes tipos de existencias representado formalmente con la fórmula de la Tabla B.5.

Tabla B.5 Fórmula ALC_{LTL} del RF Separar diferentes tipos de existencias.

$\mathcal{G}(\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{separa.Articulo}) \rightarrow (\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{verificaExistenciaDe.Articulo}) \wedge \mathcal{F}(\neg \mathcal{X}\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{actualiza.Articulo} \vee (\text{Empleado} \sqsubseteq \exists \text{registra.Articulo})) \wedge \neg \mathcal{X}\mathcal{X}\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{actualiza.Inventario})))$
--

1.2.1.1. Casos de éxito

En la Tabla B.6 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la *ABox*) basadas en casos de éxito, que conforman una secuencia que representa el comportamiento dinámico en la separación de diferentes tipos de artículos para un inventario. Una representación gráfica simplificada de la secuencia se muestra en la Figura B.2. A manera de ejemplo, en la Figura B.3 se muestra el detalle de las aserciones A_1 , A_2 , A_3 y A_4 en el caso de éxito número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de razonamiento automatizado, los demás casos se definen de manera similar, sin embargo, difieren en los individuos involucrados.

Tabla B.6 *ABox* de casos de éxito en el RF Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
Caso de éxito 1 (A_1, A_2, A_3, A_4):	
A_1 : $[X(\text{Empleado} \sqsubseteq \exists \text{separa.Articulo})]$: El empleado Valentino separa el artículo Libro1.	
$A_1 = \{$	
-- Aserciones de clase --	
$\text{Empleado}(\text{Valentino})$	$\Rightarrow \text{ClassAssertion}(\text{domain-pai:Cliente domain-pai:Valentino})$
$\text{Articulo}(\text{Libro1})$	$\Rightarrow \text{ClassAssertion}(\text{domain-pai:Articulo domain-pai:Libro1})$
$\text{Event}(\text{Req1-Ev1})$	$\Rightarrow \text{ClassAssertion}(\text{timeEvents:Event domain-pai:Req1-Ev1})$
$\text{DateTimeInterval}(\text{Req1-DT11})$	$\Rightarrow \text{ClassAssertion}(\text{time:DateTimeInterval domain-pai:Req1-DT11})$

Tabla B.6 ABox de casos de éxito en el RF Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Instant(Req1-DT11-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DT11-Sins)</i>
<i>Instant(Req1-DT11-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DT11-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>separa(Valentino,Req1-Ev1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:separa domain-pai:Valentino domain-pai:Req1-Ev1)</i>
<i>separa(Req1-Ev1,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:separa domain-pai:Req1-Ev1 domain-pai:Libro1)</i>
<i>during(Req1-Ev1,Req1-DT11)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req1-Ev1 domain-pai:Req1-DT11)</i>
<i>hasBeginning(Req1-DT11,Req1-DT11-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req1-DT11 domain-pai:Req1-DT11-Sins)</i>
<i>hasEnd(Req1-DT11,Req1-DT11-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req1-DT11 domain-pai:Req1-DT11-Eins)</i>
}	\Rightarrow
A₂: [XX (Empleado \sqsubseteq \existsverificaExistenciaDe.Articulo)] : El empleado Valentino verifica la existencia del artículo Libro1	
A ₂ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Empleado domain-pai:Valentino)</i>
<i>Articulo(Libro1)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro1)</i>
<i>Event(Req1-Ev2)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req1-Ev2)</i>
<i>DateTimeInterval(Req1-DT12)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req1-DT12)</i>
<i>Instant(Req1-DT12-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DT12-Sins)</i>
<i>Instant(Req1-DT12-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DT12-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>verificaExistenciaDe(Valentino,Req1-Ev2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:verificaExistenciaDe domain-pai:Valentino domain-pai:Req1-Ev2)</i>
<i>verificaExistenciaDe(Req1-Ev2,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:verificaExistenciaDe domain-pai:Req1-Ev2 domain-pai:Libro1)</i>
<i>during(Req1-Ev2,Req1-DT12)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req1-Ev2 domain-pai:Req1-DT12)</i>
<i>hasBeginning(Req1-DT12,Req1-DT12-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req1-DT12 domain-pai:Req1-DT12-Sins)</i>
<i>hasEnd(Req1-DT12,Req1-DT12-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req1-DT12 domain-pai:Req1-DT12-Eins)</i>

Tabla B.6 ABox de casos de éxito en el RF Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Eins</i>	<i>pai:Req1-DTI2 domain-pai:Req1-DTI2-Eins</i>
}	
A₃: [XXX((Empleado \sqsubseteq \existsactualiza.Articulo) \vee (Empleado \sqsubseteq \existsregistra.Articulo))] : El empleado Valentino actualiza el artículo Libro1 o registra el artículo Libro1.	
A ₃ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Empleado domain-pai:Valentino)</i>
<i>Articulo(Libro1)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro1)</i>
<i>Event(Req1-Ev3)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req1-Ev3)</i>
<i>DateTimeInterval(Req1-DTI3)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req1-DTI3)</i>
<i>Instant(Req1-DTI3-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DTI3-Sins)</i>
<i>Instant(Req1-DTI3-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DTI3-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>registra(Valentino,Req1-Ev3)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:registra domain-pai:Valentino domain-pai:Req1-Ev3)</i>
<i>registra(Req1-Ev3,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:registra domain-pai:Req1-Ev3 domain-pai:Libro1)</i>
<i>during(Req1-Ev3,Req1-DTI3)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req1-Ev3 domain-pai:Req1-DTI3)</i>
<i>hasBeginning(Req1-DTI3,Req1-DTI3-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req1-DTI3 domain-pai:Req1-DTI3-Sins)</i>
<i>hasEnd(Req1-DTI3,Req1-DTI3-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req1-DTI3 domain-pai:Req1-DTI3-Eins)</i>
}	
A₄: [XXXX(Empleado \sqsubseteq \existsactualiza.Inventario)] : El empleado Valentino actualiza el inventario Inventario_CENIDET.	
A ₄ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Empleado domain-pai:Valentino)</i>
<i>Inventario(Inventario_CENIDET)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Inventario domain-pai:Inventario_CENIDET)</i>
<i>Event(Req1-Ev4)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req1-Ev4)</i>
<i>DateTimeInterval(Req1-DTI4)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req1-DTI4)</i>
<i>Instant(Req1-DTI4-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DTI4-Sins)</i>

Tabla B.6 ABox de casos de éxito en el RF Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Instant(Req1-DTI4-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DTI4-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>actualiza(Valentino,Req1-Ev4)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:actualiza domain-pai:Valentino domain-pai:Req1-Ev4)</i>
<i>actualiza(Req1-Ev4,Inventario_CENIDET)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:actualiza domain-pai:Req1-Ev4 domain-pai:cInventario_CENIDET)</i>
<i>during(Req1-Ev4,Req1-DTI4)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req1-Ev4 domain-pai:Req1-DTI4)</i>
<i>hasBeginning(Req1-DTI4,Req1-DTI4-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req1-DTI4 domain-pai:Req1-DTI4-Sins)</i>
<i>hasEnd(Req1-DTI4,Req1-DTI4-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req1-DTI4 domain-pai:Req1-DTI4-Eins)</i>
}	
Caso de éxito 2 (A₅, A₆, A₇, A₈):	
A ₅ : [X(Empleado \sqsubseteq \exists separa.Articulo)]: El empleado Orlando separa el artículo Libro2.	
A ₅ = {	
-- Aserciones de clase --	
<i>Empleado(Orlando)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Empleado domain-pai:Orlando)</i>
<i>Articulo(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro2)</i>
<i>Event(Req1-Ev5)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req1-Ev5)</i>
<i>DateTimeInterval(Req1-DTI5)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req1-DTI5)</i>
<i>Instant(Req1-DTI5-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DTI5-Sins)</i>
<i>Instant(Req1-DTI5-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DTI5-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>separa(Orlando,Req1-Ev5)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:separa domain-pai:Orlando domain-pai:Req1-Ev5)</i>
<i>separa(Req1-Ev5,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:separa domain-pai:Req1-Ev5 domain-pai:Libro2)</i>
<i>during(Req1-Ev5,Req1-DTI5)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req1-Ev5 domain-pai:Req1-DTI5)</i>
<i>hasBeginning(Req1-DTI5,Req1-DTI5-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req1-DTI5 domain-pai:Req1-DTI5-Sins)</i>
<i>hasEnd(Req1-DTI5,Req1-DTI5-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req1-DTI5 domain-pai:Req1-DTI5-Eins)</i>
}	\Rightarrow
A ₆ : [XX (Empleado \sqsubseteq \exists verificaExistenciaDe.Articulo)]: El empleado Orlando verifica la existencia del artículo Libro2	

Tabla B.6 ABox de casos de éxito en el RF Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
$A_6 = \{$	
-- Aserciones de clase --	
<i>Empleado(Orlando)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Empleado domain-pai:Orlando)</i>
<i>Articulo(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro2)</i>
<i>Event(Req1-Ev6)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req1-Ev6)</i>
<i>DateTimeInterval(Req1-DTI6)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req1-DTI6)</i>
<i>Instant(Req1-DTI6-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DTI6-Sins)</i>
<i>Instant(Req1-DTI6-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DTI6-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>verificaExistenciaDe(Orlando,Req1-Ev6)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:verificaExistenciaDe domain-pai:Orlando domain-pai:Req1-Ev6)</i>
<i>verificaExistenciaDe(Req1-Ev6,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:verificaExistenciaDe domain-pai:Req1-Ev6 domain-pai:Libro2)</i>
<i>during(Req1-Ev6,Req1-DTI6)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req1-Ev6 domain-pai:Req1-DTI6)</i>
<i>hasBeginning(Req1-DTI6,Req1-DTI6-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req1-DTI6 domain-pai:Req1-DTI6-Sins)</i>
<i>hasEnd(Req1-DTI6,Req1-DTI6-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req1-DTI6 domain-pai:Req1-DTI6-Eins)</i>
$\}$	
$A_7: [XXX(Empleado \sqsubseteq \exists actualiza.Articulo) \vee (Empleado \sqsubseteq \exists registra.Articulo)] : El empleado Orlando actualiza el artículo Libro2 o registra el artículo Libro2.$	
$A_7 = \{$	
-- Aserciones de clase --	
<i>Empleado(Orlando)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Empleado domain-pai:Orlando)</i>
<i>Articulo(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro2)</i>
<i>Event(Req1-Ev7)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req1-Ev7)</i>
<i>DateTimeInterval(Req1-DTI7)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req1-DTI7)</i>
<i>Instant(Req1-DTI7-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DTI7-Sins)</i>
<i>Instant(Req1-DTI7-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DTI7-Eins)</i>

Tabla B.6 ABox de casos de éxito en el RF Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
-- Aserciones de propiedades de objeto --	
<i>actualiza(Orlando,Req1-Ev7)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:actualiza domain-pai:Orlando domain-pai:Req1-Ev7)</i>
<i>actualiza(Req1-Ev7,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:actualiza domain-pai:Req1-Ev7 domain-pai:Libro2)</i>
<i>during(Req1-Ev7,Req1-DTI7)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req1-Ev7 domain-pai:Req1-DTI7)</i>
<i>hasBeginning(Req1-DTI7,Req1-DTI7-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req1-DTI7 domain-pai:Req1-DTI7-Sins)</i>
<i>hasEnd(Req1-DTI7,Req1-DTI7-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req1-DTI7 domain-pai:Req1-DTI7-Eins)</i>
}	
A₈: [XXXX (Empleado \sqsubseteq \existsactualiza.Inventario)]: El empleado Orlando actualiza el inventario Inventario_CENIDET.	
A ₈ = {	
-- Aserciones de clase --	
<i>Empleado(Orlando)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Empleado domain-pai:Orlando)</i>
<i>Inventario(Inventario_CENIDET)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Inventario domain-pai:Inventario_CENIDET)</i>
<i>Event(Req1-Ev8)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req1-Ev8)</i>
<i>DateTimeInterval(Req1-DTI8)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req1-DTI8)</i>
<i>Instant(Req1-DTI8-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DTI8-Sins)</i>
<i>Instant(Req1-DTI8-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DTI8-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>actualiza(Orlando,Req1-Ev8)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:actualiza domain-pai:Orlando domain-pai:Req1-Ev8)</i>
<i>actualiza(Req1-Ev8,Inventario_CENIDET)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:actualiza domain-pai:Req1-Ev8 domain-pai:cInventario_CENIDET)</i>
<i>during(Req1-Ev8,Req1-DTI8)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req1-Ev8 domain-pai:Req1-DTI8)</i>
<i>hasBeginning(Req1-DTI8,Req1-DTI8-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req1-DTI8 domain-pai:Req1-DTI8-Sins)</i>
<i>hasEnd(Req1-DTI8,Req1-DTI8-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req1-DTI8 domain-pai:Req1-DTI8-Eins)</i>
}	
Caso de éxito 3 (A₉, A₁₀, A₁₁, A₁₂):	
A₉: [X(Empleado \sqsubseteq \existssepara.Articulo)]: El empleado Julia separa el articulo Libro3.	
...	
A₁₀: [XX (Empleado \sqsubseteq \existsverificaExistenciaDe.Articulo)] : El empleado Julia verifica la existencia	

Tabla B.6 ABox de casos de éxito en el RF Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
<p>del artículo Libro3</p> <p>...</p> <p>A_{11}: $[XXX(Empleado \sqsubseteq \exists actualiza.Articulo) \vee (Empleado \sqsubseteq \exists registra.Articulo)]$: El empleado Julia actualiza el artículo Libro3 o registra el artículo Libro3.</p> <p>...</p> <p>A_{12}: $[XXXX(Empleado \sqsubseteq \exists actualiza.Inventario)]$: El empleado Julia actualiza el inventario Inventario_CENIDET.</p> <p>...</p>	
<p>Caso de éxito 4 ($A_{13}, A_{14}, A_{15}, A_{16}$):</p> <p>$A_{13}$: $[X(Empleado \sqsubseteq \exists separa.Articulo)]$: El empleado Juan separa el artículo Libro4.</p> <p>...</p> <p>A_{14}: $[XX(Empleado \sqsubseteq \exists verificaExistenciaDe.Articulo)]$: El empleado Juan verifica la existencia del artículo Libro4.</p> <p>...</p> <p>A_{15}: $[XXX(Empleado \sqsubseteq \exists actualiza.Articulo) \vee (Empleado \sqsubseteq \exists registra.Articulo)]$: El empleado Juan actualiza el artículo Libro3 o registra el artículo Libro4.</p> <p>...</p> <p>A_{16}: $[XXXX(Empleado \sqsubseteq \exists actualiza.Inventario)]$: El empleado Juan actualiza el inventario Inventario_CENIDET.</p> <p>...</p>	
<p>Caso de éxito n</p> <p>A_n</p> <p>...</p>	

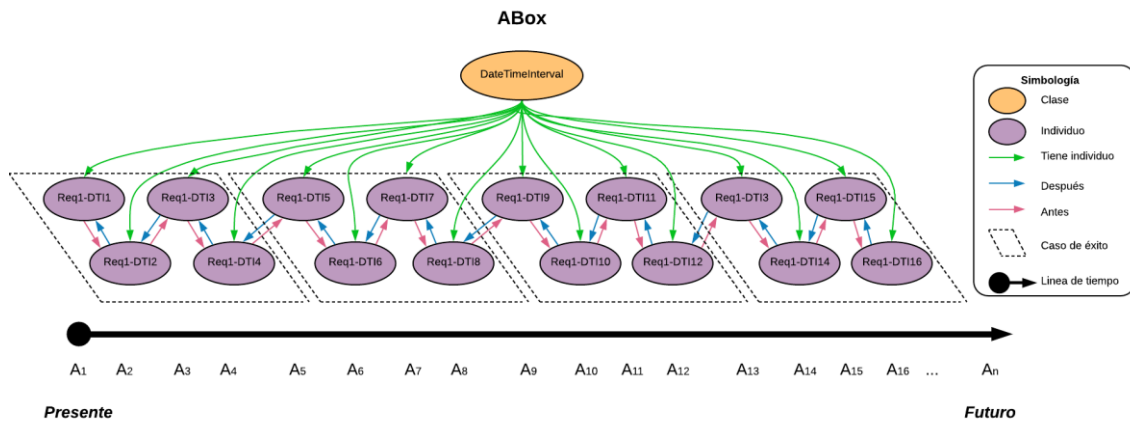


Figura B.2 SoIs en la ABox de casos de éxito de acuerdo al RF Separar diferentes tipos de existencias.

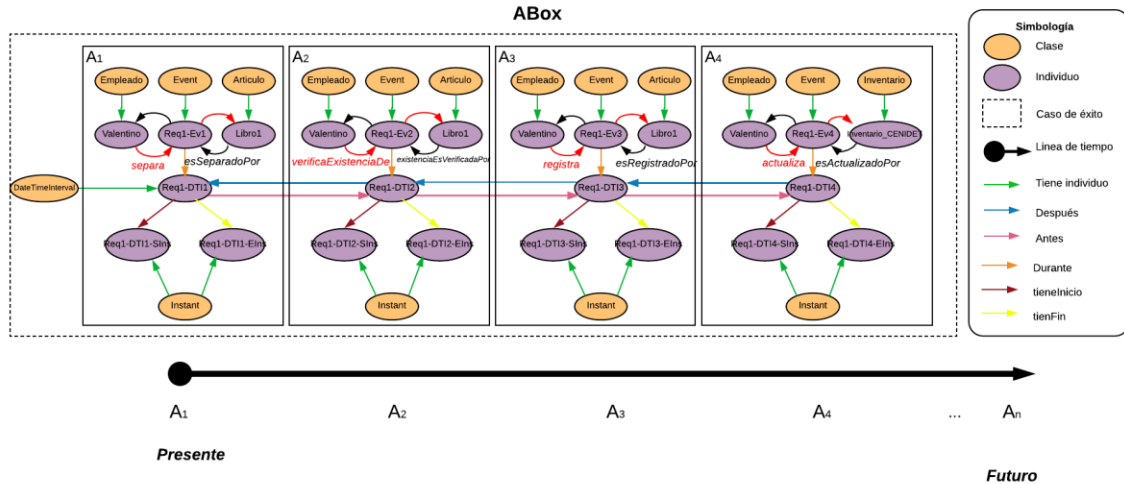


Figura B.3 Caso de éxito 1 en la ABox de casos de éxito de acuerdo al RF Separar diferentes tipos de existencias.

1.2.1.2. Casos de éxito y fracaso

En la Tabla B.7 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la *ABox*) basadas en casos de éxito y fracaso, que conforman una secuencia que representa el comportamiento dinámico en la separación de diferentes tipos de artículos para un inventario. Una representación gráfica simplificada de la secuencia se muestra en la Figura B.4. A manera de ejemplo, en la Figura B.5 se muestra el detalle de las aserciones A_1 , A_2 , A_3 y A_4 en el caso de fracaso número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de razonamiento automatizado. Los demás casos se definen de manera similar, sin embargo, difieren en los individuos involucrados.

En la Tabla B.8 se describen las reglas SWRL utilizadas para restringir la existencia de afirmaciones de propiedades de objeto en el intervalo de tiempo correspondiente.

Tabla B.7 ABox de casos de éxito y fracaso en el RF Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
Caso de fracaso 1 (A_1, A_2, A_3, A_4):	
$A_1: [X(\text{Empleado} \sqsubseteq \exists \text{separa}.\text{Articulo})]:$ El empleado Valentino separa el artículo Libro1.	
$A_1 = \{$	
-- Aserciones de clase --	
$\text{Empleado}(\text{Valentino})$	$\Rightarrow \text{ClassAssertion}(\text{domain-pai:Cliente domain-pai:Valentino})$
$\text{Articulo}(\text{Libro1})$	$\Rightarrow \text{ClassAssertion}(\text{domain-pai:Articulo domain-pai:Libro1})$
$\text{Event}(\text{Req1-Ev1})$	$\Rightarrow \text{ClassAssertion}(\text{timeEvents:Event domain-pai:Req1-Ev1})$
$\text{DateTimeInterval}(\text{Req1-DT11})$	$\Rightarrow \text{ClassAssertion}(\text{time:DateTimeInterval domain-}$

Tabla B.7 ABox de casos de éxito y fracaso en el RF Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Instant(Req1-DT11-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DT11-Sins)</i>
<i>Instant(Req1-DT11-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DT11-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>separa(Valentino,Req1-Ev1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:separa domain-pai:Valentino domain-pai:Req1-Ev1)</i>
<i>separa(Req1-Ev1,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:separa domain-pai:Req1-Ev1 domain-pai:Libro1)</i>
<i>during(Req1-Ev1,Req1-DT11)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req1-Ev1 domain-pai:Req1-DT11)</i>
<i>hasBeginning(Req1-DT11,Req1-DT11-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req1-DT11 domain-pai:Req1-DT11-Sins)</i>
<i>hasEnd(Req1-DT11,Req1-DT11-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req1-DT11 domain-pai:Req1-DT11-Eins)</i>
}	\Rightarrow
A₂: [XX (Empleado \sqsubseteq \existsverificaExistenciaDe.Articulo)] : El empleado Valentino verifica la existencia del artículo Libro1.	
A ₂ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Empleado domain-pai:Valentino)</i>
<i>Articulo(Libro1)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro1)</i>
<i>Event(Req1-Ev2)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req1-Ev2)</i>
<i>DateTimeInterval(Req1-DT12)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req1-DT12)</i>
<i>Instant(Req1-DT12-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DT12-Sins)</i>
<i>Instant(Req1-DT12-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DT12-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>verificaExistenciaDe(Valentino,Req1-Ev2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:verificaExistenciaDe domain-pai:Valentino domain-pai:Req1-Ev2)</i>
<i>verificaExistenciaDe(Req1-Ev2,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:verificaExistenciaDe domain-pai:Req1-Ev2 domain-pai:Libro1)</i>
<i>during(Req1-Ev2,Req1-DT12)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req1-Ev2 domain-pai:Req1-DT12)</i>
<i>hasBeginning(Req1-DT12,Req1-DT12-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req1-DT12 domain-pai:Req1-DT12-Sins)</i>
<i>hasEnd(Req1-DT12,Req1-DT12-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req1-DT12 domain-pai:Req1-DT12-Eins)</i>

Tabla B.7 ABox de casos de éxito y fracaso en el RF Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
<p>} A₃: [\negXXX(Empleado \sqsubseteq \existsactualiza.Articulo) \vee (Empleado \sqsubseteq \existsregistra.Articulo)]: El empleado Valentino no actualiza el artículo Libro1 o no registra el artículo Libro1. A₃ = {</p>	
<p>-- Aserciones de clase --</p>	
Event(Req1-Ev3)	\Rightarrow ClassAssertion(timeEvents:Event domain-pai:Req1-Ev3)
DateTimeInterval(Req1-DTI3)	\Rightarrow ClassAssertion(time:DateTimeInterval domain-pai:Req1-DTI3)
Instant(Req1-DTI3-Sins)	\Rightarrow ClassAssertion(time:Instant domain-pai:Req1-DTI3-Sins)
Instant(Req1-DTI3-Eins)	\Rightarrow ClassAssertion(time:Instant domain-pai:Req1-DTI3-Eins)
<p>-- Aserciones de propiedades de objeto --</p>	
during(Req1-Ev3,Req1-DTI3)	\Rightarrow ObjectPropertyAssertion(timeEvents:during domain-pai:Req1-Ev3 domain-pai:Req1-DTI3)
hasBeginning(Req1-DTI3,Req1-DTI3-Sins)	\Rightarrow ObjectPropertyAssertion(time:hasBeginning domain-pai:Req1-DTI3 domain-pai:Req1-DTI3-Sins)
hasEnd(Req1-DTI3,Req1-DTI3-Eins)	\Rightarrow ObjectPropertyAssertion(time:hasEnd domain-pai:Req1-DTI3 domain-pai:Req1-DTI3-Eins)
<p>} A₄: [\negXXXX(Empleado \sqsubseteq \existsactualiza.Inventario)]: El empleado Valentino no actualiza el inventario Inventario_CENIDET. A₄ = {</p>	
<p>-- Aserciones de clase --</p>	
Event(Req1-Ev4)	\Rightarrow ClassAssertion(timeEvents:Event domain-pai:Req1-Ev4)
DateTimeInterval(Req1-DTI4)	\Rightarrow ClassAssertion(time:DateTimeInterval domain-pai:Req1-DTI4)
Instant(Req1-DTI4-Sins)	\Rightarrow ClassAssertion(time:Instant domain-pai:Req1-DTI4-Sins)
Instant(Req1-DTI4-Eins)	\Rightarrow ClassAssertion(time:Instant domain-pai:Req1-DTI4-Eins)
<p>-- Aserciones de propiedades de objeto --</p>	
during(Req1-Ev4,Req1-DTI4)	\Rightarrow ObjectPropertyAssertion(timeEvents:during domain-pai:Req1-Ev4 domain-pai:Req1-DTI4)
hasBeginning(Req1-DTI4,Req1-DTI4-Sins)	\Rightarrow ObjectPropertyAssertion(time:hasBeginning domain-pai:Req1-DTI4 domain-pai:Req1-DTI4-Sins)
hasEnd(Req1-DTI4,Req1-DTI4-Eins)	\Rightarrow ObjectPropertyAssertion(time:hasEnd domain-pai:Req1-DTI4 domain-pai:Req1-DTI4-Eins)
<p>} Caso de éxito 1 (A₅, A₆, A₇, A₈): A₅: [X(Empleado \sqsubseteq \existssepara.Articulo)]: El empleado Orlando separa el artículo Libro2. A₅ = {</p>	
<p>-- Aserciones de clase --</p>	
Empleado(Orlando)	\Rightarrow ClassAssertion(domain-pai:Empleado domain-

Tabla B.7 ABox de casos de éxito y fracaso en el RF Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Articulo(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro2)</i>
<i>Event(Req1-Ev5)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req1-Ev5)</i>
<i>DateTimeInterval(Req1-DTI5)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req1-DTI5)</i>
<i>Instant(Req1-DTI5-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DTI5-Sins)</i>
<i>Instant(Req1-DTI5-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DTI5-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>separa(Orlando,Req1-Ev5)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:separa domain-pai:Orlando domain-pai:Req1-Ev5)</i>
<i>separa(Req1-Ev5,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:separa domain-pai:Req1-Ev5 domain-pai:Libro2)</i>
<i>during(Req1-Ev5,Req1-DTI5)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req1-Ev5 domain-pai:Req1-DTI5)</i>
<i>hasBeginning(Req1-DTI5,Req1-DTI5-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req1-DTI5 domain-pai:Req1-DTI5-Sins)</i>
<i>hasEnd(Req1-DTI5,Req1-DTI5-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req1-DTI5 domain-pai:Req1-DTI5-Eins)</i>
}	\Rightarrow
A₆: [XX (Empleado \sqsubseteq \existsverificaExistenciaDe.Articulo)] : El empleado Orlando verifica la existencia del artículo Libro2	
A ₆ = {	
-- Aserciones de clase --	
<i>Empleado(Orlando)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Empleado domain-pai:Orlando)</i>
<i>Articulo(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro2)</i>
<i>Event(Req1-Ev6)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req1-Ev6)</i>
<i>DateTimeInterval(Req1-DTI6)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req1-DTI6)</i>
<i>Instant(Req1-DTI6-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DTI6-Sins)</i>
<i>Instant(Req1-DTI6-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DTI6-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>verificaExistenciaDe(Orlando,Req1-Ev6)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:verificaExistenciaDe domain-pai:Orlando domain-pai:Req1-Ev6)</i>
<i>verificaExistenciaDe(Req1-Ev6,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:verificaExistenciaDe domain-pai:Req1-Ev6 domain-pai:Libro2)</i>

Tabla B.7 ABox de casos de éxito y fracaso en el RF Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
<i>during</i> (Req1-Ev6,Req1-DTI6)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-pai:Req1-Ev6 domain-pai:Req1-DTI6)
<i>hasBeginning</i> (Req1-DTI6,Req1-DTI6-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-pai:Req1-DTI6 domain-pai:Req1-DTI6-Sins)
<i>hasEnd</i> (Req1-DTI6,Req1-DTI6-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-pai:Req1-DTI6 domain-pai:Req1-DTI6-Eins)
}	
<i>A7</i> : [XXX((Empleado \sqsubseteq \exists actualiza.Articulo) \vee (Empleado \sqsubseteq \exists registra.Articulo))] : El empleado Orlando actualiza el artículo Libro2 o registra el artículo Libro2.	
<i>A7</i> = {	
-- Aserciones de clase --	
<i>Empleado</i> (Orlando)	\Rightarrow <i>ClassAssertion</i> (domain-pai:Empleado domain-pai:Orlando)
<i>Articulo</i> (Libro2)	\Rightarrow <i>ClassAssertion</i> (domain-pai:Articulo domain-pai:Libro2)
<i>Event</i> (Req1-Ev7)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-pai:Req1-Ev7)
<i>DateTimeInterval</i> (Req1-DTI7)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-pai:Req1-DTI7)
<i>Instant</i> (Req1-DTI7-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-pai:Req1-DTI7-Sins)
<i>Instant</i> (Req1-DTI7-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-pai:Req1-DTI7-Eins)
-- Aserciones de propiedades de objeto --	
<i>actualiza</i> (Orlando,Req1-Ev7)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-pai:actualiza domain-pai:Orlando domain-pai:Req1-Ev7)
<i>actualiza</i> (Req1-Ev7,Libro2)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-pai:actualiza domain-pai:Req1-Ev7 domain-pai:Libro2)
<i>during</i> (Req1-Ev7,Req1-DTI7)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-pai:Req1-Ev7 domain-pai:Req1-DTI7)
<i>hasBeginning</i> (Req1-DTI7,Req1-DTI7-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-pai:Req1-DTI7 domain-pai:Req1-DTI7-Sins)
<i>hasEnd</i> (Req1-DTI7,Req1-DTI7-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-pai:Req1-DTI7 domain-pai:Req1-DTI7-Eins)
}	
<i>A8</i> : [XXXX (Empleado \sqsubseteq \exists actualiza.Inventario)]: El empleado Orlando actualiza el inventario Inventario_CENIDET.	
<i>A8</i> = {	
-- Aserciones de clase --	
<i>Empleado</i> (Orlando)	\Rightarrow <i>ClassAssertion</i> (domain-pai:Empleado domain-pai:Orlando)
<i>Inventario</i> (Inventario_CENIDET)	\Rightarrow <i>ClassAssertion</i> (domain-pai:Inventario domain-pai:Inventario_CENIDET)
<i>Event</i> (Req1-Ev8)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-pai:Req1-Ev8)
<i>DateTimeInterval</i> (Req1-DTI8)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-

Tabla B.7 ABox de casos de éxito y fracaso en el RF Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Instant(Req1-DT18-Sins)</i>	\Rightarrow <i>pai:Req1-DT18)</i> <i>ClassAssertion(time:Instant domain-pai:Req1-DT18-Sins)</i>
<i>Instant(Req1-DT18-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req1-DT18-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>actualiza(Orlando,Req1-Ev8)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:actualiza domain-pai:Orlando domain-pai:Req1-Ev8)</i>
<i>actualiza(Req1-Ev8,Inventario_CENIDET)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:actualiza domain-pai:Req1-Ev8 domain-pai:cInventario_CENIDET)</i>
<i>during(Req1-Ev8,Req1-DT18)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req1-Ev8 domain-pai:Req1-DT18)</i>
<i>hasBeginning(Req1-DT18,Req1-DT18-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req1-DT18 domain-pai:Req1-DT18-Sins)</i>
<i>hasEnd(Req1-DT18,Req1-DT18-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req1-DT18 domain-pai:Req1-DT18-Eins)</i>
}	
Caso de fracaso 2 (A₉, A₁₀, A₁₁, A₁₂):	
<i>A₉: [X(Empleado \sqsubseteq \existssepara.Articulo)]: El empleado Julia separa el artículo Libro3.</i>	
...	
<i>A₁₀: [XX (Empleado \sqsubseteq \existsverificaExistenciaDe.Articulo)] : El empleado Julia verifica la existencia del artículo Libro3.</i>	
...	
<i>A₁₁: [\negXXX((Empleado \sqsubseteq \existsactualiza.Articulo) \vee (Empleado \sqsubseteq \existsregistra.Articulo))] : El empleado Julia no actualiza el artículo Libro3 o no registra el artículo Libro3.</i>	
...	
<i>A₁₂: [\negXXXX (Empleado \sqsubseteq \existsactualiza.Inventario)]: El empleado Julia no actualiza el inventario Inventario_CENIDET.</i>	
Caso de éxito 2 (A₁₃, A₁₄, A₁₅, A₁₆):	
<i>A₁₃: [X(Empleado \sqsubseteq \existssepara.Articulo)]: El empleado Juan separa el artículo Libro4.</i>	
...	
<i>A₁₄: [XX (Empleado \sqsubseteq \existsverificaExistenciaDe.Articulo)] : El empleado Juan verifica la existencia del artículo Libro4</i>	
...	
<i>A₁₅: [XXX((Empleado \sqsubseteq \existsactualiza.Articulo) \vee (Empleado \sqsubseteq \existsregistra.Articulo))] : El empleado Juan actualiza el artículo Libro4 o registra el artículo Libro4.</i>	
...	
<i>A₁₆: [XXXX (Empleado \sqsubseteq \existsactualiza.Inventario)]: El empleado Juan actualiza el inventario Inventario_CENIDET.</i>	
Caso de éxito o fracaso n	
<i>A_n</i>	
...	

Tabla B.8 Reglas SWRL para la SoIs de casos de éxito y fracaso de acuerdo al RF Separar diferentes tipos de existencias. (cont.)

ABox	Regla SWRL
A ₃	<i>Event(?e), during(?e, ?i), overlaps(?i, Req1-DTI3), registra(?e, Libro1), registra(Valentino, ?e) -> Nothing(Valentino)</i>
A ₄	<i>Event(?e), during(?e, ?i), overlaps(?i, Req1-DTI4), actualiza(?e, Inventario_CENIDET), actualiza(Valentino, ?e) -> Nothing(Valentino)</i>
A ₁₁	<i>Event(?e), during(?e, ?i), overlaps(?i, Req1-DTI11), registra(?e, Libro3), registra(Julia, ?e) -> Nothing(Julia)</i>
A ₁₂	<i>Event(?e), during(?e, ?i), overlaps(?i, Req1-DTI12), actualiza(?e, Inventario_CENIDET), actualiza(Julia, ?e) -> Nothing(Julia)</i>

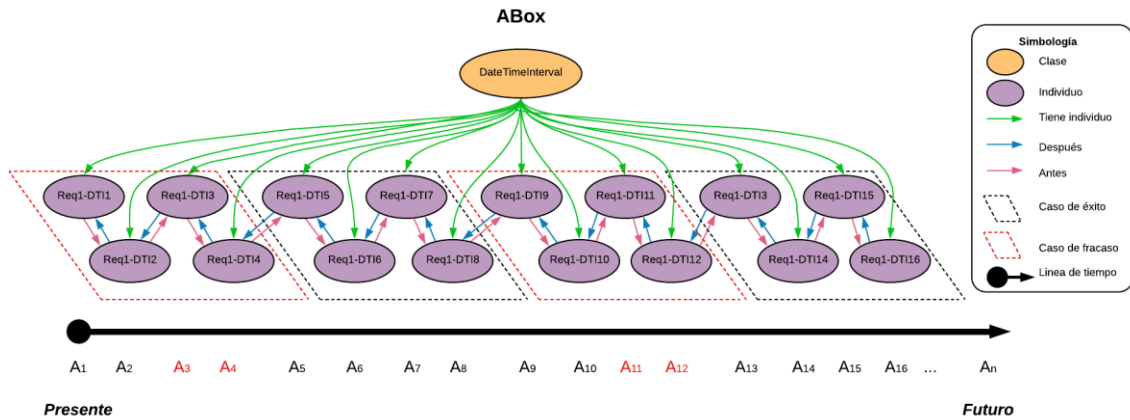


Figura B.4 SoIs en la ABox de casos de éxito y fracaso de acuerdo al RF Separar diferentes tipos de existencias.

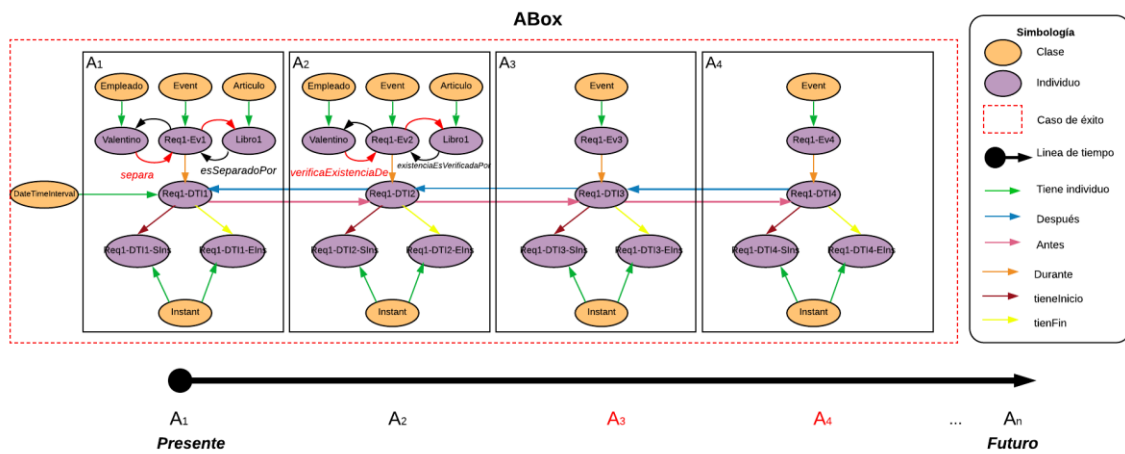


Figura B.5 Caso de fracaso 1 en la ABox de casos de éxito y fracaso de acuerdo al RF Separar diferentes tipos de existencias.

1.2.2. RF Realizar seguimiento de cantidades de artículos en existencias

Las secuencias de interpretaciones descritas en esta sección satisfacen/cumplen con el comportamiento dinámico del RF Realizar seguimiento de cantidades de artículos en existencias representado formalmente con la fórmula de la Tabla B.9.

Tabla B.9 Fórmula ALC_{LTL} del RF Realizar seguimiento de cantidades de artículos en existencias.

$$\mathcal{G}(\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{buscaExistenciasDe.Articulo}) \rightarrow (\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{verificaExistenciaDe.Articulo}) \wedge \mathcal{F}(\neg \mathcal{X}\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{especificaTipoExistenciasDe.Articulo}) \wedge \neg \mathcal{X}\mathcal{X}\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{obtieneExistenciasDe.Articulo})))$$

1.2.2.1. Casos de éxito

En la Tabla B.10 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la *ABox*) basadas en casos de éxito, que conforman una secuencia que representa el comportamiento dinámico durante el seguimiento de la cantidad de existencias del artículo Libro1 de un inventario. Una representación gráfica simplificada de la secuencia se muestra en la Figura B.6. A manera de ejemplo, en la Figura B.7 se muestra el detalle de las aserciones A_1 , A_2 , A_3 y A_4 en el caso de éxito número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de razonamiento automatizado, los demás casos se definen de manera similar. Sin embargo, difieren en los individuos involucrados.

Tabla B.10 *ABox* de casos de éxito en el RF Realizar seguimiento de cantidades de artículos en existencias. (cont.)

Sintaxis DL	Sintaxis Funcional
Caso de éxito 1 (A_1, A_2, A_3, A_4):	
A_1 : $[X(\text{Empleado} \sqsubseteq \exists \text{buscaExistenciasDe.Articulo})]$: El empleado Valentino busca existencias del artículo Libro1.	
$A_1 = \{$	
<i>-- Aserciones de clase --</i>	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-pai:Empleado domain-pai:Valentino)
<i>Articulo</i> (Libro1)	\Rightarrow <i>ClassAssertion</i> (domain-pai:Articulo domain-pai:Libro1)
<i>Event</i> (Req2-Ev1)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-pai:Req2-Ev1)
<i>DateTimeInterval</i> (Req2-DT11)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-pai:Req2-DT11)

Tabla B.10 ABox de casos de éxito en el RF Realizar seguimiento de cantidades de artículos en existencias.
(cont.)

Sintaxis DL	Sintaxis Funcional
<i>Instant(Req2-DT11-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req2-DT11-Sins)</i>
<i>Instant(Req2-DT11-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req2-DT11-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>buscaExistenciasDe(Valentino,Req2-Ev1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:buscaExistenciasDe domain-pai:Valentino domain-pai:Req2-Ev1)</i>
<i>buscaExistenciasDe(Req2-Ev1,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:buscaExistenciasDe domain-pai:Req2-Ev1 domain-pai:Libro1)</i>
<i>during(Req2-Ev1,Req2-DT11)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req2-Ev1 domain-pai:Req2-DT11)</i>
<i>hasBeginning(Req2-DT11,Req2-DT11-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req2-DT11 domain-pai:Req2-DT11-Sins)</i>
<i>hasEnd(Req2-DT11,Req2-DT11-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req2-DT11 domain-pai:Req2-DT11-Eins)</i>
}	\Rightarrow
A₂: [XX (Empleado \sqsubseteq \existsverificaExistenciaDe.Articulo)] : El empleado Valentino verifica la existencia del artículo Libro1	
A ₂ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Empleado domain-pai:Valentino)</i>
<i>Articulo(Libro1)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro1)</i>
<i>Event(Req2-Ev2)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req2-Ev2)</i>
<i>DateTimeInterval(Req2-DT12)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req2-DT12)</i>
<i>Instant(Req2-DT12-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req2-DT12-Sins)</i>
<i>Instant(Req2-DT12-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req2-DT12-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>verificaExistenciaDe(Valentino,Req2-Ev2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:verificaExistenciaDe domain-pai:Valentino domain-pai:Req2-Ev2)</i>
<i>verificaExistenciaDe(Req2-Ev2,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:verificaExistenciaDe domain-pai:Req2-Ev2 domain-pai:Libro1)</i>
<i>during(Req2-Ev2,Req2-DT12)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req2-Ev2 domain-pai:Req2-DT12)</i>
<i>hasBeginning(Req2-DT12,Req2-DT12-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req2-DT12 domain-pai:Req2-DT12-Sins)</i>

Tabla B.10 ABox de casos de éxito en el RF Realizar seguimiento de cantidades de artículos en existencias.
(cont.)

Sintaxis DL	Sintaxis Funcional
<i>hasEnd(Req2-DTI2,Req2-DTI2-Eins)</i>	<i>Sins)</i> ⇒ <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req2-DTI2 domain-pai:Req2-DTI2-Eins)</i>
}	
A₃: [XXX (Empleado ⊆ ∃especificaTipoExistenciasDe.Articulo)] : El empleado Valentino especifica el tipo de existencias (enMano, enOrden, reservadas) del artículo Libro1.	
A ₃ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	⇒ <i>ClassAssertion(domain-pai:Empleado domain-pai:Valentino)</i>
<i>Articulo(Libro1)</i>	⇒ <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro1)</i>
<i>Event(Req2-Ev3)</i>	⇒ <i>ClassAssertion(timeEvents:Event domain-pai:Req2-Ev3)</i>
<i>DateTimeInterval(Req2-DTI3)</i>	⇒ <i>ClassAssertion(time:DateTimeInterval domain-pai:Req2-DTI3)</i>
<i>Instant(Req2-DTI3-Sins)</i>	⇒ <i>ClassAssertion(time:Instant domain-pai:Req2-DTI3-Sins)</i>
<i>Instant(Req2-DTI3-Eins)</i>	⇒ <i>ClassAssertion(time:Instant domain-pai:Req2-DTI3-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>especificaTipoExistenciasDe(Valentino,Req2-Ev3)</i>	⇒ <i>ObjectPropertyAssertion(domain-pai:especificaTipoExistenciasDe domain-pai:Valentino domain-pai:Req2-Ev3)</i>
<i>especificaTipoExistenciasDe (Req2-Ev3,Libro1)</i>	⇒ <i>ObjectPropertyAssertion(domain-pai:especificaTipoExistenciasDe domain-pai:Req2-Ev3 domain-pai:Libro1)</i>
<i>during(Req2-Ev3,Req2-DTI3)</i>	⇒ <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req2-Ev3 domain-pai:Req2-DTI3)</i>
<i>hasBeginning(Req2-DTI3,Req2-DTI3-Sins)</i>	⇒ <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req2-DTI3 domain-pai:Req2-DTI3-Sins)</i>
<i>hasEnd(Req2-DTI3,Req2-DTI3-Eins)</i>	⇒ <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req2-DTI3 domain-pai:Req2-DTI3-Eins)</i>
}	
A₄: [XXXX (Empleado ⊆ ∃obtieneExistenciasDe.Articulo)] : El empleado Valentino obtiene la cantidad de existencias especificada del artículo Libro1.	
A ₄ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	⇒ <i>ClassAssertion(domain-pai:Empleado domain-pai:Valentino)</i>
<i>Articulo(Libro1)</i>	⇒ <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro1)</i>
<i>Event(Req2-Ev4)</i>	⇒ <i>ClassAssertion(timeEvents:Event domain-pai:Req2-Ev4)</i>
<i>DateTimeInterval(Req2-DTI4)</i>	⇒ <i>ClassAssertion(time:DateTimeInterval domain-</i>

Tabla B.10 ABox de casos de éxito en el RF Realizar seguimiento de cantidades de artículos en existencias. (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Instant(Req2-DTI4-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req2-DTI4-Sins)</i>
<i>Instant(Req2-DTI4-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req2-DTI4-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>obtieneExistenciasDe(Valentino,Req2-Ev4)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:obtieneExistenciasDe domain-pai:Valentino domain-pai:Req2-Ev4)</i>
<i>obtieneExistenciasDe(Req2-Ev4,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:obtieneExistenciasDe domain-pai:Req2-Ev4 domain-pai:Libro1)</i>
<i>during(Req2-Ev4,Req2-DTI4)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req2-Ev4 domain-pai:Req2-DTI4)</i>
<i>hasBeginning(Req2-DTI4,Req2-DTI4-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req2-DTI4 domain-pai:Req2-DTI4-Sins)</i>
<i>hasEnd(Req2-DTI4,Req2-DTI4-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req2-DTI4 domain-pai:Req2-DTI4-Eins)</i>
}	
Caso de éxito 2 (A₅, A₆, A₇, A₈):	
A ₅ : [<i>X (Empleado</i> \sqsubseteq \exists <i>buscaExistenciasDe.Articulo)</i>]: El empleado Orlando busca existencias del artículo Libro2.	
A ₅ = {	
-- Aserciones de clase --	
<i>Empleado(Orlando)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Empleado domain-pai:Orlando)</i>
<i>Articulo(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro2)</i>
<i>Event(Req2-Ev5)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req2-Ev5)</i>
<i>DateTimeInterval(Req2-DTI5)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req2-DTI5)</i>
<i>Instant(Req2-DTI5-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req2-DTI5-Sins)</i>
<i>Instant(Req2-DTI5-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req2-DTI5-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>buscaExistenciasDe(Orlando,Req2-Ev5)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:buscaExistenciasDe domain-pai:Orlando domain-pai:Req2-Ev5)</i>
<i>buscaExistenciasDe(Req2-Ev5,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:buscaExistenciasDe domain-pai:Req2-Ev5 domain-pai:Libro2)</i>
<i>during(Req2-Ev5,Req2-DTI5)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req2-Ev5 domain-pai:Req2-DTI5)</i>

Tabla B.10 ABox de casos de éxito en el RF Realizar seguimiento de cantidades de artículos en existencias. (cont.)

Sintaxis DL	Sintaxis Funcional
<i>hasBeginning</i> (Req2-DTI5,Req2-DTI5-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-pai:Req2-DTI5 domain-pai:Req2-DTI5-Sins)
<i>hasEnd</i> (Req2-DTI5,Req2-DTI5-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-pai:Req2-DTI5 domain-pai:Req2-DTI5-Eins)
}	\Rightarrow
A₆ : [XX (<i>Empleado</i> \sqsubseteq \exists verificaExistenciaDe.Articulo)] : El empleado Orlando verifica la existencia del artículo Libro2.	
A ₆ = {	
-- Aserciones de clase --	
<i>Empleado</i> (Orlando)	\Rightarrow <i>ClassAssertion</i> (domain-pai:Empleado domain-pai:Orlando)
<i>Articulo</i> (Libro2)	\Rightarrow <i>ClassAssertion</i> (domain-pai:Articulo domain-pai:Libro2)
<i>Event</i> (Req2-Ev6)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-pai:Req2-Ev6)
<i>DateTimeInterval</i> (Req2-DTI6)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-pai:Req2-DTI6)
<i>Instant</i> (Req2-DTI6-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-pai:Req2-DTI6-Sins)
<i>Instant</i> (Req2-DTI6-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-pai:Req2-DTI6-Eins)
-- Aserciones de propiedades de objeto --	
<i>verificaExistenciaDe</i> (Orlando,Req2-Ev6)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-pai:verificaExistenciaDe domain-pai:Orlando domain-pai:Req2-Ev6)
<i>verificaExistenciaDe</i> (Req2-Ev6,Libro2)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-pai:verificaExistenciaDe domain-pai:Req2-Ev6 domain-pai:Libro2)
<i>during</i> (Req2-Ev6,Req2-DTI6)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-pai:Req2-Ev6 domain-pai:Req2-DTI6)
<i>hasBeginning</i> (Req2-DTI6,Req2-DTI6-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-pai:Req2-DTI6 domain-pai:Req2-DTI6-Sins)
<i>hasEnd</i> (Req2-DTI6,Req2-DTI6-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-pai:Req2-DTI6 domain-pai:Req2-DTI6-Eins)
}	
A₇ : [XXX (<i>Empleado</i> \sqsubseteq \exists especificaTipoExistenciasDe.Articulo)] : El empleado Orlando especifica el tipo de existencias (enMano, enOrden, reservadas) del artículo Libro2.	
A ₇ = {	
-- Aserciones de clase --	
<i>Empleado</i> (Orlando)	\Rightarrow <i>ClassAssertion</i> (domain-pai:Empleado domain-pai:Orlando)
<i>Articulo</i> (Libro2)	\Rightarrow <i>ClassAssertion</i> (domain-pai:Articulo domain-pai:Libro2)
<i>Event</i> (Req2-Ev7)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-

Tabla B.10 ABox de casos de éxito en el RF Realizar seguimiento de cantidades de artículos en existencias. (cont.)

Sintaxis DL	Sintaxis Funcional
<i>DateTimeInterval(Req2-DTI7)</i>	<i>pai:Req2-Ev7</i> ⇒ <i>ClassAssertion(time:DateTimeInterval domain-pai:Req2-DTI7)</i>
<i>Instant(Req2-DTI7-Sins)</i>	⇒ <i>ClassAssertion(time:Instant domain-pai:Req2-DTI7-Sins)</i>
<i>Instant(Req2-DTI7-Eins)</i>	⇒ <i>ClassAssertion(time:Instant domain-pai:Req2-DTI7-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>especificaTipoExistenciasDe(Orlando,Req2-Ev7)</i>	⇒ <i>ObjectPropertyAssertion(domain-pai:especificaTipoExistenciasDe domain-pai:Orlando domain-pai:Req2-Ev7)</i>
<i>especificaTipoExistenciasDe (Req2-Ev7,Libro2)</i>	⇒ <i>ObjectPropertyAssertion(domain-pai:especificaTipoExistenciasDe domain-pai:Req2-Ev7 domain-pai:Libro2)</i>
<i>during(Req2-Ev7,Req2-DTI7)</i>	⇒ <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req2-Ev7 domain-pai:Req2-DTI7)</i>
<i>hasBeginning(Req2-DTI7,Req2-DTI7-Sins)</i>	⇒ <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req2-DTI7 domain-pai:Req2-DTI7-Sins)</i>
<i>hasEnd(Req2-DTI7,Req2-DTI7-Eins)</i>	⇒ <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req2-DTI7 domain-pai:Req2-DTI7-Eins)</i>
}	
A₈: [XXXX (Empleado \sqsubseteq \existsobtieneExistenciasDe.Articulo)]: El empleado Orlando obtiene la cantidad de existencias especificada del artículo Libro2.	
A ₈ = {	
-- Aserciones de clase --	
<i>Empleado(Orlando)</i>	⇒ <i>ClassAssertion(domain-pai:Empleado domain-pai:Orlando)</i>
<i>Articulo(Libro2)</i>	⇒ <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro2)</i>
<i>Event(Req2-Ev8)</i>	⇒ <i>ClassAssertion(timeEvents:Event domain-pai:Req2-Ev8)</i>
<i>DateTimeInterval(Req2-DTI8)</i>	⇒ <i>ClassAssertion(time:DateTimeInterval domain-pai:Req2-DTI8)</i>
<i>Instant(Req2-DTI8-Sins)</i>	⇒ <i>ClassAssertion(time:Instant domain-pai:Req2-DTI8-Sins)</i>
<i>Instant(Req2-DTI8-Eins)</i>	⇒ <i>ClassAssertion(time:Instant domain-pai:Req2-DTI8-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>obtieneExistenciasDe(Orlando,Req2-Ev8)</i>	⇒ <i>ObjectPropertyAssertion(domain-pai:obtieneExistenciasDe domain-pai:Orlando domain-pai:Req2-Ev8)</i>
<i>obtieneExistenciasDe(Req2-Ev8,Libro2)</i>	⇒ <i>ObjectPropertyAssertion(domain-pai:obtieneExistenciasDe domain-pai:Req2-Ev8 domain-pai:Libro2)</i>
<i>during(Req2-Ev8,Req2-DTI8)</i>	⇒ <i>ObjectPropertyAssertion(timeEvents:during</i>

Tabla B.10 ABox de casos de éxito en el RF Realizar seguimiento de cantidades de artículos en existencias.
(cont.)

Sintaxis DL	Sintaxis Funcional
$hasBeginning(Req2-DTI8, Req2-DTI8-Sins)$ $hasEnd(Req2-DTI8, Req2-DTI8-Eins)$ }	$domain-pai:Req2-Ev8 domain-pai:Req2-DTI8)$ $\Rightarrow ObjectPropertyAssertion(time:hasBeginning$ $domain-pai:Req2-DTI8 domain-pai:Req2-DTI8-$ $Sins)$ $\Rightarrow ObjectPropertyAssertion(time:hasEnd domain-$ $pai:Req2-DTI8 domain-pai:Req2-DTI8-Eins)$
<p>Caso de éxito 3 (A₉, A₁₀, A₁₁, A₁₂):</p> <p>A₉: [X (Empleado \sqsubseteq \existsbuscaExistenciasDe.Articulo)]: El empleado Julia busca existencias del artículo Libro3. ... A₁₀: [XX (Empleado \sqsubseteq \existsverificaExistenciaDe.Articulo)]: El empleado Julia verifica la existencia del artículo Libro3. ... A₁₁: [XXX (Empleado \sqsubseteq \existsespecificaTipoExistenciasDe.Articulo)]: El empleado Julia especifica el tipo de existencias (enMano, enOrden, reservadas) del artículo Libro3. ... A₁₂: [XXXX (Empleado \sqsubseteq \existsobtieneExistenciasDe.Articulo)]: El empleado Julia obtiene la cantidad de existencias especificada del artículo Libro3. ...</p>	
<p>Caso de éxito 4 (A₁₃, A₁₄, A₁₅, A₁₆):</p> <p>A₁₃: [X (Empleado \sqsubseteq \existsbuscaExistenciasDe.Articulo)]: El empleado Juan busca existencias del artículo Libro4. ... A₁₄: [XX (Empleado \sqsubseteq \existsverificaExistenciaDe.Articulo)]: El empleado Juan verifica la existencia del artículo Libro4. ... A₁₅: [XXX (Empleado \sqsubseteq \existsespecificaTipoExistenciasDe.Articulo)]: El empleado Juan especifica el tipo de existencias (enMano, enOrden, reservadas) del artículo Libro4. ... A₁₆: [XXXX (Empleado \sqsubseteq \existsobtieneExistenciasDe.Articulo)]: El empleado Juan obtiene la cantidad de existencias especificada del artículo Libro4. ...</p>	
<p>Caso de éxito n</p> <p>A_n ...</p>	

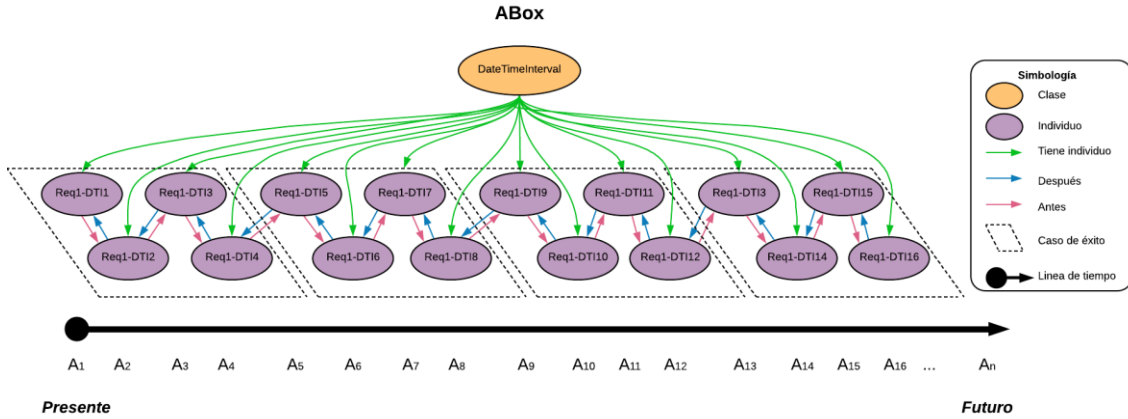


Figura B.6 SoIs en la ABox de casos de éxito de acuerdo al RF Realizar seguimiento de cantidades de artículos en existencias.

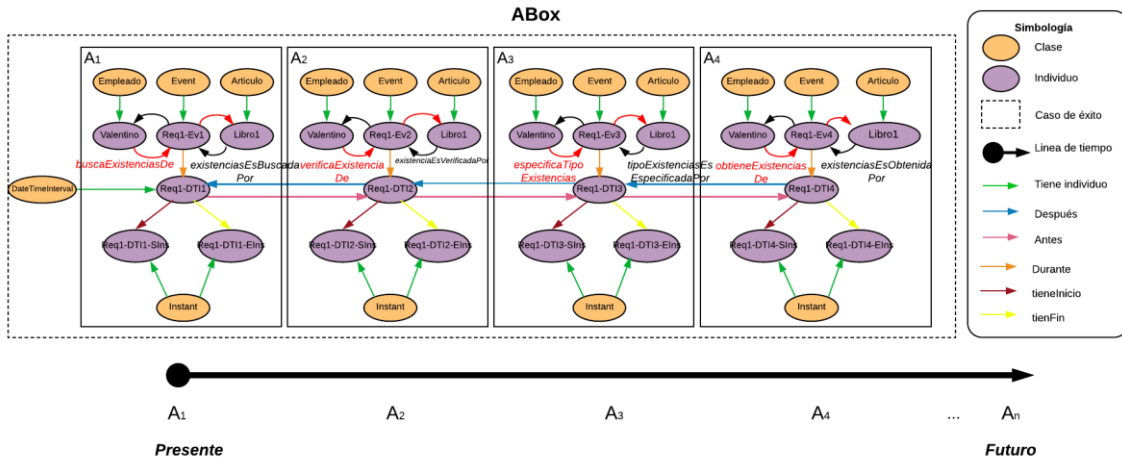


Figura B.7 Caso de éxito 1 en la ABox de casos de éxito de acuerdo al RF Realizar seguimiento de las cantidades de artículos en existencias.

1.2.2.2. Casos de éxito y fracaso

En la Tabla B.11 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la *ABox*) basadas en casos de éxito y fracaso, que conforman una secuencia que representa el comportamiento dinámico durante el seguimiento de la cantidad de existencias del artículo Libro1 de un inventario. Una representación gráfica simplificada de la secuencia se muestra en la Figura B.8. A manera de ejemplo, en la Figura B.9 se muestra el detalle de las aserciones A_1 , A_2 , A_3 y A_4 en el caso de fracaso número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de razonamiento automatizado los demás casos se definen de manera similar, sin embargo, difieren en los individuos involucrados.

En la Tabla B.12 se describen las reglas SWRL utilizadas para restringir la existencia de afirmaciones de propiedades de objeto en el intervalo de tiempo correspondiente.

Tabla B.11 ABox de casos de éxito y fracaso en el RF Realizar seguimiento de cantidades de artículos en existencias (cont.)

Sintaxis DL	Sintaxis Funcional
Caso de fracaso 1 (A_1, A_2, A_3, A_4):	
A_1 : [X ($\text{Empleado} \sqsubseteq \exists \text{buscaExistenciasDe.Articulo}$)]: El empleado Valentino busca existencias del artículo Libro1.	
$A_1 = \{$	
-- Aserciones de clase --	
$\text{Empleado}(\text{Valentino})$	$\Rightarrow \text{ClassAssertion}(\text{domain-pai:Empleado domain-pai:Valentino})$
$\text{Articulo}(\text{Libro1})$	$\Rightarrow \text{ClassAssertion}(\text{domain-pai:Articulo domain-pai:Libro1})$
$\text{Event}(\text{Req2-Ev1})$	$\Rightarrow \text{ClassAssertion}(\text{timeEvents:Event domain-pai:Req2-Ev1})$
$\text{DateTimeInterval}(\text{Req2-DT11})$	$\Rightarrow \text{ClassAssertion}(\text{time:DateTimeInterval domain-pai:Req2-DT11})$
$\text{Instant}(\text{Req2-DT11-Sins})$	$\Rightarrow \text{ClassAssertion}(\text{time:Instant domain-pai:Req2-DT11-Sins})$
$\text{Instant}(\text{Req2-DT11-Eins})$	$\Rightarrow \text{ClassAssertion}(\text{time:Instant domain-pai:Req2-DT11-Eins})$
-- Aserciones de propiedades de objeto --	
$\text{buscaExistenciasDe}(\text{Valentino}, \text{Req2-Ev1})$	$\Rightarrow \text{ObjectPropertyAssertion}(\text{domain-pai:buscaExistenciasDe domain-pai:Valentino domain-pai:Req2-Ev1})$
$\text{buscaExistenciasDe}(\text{Req2-Ev1}, \text{Libro1})$	$\Rightarrow \text{ObjectPropertyAssertion}(\text{domain-pai:buscaExistenciasDe domain-pai:Req2-Ev1 domain-pai:Libro1})$
$\text{during}(\text{Req2-Ev1}, \text{Req2-DT11})$	$\Rightarrow \text{ObjectPropertyAssertion}(\text{timeEvents:during domain-pai:Req2-Ev1 domain-pai:Req2-DT11})$
$\text{hasBeginning}(\text{Req2-DT11}, \text{Req2-DT11-Sins})$	$\Rightarrow \text{ObjectPropertyAssertion}(\text{time:hasBeginning domain-pai:Req2-DT11 domain-pai:Req2-DT11-Sins})$
$\text{hasEnd}(\text{Req2-DT11}, \text{Req2-DT11-Eins})$	$\Rightarrow \text{ObjectPropertyAssertion}(\text{time:hasEnd domain-pai:Req2-DT11 domain-pai:Req2-DT11-Eins})$
$\}$	
A_2 : [XX ($\text{Empleado} \sqsubseteq \exists \text{verificaExistenciaDe.Articulo}$)]: El empleado Valentino verifica la existencia del artículo Libro1	
$A_2 = \{$	
-- Aserciones de clase --	
$\text{Empleado}(\text{Valentino})$	$\Rightarrow \text{ClassAssertion}(\text{domain-pai:Empleado domain-pai:Valentino})$
$\text{Articulo}(\text{Libro1})$	$\Rightarrow \text{ClassAssertion}(\text{domain-pai:Articulo domain-pai:Libro1})$
$\text{Event}(\text{Req2-Ev2})$	$\Rightarrow \text{ClassAssertion}(\text{timeEvents:Event domain-pai:Req2-}$

Tabla B.11 ABox de casos de éxito y fracaso en el RF Realizar seguimiento de cantidades de artículos en existencias (cont.)

Sintaxis DL	Sintaxis Funcional
	Ev2)
<i>DateTimeInterval(Req2-DTI2)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req2-DTI2)</i>
<i>Instant(Req2-DTI2-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req2-DTI2-Sins)</i>
<i>Instant(Req2-DTI2-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req2-DTI2-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>verificaExistenciaDe(Valentino,Req2-Ev2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:verificaExistenciaDe domain-pai:Valentino domain-pai:Req2-Ev2)</i>
<i>verificaExistenciaDe(Req2-Ev2,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:verificaExistenciaDe domain-pai:Req2-Ev2 domain-pai:Libro1)</i>
<i>during(Req2-Ev2,Req2-DTI2)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req2-Ev2 domain-pai:Req2-DTI2)</i>
<i>hasBeginning(Req2-DTI2,Req2-DTI2-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req2-DTI2 domain-pai:Req2-DTI2-Sins)</i>
<i>hasEnd(Req2-DTI2,Req2-DTI2-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req2-DTI2 domain-pai:Req2-DTI2-Eins)</i>
}	
A₃: [\negXXX (Empleado \sqsubseteq \existsespecificaTipoExistenciasDe.Articulo)]: El empleado Valentino no especifica el tipo de existencias (enMano, enOrden, reservadas) del articulo Libro1.	
A ₃ = {	
-- Aserciones de clase --	
<i>Event(Req2-Ev3)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req2-Ev3)</i>
<i>DateTimeInterval(Req2-DTI3)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req2-DTI3)</i>
<i>Instant(Req2-DTI3-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req2-DTI3-Sins)</i>
<i>Instant(Req2-DTI3-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req2-DTI3-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>during(Req2-Ev3,Req2-DTI3)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req2-Ev3 domain-pai:Req2-DTI3)</i>
<i>hasBeginning(Req2-DTI3,Req2-DTI3-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req2-DTI3 domain-pai:Req2-DTI3-Sins)</i>
<i>hasEnd(Req2-DTI3,Req2-DTI3-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req2-DTI3 domain-pai:Req2-DTI3-Eins)</i>
}	
A₄: [\negXXXX (Empleado \sqsubseteq \existsobtieneExistenciasDe.Articulo)]: El empleado Valentino no obtiene la cantidad de existencias especificada del articulo Libro1.	
A ₄ = {	

Tabla B.11 ABox de casos de éxito y fracaso en el RF Realizar seguimiento de cantidades de artículos en existencias (cont.)

Sintaxis DL	Sintaxis Funcional
-- Aserciones de clase --	
<i>Event(Req2-Ev4)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req2-Ev4)</i>
<i>DateTimeInterval(Req2-DTI4)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req2-DTI4)</i>
<i>Instant(Req2-DTI4-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req2-DTI4-Sins)</i>
<i>Instant(Req2-DTI4-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req2-DTI4-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>during(Req2-Ev4,Req2-DTI4)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req2-Ev4 domain-pai:Req2-DTI4)</i>
<i>hasBeginning(Req2-DTI4,Req2-DTI4-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req2-DTI4 domain-pai:Req2-DTI4-Sins)</i>
<i>hasEnd(Req2-DTI4,Req2-DTI4-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req2-DTI4 domain-pai:Req2-DTI4-Eins)</i>
}	
Caso de éxito 1 (A₅, A₆, A₇, A₈):	
<i>A₅: [X (Empleado \sqsubseteq \existsbuscaExistenciasDe.Articulo)]: El empleado Orlando busca existencias del articulo Libro2.</i>	
<i>A₅ = {</i>	
-- Aserciones de clase --	
<i>Empleado(Orlando)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Empleado domain-pai:Orlando)</i>
<i>Articulo(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro2)</i>
<i>Event(Req2-Ev5)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req2-Ev5)</i>
<i>DateTimeInterval(Req2-DTI5)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req2-DTI5)</i>
<i>Instant(Req2-DTI5-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req2-DTI5-Sins)</i>
<i>Instant(Req2-DTI5-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req2-DTI5-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>buscaExistenciasDe(Orlando,Req2-Ev5)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:buscaExistenciasDe domain-pai:Orlando domain-pai:Req2-Ev5)</i>
<i>buscaExistenciasDe(Req2-Ev5,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:buscaExistenciasDe domain-pai:Req2-Ev5 domain-pai:Libro2)</i>
<i>during(Req2-Ev5,Req2-DTI5)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req2-Ev5 domain-pai:Req2-DTI5)</i>
<i>hasBeginning(Req2-DTI5,Req2-DTI5-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req2-DTI5 domain-pai:Req2-DTI5-Sins)</i>

Tabla B.11 ABox de casos de éxito y fracaso en el RF Realizar seguimiento de cantidades de artículos en existencias (cont.)

Sintaxis DL	Sintaxis Funcional
$hasEnd(Req2-DTI5, Req2-DTI5-Eins)$ }	$Sins)$ $\Rightarrow ObjectPropertyAssertion(time:hasEnd domain-pai:Req2-DTI5 domain-pai:Req2-DTI5-Eins)$
$A_6: [XX (Empleado \sqsubseteq \exists verificaExistenciaDe.Articulo)]$: El empleado Orlando verifica la existencia del artículo Libro2 $A_6 = \{$	
-- Aserciones de clase -- $Empleado(Orlando)$	$\Rightarrow ClassAssertion(domain-pai:Empleado domain-pai:Orlando)$
$Articulo(Libro2)$	$\Rightarrow ClassAssertion(domain-pai:Articulo domain-pai:Libro2)$
$Event(Req2-Ev6)$	$\Rightarrow ClassAssertion(timeEvents:Event domain-pai:Req2-Ev6)$
$DateTimeInterval(Req2-DTI6)$	$\Rightarrow ClassAssertion(time:DateTimeInterval domain-pai:Req2-DTI6)$
$Instant(Req2-DTI6-Sins)$	$\Rightarrow ClassAssertion(time:Instant domain-pai:Req2-DTI6-Sins)$
$Instant(Req2-DTI6-Eins)$	$\Rightarrow ClassAssertion(time:Instant domain-pai:Req2-DTI6-Eins)$
-- Aserciones de propiedades de objeto -- $verificaExistenciaDe(Orlando, Req2-Ev6)$	$\Rightarrow ObjectPropertyAssertion(domain-pai:verificaExistenciaDe domain-pai:Orlando domain-pai:Req2-Ev6)$
$verificaExistenciaDe(Req2-Ev6, Libro2)$	$\Rightarrow ObjectPropertyAssertion(domain-pai:verificaExistenciaDe domain-pai:Req2-Ev6 domain-pai:Libro2)$
$during(Req2-Ev6, Req2-DTI6)$	$\Rightarrow ObjectPropertyAssertion(timeEvents:during domain-pai:Req2-Ev6 domain-pai:Req2-DTI6)$
$hasBeginning(Req2-DTI6, Req2-DTI6-Sins)$	$\Rightarrow ObjectPropertyAssertion(time:hasBeginning domain-pai:Req2-DTI6 domain-pai:Req2-DTI6-Sins)$
$hasEnd(Req2-DTI6, Req2-DTI6-Eins)$	$\Rightarrow ObjectPropertyAssertion(time:hasEnd domain-pai:Req2-DTI6 domain-pai:Req2-DTI6-Eins)$
} $A_7: [XXX (Empleado \sqsubseteq \exists especificaTipoExistenciasDe.Articulo)]$: El empleado Orlando especifica el tipo de existencias (enMano, enOrden, reservadas) del artículo Libro2. $A_7 = \{$	
-- Aserciones de clase -- $Empleado(Orlando)$	$\Rightarrow ClassAssertion(domain-pai:Empleado domain-pai:Orlando)$
$Articulo(Libro2)$	$\Rightarrow ClassAssertion(domain-pai:Articulo domain-pai:Libro2)$
$Event(Req2-Ev7)$	$\Rightarrow ClassAssertion(timeEvents:Event domain-pai:Req2-Ev7)$
$DateTimeInterval(Req2-DTI7)$	$\Rightarrow ClassAssertion(time:DateTimeInterval domain-$

Tabla B.11 ABox de casos de éxito y fracaso en el RF Realizar seguimiento de cantidades de artículos en existencias (cont.)

Sintaxis DL	Sintaxis Funcional
	<i>pai:Req2-DTI7)</i>
<i>Instant(Req2-DTI7-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req2-DTI7-Sins)</i>
<i>Instant(Req2-DTI7-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req2-DTI7-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>especificaTipoExistenciasDe(Orlando,Req2-Ev7)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:especificaTipoExistenciasDe domain-pai:Orlando domain-pai:Req2-Ev7)</i>
<i>especificaTipoExistenciasDe (Req2-Ev7,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:especificaTipoExistenciasDe domain-pai:Req2-Ev7 domain-pai:Libro2)</i>
<i>during(Req2-Ev7,Req2-DTI7)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req2-Ev7 domain-pai:Req2-DTI7)</i>
<i>hasBeginning(Req2-DTI7,Req2-DTI7-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req2-DTI7 domain-pai:Req2-DTI7-Sins)</i>
<i>hasEnd(Req2-DTI7,Req2-DTI7-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req2-DTI7 domain-pai:Req2-DTI7-Eins)</i>
}	
<i>A₈: [XXXX (Empleado \sqsubseteq \existsobtieneExistenciasDe.Articulo)]: El empleado Orlando obtiene la cantidad de existencias especificada del articulo Libro2.</i>	
<i>A₈ = {</i>	
-- Aserciones de clase --	
<i>Empleado(Orlando)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Empleado domain-pai:Orlando)</i>
<i>Articulo(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro2)</i>
<i>Event(Req2-Ev8)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req2-Ev8)</i>
<i>DateTimeInterval(Req2-DTI8)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req2-DTI8)</i>
<i>Instant(Req2-DTI8-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req2-DTI8-Sins)</i>
<i>Instant(Req2-DTI8-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req2-DTI8-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>obtieneExistenciasDe(Orlando,Req2-Ev8)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:obtieneExistenciasDe domain-pai:Orlando domain-pai:Req2-Ev8)</i>
<i>obtieneExistenciasDe(Req2-Ev8,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:obtieneExistenciasDe domain-pai:Req2-Ev8 domain-pai:Libro2)</i>
<i>during(Req2-Ev8,Req2-DTI8)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req2-Ev8 domain-pai:Req2-DTI8)</i>
<i>hasBeginning(Req2-DTI8,Req2-DTI8-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning</i>

Tabla B.11 ABox de casos de éxito y fracaso en el RF Realizar seguimiento de cantidades de artículos en existencias (cont.)

Sintaxis DL	Sintaxis Funcional
$hasEnd(Req2-DTI8, Req2-DTI8-Eins)$ }	$domain-pai:Req2-DTI8 \text{ domain-pai:Req2-DTI8-Sins}$ $\Rightarrow ObjectPropertyAssertion(time:hasEnd \text{ domain-pai:Req2-DTI8} \text{ domain-pai:Req2-DTI8-Eins})$
<p>Caso de fracaso 2 (A₉, A₁₀, A₁₁, A₁₂):</p> <p>A₉: [<i>X</i> (Empleado $\sqsubseteq \exists buscaExistenciasDe.Articulo$): El empleado Julia busca existencias del artículo Libro3. ... A₁₀: [<i>XX</i> (Empleado $\sqsubseteq \exists verificaExistenciaDe.Articulo$): El empleado Julia verifica la existencia del artículo Libro3. ... A₁₁: [<i>¬XXX</i> (Empleado $\sqsubseteq \exists especificaTipoExistenciasDe.Articulo$): El empleado Julia no especifica el tipo de existencias (enMano, enOrden, reservadas) del artículo Libro3. ... A₁₂: [<i>¬XXXX</i> (Empleado $\sqsubseteq \exists obtieneExistenciasDe.Articulo$): El empleado Julia no obtiene la cantidad de existencias especificada del artículo Libro3.</p>	
<p>Caso de éxito 2 (A₁₃, A₁₄, A₁₅, A₁₆):</p> <p>A₁₃: [<i>X</i> (Empleado $\sqsubseteq \exists buscaExistenciasDe.Articulo$): El empleado Juan busca existencias del artículo Libro4. ... A₁₄: [<i>XX</i> (Empleado $\sqsubseteq \exists verificaExistenciaDe.Articulo$): El empleado Juan verifica la existencia del artículo Libro4. ... A₁₅: [<i>XXX</i> (Empleado $\sqsubseteq \exists especificaTipoExistenciasDe.Articulo$): El empleado Julia especifica el tipo de existencias (enMano, enOrden, reservadas) del artículo Libro4. ... A₁₆: [<i>XXXX</i> (Empleado $\sqsubseteq \exists obtieneExistenciasDe.Articulo$): El empleado Julia obtiene la cantidad de existencias especificada del artículo Libro4.</p>	
<p>Caso de éxito o fracaso n</p> <p>A_n ...</p>	

Tabla B.12 Reglas SWRL para la SoIs de casos de éxito y fracaso de acuerdo al RF Realizar seguimiento de cantidades de artículos en existencias (cont.)

ABox	Regla SWRL
A ₃	$Event(?e), \text{ during}(?e, ?i), \text{ overlaps}(?i, Req2-DTI3), \text{ especificaTipoExistenciasDe}(?e, Libro1), \text{ especificaTipoExistenciasDe}(Valentino, ?e) \rightarrow \text{ Nothing}(Valentino)$
A ₄	$Event(?e), \text{ during}(?e, ?i), \text{ overlaps}(?i, Req2-DTI4), \text{ obtieneExistenciasDe}(?e, Libro1), \text{ obtieneExistenciasDe}(Valentino, ?e) \rightarrow \text{ Nothing}(Valentino)$
A ₁₁	$Event(?e), \text{ during}(?e, ?i), \text{ overlaps}(?i, Req2-DTI1), \text{ especificaTipoExistenciasDe}(?e, Libro3), \text{ especificaTipoExistenciasDe}(Julia, ?e) \rightarrow \text{ Nothing}(Julia)$

Tabla B.12 Reglas SWRL para la SoIs de casos de éxito y fracaso de acuerdo al RF Realizar seguimiento de cantidades de artículos en existencias (cont.)

ABox	Regla SWRL
A12	$Event(?e), \text{during}(?e, ?i), \text{overlaps}(?i, Req2-DT112), \text{obtieneExistenciasDe}(?e, Libro3), \text{obtieneExistenciasDe}(Julia, ?e) \rightarrow Nothing(Julia)$

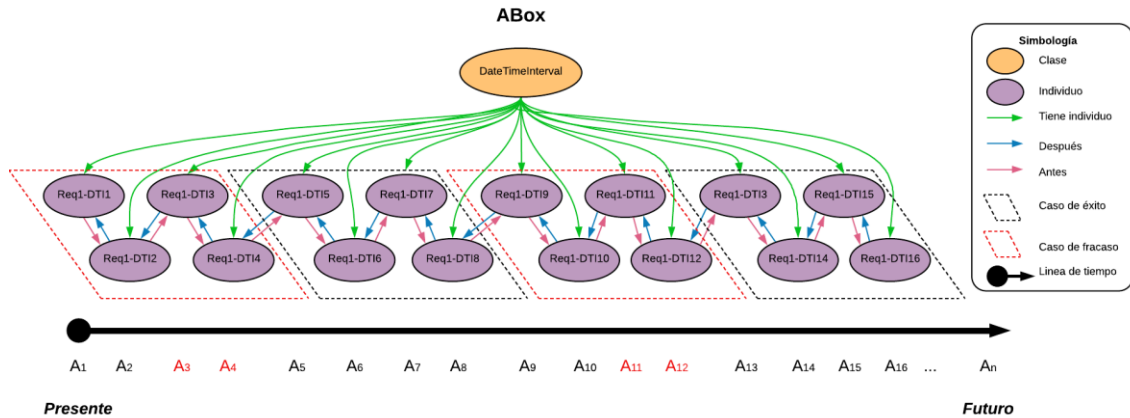


Figura B.8 SoIs en la ABox de casos de éxito y fracaso de acuerdo al RF Realizar seguimiento de cantidades de artículos en existencias.

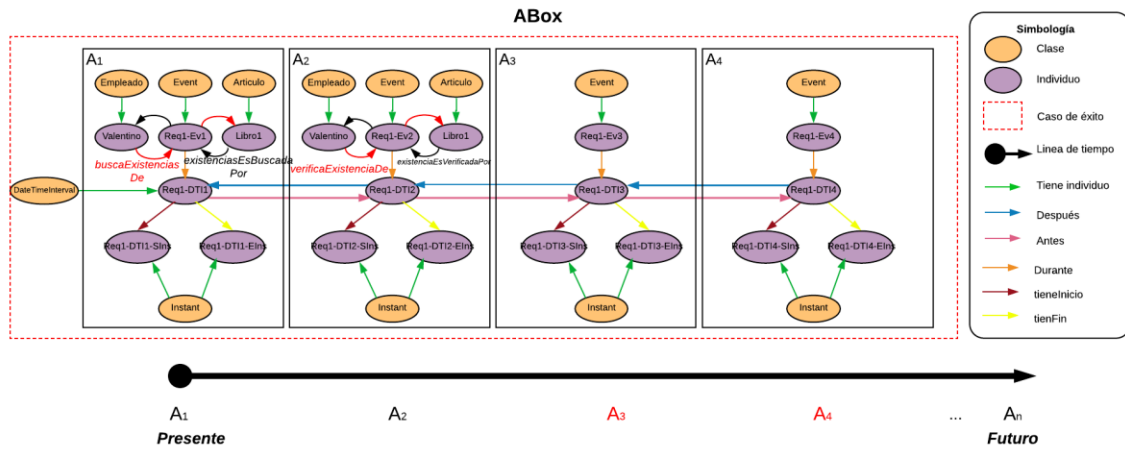


Figura B.9 Caso de fracaso 1 en la ABox de casos de éxito y fracaso de acuerdo al RF Realizar el seguimiento de las cantidades de artículos en existencias.

1.2.3. RF Realizar seguimiento de las ubicaciones de los artículos

Las secuencias de interpretaciones descritas en esta sección satisfacen/cumplen con el comportamiento dinámico del RF Realizar seguimiento de las ubicaciones de los artículos representado formalmente con la fórmula de la Tabla B.13.

Tabla B.13 Fórmula ALC_{LTL} del RF Realizar seguimiento de las ubicaciones de los artículos.

$$\mathcal{G} (\mathcal{X} (Empleado \sqsubseteq \exists \text{ubica.Articulo}) \rightarrow (\mathcal{X}\mathcal{X} (Empleado \sqsubseteq \exists \text{verificaExistenciaDe.Articulo}) \wedge \mathcal{F} (\neg \mathcal{X}\mathcal{X}\mathcal{X} (Empleado \sqsubseteq \exists \text{solicitaUbicacionDe.Articulo}) \wedge \neg \mathcal{X}\mathcal{X}\mathcal{X}\mathcal{X} (Empleado \sqsubseteq \exists \text{obtieneUbicacionDe.Articulo})))))$$

1.2.3.1. Casos de éxito

En la Tabla B.14 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la *ABox*) basadas en casos de éxito, que conforman una secuencia que representa el comportamiento dinámico durante el seguimiento de la ubicación del artículo Libro1 de un inventario. Una representación gráfica simplificada de la secuencia se muestra en la Figura B.10. A manera de ejemplo, en la Figura B.11 se muestra el detalle de las aserciones A_1 , A_2 , A_3 y A_4 en el caso de éxito número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de razonamiento automatizado, los demás casos se definen de manera similar, sin embargo, difieren en los individuos involucrados.

Tabla B.14 ABox de casos de éxito en el RF Realizar seguimiento de las ubicaciones de los artículos. (cont.)

Sintaxis DL	Sintaxis Funcional
Caso de éxito 1 (A_1, A_2, A_3, A_4):	
$A_1: [X (Empleado \sqsubseteq \exists \text{ubica.Articulo})]:$ El empleado Valentino ubica el artículo Libro1.	
$A_1 = \{$	
-- Aserciones de clase --	
$Empleado(Valentino)$	\Rightarrow $ClassAssertion(domain-pai:Empleado domain-pai:Valentino)$
$Articulo(Libro1)$	\Rightarrow $ClassAssertion(domain-pai:Articulo domain-pai:Libro1)$
$Event(Req3-Ev1)$	\Rightarrow $ClassAssertion(timeEvents:Event domain-pai:Req3-Ev1)$
$DateTimeInterval(Req3-DT11)$	\Rightarrow $ClassAssertion(time:DateTimeInterval domain-pai:Req3-DT11)$
$Instant(Req3-DT11-Sins)$	\Rightarrow $ClassAssertion(time:Instant domain-pai:Req3-DT11-Sins)$
$Instant(Req3-DT11-Eins)$	\Rightarrow $ClassAssertion(time:Instant domain-pai:Req3-DT11-Eins)$
-- Aserciones de propiedades de objeto --	
$ubica(Valentino,Req3-Ev1)$	\Rightarrow $ObjectPropertyAssertion(domain-pai: ubica domain-pai:Valentino domain-pai:Req3-Ev1)$
$ubica(Req3-Ev1,Libro1)$	\Rightarrow $ObjectPropertyAssertion(domain-pai: ubica domain-pai:Req3-Ev1 domain-pai:Libro1)$
$during(Req3-Ev1,Req3-DT11)$	\Rightarrow $ObjectPropertyAssertion(timeEvents:during domain-$

Tabla B.14 ABox de casos de éxito en el RF Realizar seguimiento de las ubicaciones de los artículos. (cont.)

Sintaxis DL	Sintaxis Funcional
<i>hasBeginning</i> (Req3-DT11,Req3-DT11-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-pai:Req3-DT11 domain-pai:Req3-DT11-Sins)
<i>hasEnd</i> (Req3-DT11,Req3-DT11-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-pai:Req3-DT11 domain-pai:Req3-DT11-Eins)
}	
<i>A₂</i> : [XX (Empleado \sqsubseteq \exists verificaExistenciaDe.Articulo)]: El empleado Valentino verifica la existencia del artículo Libro1.	
<i>A₂</i> = {	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-pai:Empleado domain-pai:Valentino)
<i>Articulo</i> (Libro1)	\Rightarrow <i>ClassAssertion</i> (domain-pai:Articulo domain-pai:Libro1)
<i>Event</i> (Req3-Ev2)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-pai:Req3-Ev2)
<i>DateTimeInterval</i> (Req3-DT12)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-pai:Req3-DT12)
<i>Instant</i> (Req3-DT12-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-pai:Req3-DT12-Sins)
<i>Instant</i> (Req3-DT12-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-pai:Req3-DT12-Eins)
-- Aserciones de propiedades de objeto --	
<i>verificaExistenciaDe</i> (Valentino,R	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-pai:verificaExistenciaDe domain-pai:Valentino domain-pai:Req3-Ev2)
<i>verificaExistenciaDe</i> (Req3-Ev2,Libro1)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-pai:verificaExistenciaDe domain-pai:Req3-Ev2 domain-pai:Libro1)
<i>during</i> (Req3-Ev2,Req3-DT12)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-pai:Req3-Ev2 domain-pai:Req3-DT12)
<i>hasBeginning</i> (Req3-DT12,Req3-DT12-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-pai:Req3-DT12 domain-pai:Req3-DT12-Sins)
<i>hasEnd</i> (Req3-DT12,Req3-DT12-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-pai:Req3-DT12 domain-pai:Req3-DT12-Eins)
}	
<i>A₃</i> : [XXX (Empleado \sqsubseteq \exists solicitaUbicacionDe.Articulo)]: El empleado Valentino solicita al inventario la ubicacion del articulo Libro1.	
<i>A₃</i> = {	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-pai:Empleado domain-pai:Valentino)
<i>Articulo</i> (Libro1)	\Rightarrow <i>ClassAssertion</i> (domain-pai:Articulo domain-pai:Libro1)
<i>Event</i> (Req3-Ev3)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-pai:Req3-Ev3)

Tabla B.14 ABox de casos de éxito en el RF Realizar seguimiento de las ubicaciones de los artículos. (cont.)

Sintaxis DL	Sintaxis Funcional
<i>DateTimeInterval(Req3-DTI3)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req3-DTI3)</i>
<i>Instant(Req3-DTI3-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req3-DTI3-Sins)</i>
<i>Instant(Req3-DTI3-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req3-DTI3-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>solicitaUbicacionDe(Valentino,Req3-Ev3)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:solicitaUbicacionDe domain-pai:Valentino domain-pai:Req3-Ev3)</i>
<i>solicitaUbicacionDe(Req3-Ev3,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:solicitaUbicacionDe domain-pai:Req3-Ev3 domain-pai:Libro1)</i>
<i>during(Req3-Ev3,Req3-DTI3)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req3-Ev3 domain-pai:Req3-DTI3)</i>
<i>hasBeginning(Req3-DTI3,Req3-DTI3-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req3-DTI3 domain-pai:Req3-DTI3-Sins)</i>
<i>hasEnd(Req3-DTI3,Req3-DTI3-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req3-DTI3 domain-pai:Req3-DTI3-Eins)</i>
}	
A₄: [XXXX (Empleado \sqsubseteq \existsobtieneUbicacionDe.Articulo)]: El empleado Valentino obtiene la ubicación del artículo Libro1.	
A ₄ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Empleado domain-pai:Valentino)</i>
<i>Articulo(Libro1)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro1)</i>
<i>Event(Req3-Ev4)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req3-Ev4)</i>
<i>DateTimeInterval(Req3-DTI4)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req3-DTI4)</i>
<i>Instant(Req3-DTI4-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req3-DTI4-Sins)</i>
<i>Instant(Req3-DTI4-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req3-DTI4-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>obtieneUbicacionDe(Valentino,Req3-Ev4)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:obtieneUbicacionDe domain-pai:Valentino domain-pai:Req3-Ev4)</i>
<i>obtieneUbicacionDe(Req3-Ev4,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:obtieneUbicacionDe domain-pai:Req3-Ev4 domain-pai:Libro1)</i>
<i>during(Req3-Ev4,Req3-DTI4)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req3-Ev4 domain-pai:Req3-DTI4)</i>
<i>hasBeginning(Req3-DTI4,Req3-DTI4-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req3-DTI4 domain-pai:Req3-DTI4-Sins)</i>

Tabla B.14 ABox de casos de éxito en el RF Realizar seguimiento de las ubicaciones de los artículos. (cont.)

Sintaxis DL	Sintaxis Funcional
<i>hasEnd(Req3-DTI4,Req3-DTI4-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req3-DTI4 domain-pai:Req3-DTI4-Eins)</i>
}	
Caso de éxito 2 (A₅, A₆, A₇, A₈):	
A₅: [<i>X (Empleado \sqsubseteq \existsubica.Articulo)</i>]: El empleado Orlando ubica el artículo Libro2.	
A ₅ = {	
-- Aserciones de clase --	
<i>Empleado(Orlando)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Empleado domain-pai:Orlando)</i>
<i>Articulo(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro2)</i>
<i>Event(Req3-Ev5)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req3-Ev5)</i>
<i>DateTimeInterval(Req3-DTI5)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req3-DTI5)</i>
<i>Instant(Req3-DTI5-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req3-DTI5-Sins)</i>
<i>Instant(Req3-DTI5-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req3-DTI5-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>ubica(Orlando,Req3-Ev5)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:ubica domain-pai:Orlando domain-pai:Req3-Ev5)</i>
<i>ubica(Req3-Ev5,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:ubica domain-pai:Req3-Ev5 domain-pai:Libro2)</i>
<i>during(Req3-Ev5,Req3-DTI5)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req3-Ev5 domain-pai:Req3-DTI5)</i>
<i>hasBeginning(Req3-DTI5,Req3-DTI5-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req3-DTI5 domain-pai:Req3-DTI5-Sins)</i>
<i>hasEnd(Req3-DTI5,Req3-DTI5-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req3-DTI5 domain-pai:Req3-DTI5-Eins)</i>
}	
A₆: [<i>XX (Empleado \sqsubseteq \existsverificaExistenciaDe.Articulo)</i>] : El empleado Orlando verifica la existencia del artículo Libro2.	
A ₆ = {	
-- Aserciones de clase --	
<i>Empleado(Orlando)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Empleado domain-pai:Orlando)</i>
<i>Articulo(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro2)</i>
<i>Event(Req3-Ev6)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req3-Ev6)</i>
<i>DateTimeInterval(Req3-DTI6)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req3-DTI6)</i>
<i>Instant(Req3-DTI6-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req3-DTI6-Sins)</i>
<i>Instant(Req3-DTI6-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req3-DTI6-Eins)</i>

Tabla B.14 ABox de casos de éxito en el RF Realizar seguimiento de las ubicaciones de los artículos. (cont.)

Sintaxis DL	Sintaxis Funcional
	Eins)
-- Aserciones de propiedades de objeto --	
<i>verificaExistenciaDe(Orlando,Req3-Ev6)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:verificaExistenciaDe domain-pai:Orlando domain-pai:Req3-Ev6)</i>
<i>verificaExistenciaDe(Req3-Ev6,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:verificaExistenciaDe domain-pai:Req3-Ev6 domain-pai:Libro2)</i>
<i>during(Req3-Ev6,Req3-DTI6)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req3-Ev6 domain-pai:Req3-DTI6)</i>
<i>hasBeginning(Req3-DTI6,Req3-DTI6-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req3-DTI6 domain-pai:Req3-DTI6-Sins)</i>
<i>hasEnd(Req3-DTI6,Req3-DTI6-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req3-DTI6 domain-pai:Req3-DTI6-Eins)</i>
}	
<i>A7: [XXX (Empleado \sqsubseteq \existssolicitaUbicacionDe.Articulo)]: El empleado Orlando solicita al inventario la ubicacion del articulo Libro2.</i>	
<i>A7 = {</i>	
-- Aserciones de clase --	
<i>Empleado(Orlando)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Empleado domain-pai:Orlando)</i>
<i>Articulo(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro2)</i>
<i>Event(Req3-Ev7)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req3-Ev7)</i>
<i>DateTimeInterval(Req3-DTI7)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req3-DTI7)</i>
<i>Instant(Req3-DTI7-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req3-DTI7-Sins)</i>
<i>Instant(Req3-DTI7-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req3-DTI7-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>solicitaUbicacionDe(Orlando,Req3-Ev7)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:solicitaUbicacionDe domain-pai:Orlando domain-pai:Req3-Ev7)</i>
<i>solicitaUbicacionDe(Req3-Ev7,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:solicitaUbicacionDe domain-pai:Req3-Ev7 domain-pai:Libro2)</i>
<i>during(Req3-Ev7,Req3-DTI7)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req3-Ev7 domain-pai:Req3-DTI7)</i>
<i>hasBeginning(Req3-DTI7,Req3-DTI7-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req3-DTI7 domain-pai:Req3-DTI7-Sins)</i>
<i>hasEnd(Req3-DTI7,Req3-DTI7-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req3-DTI7 domain-pai:Req3-DTI7-Eins)</i>
}	
<i>A8: [XXXX (Empleado \sqsubseteq \existsobtieneUbicacionDe.Articulo)]: El empleado Orlando obtiene la</i>	

Tabla B.14 ABox de casos de éxito en el RF Realizar seguimiento de las ubicaciones de los artículos. (cont.)

Sintaxis DL	Sintaxis Funcional
ubicación del artículo Libro2.	
$A_8 = \{$	
-- Aserciones de clase --	
$Empleado(Orlando)$	$\Rightarrow ClassAssertion(domain-pai:Empleado domain-pai:Orlando)$
$Articulo(Libro2)$	$\Rightarrow ClassAssertion(domain-pai:Articulo domain-pai:Libro2)$
$Event(Req3-Ev8)$	$\Rightarrow ClassAssertion(timeEvents:Event domain-pai:Req3-Ev8)$
$DateTimeInterval(Req3-DTI8)$	$\Rightarrow ClassAssertion(time:DateTimeInterval domain-pai:Req3-DTI8)$
$Instant(Req3-DTI8-Sins)$	$\Rightarrow ClassAssertion(time:Instant domain-pai:Req3-DTI8-Sins)$
$Instant(Req3-DTI8-Eins)$	$\Rightarrow ClassAssertion(time:Instant domain-pai:Req3-DTI8-Eins)$
-- Aserciones de propiedades de objeto --	
$obtieneUbicacionDe(Orlando,Req3-Ev8)$	$\Rightarrow ObjectPropertyAssertion(domain-pai:obtieneUbicacionDe domain-pai:Orlando domain-pai:Req3-Ev8)$
$obtieneUbicacionDe(Req3-Ev8,Libro2)$	$\Rightarrow ObjectPropertyAssertion(domain-pai:obtieneUbicacionDe domain-pai:Req3-Ev8 domain-pai:Libro2)$
$during(Req3-Ev8,Req3-DTI8)$	$\Rightarrow ObjectPropertyAssertion(timeEvents:during domain-pai:Req3-Ev8 domain-pai:Req3-DTI8)$
$hasBeginning(Req3-DTI8,Req3-DTI8-Sins)$	$\Rightarrow ObjectPropertyAssertion(time:hasBeginning domain-pai:Req3-DTI8 domain-pai:Req3-DTI8-Sins)$
$hasEnd(Req3-DTI8,Req3-DTI8-Eins)$	$\Rightarrow ObjectPropertyAssertion(time:hasEnd domain-pai:Req3-DTI8 domain-pai:Req3-DTI8-Eins)$
$\}$	
Caso de éxito 3 ($A_9, A_{10}, A_{11}, A_{12}$):	
$A_9: [X (Empleado \sqsubseteq \exists ubica.Articulo)]:$ El empleado Julia ubica el artículo Libro3.	
...	
$A_{10}: [XX (Empleado \sqsubseteq \exists verificaExistenciaDe.Articulo)]:$ El empleado Julia verifica la existencia del artículo Libro3.	
...	
$A_{11}: [XXX (Empleado \sqsubseteq \exists solicitaUbicacionDe.Articulo)]:$ El empleado Julia solicita al inventario la ubicación del artículo Libro3.	
...	
$A_{12}: [XXXX (Empleado \sqsubseteq \exists obtieneUbicacionDe.Articulo)]:$ El empleado Julia obtiene la ubicación del artículo Libro3.	
...	
Caso de éxito 4 ($A_{13}, A_{14}, A_{15}, A_{16}$):	
$A_{13}: [X (Empleado \sqsubseteq \exists ubica.Articulo)]:$ El empleado Juan ubica el artículo Libro4.	
...	
$A_{14}: [XX (Empleado \sqsubseteq \exists verificaExistenciaDe.Articulo)]:$ El empleado Juan verifica la existencia del artículo Libro4.	

Tabla B.14 ABox de casos de éxito en el RF Realizar seguimiento de las ubicaciones de los artículos. (cont.)

Sintaxis DL	Sintaxis Funcional
...	...
$A_{15}: [XXX (Empleado \sqsubseteq \exists \text{solicitaUbicacionDe.Articulo})]$: El empleado Juan solicita al inventario la ubicación del artículo Libro4.	El empleado Juan solicita al inventario la ubicación del artículo Libro4.
...	...
$A_{16}: [XXXX (Empleado \sqsubseteq \exists \text{obtieneUbicacionDe.Articulo})]$: El empleado Juan obtiene la ubicación del artículo Libro4.	El empleado Juan obtiene la ubicación del artículo Libro4.
...	...
Caso de éxito n	
A_n	
...	

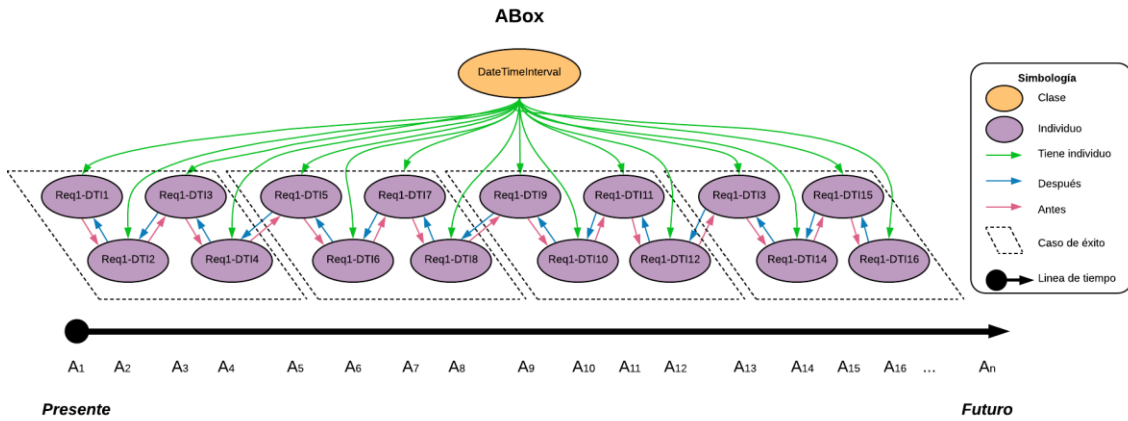


Figura B.10 SoIs en la ABox de casos de éxito de acuerdo al RF Realizar seguimiento de las ubicaciones de los artículos.

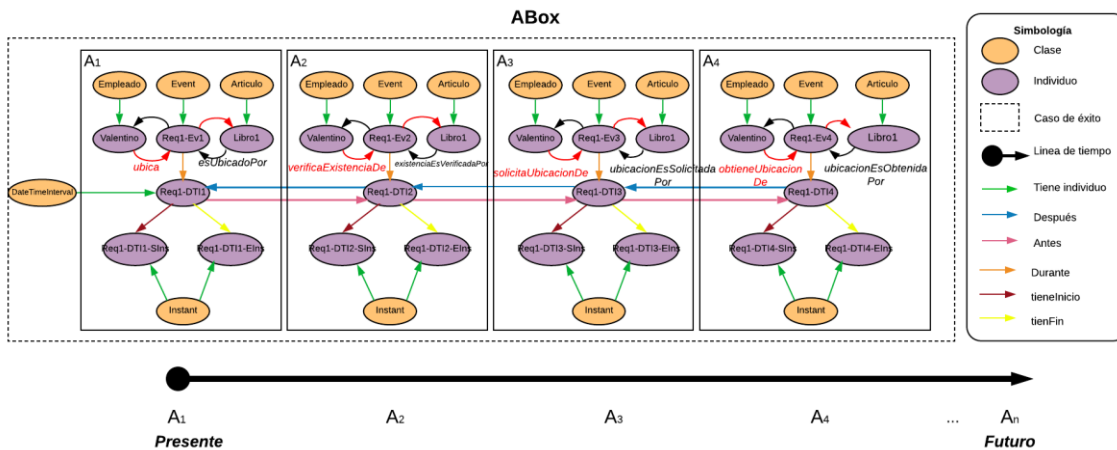


Figura B.11 Caso de éxito 1 en la ABox de casos de éxito de acuerdo al RF Realizar seguimiento de las ubicaciones de los artículos.

1.2.3.2. Casos de éxito y fracaso

En la Tabla B.15 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la *ABox*) basadas en casos de éxito y fracaso, que conforman una secuencia que representa el comportamiento dinámico durante el seguimiento de la ubicación del artículo Libro1 de un inventario. Una representación gráfica simplificada de la secuencia se muestra en la Figura B.12. A manera de ejemplo, en la Figura B.13 se muestra el detalle de las aserciones A_1 , A_2 , A_3 y A_4 en el caso de fracaso número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de razonamiento automatizado, los demás casos se definen de manera similar, sin embargo, difieren en los individuos involucrados.

En la Tabla B.16 se describen las reglas SWRL utilizadas para restringir la existencia de afirmaciones de propiedades de objeto en el intervalo de tiempo correspondiente.

Tabla B.15 ABox de casos de éxito y fracaso en el RF Realizar seguimiento de cantidades de artículos en existencias (cont.)

Sintaxis DL	Sintaxis Funcional
Caso de fracaso 1 (A_1, A_2, A_3, A_4):	
A_1 : [X (<i>Empleado</i> \sqsubseteq \exists ubica.Articulo)]: El empleado Valentino ubica el artículo Libro1.	
$A_1 = \{$	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-pai:Empleado domain-pai:Valentino)
<i>Articulo</i> (Libro1)	\Rightarrow <i>ClassAssertion</i> (domain-pai:Articulo domain-pai:Libro1)
<i>Event</i> (Req3-Ev1)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-pai:Req3-Ev1)
<i>DateTimeInterval</i> (Req3-DTII)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-pai:Req3-DTII)
<i>Instant</i> (Req3-DTII-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-pai:Req3-DTII-Sins)
<i>Instant</i> (Req3-DTII-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-pai:Req3-DTII-Eins)
-- Aserciones de propiedades de objeto --	
<i>ubica</i> (Valentino,Req3-Ev1)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-pai:ubica domain-pai:Valentino domain-pai:Req3-Ev1)
<i>ubica</i> (Req3-Ev1,Libro1)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-pai:ubica domain-pai:Req3-Ev1 domain-pai:Libro1)
<i>during</i> (Req3-Ev1,Req3-DTII)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-pai:Req3-Ev1 domain-pai:Req3-DTII)
<i>hasBeginning</i> (Req3-DTII,Req3-DTII-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-pai:Req3-DTII domain-pai:Req3-DTII-Sins)
<i>hasEnd</i> (Req3-DTII,Req3-DTII-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-pai:Req3-DTII domain-pai:Req3-DTII-Eins)
$\}$	
A_2 : [XX (<i>Empleado</i> \sqsubseteq \exists verificaExistenciaDe.Articulo)]: El empleado Valentino verifica la existencia del artículo Libro1.	
$A_2 = \{$	

Tabla B.15 ABox de casos de éxito y fracaso en el RF Realizar seguimiento de cantidades de artículos en existencias (cont.)

Sintaxis DL	Sintaxis Funcional
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-pai:Empleado domain-pai:Valentino)
<i>Articulo</i> (Libro1)	\Rightarrow <i>ClassAssertion</i> (domain-pai:Articulo domain-pai:Libro1)
<i>Event</i> (Req3-Ev2)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-pai:Req3-Ev2)
<i>DateTimeInterval</i> (Req3-DTI2)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-pai:Req3-DTI2)
<i>Instant</i> (Req3-DTI2-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-pai:Req3-DTI2-Sins)
<i>Instant</i> (Req3-DTI2-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-pai:Req3-DTI2-Eins)
-- Aserciones de propiedades de objeto --	
<i>verificaExistenciaDe</i> (Valentino,Req3-Ev2)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-pai:verificaExistenciaDe domain-pai:Valentino domain-pai:Req3-Ev2)
<i>verificaExistenciaDe</i> (Req3-Ev2,Libro1)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-pai:verificaExistenciaDe domain-pai:Req3-Ev2 domain-pai:Libro1)
<i>during</i> (Req3-Ev2,Req3-DTI2)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-pai:Req3-Ev2 domain-pai:Req3-DTI2)
<i>hasBeginning</i> (Req3-DTI2,Req3-DTI2-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-pai:Req3-DTI2 domain-pai:Req3-DTI2-Sins)
<i>hasEnd</i> (Req3-DTI2,Req3-DTI2-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-pai:Req3-DTI2 domain-pai:Req3-DTI2-Eins)
}	
A₃ : [\neg XXX (<i>Empleado</i> \sqsubseteq \exists solicitaUbicacionDe.Articulo)]: El empleado Valentino no solicita al inventario la ubicacion del articulo Libro1.	
A ₃ = {	
-- Aserciones de clase --	
<i>Event</i> (Req3-Ev3)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-pai:Req3-Ev3)
<i>DateTimeInterval</i> (Req3-DTI3)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-pai:Req3-DTI3)
<i>Instant</i> (Req3-DTI3-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-pai:Req3-DTI3-Sins)
<i>Instant</i> (Req3-DTI3-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-pai:Req3-DTI3-Eins)
-- Aserciones de propiedades de objeto --	
<i>during</i> (Req3-Ev3,Req3-DTI3)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-pai:Req3-Ev3 domain-pai:Req3-DTI3)
<i>hasBeginning</i> (Req3-DTI3,Req3-DTI3-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-pai:Req3-DTI3 domain-pai:Req3-DTI3-Sins)
<i>hasEnd</i> (Req3-DTI3,Req3-DTI3-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-pai:Req3-DTI3 domain-pai:Req3-DTI3-Eins)
}	
A₄ : [\neg XXXX (<i>Empleado</i> \sqsubseteq \exists obtieneUbicacionDe.Articulo)]: El empleado Valentino no obtiene la	

Tabla B.15 ABox de casos de éxito y fracaso en el RF Realizar seguimiento de cantidades de artículos en existencias (cont.)

Sintaxis DL	Sintaxis Funcional
ubicación del artículo Libro1.	
$A_4 = \{$	
-- Aserciones de clase --	
$Event(Req3-Ev4)$	$\Rightarrow ClassAssertion(timeEvents:Event domain-pai:Req3-Ev4)$
$DateTimeInterval(Req3-DTI4)$	$\Rightarrow ClassAssertion(time:DateTimeInterval domain-pai:Req3-DTI4)$
$Instant(Req3-DTI4-Sins)$	$\Rightarrow ClassAssertion(time:Instant domain-pai:Req3-DTI4-Sins)$
$Instant(Req3-DTI4-Eins)$	$\Rightarrow ClassAssertion(time:Instant domain-pai:Req3-DTI4-Eins)$
-- Aserciones de propiedades de objeto --	
$during(Req3-Ev4,Req3-DTI4)$	$\Rightarrow ObjectPropertyAssertion(timeEvents:during domain-pai:Req3-Ev4 domain-pai:Req3-DTI4)$
$hasBeginning(Req3-DTI4,Req3-DTI4-Sins)$	$\Rightarrow ObjectPropertyAssertion(time:hasBeginning domain-pai:Req3-DTI4 domain-pai:Req3-DTI4-Sins)$
$hasEnd(Req3-DTI4,Req3-DTI4-Eins)$	$\Rightarrow ObjectPropertyAssertion(time:hasEnd domain-pai:Req3-DTI4 domain-pai:Req3-DTI4-Eins)$
$\}$	
Caso de éxito 1 (A_5, A_6, A_7, A_8):	
$A_5: [X (Empleado \sqsubseteq \exists ubica.Articulo)]:$ El empleado Orlando ubica el artículo Libro2.	
$A_5 = \{$	
-- Aserciones de clase --	
$Empleado(Orlando)$	$\Rightarrow ClassAssertion(domain-pai:Empleado domain-pai:Orlando)$
$Articulo(Libro2)$	$\Rightarrow ClassAssertion(domain-pai:Articulo domain-pai:Libro2)$
$Event(Req3-Ev5)$	$\Rightarrow ClassAssertion(timeEvents:Event domain-pai:Req3-Ev5)$
$DateTimeInterval(Req3-DTI5)$	$\Rightarrow ClassAssertion(time:DateTimeInterval domain-pai:Req3-DTI5)$
$Instant(Req3-DTI5-Sins)$	$\Rightarrow ClassAssertion(time:Instant domain-pai:Req3-DTI5-Sins)$
$Instant(Req3-DTI5-Eins)$	$\Rightarrow ClassAssertion(time:Instant domain-pai:Req3-DTI5-Eins)$
-- Aserciones de propiedades de objeto --	
$ubica(Orlando,Req3-Ev5)$	$\Rightarrow ObjectPropertyAssertion(domain-pai: ubica domain-pai:Orlando domain-pai:Req3-Ev5)$
$ubica(Req3-Ev5,Libro2)$	$\Rightarrow ObjectPropertyAssertion(domain-pai: ubica domain-pai:Req3-Ev5 domain-pai:Libro2)$
$during(Req3-Ev5,Req3-DTI5)$	$\Rightarrow ObjectPropertyAssertion(timeEvents:during domain-pai:Req3-Ev5 domain-pai:Req3-DTI5)$
$hasBeginning(Req3-DTI5,Req3-DTI5-Sins)$	$\Rightarrow ObjectPropertyAssertion(time:hasBeginning domain-pai:Req3-DTI5 domain-pai:Req3-DTI5-Sins)$
$hasEnd(Req3-DTI5,Req3-DTI5-Eins)$	$\Rightarrow ObjectPropertyAssertion(time:hasEnd domain-pai:Req3-DTI5 domain-pai:Req3-DTI5-Eins)$
$\}$	
$A_6: [XX (Empleado \sqsubseteq \exists verificaExistenciaDe.Articulo)]:$ El empleado Orlando verifica la existencia	

Tabla B.15 ABox de casos de éxito y fracaso en el RF Realizar seguimiento de cantidades de artículos en existencias (cont.)

Sintaxis DL	Sintaxis Funcional
<i>del artículo Libro2.</i>	
$A_6 = \{$	
-- Aserciones de clase --	
<i>Empleado(Orlando)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Empleado domain-pai:Orlando)</i>
<i>Articulo(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro2)</i>
<i>Event(Req3-Ev6)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req3-Ev6)</i>
<i>DateTimeInterval(Req3-DTI6)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req3-DTI6)</i>
<i>Instant(Req3-DTI6-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req3-DTI6-Sins)</i>
<i>Instant(Req3-DTI6-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req3-DTI6-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>verificaExistenciaDe(Orlando,Req3-Ev6)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:verificaExistenciaDe domain-pai:Orlando domain-pai:Req3-Ev6)</i>
<i>verificaExistenciaDe(Req3-Ev6,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:verificaExistenciaDe domain-pai:Req3-Ev6 domain-pai:Libro2)</i>
<i>during(Req3-Ev6,Req3-DTI6)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req3-Ev6 domain-pai:Req3-DTI6)</i>
<i>hasBeginning(Req3-DTI6,Req3-DTI6-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req3-DTI6 domain-pai:Req3-DTI6-Sins)</i>
<i>hasEnd(Req3-DTI6,Req3-DTI6-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req3-DTI6 domain-pai:Req3-DTI6-Eins)</i>
$\}$	
A7: [XXX (Empleado \sqsubseteq \existssolicitaUbicacionDe.Articulo)]: El empleado Orlando solicita al inventario la ubicacion del articulo Libro2.	
$A_7 = \{$	
-- Aserciones de clase --	
<i>Empleado(Orlando)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Empleado domain-pai:Orlando)</i>
<i>Articulo(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro2)</i>
<i>Event(Req3-Ev7)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req3-Ev7)</i>
<i>DateTimeInterval(Req3-DTI7)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req3-DTI7)</i>
<i>Instant(Req3-DTI7-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req3-DTI7-Sins)</i>
<i>Instant(Req3-DTI7-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req3-DTI7-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>solicitaUbicacionDe(Orlando,Req3-Ev7)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:solicitaUbicacionDe domain-pai:Orlando domain-pai:Req3-Ev7)</i>

Tabla B.15 ABox de casos de éxito y fracaso en el RF Realizar seguimiento de cantidades de artículos en existencias (cont.)

Sintaxis DL	Sintaxis Funcional
<i>solicitaUbicacionDe(Req3-Ev7,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:solicitaUbicacionDe domain-pai:Req3-Ev7 domain-pai:Libro2)</i>
<i>during(Req3-Ev7,Req3-DTI7)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req3-Ev7 domain-pai:Req3-DTI7)</i>
<i>hasBeginning(Req3-DTI7,Req3-DTI7-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req3-DTI7 domain-pai:Req3-DTI7-Sins)</i>
<i>hasEnd(Req3-DTI7,Req3-DTI7-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req3-DTI7 domain-pai:Req3-DTI7-Eins)</i>
}	
A₈: [XXXX (Empleado \sqsubseteq \exists obtieneUbicacionDe.Articulo)]: El empleado Orlando obtiene la ubicación del artículo Libro2.	
A ₈ = {	
-- Aserciones de clase --	
<i>Empleado(Orlando)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Empleado domain-pai:Orlando)</i>
<i>Articulo(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-pai:Articulo domain-pai:Libro2)</i>
<i>Event(Req3-Ev8)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-pai:Req3-Ev8)</i>
<i>DateTimeInterval(Req3-DTI8)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-pai:Req3-DTI8)</i>
<i>Instant(Req3-DTI8-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req3-DTI8-Sins)</i>
<i>Instant(Req3-DTI8-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-pai:Req3-DTI8-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>obtieneUbicacionDe(Orlando,Req3-Ev8)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:obtieneUbicacionDe domain-pai:Orlando domain-pai:Req3-Ev8)</i>
<i>obtieneUbicacionDe(Req3-Ev8,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-pai:obtieneUbicacionDe domain-pai:Req3-Ev8 domain-pai:Libro2)</i>
<i>during(Req3-Ev8,Req3-DTI8)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-pai:Req3-Ev8 domain-pai:Req3-DTI8)</i>
<i>hasBeginning(Req3-DTI8,Req3-DTI8-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-pai:Req3-DTI8 domain-pai:Req3-DTI8-Sins)</i>
<i>hasEnd(Req3-DTI8,Req3-DTI8-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-pai:Req3-DTI8 domain-pai:Req3-DTI8-Eins)</i>
}	
Caso de fracaso 2 (A₉, A₁₀, A₁₁, A₁₂):	
A₉: [X (Empleado \sqsubseteq \exists ubica.Articulo)]: El empleado Julia ubica el artículo Libro3.	
...	
A₁₀: [XX (Empleado \sqsubseteq \exists verificaExistenciaDe.Articulo)]: El empleado Julia verifica la existencia del artículo Libro3.	
...	
A₁₁: [\neg XXX (Empleado \sqsubseteq \exists solicitaUbicacionDe.Articulo)]: El empleado Julia no solicita al	

Tabla B.15 ABox de casos de éxito y fracaso en el RF Realizar seguimiento de cantidades de artículos en existencias (cont.)

Sintaxis DL	Sintaxis Funcional
<p><i>inventario la ubicacion del articulo Libro3.</i></p> <p>...</p> <p><i>A₁₂: [¬XXXX (Empleado ⊆ ∃obtieneUbicacionDe.Articulo)]: El empleado Julia no obtiene la ubicación del articulo Libro3.</i></p>	
<p>Caso de éxito 2 (A₁₃, A₁₄, A₁₅, A₁₆):</p> <p><i>A₁₃: [X (Empleado ⊆ ∃ubica.Articulo)]: El empleado Juan ubica el articulo Libro4.</i></p> <p>...</p> <p><i>A₁₄: [XX (Empleado ⊆ ∃verificaExistenciaDe.Articulo)]: El empleado Juan verifica la existencia del artículo Libro4.</i></p> <p>...</p> <p><i>A₁₅: [XXX (Empleado ⊆ ∃solicitaUbicacionDe.Articulo)]: El empleado Juan solicita al inventario la ubicacion del articulo Libro4.</i></p> <p>...</p> <p><i>A₁₆: [XXXX (Empleado ⊆ ∃obtieneUbicacionDe.Articulo)]: El empleado Juan obtiene la ubicación del articulo Libro4.</i></p>	
<p>Caso de éxito o fracaso n</p> <p><i>A_n</i></p> <p>...</p>	

Tabla B.16 Reglas SWRL para la SoIs de casos de éxito y fracaso de acuerdo al RF Realizar seguimiento de las ubicaciones de los artículos (cont.)

ABox	Regla SWRL
A ₃	<i>Event(?e), during(?e, ?i), overlaps(?i, Req3-DTI3), solicitaUbicacionDe(?e, Libro1), solicitaUbicacionDe(Valentino, ?e) -> Nothing(Valentino)</i>
A ₄	<i>Event(?e), during(?e, ?i), overlaps(?i, Req3-DTI4), obtieneUbicacionDe(?e, Libro1), obtieneUbicacionDe(Valentino, ?e) -> Nothing(Valentino)</i>
A ₁₁	<i>Event(?e), during(?e, ?i), overlaps(?i, Req3-DTI11), solicitaUbicacionDe(?e, Libro3), solicitaUbicacionDe(Julia, ?e) -> Nothing(Julia)</i>
A ₁₂	<i>Event(?e), during(?e, ?i), overlaps(?i, Req3-DTI12), obtieneUbicacionDe(?e, Libro3), obtieneUbicacionDe(Julia, ?e) -> Nothing(Julia)</i>

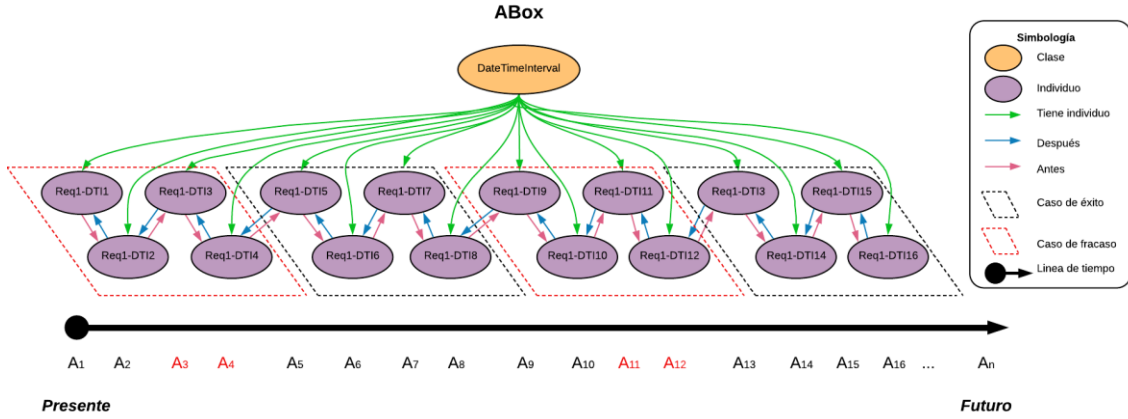


Figura B.12 SoIs en la ABox de casos de éxito y fracaso de acuerdo al RF Realizar seguimiento de las ubicaciones de los artículos.

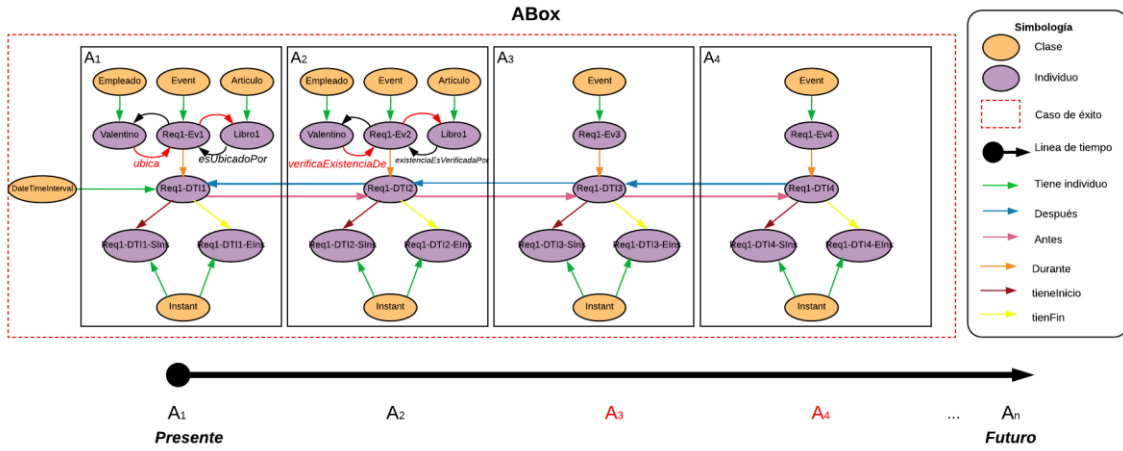


Figura B.13 Caso de fracaso 1 en la ABox de casos de éxito y fracaso de acuerdo al RF Realizar seguimiento de las ubicaciones de los artículos.

Anexo C. Ontología OPAE del patrón de análisis PAIyER

En este anexo, se presenta la última versión de la ontología OPAE del patrón de análisis PAIyER, resultado de la combinación de la OTPAE del PAI con la OTPAE del PAER.

1. Base de conocimiento

En esta sección se presentan las características de la KB (OPAAE) del patrón de análisis PAIyER.

La *TBox* (o componente intencional) y la *ABox* (componente extensional) se describen bajo el poder de expresividad de la DL *ALCIF(D)*. En ambos componentes se incluyen las equivalencias correspondientes en la sintaxis funcional de OWL-DL.

Una versión completa de la KB se encuentra en el directorio *Ontologías/OPAAE PAIyER/*, almacenado en el disco que acompaña esta tesis (para mayor referencia consultar el *Anexo G. Contenido del disco*).

1.1. Componente intencional

Para obtener el componente *TBox* descrito en la Tabla C.1, se utilizaron:

1. Clases (en OWL-DL o nombres de conceptos N_C en las DLs):
{Contrato, Entidad, Institucion, Inventario, Persona, Cliente, Empleado, Propietario, Reservacion}

a partir de las cuales se generó la jerarquía de clases de la Figura C.1, que modela el conocimiento central del patrón de análisis PAIyER.

2. Propiedades de objeto (en OWL-DL o nombres de roles abstractos N_{RA} en las DLs):

{actualiza, adquiere, aplicaPolíticas, buscaExistenciasDe, cancela, crea, daDeBaja, devuelve, seActualizaPor, esAdquiridoPor, esCanceladaPor, esCreadaPor, esDadaDeBajaPor, esDevoluciónPor, esDuenyoDe, esExpedidoPor, esModificadaPor, especificaTipoExistenciasDe, seRegistraPor, esReservadaPor, seSeparaPor, seUbicaPor, esUsadaPor, esVerificadaPor, existenciasSeBuscaPor, existenciasSeObtienePor, expide, laboraEn, modifica, obtieneExistenciasDe, obtieneUbicaciónDe, perteneceA, registra, registraUsoDe, reserva, separa, solicitaUbicaciónDe, suUsoEsRegistradaPor, tieneDatosDe, tipoExistenciasSeEspecificaPor, ubica, ubicaciónSeObtienePor, ubicaciónSeSolicitaPor, usa, verificaDisponibilidadDe}

3. Propiedades de datos (en OWL-DL o nombres de roles concretos N_{RC} en las DLs):

{tieneCantidad, tieneComportamiento, tieneDirección, tieneDisponibilidad, tieneEdad, tieneEspecificidad, tieneFechaFin, tieneFechaInicio, tieneFolio, tieneGiro, tieneId, tieneNivelEstabilidad, tieneNoConfirmación, tieneNombre, tieneNoReservación, tienePercepción, tienePrecio, tieneRfc, tieneTeléfono, tieneTipo, tieneTipoPago, tieneUbicación }

4. Espacios de nombres (o prefijos en OWL para abreviar URIs extensas):

- Ontología que describe parcialmente las clases y propiedades integradas que forman la base de la sintaxis RDF / XML de OWL 2.
Prefix(owl:=<http://www.w3.org/2002/07/owl#>)
- Esquema RDF para el vocabulario RDF definido en el espacio de nombres RDF.
Prefix(rdf:=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
- El espacio de nombres XML proporciona un método simple para calificar los nombres de elementos y atributos utilizados en documentos XML al asociarlos con espacios de nombres identificados por referencias de URI.
Prefix(xml:=<http://www.w3.org/XML/1998/namespace>)
- XML Schema describe la estructura de un documento XML, también se conoce como definición de esquema XML (XSD).
Prefix(xsd:=<http://www.w3.org/2001/XMLSchema#>)
- RDF Schema proporciona un vocabulario de modelado de datos para datos RDF. El esquema RDF es una extensión del vocabulario RDF básico.
Prefix(rdfs:=<http://www.w3.org/2000/01/rdf-schema#>)
- Vocabulario en el dominio del patrón de análisis PAIyER (aspectos estáticos).
Prefix(domain-PAIyER:=<https://www.cenidet.edu.mx/SoftwareEngineering/AnalysisPatterns/StaticOntology/PAIyER#>)

El componente *TBox*, de la KB del patrón de análisis PAIyER, se describe en la Tabla C.1.

Tabla C.1 TBox de la KB del patrón de análisis PAIyER(cont.)

	Sintaxis DL	Sintaxis Funcional
<i>TBox</i> = {		
-- Clases --		
<i>T</i>		⇒ Declaration(Class(owl:Thing))
<i>Contrato</i> ⊆ <i>T</i>		⇒ Declaration(Class(domain-PAIyER:Contrato)) SubClassOf(domain-PAIyER:Contrato owl:Thing)
<i>Entidad</i> ⊆ <i>T</i>		⇒ Declaration(Class(domain-PAIyER:Entidad)) SubClassOf(domain-PAIyER:Entidad owl:Thing)
<i>Institucion</i> ⊆ <i>T</i>		⇒ Declaration(Class(domain-PAIyER:Institucion)) SubClassOf(domain-PAIyER:Institucion owl:Thing)
<i>Inventario</i> ⊆ <i>T</i>		⇒ Declaration(Class(domain-PAIyER:Inventario)) SubClassOf(domain-PAIyER:Inventario owl:Thing)
<i>Persona</i> ⊆ <i>T</i>		⇒ Declaration(Class(domain-PAIyER:Persona)) SubClassOf(domain-PAIyER:Persona owl:Thing)

Tabla C.1 TBox de la KB del patrón de análisis PAIyER(cont.)

Sintaxis DL	Sintaxis Funcional
$Propietario \sqsubseteq T$	\Rightarrow Declaration(Class(domain-PAIyER:Propietario)) SubClassOf(domain-PAIyER:Propietario owl:Thing)
$Reservacion \sqsubseteq T$	\Rightarrow Declaration(Class(domain-PAIyER:Reservacion)) SubClassOf(domain-PAIyER:Reservacion owl:Thing)
$Empleado \sqsubseteq Persona$	\Rightarrow Declaration(Class(domain-PAIyER:Empleado)) SubClassOf(domain-PAIyER:Empleado domain-PAIyER:Persona)
$Cliente \sqsubseteq Persona$	\Rightarrow Declaration(Class(domain-PAIyER:Cliente)) SubClassOf(domain-PAIyER:Cliente domain-PAIyER:Persona)
-- Clases disjuntas --	
$Contrato \sqcap Entidad \sqcap Institucion \sqcap Inventario \sqcap Cliente \sqcap Empleado \sqcap Propietario \sqcap Reservacion \sqsubseteq \perp$	\Rightarrow DisjointClasses(domain-PAIyER:Contrato domain-PAIyER:Entidad domain-PAIyER:Institucion domain-PAIyER:Inventario domain-PAIyER:Cliente domain-PAIyER:Empleado domain-PAIyER:Propietario domain-PAIyER:Reservacion)
-- Restricciones existenciales y universales --	
$Empleado \sqsubseteq \exists separa.Entidad$	\Rightarrow SubClassOf(domain-PAIyER:Empleado ObjectAllValuesFrom(domain-PAIyER:separa Domain-PAIyER:Entidad))
$Empleado \sqsubseteq \exists registra.Entidad$	\Rightarrow SubClassOf(domain-PAIyER:Empleado ObjectAllValuesFrom(domain-PAIyER:registra Domain-PAIyER:Entidad))
$Empleado \sqsubseteq \exists actualiza.Inventario$	\Rightarrow SubClassOf(domain-PAIyER:Empleado ObjectAllValuesFrom(domain-PAIyER:actualiza Domain-PAIyER:Inventario))
$Empleado \sqsubseteq \exists buscaExistenciasDe.Entidad$	\Rightarrow SubClassOf(domain-PAIyER:Empleado ObjectAllValuesFrom(domain-PAIyER:buscaExistenciasDe Domain-PAIyER:Entidad))
$Empleado \sqsubseteq \exists verificaExistenciaDe.Entidad$	\Rightarrow SubClassOf(domain-PAIyER:Empleado ObjectAllValuesFrom(domain-PAIyER:verificaExistenciaDe Domain-PAIyER:Entidad))
$Empleado \sqsubseteq \exists obtieneUbicacionDe.Entidad$	\Rightarrow SubClassOf(domain-PAIyER:Empleado ObjectAllValuesFrom(domain-PAIyER:obtieneUbicacionDe Domain-PAIyER:Entidad))
$Empleado \sqsubseteq \exists especificaTipoExistenciasDe.Entidad$	\Rightarrow SubClassOf(domain-PAIyER:Empleado ObjectAllValuesFrom(domain-PAIyER:especificaTipoExistenciasDe Domain-PAIyER:Entidad))
$Empleado \sqsubseteq \exists obtieneExistenciasDe.Entidad$	\Rightarrow SubClassOf(domain-PAIyER:Empleado ObjectAllValuesFrom(domain-PAIyER:obtieneExistenciasDe Domain-PAIyER:Entidad))
$Empleado \sqsubseteq \exists ubica.Entidad$	\Rightarrow SubClassOf(domain-PAIyER:Empleado ObjectAllValuesFrom(domain-PAIyER:ubica

Tabla C.1 TBox de la KB del patrón de análisis PAIyER(cont.)

Sintaxis DL	Sintaxis Funcional
$Empleado \sqsubseteq \exists solicitaUbicacionDe.Entidad$	$\Rightarrow SubClassOf(domain-PAIyER:Entidad)$ $\Rightarrow SubClassOf(domain-PAIyER:Empleado$ $ObjectAllValuesFrom(domain-PAIyER:$ $solicitaUbicacionDe Domain-PAIyER:Entidad$ $))$
$Empleado \sqsubseteq \exists cancela.Reservacion$	$\Rightarrow SubClassOf(domain-PAIyER:Empleado$ $ObjectAllValuesFrom(domain-PAIyER: cancela$ $Domain-PAIyER:Reservacion)$ $))$
$Empleado \sqsubseteq \exists registraUsoDe.Entidad$	$\Rightarrow SubClassOf(domain-PAIyER:Empleado$ $ObjectAllValuesFrom(domain-PAIyER:$ $registraUsoDe Domain-PAIyER:Entidad)$ $))$
$Empleado \sqsubseteq \exists verificaDisponibilidadDe.Entidad$	$\Rightarrow SubClassOf(domain-PAIyER:Empleado$ $ObjectAllValuesFrom(domain-PAIyER:$ $verificaDisponibilidadDe Domain-$ $PAIyER:Entidad)$ $))$
$Cliente \sqsubseteq \exists adquiere.Entidad$	$\Rightarrow SubClassOf(domain-PAIyER:Cliente$ $ObjectAllValuesFrom(domain-PAIyER:$ $adquiere Domain-PAIyER:Entidad)$ $))$
$Cliente \sqsubseteq \exists daDeBaja.Reservacion$	$\Rightarrow SubClassOf(domain-PAIyER:Cliente$ $ObjectAllValuesFrom(domain-PAIyER:$ $daDeBaja Domain-PAIyER:Reservacion)$ $))$
$Cliente \sqsubseteq \exists devuelve.Entidad$	$\Rightarrow SubClassOf(domain-PAIyER:Cliente$ $ObjectAllValuesFrom(domain-PAIyER:$ $devuelve Domain-PAIyER:Entidad)$ $))$
$Cliente \sqsubseteq \exists modifica.Reservacion$	$\Rightarrow SubClassOf(domain-PAIyER:Cliente$ $ObjectAllValuesFrom(domain-PAIyER:$ $modifica Domain-PAIyER:Reservacion)$ $))$
$Cliente \sqsubseteq \exists reserva.Entidad$	$\Rightarrow SubClassOf(domain-PAIyER:Cliente$ $ObjectAllValuesFrom(domain-PAIyER: reserva$ $Domain-PAIyER:Entidad)$ $))$
$Cliente \sqsubseteq \exists usa.Entidad$	$\Rightarrow SubClassOf(domain-PAIyER:Cliente$ $ObjectAllValuesFrom(domain-PAIyER: usa$ $Domain-PAIyER:Entidad)$ $))$
$Institucion \sqsubseteq \exists tieneDatosDe.Empleado$	$\Rightarrow SubClassOf(domain-PAIyER:Institucion$ $ObjectAllValuesFrom(domain-$ $PAIyER:tieneDatosDe domain-$ $PAIyER:Empleado))$
$Propietario \sqsubseteq \exists esDuenyoDe.Entidad$	$\Rightarrow SubClassOf(domain-PAIyER:Propietario$ $ObjectAllValuesFrom(domain-PAIyER:$ $esDuenyoDe Domain-PAIyER:Entidad)$ $))$
$Entidad \sqsubseteq \forall seSeparaPor.Empleado$	$\Rightarrow SubClassOf(domain-PAIyER:Entidad$ $ObjectAllValuesFrom(domain-$ $PAIyER:seSeparaPor domain-$ $PAIyER:Empleado))$
$Entidad \sqsubseteq \forall seRegistraPor.Empleado$	$\Rightarrow SubClassOf(domain-PAIyER:Entidad$ $ObjectAllValuesFrom(domain-$ $PAIyER:seRegistraPor domain-$ $PAIyER:Empleado))$
$Entidad \sqsubseteq \forall esDevueltaPor.Cliente$	$\Rightarrow SubClassOf(domain-PAIyER:Entidad$ $ObjectAllValuesFrom(domain-$ $PAIyER:esDevueltaPor domain-$ $PAIyER:Cliente))$
$Entidad \sqsubseteq \forall esUsadaPor.Cliente$	$\Rightarrow SubClassOf(domain-PAIyER:Entidad$ $ObjectAllValuesFrom(domain-$

Tabla C.1 TBox de la KB del patrón de análisis PAIyER(cont.)

Sintaxis DL	Sintaxis Funcional
$Entidad \sqsubseteq \forall esVerificadoPor.Empleado$	$\Rightarrow PAIyER:esUsadaPor \text{ domain-PAIyER:Cliente})$ $\Rightarrow SubClassOf(\text{domain-PAIyER:Entidad}$ $ObjectAllValuesFrom(\text{domain-}$ $PAIyER:esVerificadoPor \text{ domain-}$ $PAIyER:Empleado))$
$Entidad \sqsubseteq \forall perteneceA.Propietario$	$\Rightarrow SubClassOf(\text{domain-PAIyER:Entidad}$ $ObjectAllValuesFrom(\text{domain-}$ $PAIyER:perteneceA \text{ domain-}$ $PAIyER:Propietario))$
$Contrato \sqsubseteq \forall esAdquiridoPor.Cliente$	$\Rightarrow SubClassOf(\text{domain-PAIyER:Contrato}$ $ObjectAllValuesFrom(\text{domain-}$ $PAIyER:esAdquiridoPor \text{ domain-}$ $PAIyER:Cliente))$
$Contrato \sqsubseteq \forall esExpedidoPor.Empleado$	$\Rightarrow SubClassOf(\text{domain-PAIyER:Contrato}$ $ObjectAllValuesFrom(\text{domain-}$ $PAIyER:esExpedidoPor \text{ domain-}$ $PAIyER:Empleado))$
$Empleado \sqsubseteq \forall actualiza.Reservacion$	$\Rightarrow SubClassOf(\text{domain-PAIyER:Empleado}$ $ObjectAllValuesFrom(\text{domain-}$ $PAIyER:actualiza \text{ domain-PAIyER:Inventario}))$
$Empleado \sqsubseteq \forall laboraEn.Institucion$	$\Rightarrow SubClassOf(\text{domain-PAIyER:Empleado}$ $ObjectAllValuesFrom(\text{domain-}$ $PAIyER:laboraEn \text{ domain-PAIyER:Institucion}))$
$Empleado \sqsubseteq \forall crea.Reservacion$	$\Rightarrow SubClassOf(\text{domain-PAIyER:Empleado}$ $ObjectAllValuesFrom(\text{domain-PAIyER:crea}$ $\text{ domain-PAIyER:Reservacion}))$
$Empleado \sqsubseteq \forall aplicaPoliticlas.(Reservacion \sqcup$ $Cliente)$	$\Rightarrow SubClassOf(\text{domain-PAIyER:Empleado}$ $ObjectAllValuesFrom(\text{domain-}$ $PAIyER:aplicaPoliticlas \text{ domain-}$ $PAIyER:Reservacion \text{ domain-}$ $PAIyER:Cliente))$
$Contrato \sqsubseteq \forall tieneDatosDe.(Cliente \sqcup Empleado \sqcup$ $Propietario \sqcup Reservacion \sqcup Institucion \sqcup Entidad)$	$\Rightarrow SubClassOf(\text{domain-PAIyER:Contrato}$ $ObjectAllValuesFrom(\text{domain-}$ $PAIyER:tieneDatosDe \text{ domain-PAIyER:Cliente}$ $\text{ domain-PAIyER:Empleado \text{ domain-}$ $PAIyER:Entidad \text{ domain-PAIyER:Institucion}$ $\text{ domain-PAIyER:Propietario \text{ domain-}$ $PAIyER:Reservacion}))$
$Empleado \sqsubseteq \forall crea.Reservacion$	$\Rightarrow SubClassOf(\text{domain-PAIyER:Empleado}$ $ObjectAllValuesFrom(\text{domain-PAIyER:crea}$ $\text{ domain-PAIyER:Reservacion}))$
$Empleado \sqsubseteq \forall expide.Contrato$	$\Rightarrow SubClassOf(\text{domain-PAIyER:Empleado}$ $ObjectAllValuesFrom(\text{domain-PAIyER:expide}$ $\text{ domain-PAIyER:Contrato}))$
$Reservacion \sqsubseteq \forall esCreadaPor.Empleado$	$\Rightarrow SubClassOf(\text{domain-PAIyER:Reservacion}$ $ObjectAllValuesFrom(\text{domain-}$ $PAIyER:esCreadaPor \text{ domain-}$ $PAIyER:Empleado))$
-- Propiedades de objeto --	
<i>actualiza</i>	$\Rightarrow Declaration(ObjectProperty(\text{domain-}$ $PAIyER:actualiza))$
<i>seActualizaPor</i>	$\Rightarrow Declaration(ObjectProperty(\text{domain-}$ $PAIyER:seActualizaPor))$

Tabla C.1 TBox de la KB del patrón de análisis PAIyER(cont.)

Sintaxis DL	Sintaxis Funcional
<i>obtieneExistenciasDe</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:obtieneExistenciasDe))</i>
<i>existenciasSeObtienePor</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:existenciasSeObtienePor))</i>
<i>buscaExistenciasDe</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:buscaExistenciasDe))</i>
<i>existenciasSeBuscaPor</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:existenciasSeBuscaPor))</i>
<i>especificaTipoExistenciasDe</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:especificaTipoExistenciasDe))</i>
<i>tipoExistenciasSeEspecificaPor</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:tipoExistenciasSeEspecificaPor))</i>
<i>registra</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:tipoExistenciasSeEspecificaPor))</i>
<i>seRegistraPor</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:seRegistraPor))</i>
<i>separa</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:separa))</i>
<i>seSeparaPor</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:seSeparaPor))</i>
<i>ubica</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:ubica))</i>
<i>seUbicaPor</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:seUbicaPor))</i>
<i>verificaExistenciaDe</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:verificaExistenciaDe))</i>
<i>existenciaSeVerificaPor</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:existenciaSeVerificaPor))</i>
<i>laboraEn</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:laboraEn))</i>
<i>obtieneUbicacionDe</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:obtieneUbicacionDe))</i>
<i>ubicacionSeObtienePor</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:ubicacionSeObtienePor))</i>
<i>solicitaUbicacionDe</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:solicitaUbicacionDe))</i>
<i>ubicacionSeSolicitaPor</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:ubicacionSeSolicitaPor))</i>
<i>tieneDatosDe</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:tieneDatosDe))</i>
<i>adquiere</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:adquiere))</i>
<i>esAdquiridoPor</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:esAdquiridoPor))</i>
<i>aplicaPolíticas</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:aplicaPolíticas))</i>
<i>cancela</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:cancela))</i>
<i>esCanceladaPor</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:esCanceladaPor))</i>
<i>crea</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:crea))</i>
<i>esCreadaPor</i>	⇒ <i>Declaration(ObjectProperty(domain-PAIyER:esCreadaPor))</i>

Tabla C.1 TBox de la KB del patrón de análisis PAIyER(cont.)

Sintaxis DL	Sintaxis Funcional
<i>daDeBaja</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-PAIyER: daDeBaja))</i>
<i>esDadaDeBajaPor</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-PAIyER: esDadaDeBajaPor))</i>
<i>devuelve</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-PAIyER: devuelve))</i>
<i>esDevueltaPor</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-PAIyER: esDevueltaPor))</i>
<i>esDuenyoDe</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-PAIyER: esDuenyoDe))</i>
<i>expide</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-PAIyER: expide))</i>
<i>esExpedidoPor</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-PAIyER: esExpedidoPor))</i>
<i>modifica</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-PAIyER: modifica))</i>
<i>esModificadaPor</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-PAIyER: esModificadaPor))</i>
<i>reserva</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-PAIyER: reserva))</i>
<i>esReservadaPor</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-PAIyER: esReservadaPor))</i>
<i>usa</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-PAIyER: usa))</i>
<i>esUsadaPor</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-PAIyER: esUsadaPor))</i>
<i>verificaDisponibilidadDe</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-PAIyER: verificaDisponibilidadDe))</i>
<i>esVerificadaPor</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-PAIyER: esVerificadaPor))</i>
<i>perteneceA</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-PAIyER: perteneceA))</i>
<i>registraUsoDe</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-PAIyER: registraUsoDe))</i>
<i>suUsoSeRegistraPor</i>	\Rightarrow <i>Declaration(ObjectProperty(domain-PAIyER: suUsoSeRegistraPor))</i>
-- Restricciones y características de propiedades de objeto --	
\exists actualiza.T \sqsubseteq Empleado	\Rightarrow <i>ObjectPropertyDomain(domain-PAIyER:actualiza domain-PAIyER:Empleado))</i>
$T \sqsubseteq \forall$ actualiza.Inventario	\Rightarrow <i>ObjectPropertyRange(domain-PAIyER:actualiza domain-PAIyER:Inventario))</i>
$T \sqsubseteq \leq_1$ actualiza	\Rightarrow <i>FunctionalObjectProperty(domain-PAIyER:actualiza)</i>
\exists seActualizaPor.T \sqsubseteq Inventario	\Rightarrow <i>ObjectPropertyDomain(domain-PAIyER:seActualizaPor domain-PAIyER:Inventario))</i>
$T \sqsubseteq \forall$ seActualizaPor. Empleado	\Rightarrow <i>ObjectPropertyRange(domain-PAIyER:seActualizaPor domain-PAIyER:Empleado))</i>
$T \sqsubseteq \leq_1$ seActualizaPor ⁻	\Rightarrow <i>InverseFunctionalObjectProperty(domain-PAIyER: seActualizaPor)</i>
\exists obtieneExistenciasDe.T \sqsubseteq Empleado	\Rightarrow <i>ObjectPropertyDomain(domain-PAIyER: obtieneExistenciasDe domain-PAIyER:Empleado))</i>

Tabla C.1 TBox de la KB del patrón de análisis PAIyER(cont.)

Sintaxis DL	Sintaxis Funcional
$T \sqsubseteq \forall \text{obtieneExistenciasDe. Entidad}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: obtieneExistenciasDe Domain-PAIyER:Entidad})$
$\exists \text{existenciasSeObtienePor. T} \sqsubseteq \text{Entidad}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: existenciasSeObtienePor Domain-PAIyER:Entidad})$
$T \sqsubseteq \forall \text{existenciasSeObtienePor. Empleado}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: existenciasSeObtienePor domain-PAIyER:Empleado})$
$\exists \text{buscaExistenciasDe. T} \sqsubseteq \text{Empleado}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: buscaExistenciasDe domain-PAIyER:Empleado})$
$T \sqsubseteq \forall \text{buscaExistenciasDe. Entidad}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: buscaExistenciasDe Domain-PAIyER:Entidad})$
$\exists \text{existenciasSeBuscaPor. T} \sqsubseteq \text{Entidad}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: existenciasSeBuscaPor Domain-PAIyER:Entidad})$
$T \sqsubseteq \forall \text{existenciasSeBuscaPor. Empleado}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: existenciasSeBuscaPor domain-PAIyER:Empleado})$
$\exists \text{especificaTipoExistenciasDe. T} \sqsubseteq \text{Empleado}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: especificaTipoExistenciasDe domain-PAIyER:Empleado})$
$T \sqsubseteq \forall \text{especificaTipoExistenciasDe. Entidad}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: especificaTipoExistenciasDe Domain-PAIyER:Entidad})$
$\exists \text{tipoExistenciasSeEspecificaPor. T} \sqsubseteq \text{Entidad}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: tipoExistenciasSeEspecificaPor Domain-PAIyER:Entidad})$
$T \sqsubseteq \forall \text{tipoExistenciasSeEspecificaPor. Empleado}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: tipoExistenciasSeEspecificaPor domain-PAIyER:Empleado})$
$\exists \text{registra. T} \sqsubseteq \text{Empleado}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: registra domain-PAIyER:Empleado})$
$T \sqsubseteq \forall \text{registra. Entidad}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: registra Domain-PAIyER:Entidad})$
$T \sqsubseteq \leq_1 \text{registra}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER: registra})$
$\exists \text{seRegistraPor. T} \sqsubseteq \text{Entidad}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: seRegistraPor Domain-PAIyER:Entidad})$
$T \sqsubseteq \forall \text{seRegistraPor. Empleado}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: seRegistraPor domain-PAIyER:Empleado})$
$T \sqsubseteq \leq_1 \text{seRegistraPor}$	$\Rightarrow \text{FunctionalObjectProperty}(\text{domain-PAIyER: seRegistraPor})$
$\exists \text{separa. T} \sqsubseteq \text{Empleado}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: separa domain-PAIyER:Empleado})$
$T \sqsubseteq \forall \text{separa. Entidad}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: separa Domain-PAIyER:Entidad})$
$T \sqsubseteq \leq_1 \text{separa}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER: separa})$
$\exists \text{seSeparaPor. T} \sqsubseteq \text{Entidad}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: seSeparaPor Domain-PAIyER:Entidad})$
$T \sqsubseteq \forall \text{seSeparaPor. Empleado}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: seSeparaPor domain-PAIyER:Empleado})$

Tabla C.1 TBox de la KB del patrón de análisis PAIyER(cont.)

Sintaxis DL	Sintaxis Funcional
$T \sqsubseteq_{\leq 1} \text{seSeparaPor}$	\Rightarrow <i>FunctionalObjectProperty(domain-PAIyER: seSeparaPor)</i>
$\exists \text{ubica}.T \sqsubseteq \text{Empleado}$	\Rightarrow <i>ObjectPropertyDomain(domain-PAIyER: ubica domain-PAIyER:Empleado)</i>
$T \sqsubseteq \forall \text{ubica}. \text{Entidad}$	\Rightarrow <i>ObjectPropertyRange(domain-PAIyER: ubica Domain-PAIyER:Entidad)</i>
$\exists \text{seUbicaPor}.T \sqsubseteq \text{Entidad}$	\Rightarrow <i>ObjectPropertyDomain(domain-PAIyER: seUbicaPor Domain-PAIyER:Entidad)</i>
$T \sqsubseteq \forall \text{seUbicaPor}. \text{Empleado}$	\Rightarrow <i>ObjectPropertyRange(domain-PAIyER: seUbicaPor domain-PAIyER:Empleado))</i>
$\exists \text{verificaExistenciaDe}.T \sqsubseteq \text{Empleado}$	\Rightarrow <i>ObjectPropertyDomain(domain-PAIyER: verificaExistenciaDe domain-PAIyER:Empleado))</i>
$T \sqsubseteq \forall \text{verificaExistenciaDe}. \text{Entidad}$	\Rightarrow <i>ObjectPropertyRange(domain-PAIyER: verificaExistenciaDe Domain-PAIyER:Entidad)</i>
$\exists \text{existenciaSeVerificaPor}.T \sqsubseteq \text{Entidad}$	\Rightarrow <i>ObjectPropertyDomain(domain-PAIyER: existenciaSeVerificaPor Domain-PAIyER:Entidad)</i>
$T \sqsubseteq \forall \text{existenciaSeVerificaPor}. \text{Empleado}$	\Rightarrow <i>ObjectPropertyRange(domain-PAIyER: existenciaSeVerificaPor domain-PAIyER:Empleado))</i>
$\exists \text{obtieneUbicacionDe}.T \sqsubseteq \text{Empleado}$	\Rightarrow <i>ObjectPropertyDomain(domain-PAIyER: obtieneUbicacionDe domain-PAIyER:Empleado))</i>
$T \sqsubseteq \forall \text{obtieneUbicacionDe}. \text{Entidad}$	\Rightarrow <i>ObjectPropertyRange(domain-PAIyER: obtieneUbicacionDe Domain-PAIyER:Entidad)</i>
$\exists \text{ubicacionSeObtienePor}.T \sqsubseteq \text{Entidad}$	\Rightarrow <i>ObjectPropertyDomain(domain-PAIyER: ubicacionSeObtienePor Domain-PAIyER:Entidad)</i>
$T \sqsubseteq \forall \text{ubicacionSeObtienePor}. \text{Empleado}$	\Rightarrow <i>ObjectPropertyRange(domain-PAIyER: ubicacionSeObtienePor domain-PAIyER:Empleado))</i>
$\exists \text{solicitaUbicacionDe}.T \sqsubseteq \text{Empleado}$	\Rightarrow <i>ObjectPropertyDomain(domain-PAIyER: solicitaUbicacionDe domain-PAIyER:Empleado))</i>
$T \sqsubseteq \forall \text{solicitaUbicacionDe}. \text{Entidad}$	\Rightarrow <i>ObjectPropertyRange(domain-PAIyER: solicitaUbicacionDe Domain-PAIyER:Entidad)</i>
$\exists \text{ubicacionSeSolicitaPor}.T \sqsubseteq \text{Entidad}$	\Rightarrow <i>ObjectPropertyDomain(domain-PAIyER: ubicacionSeSolicitaPor Domain-PAIyER:Entidad)</i>
$T \sqsubseteq \forall \text{ubicacionSeSolicitaPor}. \text{Empleado}$	\Rightarrow <i>ObjectPropertyRange(domain-PAIyER: ubicacionSeSolicitaPor domain-PAIyER:Empleado))</i>
$\exists \text{tieneDatosDe}.T \sqsubseteq \text{Institucion}$	\Rightarrow <i>ObjectPropertyDomain(domain-PAIyER: tieneDatosDe domain-PAIyER:Institucion))</i>
$T \sqsubseteq \forall \text{tieneDatosDe}. \text{Empleado}$	\Rightarrow <i>ObjectPropertyRange(domain-PAIyER: tieneDatosDe domain-PAIyER:Empleado))</i>
$\exists \text{laboraEn}.T \sqsubseteq \text{Empleado}$	\Rightarrow <i>ObjectPropertyDomain(domain-PAIyER:laboraEn domain-PAIyER:Empleado)</i>
$T \sqsubseteq \forall \text{laboraEn}. \text{Institucion}$	\Rightarrow <i>ObjectPropertyRange(domain-PAIyER:laboraEn domain-PAIyER:Institucion)</i>
$\exists \text{actualiza}.T \sqsubseteq \text{Empleado}$	\Rightarrow <i>ObjectPropertyDomain(domain-PAIyER:actualiza domain-PAIyER:Empleado)</i>

Tabla C.1 TBox de la KB del patrón de análisis PAIyER(cont.)

Sintaxis DL	Sintaxis Funcional
$T \sqsubseteq \forall \text{actualiza.Reservacion}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER:actualiza domain-PAIyER:Reservacion})$
$T \sqsubseteq \leq I \text{ actualiza}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER:actualiza})$
$\exists \text{adquiere.T} \sqsubseteq \text{Cliente}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER:adquiere domain-PAIyER:Cliente})$
$T \sqsubseteq \forall \text{adquiere.Contrato}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER:adquiere domain-PAIyER:Contrato})$
$T \sqsubseteq \leq I \text{ adquiere}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER:adquiere})$
$\exists \text{aplicaPoliticlas.T} \sqsubseteq \text{Empleado}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER:aplicaPoliticlas domain-PAIyER:Empleado})$
$T \sqsubseteq \forall \text{aplicaPoliticlas.}(\text{Reservacion} \sqcup \text{Cliente})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER:aplicaPoliticlas domain-PAIyER:Reservacion domain-PAIyER:Cliente})$
$\exists \text{cancela.T} \sqsubseteq \text{Empleado}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER:cancela domain-PAIyER:Empleado})$
$T \sqsubseteq \forall \text{cancela.Reservacion}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER:cancela domain-PAIyER:Reservacion})$
$T \sqsubseteq \leq I \text{ cancela}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER:cancela})$
$\exists \text{crea.T} \sqsubseteq \text{Empleado}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER:crea domain-PAIyER:Empleado})$
$T \sqsubseteq \forall \text{crea.Reservacion}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER:crea domain-PAIyER:Reservacion})$
$T \sqsubseteq \leq I \text{ crea}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER:crea})$
$\exists \text{esCreadaPor.T} \sqsubseteq \text{Reservacion}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER:esCreadaPor domain-PAIyER:Reservacion})$
$T \sqsubseteq \forall \text{esCreadaPor.Empleado}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER:esCreadaPor domain-PAIyER:Empleado})$
$T \sqsubseteq \leq I \text{ esCreadaPor}$	$\Rightarrow \text{FunctionalObjectProperty}(\text{domain-PAIyER:esCreadaPor})$
$\exists \text{devuelve.T} \sqsubseteq \text{Cliente}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER:devuelve domain-PAIyER:Cliente})$
$T \sqsubseteq \forall \text{devuelve.Entidad}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER:devuelve domain-PAIyER:Entidad})$
$T \sqsubseteq \leq I \text{ devuelve}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER:devuelve})$
$\exists \text{seActualizaPor.T} \sqsubseteq \text{Reservacion}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER:seActualizaPor domain-PAIyER:Reservacion})$
$T \sqsubseteq \forall \text{seActualizaPor.Empleado}$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER:seActualizaPor domain-PAIyER:Empleado})$
$T \sqsubseteq \leq I \text{ seActualizaPor}$	$\Rightarrow \text{FunctionalObjectProperty}(\text{domain-PAIyER:seActualizaPor})$
$\exists \text{esAdquiridoPor.T} \sqsubseteq \text{Contrato}$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER:esAdquiridoPor domain-PAIyER:Contrato})$

Tabla C.1 TBox de la KB del patrón de análisis PAIyER(cont.)

Sintaxis DL	Sintaxis Funcional
$T \sqsubseteq \forall esAdquiridoPor.Cliente$	$\Rightarrow ObjectPropertyRange(domain-PAIyER:esAdquiridoPor domain-PAIyER:Cliente)$
$T \sqsubseteq \leq I esAdquiridoPor$	$\Rightarrow FunctionalObjectProperty(domain-PAIyER:esAdquiridoPor)$
$\exists esCanceladaPor.T \sqsubseteq Reservacion$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER:Reservacion)$
$T \sqsubseteq \forall esCanceladaPor.Empleado$	$\Rightarrow ObjectPropertyRange(domain-PAIyER:esCanceladaPor domain-PAIyER:Empleado)$
$T \sqsubseteq \leq I esCanceladaPor$	$\Rightarrow FunctionalObjectProperty(domain-PAIyER:esCanceladaPor)$
$\exists esDadaDeBajaPor.T \sqsubseteq Reservacion$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER:Reservacion)$
$T \sqsubseteq \forall esDadaDeBajaPor.Cliente$	$\Rightarrow ObjectPropertyRange(domain-PAIyER:esDadaDeBajaPor domain-PAIyER:Cliente)$
$T \sqsubseteq \leq I esDadaDeBajaPor$	$\Rightarrow FunctionalObjectProperty(domain-PAIyER:esDadaDeBajaPor)$
$\exists esDevoltaPor.T \sqsubseteq Entidad$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER:Entidad)$
$T \sqsubseteq \forall esDevoltaPor.Cliente$	$\Rightarrow ObjectPropertyRange(domain-PAIyER:esDevoltaPor domain-PAIyER:Cliente)$
$T \sqsubseteq \leq I esDevoltaPor$	$\Rightarrow FunctionalObjectProperty(domain-PAIyER:esDevoltaPor)$
$\exists esDuenyoDe.T \sqsubseteq Propietario$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER:Propietario)$
$T \sqsubseteq \forall esDuenyoDe.Entidad$	$\Rightarrow ObjectPropertyRange(domain-PAIyER:esDuenyoDe domain-PAIyER:Entidad)$
$\exists esExpedidoPor.T \sqsubseteq Contrato$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER:Contrato)$
$T \sqsubseteq \forall esExpedidoPor.Empleado$	$\Rightarrow ObjectPropertyRange(domain-PAIyER:esExpedidoPor domain-PAIyER:Empleado)$
$T \sqsubseteq \leq I esExpedidoPor$	$\Rightarrow FunctionalObjectProperty(domain-PAIyER:esExpedidoPor)$
$\exists esModificadaPor.T \sqsubseteq Reservacion$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER:Reservacion)$
$T \sqsubseteq \forall esModificadaPor.Cliente$	$\Rightarrow ObjectPropertyRange(domain-PAIyER:esModificadaPor domain-PAIyER:Cliente)$
$T \sqsubseteq \leq I esModificadaPor$	$\Rightarrow FunctionalObjectProperty(domain-PAIyER:esModificadaPor)$
$\exists esReservadaPor.T \sqsubseteq Entidad$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER:Entidad)$
$T \sqsubseteq \forall esReservadaPor.Cliente$	$\Rightarrow ObjectPropertyRange(domain-$

Tabla C.1 TBox de la KB del patrón de análisis PAIyER(cont.)

Sintaxis DL	Sintaxis Funcional
	<i>PAIyER:esReservadaPor</i> domain- <i>PAIyER:Cliente</i>)
$T \sqsubseteq \leq I$ <i>esReservadaPor</i>	\Rightarrow <i>FunctionalObjectProperty</i> (domain- <i>PAIyER:esReservadaPor</i>)
\exists <i>esUsadaPor</i> . $T \sqsubseteq$ <i>Entidad</i>	\Rightarrow <i>ObjectPropertyDomain</i> (domain- <i>PAIyER:esUsadaPor</i> domain- <i>PAIyER:Entidad</i>)
$T \sqsubseteq \forall$ <i>esUsadaPor</i> . <i>Cliente</i>	\Rightarrow <i>ObjectPropertyRange</i> (domain- <i>PAIyER:esUsadaPor</i> domain- <i>PAIyER:Cliente</i>)
$T \sqsubseteq \leq I$ <i>esUsadaPor</i>	\Rightarrow <i>FunctionalObjectProperty</i> (domain- <i>PAIyER:esUsadaPor</i>)
	\Rightarrow <i>ObjectPropertyDomain</i> (domain- <i>PAIyER:esVerificadaPor</i> domain- <i>PAIyER:Entidad</i>)
\exists <i>esVerificadaPor</i> . $T \sqsubseteq$ <i>Entidad</i>	\Rightarrow <i>ObjectPropertyRange</i> (domain- <i>PAIyER:esVerificadaPor</i> domain- <i>PAIyER:Empleado</i>)
$T \sqsubseteq \forall$ <i>esVerificadaPor</i> . <i>Empleado</i>	\Rightarrow <i>FunctionalObjectProperty</i> (domain- <i>PAIyER:esVerificadaPor</i>)
$T \sqsubseteq \leq I$ <i>esVerificadaPor</i>	\Rightarrow <i>ObjectPropertyDomain</i> (domain- <i>PAIyER:expide</i> domain- <i>PAIyER:Empleado</i>)
\exists <i>expide</i> . $T \sqsubseteq$ <i>Empleado</i>	\Rightarrow <i>ObjectPropertyRange</i> (domain- <i>PAIyER:expide</i> domain- <i>PAIyER:Contrato</i>)
$T \sqsubseteq \forall$ <i>expide</i> . <i>Contrato</i>	\Rightarrow <i>InverseFunctionalObjectProperty</i> (domain- <i>PAIyER:expide</i>)
$T \sqsubseteq \leq I$ <i>expide</i> ⁻	\Rightarrow <i>ObjectPropertyDomain</i> (domain- <i>PAIyER:modifica</i> domain- <i>PAIyER:Cliente</i>)
\exists <i>modifica</i> . $T \sqsubseteq$ <i>Cliente</i>	\Rightarrow <i>ObjectPropertyRange</i> (domain- <i>PAIyER:modifica</i> domain- <i>PAIyER:Reservacion</i>)
$T \sqsubseteq \forall$ <i>modifica</i> . <i>Reservacion</i>	\Rightarrow <i>InverseFunctionalObjectProperty</i> (domain- <i>PAIyER:modifica</i>)
$T \sqsubseteq \leq I$ <i>modifica</i> ⁻	\Rightarrow <i>ObjectPropertyDomain</i> (domain- <i>PAIyER:perteneceA</i> domain- <i>PAIyER:Entidad</i>)
\exists <i>perteneceA</i> . $T \sqsubseteq$ <i>Entidad</i>	\Rightarrow <i>ObjectPropertyRange</i> (domain- <i>PAIyER:perteneceA</i> domain- <i>PAIyER:Propietario</i>)
$T \sqsubseteq \forall$ <i>perteneceA</i> . <i>Propietario</i>	\Rightarrow <i>FunctionalObjectProperty</i> (domain- <i>PAIyER:perteneceA</i>)
$T \sqsubseteq \leq I$ <i>perteneceA</i>	\Rightarrow <i>ObjectPropertyDomain</i> (domain- <i>PAIyER:registraUsoDe</i> domain- <i>PAIyER:Empleado</i>)
\exists <i>registraUsoDe</i> . $T \sqsubseteq$ <i>Empleado</i>	\Rightarrow <i>ObjectPropertyRange</i> (domain- <i>PAIyER:registraUsoDe</i> domain- <i>PAIyER:Entidad</i>)
$T \sqsubseteq \forall$ <i>registraUsoDe</i> . <i>Entidad</i>	\Rightarrow <i>InverseFunctionalObjectProperty</i> (domain- <i>PAIyER:registraUsoDe</i>)
$T \sqsubseteq \leq I$ <i>registraUsoDe</i> ⁻	\Rightarrow <i>ObjectPropertyDomain</i> (domain- <i>PAIyER:reserva</i> domain- <i>PAIyER:Cliente</i>)
\exists <i>reserva</i> . $T \sqsubseteq$ <i>Cliente</i>	\Rightarrow <i>ObjectPropertyRange</i> (domain- <i>PAIyER:reserva</i> domain- <i>PAIyER:Entidad</i>)
$T \sqsubseteq \forall$ <i>reserva</i> . <i>Entidad</i>	\Rightarrow <i>InverseFunctionalObjectProperty</i> (domain- <i>PAIyER:reserva</i>)
$T \sqsubseteq \leq I$ <i>reserva</i> ⁻	\Rightarrow <i>ObjectPropertyDomain</i> (domain- <i>PAIyER:suUsoSeRegistraPor</i> domain-
\exists <i>suUsoSeRegistraPor</i> . $T \sqsubseteq$ <i>Entidad</i>	

Tabla C.1 TBox de la KB del patrón de análisis PAIyER(cont.)

Sintaxis DL	Sintaxis Funcional
	<i>PAIyER:Entidad</i>)
$T \sqsubseteq \forall suUsoSeRegistraPor.Empleado$	\Rightarrow <i>ObjectPropertyRange(domain-PAIyER:suUsoSeRegistraPor domain-PAIyER:Empleado)</i>
$T \sqsubseteq \leq l suUsoSeRegistraPor$	\Rightarrow <i>FunctionalObjectProperty(domain-PAIyER:suUsoSeRegistraPor)</i>
$\exists tieneDatosDe.T \sqsubseteq Contrato$	\Rightarrow <i>ObjectPropertyDomain(domain-PAIyER:tieneDatosDe domain-PAIyER:Contrato)</i>
$T \sqsubseteq \forall tieneDatosDe.(Cliente \sqcup Empleado \sqcup Propietario \sqcup Reservacion \sqcup Institucion \sqcup Entidad)$	\Rightarrow <i>ObjectPropertyRange(domain-PAIyER:tieneDatosDe domain-PAIyER:Cliente domain-PAIyER:Empleado domain-PAIyER:Propietario domain-PAIyER:Reservacion domain-PAIyER:Institucion domain-PAIyER:Entidad)</i>
$\exists usa.T \sqsubseteq Cliente$	\Rightarrow <i>ObjectPropertyDomain(domain-PAIyER:usa domain-PAIyER:Cliente)</i>
$T \sqsubseteq \forall usa.Entidad$	\Rightarrow <i>ObjectPropertyRange(domain-PAIyER:usa domain-PAIyER:Entidad)</i>
$T \sqsubseteq \leq l usa^{-}$	\Rightarrow <i>InverseFunctionalObjectProperty(domain-PAIyER:usa)</i>
$\exists verificaDisponibilidadDe.T \sqsubseteq Empleado$	\Rightarrow <i>ObjectPropertyDomain(domain-PAIyER:verificaDisponibilidadDe domain-PAIyER:Empleado)</i>
$T \sqsubseteq \forall verificaDisponibilidadDe.Entidad$	\Rightarrow <i>ObjectPropertyRange(domain-PAIyER:verificaDisponibilidadDe domain-PAIyER:Entidad)</i>
$T \sqsubseteq \leq l verificaDisponibilidadDe^{-}$	\Rightarrow <i>InverseFunctionalObjectProperty(domain-PAIyER:verificaDisponibilidadDe)</i>
$actualiza \equiv seActualizaPor^{-}$	\Rightarrow <i>InverseObjectProperties(domain-PAIyER:actualiza domain-PAIyER:seActualizaPor)</i>
$obtieneExistenciasDe \equiv existenciasSeObtienePor^{-}$	\Rightarrow <i>InverseObjectProperties(domain-PAIyER:obtieneExistenciasDe domain-PAIyER:existenciasSeObtienePor)</i>
$buscaExistenciasDe \equiv existenciasSeBuscaPor^{-}$	\Rightarrow <i>InverseObjectProperties(domain-PAIyER:buscaExistenciasDe domain-PAIyER:existenciasSeBuscaPor)</i>
$especificaTipoExistenciasDe \equiv tipoExistenciasSeEspecificaPor^{-}$	\Rightarrow <i>InverseObjectProperties(domain-PAIyER:especificaTipoExistenciasDe domain-PAIyER:tipoExistenciasSeEspecificaPor)</i>
$registra \equiv seRegistraPor^{-}$	\Rightarrow <i>InverseObjectProperties(domain-PAIyER:registra domain-PAIyER: seRegistraPor)</i>
$separa \equiv seSeparaPor^{-}$	\Rightarrow <i>InverseObjectProperties(domain-PAIyER:separa domain-PAIyER: seSeparaPor)</i>
$ubica \equiv seUbicaPor^{-}$	\Rightarrow <i>InverseObjectProperties(domain-PAIyER:ubica domain-PAIyER: seUbicaPor)</i>
$verificaExistenciaDe \equiv existenciaSeVerificaPor^{-}$	\Rightarrow <i>InverseObjectProperties(domain-PAIyER:verificaExistenciaDe domain-PAIyER:existenciaSeVerificaPor)</i>
$obtieneUbicacionDe \equiv ubicacionSeObtienePor^{-}$	\Rightarrow <i>InverseObjectProperties(domain-PAIyER:obtieneUbicacionDe domain-PAIyER:ubicacionSeObtienePor)</i>
$solicitaUbicacionDe \equiv ubicacionSeSolicitaPor^{-}$	\Rightarrow <i>InverseObjectProperties(domain-PAIyER:)</i>

Tabla C.1 TBox de la KB del patrón de análisis PAIyER(cont.)

Sintaxis DL	Sintaxis Funcional
<i>adquiere</i> \equiv <i>esAdquiridoPor</i> ⁻	\Rightarrow <i>solicitaUbicacionDe</i> domain-PAIyER: <i>ubicacionSeSolicitaPor</i>) \Rightarrow <i>InverseObjectProperties</i> (domain-PAIyER: <i>adquiere</i> domain-PAIyER: <i>esAdquiridoPor</i>)
<i>cancela</i> \equiv <i>esCanceladaPor</i> ⁻	\Rightarrow <i>InverseObjectProperties</i> (domain-PAIyER: <i>cancela</i> domain-PAIyER: <i>esCanceladaPor</i>)
<i>crea</i> \equiv <i>esCreadaPor</i> ⁻	\Rightarrow <i>InverseObjectProperties</i> (domain-PAIyER: <i>crea</i> domain-PAIyER: <i>esCreadaPor</i>)
<i>daDeBaja</i> \equiv <i>esDadaDeBaja</i> ⁻	\Rightarrow <i>InverseObjectProperties</i> (domain-PAIyER: <i>daDeBaja</i> domain-PAIyER: <i>esDadaDeBaja</i>)
<i>devuelve</i> \equiv <i>esDevueltaPor</i> ⁻	\Rightarrow <i>InverseObjectProperties</i> (domain-PAIyER: <i>devuelve</i> domain-PAIyER: <i>esDevueltaPor</i>)
<i>expide</i> \equiv <i>esExpedidaPor</i> ⁻	\Rightarrow <i>InverseObjectProperties</i> (domain-PAIyER: <i>expide</i> domain-PAIyER: <i>esExpedidaPor</i>)
<i>modifica</i> \equiv <i>esModificadaPor</i> ⁻	\Rightarrow <i>InverseObjectProperties</i> (domain-PAIyER: <i>modifica</i> domain-PAIyER: <i>esModificadaPor</i>)
<i>reserva</i> \equiv <i>esReservadaPor</i> ⁻	\Rightarrow <i>InverseObjectProperties</i> (domain-PAIyER: <i>reserva</i> domain-PAIyER: <i>esReservadaPor</i>)
<i>usa</i> \equiv <i>esUsadaPor</i> ⁻	\Rightarrow <i>InverseObjectProperties</i> (domain-PAIyER: <i>usa</i> domain-PAIyER: <i>esUsadaPor</i>)
<i>verificaDisponibilidadDe</i> \equiv <i>esVerificadaPor</i> ⁻	\Rightarrow <i>InverseObjectProperties</i> (domain-PAIyER: <i>verificaDisponibilidad</i> domain-PAIyER: <i>esVerificadaPor</i>)
<i>registraUsoDe</i> \equiv <i>suUsoSeRegistraPor</i> ⁻	\Rightarrow <i>InverseObjectProperties</i> (domain-PAIyER: <i>registraUsoDe</i> domain-PAIyER: <i>suUsoSeRegistraPor</i>)
-- Propiedades de datos --	
<i>tieneCantidad</i>	\Rightarrow <i>Declaration</i> (<i>DataProperty</i> (domain-PAIyER: <i>tieneCantidad</i>))
<i>tieneComportamiento</i>	\Rightarrow <i>Declaration</i> (<i>DataProperty</i> (domain-PAIyER: <i>tieneComportamiento</i>))
<i>tieneDireccion</i>	\Rightarrow <i>Declaration</i> (<i>DataProperty</i> (domain-PAIyER: <i>tieneDireccion</i>))
<i>tieneDisponibilidad</i>	\Rightarrow <i>Declaration</i> (<i>DataProperty</i> (domain-PAIyER: <i>tieneDisponibilidad</i>))
<i>tieneEdad</i>	\Rightarrow <i>Declaration</i> (<i>DataProperty</i> (domain-PAIyER: <i>tieneEdad</i>))
<i>tieneEspecificidad</i>	\Rightarrow <i>Declaration</i> (<i>DataProperty</i> (domain-PAIyER: <i>tieneEspecificidad</i>))
<i>tieneFechaInicio</i>	\Rightarrow <i>Declaration</i> (<i>DataProperty</i> (domain-PAIyER: <i>tieneFechaInicio</i>))
<i>tieneFechaFin</i>	\Rightarrow <i>Declaration</i> (<i>DataProperty</i> (domain-PAIyER: <i>tieneFechaFin</i>))
<i>tieneFolio</i>	\Rightarrow <i>Declaration</i> (<i>DataProperty</i> (domain-PAIyER: <i>tieneFolio</i>))
<i>tieneGiro</i>	\Rightarrow <i>Declaration</i> (<i>DataProperty</i> (domain-PAIyER: <i>tieneGiro</i>))
<i>tieneId</i>	\Rightarrow <i>Declaration</i> (<i>DataProperty</i> (domain-PAIyER: <i>tieneId</i>))
<i>tieneNivelEstabilidad</i>	\Rightarrow <i>Declaration</i> (<i>DataProperty</i> (domain-PAIyER: <i>tieneNivelEstabilidad</i>))
<i>tieneNoConfirmacion</i>	\Rightarrow <i>Declaration</i> (<i>DataProperty</i> (domain-PAIyER: <i>tieneNoConfirmacion</i>))
<i>tieneNombre</i>	\Rightarrow <i>Declaration</i> (<i>DataProperty</i> (domain-PAIyER:

Tabla C.1 TBox de la KB del patrón de análisis PAIyER(cont.)

Sintaxis DL	Sintaxis Funcional
<i>tieneNoReservacion</i>	\Rightarrow <i>Declaration(DataProperty(domain-PAIyER: tieneNoReservacion))</i>
<i>tienePercepcion</i>	\Rightarrow <i>Declaration(DataProperty(domain-PAIyER: tienePercepcion))</i>
<i>tienePrecio</i>	\Rightarrow <i>Declaration(DataProperty(domain-PAIyER: tienePrecio))</i>
<i>tieneRfc</i>	\Rightarrow <i>Declaration(DataProperty(domain-PAIyER: tieneRfc))</i>
<i>tieneTelefono</i>	\Rightarrow <i>Declaration(DataProperty(domain-PAIyER: tieneTelefono))</i>
<i>tieneTipo</i>	\Rightarrow <i>Declaration(DataProperty(domain-PAIyER: tieneTipo))</i>
<i>tieneTipoPago</i>	\Rightarrow <i>Declaration(DataProperty(domain-PAIyER: tieneTipoPago))</i>
<i>tieneUbicacion</i>	\Rightarrow <i>Declaration(DataProperty(domain-PAIyER: tieneUbicacion))</i>
-- Restricciones y características de propiedades de datos --	
≥ 1 <i>tieneCantidad</i> \sqsubseteq Entidad	\Rightarrow <i>DataPropertyDomain(domain-PAIyER: tieneCantidad Domain-PAIyER:Entidad)</i>
$T \sqsubseteq \leq 1$ <i>tieneCantidad</i>	\Rightarrow <i>FunctionalDataProperty(domain-PAIyER: tieneCantidad)</i>
$T \sqsubseteq \forall$ <i>tieneCantidad.integer</i>	\Rightarrow <i>DataPropertyRange(domain-PAIyER: tieneCantidad xsd:integer)</i>
≥ 1 <i>tieneComportamiento</i> \sqsubseteq (Entidad \sqcup Empleado \sqcup Inventario \sqcup Institucion)	\Rightarrow <i>DataPropertyDomain(domain-PAIyER: tieneComportamiento domain-PAIyER:Entidad domain-PAIyER:Empleado domain-PAIyER:Inventario domain-PAIyER:Institucion))</i>
$T \sqsubseteq \forall$ <i>tieneComportamiento.string</i>	\Rightarrow <i>DataPropertyRange(domain-PAIyER: tieneComportamiento xsd:string)</i>
$T \sqsubseteq \leq$ <i>tieneComportamiento</i>	\Rightarrow <i>FunctionalDataProperty(domain-PAIyER: tieneComportamiento)</i>
≥ 1 <i>tieneEdad</i> \sqsubseteq Empleado	\Rightarrow <i>DataPropertyDomain(domain-PAIyER: tieneEdad domain-PAIyER:Empleado)</i>
$T \sqsubseteq \forall$ <i>tieneEdad.integer</i>	\Rightarrow <i>DataPropertyRange(domain-PAIyER: tieneEdad xsd:integer)</i>
$T \sqsubseteq \leq$ <i>tieneEdad</i>	\Rightarrow <i>FunctionalDataProperty(domain-PAIyER: tieneEdad)</i>
≥ 1 <i>tieneEspecificidad</i> \sqsubseteq (Entidad \sqcup Empleado \sqcup Inventario \sqcup Institucion)	\Rightarrow <i>DataPropertyDomain(domain-PAIyER:tieneEspecificidad domain-PAIyER:Entidad domain-PAIyER:Empleado domain-PAIyER:Inventario domain-PAIyER:Institucion))</i>
$T \sqsubseteq \forall$ <i>tieneEspecificidad.string</i>	\Rightarrow <i>DataPropertyRange(domain-PAIyER:tieneEspecificidad xsd:string)</i>
$T \sqsubseteq \leq$ <i>tieneEspecificidad</i>	\Rightarrow <i>FunctionalDataProperty(domain-PAIyER: tieneEspecificidad)</i>
≥ 1 <i>tieneGiro</i> \sqsubseteq Institucion	\Rightarrow <i>DataPropertyDomain(domain-PAIyER: tieneGiro domain-PAIyER:Institucion)</i>
$T \sqsubseteq \forall$ <i>tieneGiro.string</i>	\Rightarrow <i>DataPropertyRange(domain-PAIyER: tieneGiro xsd:string)</i>
$T \sqsubseteq \leq$ <i>tieneGiro</i>	\Rightarrow <i>FunctionalDataProperty(domain-PAIyER: tieneGiro)</i>

Tabla C.1 TBox de la KB del patrón de análisis PAIyER(cont.)

Sintaxis DL	Sintaxis Funcional
$\geq I \text{ tieneId} \sqsubseteq (\text{Entidad} \sqcup \text{Empleado} \sqcup \text{Inventario} \sqcup \text{Institucion})$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-PAIyER: tieneId} \text{ domain-PAIyER:Entidad} \text{ domain-PAIyER:Empleado} \text{ domain-PAIyER:Inventario} \text{ domain-PAIyER:Institucion})$
$T \sqsubseteq \forall \text{ tieneId.string}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-PAIyER: tieneId} \text{ xsd:string})$
$T \sqsubseteq \leq \text{ tieneId}$	$\Rightarrow \text{FunctionalDataProperty}(\text{domain-PAIyER: tieneId})$
$\geq I \text{ tieneNivelEstabilidad} \sqsubseteq (\text{Entidad} \sqcup \text{Empleado} \sqcup \text{Inventario} \sqcup \text{Institucion})$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-PAIyER: tieneNivelEstabilidad} \text{ domain-PAIyER:Entidad} \text{ domain-PAIyER:Empleado} \text{ domain-PAIyER:Inventario} \text{ domain-PAIyER:Institucion})$
$T \sqsubseteq \forall \text{ tieneNivelEstabilidad.string}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-PAIyER: tieneNivelEstabilidad} \text{ xsd:string})$
$T \sqsubseteq \leq \text{ tieneNivelEstabilidad}$	$\Rightarrow \text{FunctionalDataProperty}(\text{domain-PAIyER: tieneNivelEstabilidad})$
$\geq I \text{ tieneNombre} \sqsubseteq (\text{Entidad} \sqcup \text{Empleado} \sqcup \text{Inventario} \sqcup \text{Institucion})$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-PAIyER: tieneNombre} \text{ domain-PAIyER:Entidad} \text{ domain-PAIyER:Empleado} \text{ domain-PAIyER:Inventario} \text{ domain-PAIyER:Institucion})$
$T \sqsubseteq \forall \text{ tieneNombre.string}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-PAIyER: tieneNombre} \text{ xsd:string})$
$T \sqsubseteq \leq \text{ tieneNombre}$	$\Rightarrow \text{FunctionalDataProperty}(\text{domain-PAIyER: tieneNombre})$
$\geq I \text{ tienePercepcion} \sqsubseteq (\text{Entidad} \sqcup \text{Empleado} \sqcup \text{Inventario} \sqcup \text{Institucion})$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-PAIyER: tienePercepcion} \text{ domain-PAIyER:Entidad} \text{ domain-PAIyER:Empleado} \text{ domain-PAIyER:Inventario} \text{ domain-PAIyER:Institucion})$
$T \sqsubseteq \forall \text{ tienePercepcion.string}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-PAIyER: tienePercepcion} \text{ xsd:string})$
$T \sqsubseteq \leq \text{ tienePercepcion}$	$\Rightarrow \text{FunctionalDataProperty}(\text{domain-PAIyER: tienePercepcion})$
$\geq I \text{ tieneTipo} \sqsubseteq \text{Entidad}$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-PAIyER: tieneTipo} \text{ domain-PAIyER:Entidad})$
$T \sqsubseteq \forall \text{ tieneTipo.string}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-PAIyER: tieneTipo} \text{ xsd:string})$
$T \sqsubseteq \leq \text{ tieneTipo}$	$\Rightarrow \text{FunctionalDataProperty}(\text{domain-PAIyER: tieneTipo})$
$\geq I \text{ tieneUbicacion} \sqsubseteq \text{Entidad}$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-PAIyER: tieneUbicacion} \text{ Domain-PAIyER:Entidad})$
$T \sqsubseteq \forall \text{ tieneUbicacion.string}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-PAIyER: tieneUbicacion} \text{ xsd:string})$
$T \sqsubseteq \leq I \text{ tieneUbicacion}$	$\Rightarrow \text{FunctionalDataProperty}(\text{domain-PAIyER: tieneUbicacion})$
$\geq I \text{ tieneComportamiento} \sqsubseteq (\text{Contrato} \sqcup \text{Entidad} \sqcup \text{Institucion} \sqcup \text{Persona} \sqcup \text{Propietario} \sqcup \text{Reservacion})$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-PAIyER: tieneComportamiento} \text{ ObjectUnionOf}(\text{domain-PAIyER: Contrato} \text{ domain-PAIyER:Entidad} \text{ domain-PAIyER:Establecimiento} \text{ domain-PAIyER:Persona} \text{ domain-PAIyER:Propietario} \text{ domain-PAIyER:Reservacion}))$

Tabla C.1 TBox de la KB del patrón de análisis PAIyER(cont.)

Sintaxis DL	Sintaxis Funcional
$T \sqsubseteq \leq I \text{tieneComportamiento}$	\Rightarrow <i>FunctionalDataProperty(domain-PAIyER:tieneComportamiento)</i>
$T \sqsubseteq \forall \text{tieneComportamiento.string}$	\Rightarrow <i>DataPropertyRange(domain-PAIyER:tieneComportamiento xsd:string)</i>
$\geq I \text{ tieneDireccion} \sqsubseteq \text{Cliente}$	\Rightarrow <i>DataPropertyDomain(domain-PAIyER:tieneDireccion domain-PAIyER:Cliente)</i>
$T \sqsubseteq \forall \text{tieneDireccion.string}$	\Rightarrow <i>DataPropertyRange(domain-PAIyER:tieneDireccion xsd:string)</i>
$T \sqsubseteq \leq I \text{tieneDireccion}$	\Rightarrow <i>FunctionalDataProperty(domain-PAIyER:tieneDireccion)</i>
$\geq I \text{ tieneDisponibilidad} \sqsubseteq \text{Entidad}$	\Rightarrow <i>DataPropertyDomain(domain-PAIyER:tieneDisponibilidad domain-PAIyER:Entidad)</i>
$T \sqsubseteq \forall \text{tieneDisponibilidad.boolean}$	\Rightarrow <i>DataPropertyRange(domain-PAIyER:tieneDisponibilidad xsd:boolean)</i>
$T \sqsubseteq \leq I \text{tieneDisponibilidad}$	\Rightarrow <i>FunctionalDataProperty(domain-PAIyER:tieneDisponibilidad)</i>
$\geq I \text{ tieneEspecificidad} \sqsubseteq (\text{Contrato} \sqcup \text{Entidad} \sqcup \text{Institucion} \sqcup \text{Persona} \sqcup \text{Propietario} \sqcup \text{Reservacion})$	\Rightarrow <i>DataPropertyDomain(domain-PAIyER:tieneEspecificidad ObjectUnionOf(domain-PAIyER:Contrato domain-PAIyER:Entidad domain-PAIyER:Establecimiento domain-PAIyER:Persona domain-PAIyER:Propietario domain-PAIyER:Reservacion))</i>
$T \sqsubseteq \forall \text{tieneEspecificidad.string}$	\Rightarrow <i>DataPropertyRange(domain-PAIyER:tieneEspecificidad xsd:string)</i>
$\geq I \text{ tieneFechaFin} \sqsubseteq \text{Reservacion}$	\Rightarrow <i>DataPropertyDomain(domain-PAIyER:tieneFechaFin domain-PAIyER:Reservacion)</i>
$T \sqsubseteq \forall \text{tieneFechaFin.dateTime}$	\Rightarrow <i>DataPropertyRange(domain-PAIyER:tieneFechaFin xsd:dateTime)</i>
$T \sqsubseteq \leq I \text{tieneFechaFin}$	\Rightarrow <i>FunctionalDataProperty(domain-PAIyER:tieneFechaFin)</i>
$\geq I \text{ tieneFechaInicio} \sqsubseteq \text{Reservacion}$	\Rightarrow <i>DataPropertyDomain(domain-PAIyER:tieneFechaInicio domain-PAIyER:Reservacion)</i>
$T \sqsubseteq \forall \text{tieneFechaInicio.dateTime}$	\Rightarrow <i>DataPropertyRange(domain-PAIyER:tieneFechaInicio xsd:dateTime)</i>
$T \sqsubseteq \leq I \text{tieneFechaInicio}$	\Rightarrow <i>FunctionalDataProperty(domain-PAIyER:tieneFechaInicio)</i>
$\geq I \text{ tieneFolio} \sqsubseteq \text{Contrato}$	\Rightarrow <i>DataPropertyDomain(domain-PAIyER:tieneFolio domain-PAIyER:Contrato)</i>
$T \sqsubseteq \forall \text{tieneFolio.integer}$	\Rightarrow <i>DataPropertyRange(domain-PAIyER:tieneFolio xsd:integer)</i>
$T \sqsubseteq \leq I \text{tieneFolio}$	\Rightarrow <i>FunctionalDataProperty(domain-PAIyER:tieneFolio)</i>
$\geq I \text{ tieneGiro} \sqsubseteq \text{Institucion}$	\Rightarrow <i>DataPropertyDomain(domain-PAIyER:tieneGiro domain-PAIyER:Establecimiento)</i>
$T \sqsubseteq \forall \text{tieneGiro.string}$	\Rightarrow <i>DataPropertyRange(domain-PAIyER:tieneGiro xsd:string)</i>
$T \sqsubseteq \leq I \text{tieneGiro}$	\Rightarrow <i>FunctionalDataProperty(domain-PAIyER:tieneGiro)</i>

Tabla C.1 TBox de la KB del patrón de análisis PAIyER(cont.)

Sintaxis DL	Sintaxis Funcional
$\geq 1 \text{ tieneId} \sqsubseteq (\text{Entidad} \sqcup \text{Empleado})$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-PAIyER:tieneId} \text{ ObjectUnionOf}(\text{domain-PAIyER:Entidad} \text{ domain-PAIyER:Empleado}))$
$\mathcal{T} \sqsubseteq \forall \text{tieneId.string}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-PAIyER:tieneId} \text{ xsd:string})$
$\mathcal{T} \sqsubseteq \leq 1 \text{ tieneId}$	$\Rightarrow \text{FunctionalDataProperty}(\text{domain-PAIyER:tieneId})$
$\geq 1 \text{ tieneNivelEstabilidad} \sqsubseteq (\text{Contrato} \sqcup \text{Entidad} \sqcup \text{Institucion} \sqcup \text{Persona} \sqcup \text{Propietario} \sqcup \text{Reservacion})$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-PAIyER:tieneNivelEstabilidad} \text{ ObjectUnionOf}(\text{domain-PAIyER:Contrato} \text{ domain-PAIyER:Entidad} \text{ domain-PAIyER:Establecimiento} \text{ domain-PAIyER:Persona} \text{ domain-PAIyER:Propietario} \text{ domain-PAIyER:Reservacion}))$
$\mathcal{T} \sqsubseteq \forall \text{tieneNivelEstabilidad.string}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-PAIyER:tieneNivelEstabilidad} \text{ xsd:string})$
$\geq 1 \text{ tieneNoConfirmacion} \sqsubseteq \text{Reservacion}$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-PAIyER:tieneNoConfirmacion} \text{ domain-PAIyER:Reservacion})$
$\mathcal{T} \sqsubseteq \forall \text{tieneNoConfirmacion.integer}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-PAIyER:tieneNoConfirmacion} \text{ xsd:integer})$
$\mathcal{T} \sqsubseteq \leq 1 \text{ tieneNoConfirmacion}$	$\Rightarrow \text{FunctionalDataProperty}(\text{domain-PAIyER:tieneNoConfirmacion})$
$\geq 1 \text{ tieneNoReservacion} \sqsubseteq \text{Contrato}$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-PAIyER:tieneNoReservacion} \text{ domain-PAIyER:Contrato})$
$\mathcal{T} \sqsubseteq \forall \text{tieneNoReservacion.integer}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-PAIyER:tieneNoReservacion} \text{ xsd:integer})$
$\mathcal{T} \sqsubseteq \leq 1 \text{ tieneNoReservacion}$	$\Rightarrow \text{FunctionalDataProperty}(\text{domain-PAIyER:tieneNoReservacion})$
$\geq 1 \text{ tieneNombre} \sqsubseteq (\text{Entidad} \sqcup \text{Institucion} \sqcup \text{Persona} \sqcup \text{Propietario})$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-PAIyER:tieneNombre} \text{ ObjectUnionOf}(\text{domain-PAIyER:Entidad} \text{ domain-PAIyER:Establecimiento} \text{ domain-PAIyER:Persona} \text{ domain-PAIyER:Propietario}))$
$\mathcal{T} \sqsubseteq \forall \text{tieneNombre.string}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-PAIyER:tieneNombre} \text{ xsd:string})$
$\mathcal{T} \sqsubseteq \leq 1 \text{ tieneNombre}$	$\Rightarrow \text{FunctionalDataProperty}(\text{domain-PAIyER:tieneNombre})$
$\geq 1 \text{ tienePercepcion} \sqsubseteq (\text{Contrato} \sqcup \text{Entidad} \sqcup \text{Institucion} \sqcup \text{Persona} \sqcup \text{Propietario} \sqcup \text{Reservacion})$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-PAIyER:tienePercepcion} \text{ ObjectUnionOf}(\text{domain-PAIyER:Contrato} \text{ domain-PAIyER:Entidad} \text{ domain-PAIyER:Establecimiento} \text{ domain-PAIyER:Persona} \text{ domain-PAIyER:Propietario} \text{ domain-PAIyER:Reservacion}))$
$\mathcal{T} \sqsubseteq \forall \text{tienePercepcion.string}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-PAIyER:tienePercepcion} \text{ xsd:string})$
$\geq 1 \text{ tienePrecio} \sqsubseteq \text{Entidad}$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-PAIyER:tienePrecio} \text{ domain-PAIyER:Entidad})$
$\mathcal{T} \sqsubseteq \forall \text{tienePrecio.double}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-PAIyER:tienePrecio} \text{ xsd:double})$
$\mathcal{T} \sqsubseteq \leq 1 \text{ tienePrecio}$	$\Rightarrow \text{FunctionalDataProperty}(\text{domain-PAIyER:tienePrecio})$

Tabla C.1 TBox de la KB del patrón de análisis PAIyER(cont.)

Sintaxis DL	Sintaxis Funcional
$\geq 1 \text{ tieneRfc} \sqsubseteq \text{Propietario}$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-PAIyER:tieneRfc} \text{ domain-PAIyER:Propietario})$
$\mathcal{T} \sqsubseteq \forall \text{tieneRfc.string}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-PAIyER:tieneRfc} \text{ xsd:string})$
$\mathcal{T} \sqsubseteq \leq 1 \text{ tieneRfc}$	$\Rightarrow \text{FunctionalDataProperty}(\text{domain-PAIyER:tieneRfc})$
$\geq 1 \text{ tieneTelefono} \sqsubseteq \text{Cliente}$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-PAIyER:tieneTelefono} \text{ domain-PAIyER:Cliente})$
$\mathcal{T} \sqsubseteq \forall \text{tieneTelefono.string}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-PAIyER:tieneTelefono} \text{ xsd:string})$
$\mathcal{T} \sqsubseteq \leq 1 \text{ tieneTelefono}$	$\Rightarrow \text{FunctionalDataProperty}(\text{domain-PAIyER:tieneTelefono})$
$\geq 1 \text{ tieneTipo} \sqsubseteq (\text{Entidad} \sqcup \text{Contrato})$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-PAIyER:tieneTipo} \text{ ObjectUnionOf}(\text{domain-PAIyER:Entidad} \text{ domain-PAIyER:Contrato}))$
$\mathcal{T} \sqsubseteq \forall \text{tieneTipo.string}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-PAIyER:tieneTipo} \text{ xsd:string})$
$\mathcal{T} \sqsubseteq \leq 1 \text{ tieneTipo}$	$\Rightarrow \text{FunctionalDataProperty}(\text{domain-PAIyER:tieneTipo})$
$\geq 1 \text{ tieneTipoPago} \sqsubseteq \text{Reservacion}$	$\Rightarrow \text{DataPropertyDomain}(\text{domain-PAIyER:tieneTipoPago} \text{ domain-PAIyER:Reservacion})$
$\mathcal{T} \sqsubseteq \forall \text{tieneTipoPago.string}$	$\Rightarrow \text{DataPropertyRange}(\text{domain-PAIyER:tieneTipoPago} \text{ xsd:string})$
$\mathcal{T} \sqsubseteq \leq 1 \text{ tieneTipoPago}$	$\Rightarrow \text{FunctionalDataProperty}(\text{domain-PAIyER:tieneTipoPago})$

Para una mejor comprensión de las descripciones lógicas presentadas en la TBox de la Tabla C.1, a continuación se describen algunos ejemplos tomados de las secciones *Clases*, *Clases Disjuntas*, *Restricciones existenciales y universales*, *Propiedades de objeto*, *Restricciones y características de propiedades de objeto*, *Propiedades de datos y Restricciones y características de propiedades de datos* en la misma tabla:

Clases

\mathcal{T} : Conjunto de todos los individuos en el dominio.

$\text{Persona} \sqsubseteq \mathcal{T}$: La clase Persona es un tipo de individuos en el dominio.

$\text{Empleado} \sqsubseteq \text{Persona}$: La clase Empleado es un tipo de la clase Persona.

Clases Disjuntas

$\text{Contrato} \sqcap \text{Entidad} \sqcap \text{Institucion} \sqcap \text{Inventario} \sqcap \text{Empleado} \sqcap \text{Cliente} \sqcap \text{Propietario} \sqcap \text{Reservacion} \sqcap \sqsubseteq \perp$: Establece que los individuos que son miembros de una de las clases (Contrato, Entidad, Institucion, Inventario, Empleado, Cliente, Propietario y Reservación), no son instancias de otra clase.

Restricciones existenciales y universales

$\text{Inventario} \sqsubseteq \forall \text{seActualizaPor.Empleado}$: Establece que un Inventario es todo aquello que es actualizado por un Empleado.

$Empleado \sqsubseteq \exists actualiza.Inventario$: Establece que un Empleado es quien al menos actualiza un inventario.

Propiedades de objeto

$actualiza$: representa la declaración de la propiedad de objeto actualiza.

Restricciones y características de propiedades de objeto

$\exists actualiza.T \sqsubseteq Empleado$: Establece a la clase Empleado en el dominio de la propiedad de objeto actualiza.

$T \sqsubseteq \forall actualiza.Inventario$: Establece a la clase Inventario como rango de la propiedad de objeto actualiza.

$T \sqsubseteq \leq 1 actualiza$: Establece que la propiedad de objeto actualiza es una propiedad de objeto funcional.

$actualiza \equiv seActualizaPor^{-}$: Establece que la propiedad de objeto actualiza es la propiedad inversa de la propiedad de objeto seActualizaPor.

Propiedades de datos

$tieneComportamiento$: representa la declaración de la propiedad de datos tieneComportamiento.

Restricciones y características de propiedades de datos

$\geq 1 tieneComportamiento \sqsubseteq (Contrato \sqcup Entidad \sqcup Inventario \sqcup Institucion \sqcup Empleado \sqcup Cliente \sqcup Propietario \sqcup Reservacion)$: Establece a las clases Contrato, Entidad, Institucion, Inventario, Empleado, Cliente, Propietario y Reservacion en el dominio de la propiedad de datos tieneComportamiento.

$T \sqsubseteq \forall tieneComportamiento.string$: Establece el tipo de datos string como el rango de la propiedad de objeto tieneComportamiento.

$T \sqsubseteq \leq 1 tieneComportamiento$: Establece que la propiedad de datos tieneComportamiento es una propiedad de datos funcional.

Como se puede observar, la definición de algunas propiedades de datos consiste en nombres de conceptos basados en el modelo SSM (Comportamiento ($tieneComportamiento$), Especificidad ($tieneEspecificidad$), Nivel de estabilidad ($tieneNivelEstabilidad$) y Percepción ($tienePercepcion$)) y permiten relacionar individuos abstractos (individuo *Valentino* de la clase Empleado) con individuos concretos (valor “*Inestable*” de tipo string), por ejemplo, $tieneComportamiento(Valentino, “Inestable”)$. Por otra parte, existen propiedades de datos que definen atributos como: nombre, giro, edad, entre otros. Cada propiedad de datos se caracteriza por ser funcional (“ $T \sqsubseteq \leq 1$ ” en las DLs y “*FunctionalDataProperty*” en OWL-DL) y tener un dominio (“ $\geq 1 U \sqsubseteq C$ ” en las DLs “*DataPropertyDomain*” en OWL-DL) y rango (“ $T \sqsubseteq \forall U.D$ ” en las DLs y “*DataPropertyRange*” en OWL-DL) definidos.

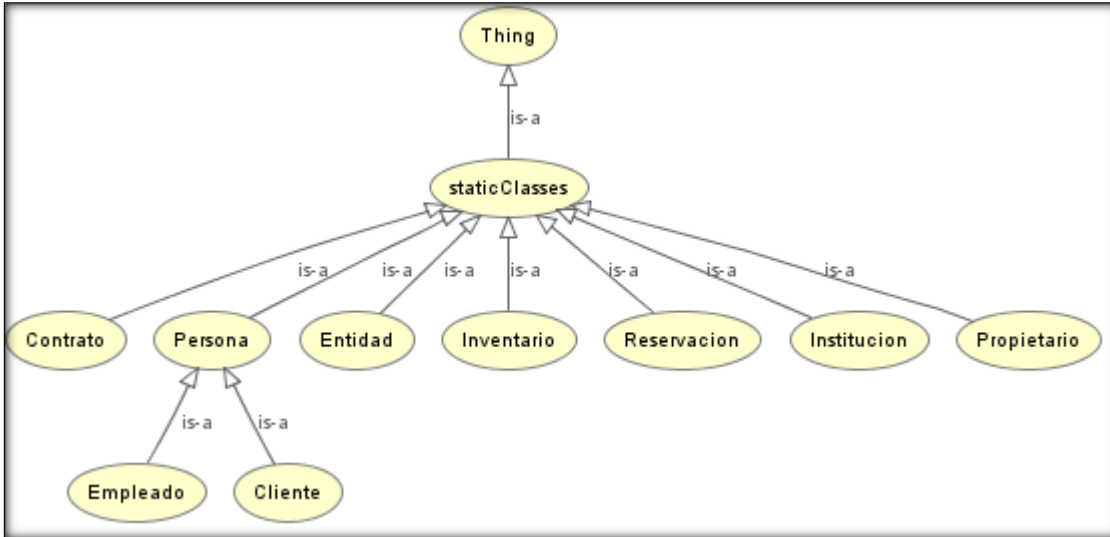


Figura C.1 Jerarquía de clases en la KB del patrón de análisis PAIyER.

1.2. Componente extensional

Para obtener el componente *ABox*, se utilizaron individuos (nombres de individuos abstractos N_{IA} y nombres de individuos concretos N_{IC} en las DLs) como:

$N_{IA} = \{ \text{Libro1, Valentino, Heidi, Biblioteca_CENIDET, CENIDET, Inventario_CENIDET, Contrato_Heidi, Reservacion_prestamo_Heidi} \}$

$N_{IC} = \{ \text{string, Boolean, integer, double, dateTime, long} \}$

en donde los individuos abstractos Libro1, Valentino, Heidi, Biblioteca_CENIDET, CENIDET, Inventario_CENIDET, Contrato_Heidi, Reservacion_prestamo_Heidi son aplicables al contexto del patrón de análisis PAIyER.

El detalle acerca de las aserciones de propiedades de datos que hacen uso de los individuos concretos en la *ABox*, pueden consultarse en la versión completa de la KB (OPAE) del patrón de análisis PAIyER que se encuentra en el directorio *Ontologías/OPAE PAIyER/*, almacenado en el disco que acompaña esta tesis (para mayor referencia consultar el *Anexo G. Contenido del disco*). Al igual que en la ontología OPAE, algunas de estas aserciones contemplan propiedades de datos basadas en el modelo SSM como son *Comportamiento, Especificidad, Nivel de estabilidad y Percepción*, entre otras.

El componente *ABox*, de la KB del patrón de análisis PAIyER, se describe en la Tabla C.2.

Tabla C.2 ABox de la KB del patrón de análisis PAIyER (cont.)

Sintaxis DL	Sintaxis Funcional
<i>ABox = {</i>	
<i>-- Aserciones de clase --</i>	
<i>Entidad(Libro1)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro1)</i>
<i>Institucion(Biblioteca_CENIDET)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Institucion domain-PAIyER:Biblioteca_CENIDET)</i>
<i>Inventario(Inventario_CENIDET)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Inventario domain-PAIyER:Inventario_CENIDET)</i>
<i>Propietario(CENIDET)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Propietario domain-PAIyER:CENIDET)</i>
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Cliente(Heidi)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Heidi)</i>
<i>Reservacion(Reservacion_prestamo_Heidi)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Reservacion domain-PAIyER:Reservacion_prestamo_Heidi)</i>
<i>Contrato(Contrato_Heidi)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Contrato domain-PAIyER:Contrato_Heidi)</i>
<i>-- Aserciones de propiedades de objeto --</i>	
<i>actualiza(Valentino, Inventario_CENIDET)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Valentino domain-PAIyER:Inventario_CENIDET)</i>
<i>seActualizaPor(Inventario_CENIDET, Valentino)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:seActualizaPor domain-PAIyER:Inventario_CENIDET domain-PAIyER:Valentino)</i>
<i>obtieneExistenciasDe(Valentino, Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:obtieneExistenciasDe domain-PAIyER:Valentino domain-PAIyER:Libro1)</i>
<i>existenciasSeObtienePor(Libro1, Valentino)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:existenciasSeObtienePor domain-PAIyER:Libro1 domain-PAIyER:Valentino)</i>
<i>buscaExistenciasDe(Valentino, Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:buscaExistenciasDe domain-PAIyER:Valentino domain-PAIyER:Libro1)</i>
<i>existenciasSeBuscaPor(Libro1, Valentino)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:existenciasSeBuscaPor domain-PAIyER:Libro1 domain-PAIyER:Valentino)</i>
<i>especificaTipoExistenciasDe(Valentino, Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:especificaTipoExistenciasDe domain-PAIyER:Valentino domain-PAIyER:Libro1)</i>
<i>tipoExistenciasSeEspecificaPor(Libro1, Valentino)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:tipoExistenciasSeEspecificaPor domain-PAIyER:Libro1 domain-PAIyER:Valentino)</i>
<i>registra(Valentino, Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:registra domain-PAIyER:Valentino domain-PAIyER:Libro1)</i>
<i>seRegistraPor(Libro1, Valentino)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:seRegistraPor domain-PAIyER:Libro1 domain-PAIyER:Valentino)</i>

Tabla C.2 ABox de la KB del patrón de análisis PAIyER (cont.)

Sintaxis DL	Sintaxis Funcional
<i>separa</i> (Valentino, Libro1)	\Rightarrow <i>domain-PAIyER:Valentino</i> \Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:separa domain-PAIyER:Valentino domain-PAIyER:Libro1)</i>
<i>seSeparaPor</i> (Libro1, Valentino)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:seSeparaPor domain-PAIyER:Libro1 domain-PAIyER:Valentino)</i>
<i>ubica</i> (Valentino, Libro1)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:ubica domain-PAIyER:Valentino domain-PAIyER:Libro1)</i>
<i>seUbicaPor</i> (Libro1, Valentino)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:seUbicaPor domain-PAIyER:Libro1 domain-PAIyER:Valentino)</i>
<i>verificaExistenciaDe</i> (Valentino, Libro1)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaExistenciaDe domain-PAIyER:Valentino domain-PAIyER:Libro1)</i>
<i>existenciaSeVerificaPor</i> (Libro1, Valentino)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:existenciaSeVerificaPor domain-PAIyER:Libro1 domain-PAIyER:Valentino)</i>
<i>laboraEn</i> (Valentino, Biblioteca_CENIDET)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:laboraEn domain-PAIyER:Valentino domain-PAIyER: Biblioteca_CENIDET)</i>
<i>obtieneUbicacionDe</i> (Valentino, Libro1)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:obtieneUbicacionDe domain-PAIyER:Valentino domain-PAIyER:Libro1)</i>
<i>ubicacionSeObtienePor</i> (Libro1, Valentino)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:ubicacionSeObtienePor domain-PAIyER:Libro1 domain-PAIyER:Valentino)</i>
<i>solicitaUbicacionDe</i> (Valentino, Libro1)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:solicitaUbicacionDe domain-PAIyER:Valentino domain-PAIyER:Libro1)</i>
<i>ubicacionSeSolicitaPor</i> (Libro1, Valentino)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:ubicacionSeSolicitaPor domain-PAIyER:Libro1 domain-PAIyER:Valentino)</i>
<i>tieneDatosDe</i> (Biblioteca_CENIDET, Valentino)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:tieneDatosDe domain-PAIyER: Biblioteca_CENIDET domain-PAIyER:Valentino)</i>
<i>crea</i> (Valentino, Reservacion_prestamo_Heidi)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:crea domain-PAIyER:Valentino domain-PAIyER:Reservacion_prestamo_Heidi)</i>
<i>esCreadaPor</i> (Reservacion_prestamo_Heidi, Valentino)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:esCreadaPor domain-PAIyER: Reservacion_prestamo_Heidi domain-PAIyER:Valentino)</i>
<i>adquiere</i> (Heidi, Contrato_Heidi)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:adquiere domain-PAIyER:Heidi domain-PAIyER:Contrato_Heidi)</i>
<i>esAdquiridoPor</i> (Contrato_Heidi, Heidi)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:esAdquiridoPor domain-</i>

Tabla C.2 ABox de la KB del patrón de análisis PAIyER (cont.)

Sintaxis DL	Sintaxis Funcional
<i>devuelve(Heidi, Libro1)</i>	\Rightarrow <i>PAIyER:Contrato_Heidi domain-PAIyER:Heidi)</i> <i>ObjectPropertyAssertion(domain-PAIyER:devuelve domain-PAIyER:Heidi domain-PAIyER:Libro1)</i>
<i>esCanceladaPor(Reservacion_prestamo_Heidi, Valentino)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:esCanceladaPor domain-PAIyER:Reservacion_prestamo_Heidi domain-PAIyER:Valentino)</i>
<i>daDeBaja(Heidi, Reservacion_prestamo_Heidi)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:daDeBaja domain-PAIyER:Heidi domain-PAIyER: Reservacion_prestamo_Heidi)</i>
<i>esDadaDeBajaPor(Reservacion_prestamo_Heidi, Heidi)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:esDadaDeBajaPor domain-PAIyER:Reservacion_prestamo_Heidi domain-PAIyER:Heidi)</i>
<i>esDevueltaPor(Libro1, Heidi)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:esDevueltaPor domain-PAIyER: Libro1 domain-PAIyER: Heidi)</i>
<i>esDuenyoDe(CENIDET, Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:esDuenyoDe domain-PAIyER:CENIDET domain-PAIyER: Libro1)</i>
<i>esExpedidoPor(Contrato_Heidi, Valentino)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:esExpedidoPor domain-PAIyER:Contrato_Heidi domain-PAIyER: Valentino)</i>
<i>esModificadaPor(Reservacion_prestamo_Heidi, Heidi)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:esModificadaPor domain-PAIyER:Reservacion_prestamo_Heidi domain-PAIyER:Heidi)</i>
<i>esReservadaPor(Libro1, Heidi)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:esReservadaPor domain-PAIyER:Libro1 domain-PAIyER: Heidi)</i>
<i>esUsadaPor(Libro1, Heidi)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:esUsadaPor domain-PAIyER: Libro1 domain-PAIyER: Heidi)</i>
<i>esVerificadaPor(Libro1, Valentino)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:esVerificadaPor domain-PAIyER:Libro1 domain-PAIyER: Valentino)</i>
<i>expide(Valentino, Contrato_Heidi)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:expide domain-PAIyER: Valentino domain-PAIyER: Contrato_Heidi)</i>
<i>cancela(Valentino, Reservacion_prestamo_Heidi)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:cancela domain-PAIyER: Valentino domain-PAIyER: Reservacion_prestamo_Heidi)</i>
<i>modifica(Heidi, Reservacion_prestamo_Heidi)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:modifica domain-PAIyER: Heidi domain-PAIyER: Reservacion_prestamo_Heidi)</i>
<i>perteneceA(Libro1, CENIDET)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:perteneceA domain-PAIyER: Libro1 domain-PAIyER: CENIDET)</i>
<i>registraUsoDe(Valentino, Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:registraUsoDe domain-PAIyER:Valentino domain-PAIyER: Libro1)</i>
<i>reserva(Heidi, Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:reserva domain-PAIyER: Heidi domain-</i>

Tabla C.2 ABox de la KB del patrón de análisis PAIyER (cont.)

Sintaxis DL	Sintaxis Funcional
<i>suUsoEsRegistradaPor(Libro1, Valentino)</i>	<i>PAIyER: Libro1</i> <i>ObjectPropertyAssertion(domain-</i> <i>PAIyER:suUsoEsRegistradaPor domain-</i> <i>PAIyER: Libro1 domain-PAIyER: Valentino)</i>
<i>usa(Heidi, Libro1)</i>	<i>ObjectPropertyAssertion(domain-PAIyER:usa</i> <i>domain-PAIyER: Heidi domain-PAIyER:</i> <i>Libro1)</i>
<i>verificaDisponibilidadDe(Valentino, Libro1)</i>	<i>ObjectPropertyAssertion(domain-</i> <i>PAIyER:verificaDisponibilidadDe domain-</i> <i>PAIyER: Valentino domain-PAIyER: Libro1)</i>
-- Aserciones de propiedades de datos --	
<i>tieneCantidad(Libro1, 2)</i>	<i>⇒ DataPropertyAssertion(domain-</i> <i>PAIyER:tieneCantidad domain-PAIyER:Libro1</i> <i>"2"^^xsd:integer)</i>
<i>tieneComportamiento(Libro1, "estable-adaptable")</i>	<i>⇒ DataPropertyAssertion(domain-</i> <i>PAIyER:tieneComportamiento domain-</i> <i>PAIyER:Libro1 "estable-adaptable"^^xsd:string)</i>
<i>tieneComportamiento(Valentino, "inestable")</i>	<i>⇒ DataPropertyAssertion(domain-</i> <i>PAIyER:tieneComportamiento domain-</i> <i>PAIyER:Valentino "inestable"^^xsd:string)</i>
<i>tieneComportamiento(Inventario_CENIDET,</i> <i>"estable-adaptable")</i>	<i>⇒ DataPropertyAssertion(domain-</i> <i>PAIyER:tieneComportamiento domain-</i> <i>PAIyER:Inventario_CENIDET "estable-</i> <i>adaptable"^^xsd:string)</i>
<i>tieneComportamiento(Biblioteca_CENIDET,</i> <i>"inestable")</i>	<i>⇒ DataPropertyAssertion(domain-</i> <i>PAIyER:tieneComportamiento domain-PAIyER:</i> <i>Biblioteca_CENIDET "inestable"^^xsd:string)</i>
<i>tieneComportamiento(Contrato_Heidi, "estable-</i> <i>adaptable")</i>	<i>⇒ DataPropertyAssertion(domain-</i> <i>PAIyER:tieneComportamiento domain-PAIyER:</i> <i>Contrato_Heidi "estable-daptable"^^xsd:string)</i>
<i>tieneComportamiento(Reservacion_prestamo_Heidi,</i> <i>"estable-adaptable")</i>	<i>⇒ DataPropertyAssertion(domain-</i> <i>PAIyER:tieneComportamiento domain-PAIyER:</i> <i>Reservacion_prestamo_Heidi "estable-daptable</i> <i>"^^xsd:string)</i>
<i>tieneComportamiento(CENIDET, "inestable")</i>	<i>⇒ DataPropertyAssertion(domain-</i> <i>PAIyER:tieneComportamiento domain-PAIyER:</i> <i>CENIDET "inestable"^^xsd:string)</i>
<i>tieneComportamiento(Heidi, "inestable")</i>	<i>⇒ DataPropertyAssertion(domain-</i> <i>PAIyER:tieneComportamiento domain-PAIyER:</i> <i>Heidi "inestable"^^xsd:string)</i>
<i>tienePercepcion (Libro1, "semi-tangible")</i>	<i>⇒ DataPropertyAssertion(domain-PAIyER:</i> <i>tienePercepcion domain-PAIyER:Libro1 "semi-</i> <i>tangible"^^xsd:string)</i>
<i>tienePercepcion (Valentino, "tangible")</i>	<i>⇒ DataPropertyAssertion(domain-PAIyER:</i> <i>tienePercepcion domain-PAIyER:Valentino</i> <i>"tangible"^^xsd:string)</i>
<i>tienePercepcion (Inventario_CENIDET, "semi-</i> <i>tangible")</i>	<i>⇒ DataPropertyAssertion(domain-PAIyER:</i> <i>tienePercepcion domain-</i> <i>PAIyER:Inventario_CENIDET "semi-</i> <i>tangible"^^xsd:string)</i>
<i>tienePercepcion (Biblioteca_CENIDET, "tangible")</i>	<i>⇒ DataPropertyAssertion(domain-PAIyER:</i> <i>tienePercepcion domain-PAIyER:</i> <i>Biblioteca_CENIDET "tangible"^^xsd:string)</i>
<i>tienePercepcion(Contrato_Heidi, "semi-tangible")</i>	<i>⇒ DataPropertyAssertion(domain-</i>

Tabla C.2 ABox de la KB del patrón de análisis PAIyER (cont.)

Sintaxis DL	Sintaxis Funcional
<i>tienePercepcion(Reservacion_prestamo_Heidi, "semi-tangible")</i>	⇒ <i>PAIyER:tienePercepcion domain-PAIyER: Contrato_Heidi "semi-tangible"^^xsd:string</i> ⇒ <i>DataPropertyAssertion(domain-PAIyER:tienePercepcion domain-PAIyER:Reservacion_prestamo_Heidi "semi-tangible"^^xsd:string)</i>
<i>tienePercepcion(CENIDET, "tangible")</i>	⇒ <i>DataPropertyAssertion(domain-PAIyER:tienePercepcion domain-PAIyER:CENIDET "tangible"^^xsd:string)</i>
<i>tienePercepcion(Heidi, "tangible")</i>	⇒ <i>DataPropertyAssertion(domain-PAIyER:tienePercepcion domain-PAIyER:Heidi "tangible"^^xsd:string)</i>
<i>tieneEspecificidad (Libro1, "general")</i>	⇒ <i>DataPropertyAssertion(domain-PAIyER:tieneEspecificidad domain-PAIyER:Libro1 "general"^^xsd:string)</i>
<i>tieneEspecificidad (Valentino, "particular")</i>	⇒ <i>DataPropertyAssertion(domain-PAIyER:tieneEspecificidad domain-PAIyER:Valentino "particular"^^xsd:string)</i>
<i>tieneEspecificidad (Inventario_CENIDET, "general")</i>	⇒ <i>DataPropertyAssertion(domain-PAIyER:tieneEspecificidad domain-PAIyER:Inventario_CENIDET "general"^^xsd:string)</i>
<i>tieneEspecificidad (Biblioteca_CENIDET, "particular")</i>	⇒ <i>DataPropertyAssertion(domain-PAIyER:tieneEspecificidad domain-PAIyER:Biblioteca_CENIDET "particular"^^xsd:string)</i>
<i>tieneEspecificidad (Contrato_Heidi, "general")</i>	⇒ <i>DataPropertyAssertion(domain-PAIyER:tieneEspecificidad domain-PAIyER:Contrato_Heidi "general"^^xsd:string)</i>
<i>tieneEspecificidad (Reservacion_prestamo_Heidi, "particular")</i>	⇒ <i>DataPropertyAssertion(domain-PAIyER:tieneEspecificidad domain-PAIyER:Reservacion_prestamo_Heidi "particular"^^xsd:string)</i>
<i>tieneEspecificidad (CENIDET, "particular")</i>	⇒ <i>DataPropertyAssertion(domain-PAIyER:tieneEspecificidad domain-PAIyER:CENIDET "particular"^^xsd:string)</i>
<i>tieneEspecificidad (Heidi, "particular")</i>	⇒ <i>DataPropertyAssertion(domain-PAIyER:tieneEspecificidad domain-PAIyER:Heidi "particular"^^xsd:string)</i>
<i>tieneNivelEstabilidad (Libro1, "BO")</i>	⇒ <i>DataPropertyAssertion(domain-PAIyER:tieneNivelEstabilidad domain-PAIyER:Libro1 "BO"^^xsd:string)</i>
<i>tieneNivelEstabilidad (Valentino, "IO")</i>	⇒ <i>DataPropertyAssertion(domain-PAIyER:tieneNivelEstabilidad domain-PAIyER:Valentino "IO"^^xsd:string)</i>
<i>tieneNivelEstabilidad (Inventario_CENIDET, "BO")</i>	⇒ <i>DataPropertyAssertion(domain-PAIyER:tieneNivelEstabilidad domain-PAIyER:Inventario_CENIDET "BO"^^xsd:string)</i>
<i>tieneNivelEstabilidad (Biblioteca_CENIDET, "IO")</i>	⇒ <i>DataPropertyAssertion(domain-PAIyER:tieneNivelEstabilidad domain-PAIyER:Biblioteca_CENIDET "IO"^^xsd:string)</i>
<i>tieneNivelEstabilidad (Contrato_Heidi, "BO")</i>	⇒ <i>DataPropertyAssertion(domain-PAIyER:tieneNivelEstabilidad domain-PAIyER:Contrato_Heidi "BO"^^xsd:string)</i>

Tabla C.2 ABox de la KB del patrón de análisis PAIyER (cont.)

Sintaxis DL	Sintaxis Funcional
<i>tieneNivelEstabilidad</i> (Reservacion_prestamo_Heidi, "BO")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneNivelEstabilidad</i> domain-PAIyER: Reservacion_prestamo_Heidi "BO"^^xsd:string)
<i>tieneNivelEstabilidad</i> (CENIDET, "IO")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneNivelEstabilidad</i> domain-PAIyER: CENIDET "IO"^^xsd:string)
<i>tieneNivelEstabilidad</i> (Heidi, "IO")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneNivelEstabilidad</i> domain-PAIyER: Biblioteca_CENIDET "IO"^^xsd:string)
<i>tieneEdad</i> (Valentino, 26)	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneEdad</i> domain-PAIyER:Valentino "26"^^xsd:integer)
<i>tieneGiro</i> (Biblioteca_CENIDET, "Préstamo de libros y gestión de Inventario")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneGiro</i> domain-PAIyER: Biblioteca_CENIDET " Préstamo de libros y gestión de Inventario"^^xsd:string)
<i>tieneId</i> (Libro1, "C000001")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneId</i> domain-PAIyER:Libro1 "C000001"^^xsd:string)
<i>tieneId</i> (Valentino, "E000001")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneId</i> domain-PAIyER:Valentino " E000001"^^xsd:string)
<i>tieneId</i> (Inventario_CENIDET, "I000001")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneId</i> domain-PAIyER:Inventario_CENIDET " I000001"^^xsd:string)
<i>tieneId</i> (Biblioteca_CENIDET, "000001")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneId</i> domain-PAIyER:CENIDET "000001"^^xsd:string)
<i>tieneNombre</i> (Libro1, "El llano en llamas")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneNombre</i> domain-PAIyER:Libro1 " El llano en llamas "^^xsd:string)
<i>tieneNombre</i> (Valentino, "Valentino Velázquez Olivera")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneNombre</i> domain-PAIyER:Valentino " Valentino Velázquez Olivera "^^xsd:string)
<i>tieneNombre</i> (Inventario_CENIDET, "Sistema de control de inventario CENIDET")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneNombre</i> domain- PAIyER:Inventario_CENIDET " Sistema de control de inventario CENIDET "^^xsd:string)
<i>tieneNombre</i> (Biblioteca_CENIDET, "Centro Nacional de Investigación y Desarrollo Tecnológico")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneNombre</i> domain-PAIyER: Biblioteca_CENIDET " Centro Nacional de Investigación y Desarrollo Tecnológico "^^xsd:string)
<i>tieneNombre</i> (CENIDET, "CENIDET")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneNombre</i> domain-PAIyER: CENIDET " CENIDET "^^xsd:string)
<i>tieneRfc</i> (CENIDET, "SEP210905778")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneRfc</i> domain-PAIyER: CENIDET " SEP210905778"^^xsd:string)
<i>tieneNombre</i> (Heidi, "Heidi Barrera")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneNombre</i> domain-PAIyER:Heidi " Heidi Barrera "^^xsd:string)
<i>tieneTelefono</i> (Heidi, "7771235467")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneTelefono</i> domain-PAIyER:Heidi " 7771235467"^^xsd:string)

Tabla C.2 ABox de la KB del patrón de análisis PAIyER (cont.)

Sintaxis DL	Sintaxis Funcional
<i>tieneDireccion</i> (Heidi, "Domicilio conocido")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneDireccion</i> domain-PAIyER:Heidi " Domicilio conocido"^^xsd:string)
<i>tieneFolio</i> (Contrato_Heidi, "1")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneFolio</i> domain-PAIyER:Heidi " 1"^^xsd:integer)
<i>tieneNoReservacion</i> (Contrato_Heidi, "000001")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneNoReservacion</i> domain-PAIyER:Heidi " 000001"^^xsd:integer)
<i>tieneTipo</i> (Libro1, "Componente")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneTipo</i> domain-PAIyER:Libro1 "Componente"^^xsd:string)
<i>tieneTipo</i> (Contrato_Heidi, "tarjeta")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneTipo</i> domain-PAIyER:Contrato_Heidi "tarjeta"^^xsd:string)
<i>tieneUbicacion</i> (Libro1, "En el pasillo #2, en el librero #4 en el stand #3")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneUbicacion</i> domain-PAIyER:Libro1 " En el pasillo #2, en el librero #4 en el stand #3"^^xsd:string)
<i>tieneFechaInicio</i> (Reservacion_prestamo_Heidi, "2019-09-02T00:00:00")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneFechaInicio</i> domain-PAIyER:Reservacion_prestamo_Heidi "2019-09-02T00:00:00"^^xsd:dateTime)
<i>tieneFechaFin</i> (Reservacion_prestamo_Heidi, "2019-09-10T00:00:00")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneFechaFin</i> domain-PAIyER:Reservacion_prestamo_Heidi "2019-09-10T00:00:00"^^xsd:dateTime)
<i>tieneTipoPago</i> (Reservacion_prestamo_Heidi, "tarjetaCENIDET")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneTipoPago</i> domain-PAIyER:Reservacion_prestamo_Heidi "tarjetaCENIDET"^^xsd:string)
<i>tieneNoConfirmacion</i> (Reservacion_prestamo_Heidi, "000001")	⇒ <i>DataPropertyAssertion</i> (domain-PAIyER: <i>tieneNoConfirmacion</i> domain-PAIyER:Reservacion_prestamo_Heidi "000001"^^xsd:integer)

Para una mejor comprensión de las descripciones lógicas presentadas en la ABox de la Tabla C.2, se describen a continuación algunos ejemplos tomados de las secciones *Aserciones de clase*, *Aserciones de propiedades de objeto* y *Aserciones de propiedades de datos* en la misma tabla:

Aserciones de clase

Entidad(Libro1): Afirma que Libro1 es un individuo de la clase Entidad.

Empleado(Valentino): Afirma que Valentino es un individuo de la clase Empleado.

Institucion(Biblioteca_CENIDET): Afirma que Biblioteca_CENIDET es un individuo de la clase Institucion.

Inventario(Inventario_CENIDET): Afirma que Inventario_CENIDET es un individuo de la clase Inventario.

Aserciones de propiedades de objeto

actualiza(Valentino,Inventario_CENIDET): Afirma que el empleado Valentino actualiza el inventario Inventario_CENIDET.

ubica(Valentino,Libro1): Afirma que el empleado Valentino ubica el Entidad Libro1.

Aserciones de propiedades de datos

tieneComportamiento(Biblioteca_CENIDET, "inestable"): Afirma que la institución Biblioteca_CENIDET tiene un comportamiento inestable en términos del modelo SSM.

tieneComportamiento(Valentino, "Inestable"): Afirma que el empleado Valentino tiene un tipo de comportamiento Inestable en términos del modelo SSM.

Anexo D. Ontología OTPAE del patrón de análisis PAIyER

En este anexo, se presenta la última versión de la ontología OTPAE del Patrón de Análisis para Inventarios y Entidades Reutilizables PAIyER, una combinación de la OTPAE del PAI con la OTPAE del PAER obtenida a partir del método COMOTPA, propuesto en este trabajo de investigación.

1. Base de conocimiento

En esta sección se presentan algunas de las clases, propiedades de objeto y propiedades de objeto temporales en la KB con aspectos temporales (OTPAE) del patrón de análisis PAIyER. Esta ontología hereda las características y restricciones de las propiedades de objeto y de las propiedades de datos definidas en la ontología OPAE descrita en el *Anexo C. Ontología OPAE del patrón de análisis PAIyER* y agrega algunas otras pertenecientes al dominio de la ontología OWL-Time. De la ontología OWL-Time, solo se hace referencia a algunas características y restricciones asociadas a sus propiedades de objeto.

Una versión completa de la KB se encuentra en el directorio *Ontologías/OTPAE PAIyER/*, almacenado en el disco que acompaña esta tesis (para mayor referencia consultar el *Anexo G. Contenido del disco*). Debido a que en una ontología temporal puede llegar a existir una extensa proliferación de aserciones de clase y de roles (individuos), se elaboraron dos ontologías temporales por cada RF para facilitar su lectura y comprensión. Estas ontologías describen el comportamiento dinámico considerando casos de éxito y una combinación de casos de éxito y fracaso/alternativo, respectivamente.

La *TBox* (o componente intencional) y la *ABox* (componente extensional) de la KB, se describen bajo el poder de expresividad de la DL *ALCOIN(D)*. En ambos componentes se incluyen las equivalencias correspondientes en la sintaxis funcional de OWL-DL.

1.1. Componente intencional

Para obtener el componente *TBox*, fue necesario modificar algunos axiomas de la ontología OPAE del patrón de análisis PAIyER, para incluir en su definición los conceptos temporales de la ontología OWL-Time. En su construcción se utilizaron:

1. Clases (en OWL-DL o nombres de conceptos N_C en las DLs):
 - *OWL-Time*: {dynamicClasses, DateTimeDescription, DayOfWeak, DurationDescription, Year, Event, TemporalEntit, Instant, Interval, ProperInterval, DateTimeInterval, TemporalUnit, TimeZone}
 - *Dominio del patrón de análisis PAIyER*: {staticClasses, Contrato, Entidad, Institucion, Inventario, Persona, Empleado, Cliente, Propietario, Reservacion }

a partir de las cuales se generó la jerarquía de clases (nombres de concepto N_C en las DLs) de la Figura D.1, que modela el conocimiento central del patrón de análisis PAIyER.

2. Propiedades de objeto estáticas (en OWL-DL o nombres de roles abstractos N_{RA} en las DLs):

- *OWL-Time*: { dynamicOP, after, before, intervalBefore, dayOfWeak, during, hasBeginning, hasDateTimeDescription, hasEnd, inDateTime, inside, intervalAfter, intervalMeets, intervalMeets, intervalMetBy, overlaps, equals, intervalContains, intervalDuring, intervalEquals, intervalFinishes, intervalOverlappedBy, intervalOverlaps, intervalStartedBy intervalStarts, timeZone, unitType }
 - *Dominio del patrón de análisis PAIyER*: { staticOP, actualiza, adquiere, aplicaPolíticas, buscaExistenciasDe, cancela, crea, daDeBaja, devuelve, seActualizaPor, esAdquiridoPor, esCanceladaPor, esCreadaPor, esDadaDeBajaPor, esDevueltaPor, esDuenyoDe, esExpedidoPor, esModificadaPor, especificaTipoExistenciasDe, seRegistraPor, esReservadaPor, seSeparaPor, seUbicaPor, esUsadaPor, esVerificadaPor, existenciaSeVerificaPor, existenciasSeBuscaPor, existenciasSeObtienePor, expide, laboraEn, modifica, obtieneExistenciasDe, obtieneUbicacionDe, perteneceA, registra, registraUsoDe, reserva, separa, solicitaUbicacionDe, suUsoEsRegistradaPor, tieneDatosDe, tipoExistenciasEsEspecificadaPor, ubica, ubicacionSeObtienePor, ubicacionSeSolicitaPor, usa, verificaDisponibilidadDe, verificaExistenciaDe }
3. Propiedades de objeto temporales
- *OWL-Time y dominio del patrón de análisis PAIyER*: { dynamicOP, participatesIn, actualiza, adquiere, buscaExistenciasDe, cancela, crea, daDeBaja, devuelve, seActualizaPor, esAdquiridoPor, esCanceladaPor, esCreadaPor, esDadaDeBajaPor, esDevueltaPor, esExpedidoPor, esModificadaPor, especificaTipoExistenciasDe, seRegistraPor, esReservadaPor, seSeparaPor, seUbicaPor, esUsadaPor, esVerificadaPor, existenciaSeVerificaPor, existenciasSeBuscaPor, existenciasSeObtienePor, expide, modifica, obtieneExistenciasDe, obtieneUbicacionDe, registra, registraUsoDe, reserva, separa, solicitaUbicacionDe, suUsoEsRegistradaPor, tipoExistenciasEsEspecificadaPor, ubica, ubicacionSeObtienePor, ubicacionSeSolicitaPor, usa, verificaDisponibilidadDe, verificaExistenciaDe }
4. Propiedades de datos estáticas (en OWL-DL o nombres de roles concretos N_{RC} en las DLs):
- *OWL-Time*: { dynamicDP, inXSDDateTime }
 - *Dominio del patrón de análisis PAIyER*: { staticDP, tieneCantidad, tieneComportamiento, tieneDireccion, tieneDisponibilidad, tieneEdad, tieneEspecificidad, tieneFechaFin, tieneFechaInicio, tieneFolio, tieneGiro, tieneId, tieneNivelEstabilidad, tieneNoConfirmacion, tieneNombre, tieneNoReservacion, tienePercepcion, tienePrecio, tieneRfc, tieneTelefono, tieneTipo, tieneTipoPago, tieneUbicacion }

5. Propiedades de datos temporales

Ninguna ($\{\emptyset\}$).

6. Espacios de nombres (complemento a los prefijos descritos en el Anexo C. Ontología OPAAE del patrón de análisis PAIyER):

- Espacio de nombres para el lenguaje de reglas de la web semántica (SWRL)
Prefix(swrl:=<http://www.w3.org/2003/11/swrl#>)
- Espacio de nombres para los términos de la ontología OWL-Time.
Prefix(time:=<http://www.w3.org/2006/time#>)
- Espacio de nombres de Incorporaciones de SWRL.
Prefix(swrlb:=<http://www.w3.org/2003/11/swrlb#>)
- Ontología de zona horaria.
Prefix(timeZone:=<http://www.w3.org/2006/timezone#>)
- Vocabulario en el dominio del patrón de análisis PAIyER (aspectos estáticos).
Prefix(domain-PAIyER:=<https://www.cenidet.edu.mx/SoftwareEngineering/AnalysisPatterns/TemporalOntology/PAIyER#>)
- Versión estable de la ontología OWL-Time de conceptos temporales proveída por el plugin Chronos.
Prefix(timeEvents:=<http://www.intelligence.tuc.gr/2011/timeEvents#>)

Las propiedades de objeto temporales fueron utilizadas en la representación del comportamiento expresado en los requerimientos funcionales del patrón de análisis PAIyER.

Debido a que cada una de las propiedades de objeto temporales cuenta con características tales como funcional o inversa funcional, en la Tabla D.3 se describen las reglas SWRL que las representan. Estas reglas son el resultado del proceso de conversión de propiedades de objeto a propiedades de objeto temporales realizado por el plugin Chronos en el editor de ontologías Protégé.

Tabla D.3 Reglas SWRL en características funcional e inversa funcional de propiedades de objeto en el patrón de análisis PAIyER (cont.)

Propiedad de objeto	Característica	Regla SWRL
<i>actualiza</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), actualiza(?e1, ?y1), actualiza(?e2, ?y2), actualiza(?x, ?e1), actualiza(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>buscaExistenciasDe</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), buscaExistenciasDe(?e1, ?y), buscaExistenciasDe(?e2, ?y),</i>

Tabla D.3 Reglas SWRL en características funcional e inversa funcional de propiedades de objeto en el patrón de análisis PAIyER (cont.)

Propiedad de objeto	Característica	Regla SWRL
		<i>buscaExistenciasDe(?x1, ?e1), buscaExistenciasDe(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>seActualizaPor</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), seActualizaPor(?e1, ?y), seActualizaPor(?e2, ?y), seActualizaPor(?x1, ?e1), seActualizaPor(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>especificaTipoExistenciasDe</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), especificaTipoExistenciasDe(?e1, ?y), especificaTipoExistenciasDe(?e2, ?y), especificaTipoExistenciasDe(?x1, ?e1), especificaTipoExistenciasDe(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>seRegistraPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), seRegistraPor(?e1, ?y1), seRegistraPor(?e2, ?y2), seRegistraPor(?x, ?e1), seRegistraPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>seSeparaPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), seSeparaPor(?e1, ?y1), seSeparaPor(?e2, ?y2), seSeparaPor(?x, ?e1), seSeparaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>seUbicaPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), seUbicaPor(?e1, ?y1), seUbicaPor(?e2, ?y2), seUbicaPor(?x, ?e1), seUbicaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>existenciaEsVerificadoPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), existenciaSeVerificaPor(?e1, ?y1), existenciaSeVerificaPor(?e2, ?y2), existenciaSeVerificaPor(?x, ?e1), existenciaSeVerificaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>existenciasSeBuscaPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), existenciasSeBuscaPor(?e1, ?y1), existenciasSeBuscaPor(?e2, ?y2), existenciasSeBuscaPor(?x, ?e1), existenciasSeBuscaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>existenciasSeObtienePor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), existenciasSeObtienePor(?e1, ?y1), existenciasSeObtienePor(?e2, ?y2), existenciasSeObtienePor(?x, ?e1), existenciasSeObtienePor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>laboraEn</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1),</i>

Tabla D.3 Reglas SWRL en características funcional e inversa funcional de propiedades de objeto en el patrón de análisis PAIyER (cont.)

Propiedad de objeto	Característica	Regla SWRL
		<i>during(?e2, ?i2), overlaps(?i1, ?i2), laboraEn(?e1, ?y1), laboraEn(?e2, ?y2), laboraEn(?x, ?e1), laboraEn(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>obtieneExistenciasDe</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), obtieneExistenciasDe(?e1, ?y), obtieneExistenciasDe(?e2, ?y), obtieneExistenciasDe(?x1, ?e1), obtieneExistenciasDe(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>obtieneUbicacionDe</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), obtieneUbicacionDe(?e1, ?y), obtieneUbicacionDe(?e2, ?y), obtieneUbicacionDe(?x1, ?e1), obtieneUbicacionDe(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>registra</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), registra(?e1, ?y), registra(?e2, ?y), registra(?x1, ?e1), registra(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>separa</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), separa(?e1, ?y), separa(?e2, ?y), separa(?x1, ?e1), separa(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>solicitaUbicacionDe</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), solicitaUbicacionDe(?e1, ?y1), solicitaUbicacionDe(?e2, ?y2), solicitaUbicacionDe(?x, ?e1), solicitaUbicacionDe(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>tipoExistenciasSeEspecificaPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), tipoExistenciasSeEspecificaPor(?e1, ?y1), tipoExistenciasSeEspecificaPor(?e2, ?y2), tipoExistenciasSeEspecificaPor(?x, ?e1), tipoExistenciasSeEspecificaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>ubica</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), ubica(?e1, ?y), ubica(?e2, ?y), ubica(?x1, ?e1), ubica(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>ubicacionSeObtienePor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), ubicacionSeObtienePor(?e1, ?y1), ubicacionSeObtienePor(?e2, ?y2), ubicacionSeObtienePor(?x, ?e1), ubicacionSeObtienePor(?x, ?e2) -> SameAs (?y1, ?y2)</i>

Tabla D.3 Reglas SWRL en características funcional e inversa funcional de propiedades de objeto en el patrón de análisis PAIyER (cont.)

Propiedad de objeto	Característica	Regla SWRL
<i>ubicacionSeSolicitaPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), ubicacionSeSolicitaPor(?e1, ?y), ubicacionSeSolicitaPor(?e2, ?y), ubicacionSeSolicitaPor(?x1, ?e1), ubicacionSeSolicitaPor(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>verificaExistenciaDe</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), verificaExistenciaDe(?e1, ?y), verificaExistenciaDe(?e2, ?y), verificaExistenciaDe(?x1, ?e1), verificaExistenciaDe(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>esCanceladaPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), esCanceladaPor(?e1, ?y1), esCanceladaPor(?e2, ?y2), esCanceladaPor(?x, ?e1), esCanceladaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>verificaDisponibilidadDe</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), verificaDisponibilidadDe(?e1, ?y), verificaDisponibilidadDe(?e2, ?y), verificaDisponibilidadDe(?x1, ?e1), verificaDisponibilidadDe(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>esAdquiridoPor</i>	<i>funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), esAdquiridoPor(?e1, ?y1), esAdquiridoPor(?e2, ?y2), esAdquiridoPor(?x, ?e1), esAdquiridoPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>esVerificadaPor</i>	<i>funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), esVerificadaPor(?e1, ?y1), esVerificadaPor(?e2, ?y2), esVerificadaPor(?x, ?e1), esVerificadaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>esDevoltaPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), esDevoltaPor(?e1, ?y1), esDevoltaPor(?e2, ?y2), esDevoltaPor(?x, ?e1), esDevoltaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>devuelve</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), devuelve(?e1, ?y), devuelve(?e2, ?y), devuelve(?x1, ?e1), devuelve(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>usa</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), usa(?e1, ?y), usa(?e2, ?y), usa(?x1, ?e1), usa(?x2, ?e2) -> SameAs (?x1, ?x2)</i>

Tabla D.3 Reglas SWRL en características funcional e inversa funcional de propiedades de objeto en el patrón de análisis PAIyER (cont.)

Propiedad de objeto	Característica	Regla SWRL
<i>daDeBaja</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), daDeBaja(?e1, ?y), daDeBaja(?e2, ?y), daDeBaja(?x1, ?e1), daDeBaja(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>crea</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), crea(?e1, ?y), crea(?e2, ?y), crea(?x1, ?e1), crea(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>suUsoEsRegistradaPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), suUsoEsRegistradaPor(?e1, ?y1), suUsoEsRegistradaPor(?e2, ?y2), suUsoEsRegistradaPor(?x, ?e1), suUsoEsRegistradaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>reserva</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), reserva(?e1, ?y), reserva(?e2, ?y), reserva(?x1, ?e1), reserva(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>esCreadaPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), esCreadaPor(?e1, ?y1), esCreadaPor(?e2, ?y2), esCreadaPor(?x, ?e1), esCreadaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>esDadaDeBajaPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), esDadaDeBajaPor(?e1, ?y1), esDadaDeBajaPor(?e2, ?y2), esDadaDeBajaPor(?x, ?e1), esDadaDeBajaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>registraUsoDe</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), registraUsoDe(?e1, ?y), registraUsoDe(?e2, ?y), registraUsoDe(?x1, ?e1), registraUsoDe(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>esExpedidoPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), esExpedidoPor(?e1, ?y1), esExpedidoPor(?e2, ?y2), esExpedidoPor(?x, ?e1), esExpedidoPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>expide</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), expide(?e1, ?y), expide(?e2, ?y), expide(?x1, ?e1), expide(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>cancela</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), cancela(?e1, ?y), cancela(?e2, ?y), cancela(?x1, ?e1), cancela(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>modifica</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2),</i>

Tabla D.3 Reglas SWRL en características funcional e inversa funcional de propiedades de objeto en el patrón de análisis PAIyER (cont.)

Propiedad de objeto	Característica	Regla SWRL
		<i>modifica(?e1, ?y), modifica(?e2, ?y), modifica(?x1, ?e1), modifica(?x2, ?e2) -> SameAs (?x1, ?x2)</i>
<i>esReservadaPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), esReservadaPor(?e1, ?y1), esReservadaPor(?e2, ?y2), esReservadaPor(?x, ?e1), esReservadaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>esModificadaPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), esModificadaPor(?e1, ?y1), esModificadaPor(?e2, ?y2), esModificadaPor(?x, ?e1), esModificadaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>esUsadaPor</i>	<i>Funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), esUsadaPor(?e1, ?y1), esUsadaPor(?e2, ?y2), esUsadaPor(?x, ?e1), esUsadaPor(?x, ?e2) -> SameAs (?y1, ?y2)</i>
<i>adquiere</i>	<i>Inversa funcional</i>	<i>Event(?e1), Event(?e2), during(?e1, ?i1), during(?e2, ?i2), overlaps(?i1, ?i2), adquiere(?e1, ?y), adquiere(?e2, ?y), adquiere(?x1, ?e1), adquiere(?x2, ?e2) -> SameAs (?x1, ?x2)</i>

El componente *TBox*, resultado de la inclusión de aspectos temporales en la KB de la ontología OPAE del patrón de análisis PAIyER, se describe en la Tabla D.4.

Tabla D.4 TBox de la KB del patrón de análisis PAIyER con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
<i>TBox = {</i>	
<i>-- Clases --</i>	
<i>T</i>	<i>⇒ Declaration(Class(owl:Thing))</i>
<i>staticClasses ⊆ T</i>	<i>⇒ Declaration(Class(domain-PAIyER:staticClasses)) SubClassOf(domain-PAIyER:staticClasses owl:Thing)</i>
<i>dynamicClasses ⊆ T</i>	<i>⇒ Declaration(Class(domain-PAIyER:dynamicClasses)) SubClassOf(domain-PAIyER:dynamicClasses owl:Thing)</i>
<i>Contrato ⊆ staticClasses</i>	<i>⇒ Declaration(Class(domain-PAIyER:Contrato)) SubClassOf(domain-PAIyER:Contrato domain- PAIyER:staticClasses)</i>
<i>Entidad ⊆ staticClasses</i>	<i>⇒ Declaration(Class(domain-PAIyER:Entidad)) SubClassOf(domain-PAIyER:Entidad domain- PAIyER:staticClasses)</i>
<i>Institucion ⊆ staticClasses</i>	<i>⇒ Declaration(Class(domain-PAIyER:Institucion)) SubClassOf(domain-PAIyER:Institucion domain- PAIyER:staticClasses)</i>
<i>Inventario ⊆ staticClasses</i>	<i>⇒ Declaration(Class(domain-PAIyER:Inventario)) SubClassOf(domain-PAIyER:Inventario domain- PAIyER:staticClasses)</i>

Tabla D.4 TBox de la KB del patrón de análisis PAIyER con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
$Persona \sqsubseteq staticClasses$	$\Rightarrow Declaration(Class(domain-PAIyER:Persona))$ $SubClassOf(domain-PAIyER:Persona domain-PAIyER:staticClasses)$
$Propietario \sqsubseteq staticClasses$	$\Rightarrow Declaration(Class(domain-PAIyER:Propietario))$ $SubClassOf(domain-PAIyER:Propietario domain-PAIyER:staticClasses)$
$Resrevacion \sqsubseteq staticClasses$	$\Rightarrow Declaration(Class(domain-PAIyER:Resrevacion))$ $SubClassOf(domain-PAIyER:Resrevacion domain-PAIyER:staticClasses)$
$Empleado \sqsubseteq Persona$	$\Rightarrow Declaration(Class(domain-PAIyER:Empleado))$ $SubClassOf(domain-PAIyER:Empleado domain-PAIyER:Persona)$
$Cliente \sqsubseteq Persona$	$\Rightarrow Declaration(Class(domain-PAIyER:Cliente))$ $SubClassOf(domain-PAIyER:Cliente domain-PAIyER:Persona)$
$Event \sqsubseteq dynamicClasses$	$\Rightarrow Declaration(Class())SubClassOf(timeEvents:Event domain-PAIyER:dynamicClasses)$
$TemporalEntity \sqsubseteq dynamicClasses$	$\Rightarrow Declaration(Class(time:TemporalEntity))$ $SubClassOf(time:TemporalEntity domain-PAIyER:dynamicClasses)$
$Instant \sqsubseteq TemporalEntity$	$\Rightarrow Declaration(Class(time:Instant))$ $SubClassOf(time:Instant time:TemporalEntity)$
$Interval \sqsubseteq TemporalEntity$	$\Rightarrow Declaration(Class(time:Interval))$ $SubClassOf(time:Interval time:TemporalEntity)$
$ProperInterval \sqsubseteq Interval$	$\Rightarrow Declaration(Class(time:ProperInterval))$ $SubClassOf(time:ProperInterval time:Interval)$
$DateTimeInterval \sqsubseteq ProperInterval$	$\Rightarrow Declaration(Class(time:DateTimeInterval))$ $SubClassOf(time:DateTimeInterval time:ProperInterval)$
-- Clases disjuntas --	
$ProperInterval \sqcap Instant \sqsubseteq \perp$	$\Rightarrow DisjointClasses(time:Instant time:ProperInterval)$
$Contrato \sqcap Entidad \sqcap Inventario \sqcap Empleado \sqcap Institucion \sqcap Cliente \sqcap Propietario \sqcap Reservacion \sqsubseteq \perp$	$\Rightarrow DisjointClasses(domain-PAIyER:Contrato domain-PAIyER:Entidad domain-PAIyER:Inventario domain-PAIyER:Empleado domain-PAIyER:Institucion domain-PAIyER:Cliente domain-PAIyER:Propietario domain-PAIyER:Reservacion)$
-- Restricciones existenciales y universales --	
$Empleado \sqsubseteq \exists separa.(Event \exists separa.Entidad)$	$\Rightarrow SubClassOf(domain-PAIyER:Empleado ObjectSomeValuesFrom(domain-PAIyER:separa ObjectIntersectionOf(ObjectSomeValuesFrom(domain-PAIyER:separa Domain-PAIyER:Entidad)))$
$Empleado \sqsubseteq \exists registra.(Event \exists registra.Entidad)$	$\Rightarrow SubClassOf(domain-PAIyER:Empleado ObjectSomeValuesFrom(domain-PAIyER:registra ObjectIntersectionOf(ObjectSomeValuesFrom(domain-PAIyER:registra Domain-PAIyER:Entidad)))$
$Empleado \sqsubseteq \exists actualiza.(Event \exists actualiza.Inventario)$	$\Rightarrow SubClassOf(domain-PAIyER:Empleado ObjectSomeValuesFrom(domain-PAIyER:actualiza ObjectIntersectionOf(ObjectSomeValuesFrom(domain-PAIyER:actualiza Domain-PAIyER:Inventario)))$
$Empleado \sqsubseteq \exists buscaExistenciasDe.(Event \exists buscaExistenciasDe.Entidad)$	$\Rightarrow SubClassOf(domain-PAIyER:Empleado ObjectSomeValuesFrom(domain-PAIyER:buscaExistenciasDe ObjectIntersectionOf(ObjectSomeValuesFrom(domain-$

Tabla D.4 TBox de la KB del patrón de análisis PAIyER con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
$Empleado \sqsubseteq \exists verificaExistenciaDe.(Event \exists verificaExistenciaDe.Entidad)$	$\Rightarrow PAIyER:buscaExistenciasDe Domain-PAIyER:Entidad))$ $SubClassOf(domain-PAIyER:Empleado ObjectSomeValuesFrom(domain-PAIyER:verificaExistenciaDe ObjectIntersectionOf(ObjectSomeValuesFrom(domain-PAIyER:verificaExistenciaDe Domain-PAIyER:Entidad))$
$Empleado \sqsubseteq \exists obtieneUbicacionDe.(Event \exists obtieneUbicacionDe.Entidad)$	$\Rightarrow SubClassOf(domain-PAIyER:Empleado ObjectSomeValuesFrom(domain-PAIyER:obtieneUbicacionDe ObjectIntersectionOf(ObjectSomeValuesFrom(domain-PAIyER:obtieneUbicacionDe Domain-PAIyER:Entidad))$
$Empleado \sqsubseteq \exists especificaTipoExistenciasDe.(Event \exists especificaTipoExistenciasDe.Entidad)$	$\Rightarrow SubClassOf(domain-PAIyER:Empleado ObjectSomeValuesFrom(domain-PAIyER:especificaTipoExistenciasDe ObjectIntersectionOf(ObjectSomeValuesFrom(domain-PAIyER:especificaTipoExistenciasDe Domain-PAIyER:Entidad))$
$Empleado \sqsubseteq \exists obtieneExistenciasDe.(Event \exists obtieneExistenciasDe.Entidad)$	$\Rightarrow SubClassOf(domain-PAIyER:Empleado ObjectSomeValuesFrom(domain-PAIyER:obtieneExistenciasDe ObjectIntersectionOf(ObjectSomeValuesFrom(domain-PAIyER:obtieneExistenciasDe Domain-PAIyER:Entidad))$
$Empleado \sqsubseteq \exists ubica.(Event \exists ubica.Entidad)$	$\Rightarrow SubClassOf(domain-PAIyER:Empleado ObjectSomeValuesFrom(domain-PAIyER:ubica ObjectIntersectionOf(ObjectSomeValuesFrom(domain-PAIyER:ubica Domain-PAIyER:Entidad))$
$Empleado \sqsubseteq \exists solicitaUbicacionDe.(Event \exists solicitaUbicacionDe.Entidad)$	$\Rightarrow SubClassOf(domain-PAIyER:Empleado ObjectSomeValuesFrom(domain-PAIyER:solicitaUbicacionDe ObjectIntersectionOf(ObjectSomeValuesFrom(domain-PAIyER:solicitaUbicacionDe Domain-PAIyER:Entidad))$
$Empleado \sqsubseteq \exists cancela.(Event \exists cancela.Reservacion)$	$\Rightarrow SubClassOf(domain-PAIyER:Empleado ObjectSomeValuesFrom(domain-PAIyER:cancela ObjectIntersectionOf(ObjectSomeValuesFrom(domain-PAIyER:cancela Domain-PAIyER:Reservacion))$
$Empleado \sqsubseteq \exists registraUsoDe.(Event \exists registraUsoDe.Entidad)$	$\Rightarrow SubClassOf(domain-PAIyER:Empleado ObjectSomeValuesFrom(domain-PAIyER:registraUsoDe ObjectIntersectionOf(ObjectSomeValuesFrom(domain-PAIyER:registraUsoDe Domain-PAIyER:Entidad))$
$Empleado \sqsubseteq \exists verificaDisponibilidadDe.(Event \exists verificaDisponibilidadDe.Entidad)$	$\Rightarrow SubClassOf(domain-PAIyER:Empleado ObjectSomeValuesFrom(domain-PAIyER:verificaDisponibilidadDe ObjectIntersectionOf(ObjectSomeValuesFrom(domain-PAIyER:verificaDisponibilidadDe Domain-PAIyER:Entidad))$
$Cliente \sqsubseteq \exists adquiere.(Event \exists adquiere.Contrato)$	$\Rightarrow SubClassOf(domain-PAIyER:Cliente ObjectSomeValuesFrom(domain-PAIyER:adquiere ObjectIntersectionOf(ObjectSomeValuesFrom(domain-$

Tabla D.4 TBox de la KB del patrón de análisis PAIyER con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
$Cliente \sqsubseteq \exists daDeBaja.(Event \exists daDeBaja.Reservacion)$	$\Rightarrow PAIyER: adquiere Domain-PAIyER:Contrato))$ $\Rightarrow SubClassOf(domain-PAIyER:Cliente$ $ObjectSomeValuesFrom(domain-PAIyER: daDeBaja$ $ObjectIntersectionOf(ObjectSomeValuesFrom(domain-$ $PAIyER: daDeBaja Domain-PAIyER:Reservacion))$
$Cliente \sqsubseteq \exists devuelve.(Event \exists devuelve.Entidad)$	$\Rightarrow SubClassOf(domain-PAIyER:Cliente$ $ObjectSomeValuesFrom(domain-PAIyER: devuelve$ $ObjectIntersectionOf(ObjectSomeValuesFrom(domain-$ $PAIyER: devuelve Domain-PAIyER:Entidad))$
$Cliente \sqsubseteq \exists modifica.(Event \exists modifica.Reservacion)$	$\Rightarrow SubClassOf(domain-PAIyER:Cliente$ $ObjectSomeValuesFrom(domain-PAIyER: modifica$ $ObjectIntersectionOf(ObjectSomeValuesFrom(domain-$ $PAIyER: modifica Domain-PAIyER:Reservacion))$
$Cliente \sqsubseteq \exists reserva.(Event \exists reserva.Entidad)$	$\Rightarrow SubClassOf(domain-PAIyER:Cliente$ $ObjectSomeValuesFrom(domain-PAIyER: reserva$ $ObjectIntersectionOf(ObjectSomeValuesFrom(domain-$ $PAIyER: reserva Domain-PAIyER:Entidad))$
$Cliente \sqsubseteq \exists usa.(Event \exists usa.Entidad)$	$\Rightarrow SubClassOf(domain-PAIyER:Cliente$ $ObjectSomeValuesFrom(domain-PAIyER: usa$ $ObjectIntersectionOf(ObjectSomeValuesFrom(domain-$ $PAIyER: usa Domain-PAIyER:Entidad))$
$Entidad \sqsubseteq \forall seSeparaPor.(Event \forall seSeparaPor.Empleado)$	$\Rightarrow SubClassOf(domain-PAIyER:Entidad$ $ObjectAllValuesFrom(domain-PAIyER: seSeparaPor$ $ObjectIntersectionOf(ObjectAllValuesFrom(domain-$ $PAIyER: seSeparaPor domain-PAIyER:Empleado))$
$Entidad \sqsubseteq \forall seRegistraPor.(Event \forall seRegistraPor.Empleado)$	$\Rightarrow SubClassOf(domain-PAIyER:Entidad$ $ObjectSomeValuesFrom(domain-PAIyER:$ $seRegistraPor$ $ObjectIntersectionOf(ObjectSomeValuesFrom(domain-$ $PAIyER: seRegistraPor domain-PAIyER:Empleado))$
$Entidad \sqsubseteq \forall esDevueltaPor.(Event \forall esDevueltaPor.Cliente)$	$\Rightarrow SubClassOf(domain-PAIyER:Entidad$ $ObjectSomeValuesFrom(domain-PAIyER:$ $esDevueltaPor$ $ObjectIntersectionOf(ObjectSomeValuesFrom(domain-$ $PAIyER: esDevueltaPor Domain-PAIyER:Cliente))$
$Entidad \sqsubseteq \forall esUsadaPor.(Event \forall esUsadaPor.Cliente)$	$\Rightarrow SubClassOf(domain-PAIyER:Entidad$ $ObjectSomeValuesFrom(domain-PAIyER: esUsadaPor$ $ObjectIntersectionOf(ObjectSomeValuesFrom(domain-$ $PAIyER: esUsadaPor Domain-PAIyER:Cliente))$
$Entidad \sqsubseteq \forall esVerificadoPor.(Event \forall esVerificadoPor.Empleado)$	$\Rightarrow SubClassOf(domain-PAIyER:Entidad$ $ObjectSomeValuesFrom(domain-PAIyER:$ $esVerificadoPor$ $ObjectIntersectionOf(ObjectSomeValuesFrom(domain-$ $PAIyER: esVerificadoPor Domain-PAIyER:Empleado$ $))$
$Contrato \sqsubseteq \forall esAdquiridoPor.(Event \forall esAdquiridoPor.Cliente)$	$\Rightarrow SubClassOf(domain-PAIyER:Contrato$ $ObjectSomeValuesFrom(domain-PAIyER:$ $esAdquiridoPor$ $ObjectIntersectionOf(ObjectSomeValuesFrom(domain-$ $PAIyER: esAdquiridoPor Domain-PAIyER:Cliente))$
$Contrato \sqsubseteq \forall esExpedidoPor.(Event \forall esExpedidoPor.Empleado)$	$\Rightarrow SubClassOf(domain-PAIyER:Contrato$ $ObjectSomeValuesFrom(domain-PAIyER:$ $esExpedidoPor$ $ObjectIntersectionOf(ObjectSomeValuesFrom(domain-$

Tabla D.4 TBox de la KB del patrón de análisis PAIyER con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
$Empleado \sqsubseteq \forall actualiza.(Event \forall actualiza.Reservacion)$	$\Rightarrow PAIyER: esExpedidoPor Domain-PAIyER:Empleado))$ $\Rightarrow SubClassOf(domain-PAIyER:Empleado$ $ObjectAllValuesFrom(domain-PAIyER: actualiza$ $ObjectIntersectionOf(ObjectAllValuesFrom(domain-$ $PAIyER: actualiza domain-PAIyER:Reservacion)))$
$Empleado \sqsubseteq \forall crea.(Event \forall crea.Reservacion)$	$\Rightarrow SubClassOf(domain-PAIyER:Empleado$ $ObjectAllValuesFrom(domain-PAIyER: crea$ $ObjectIntersectionOf(ObjectAllValuesFrom(domain-$ $PAIyER: crea domain-PAIyER:Reservacion)))$
$Empleado \sqsubseteq \forall expide.(Event \forall expide.Contrato)$	$\Rightarrow SubClassOf(domain-PAIyER:Empleado$ $ObjectAllValuesFrom(domain-PAIyER: expide$ $ObjectIntersectionOf(ObjectAllValuesFrom(domain-$ $PAIyER: expide domain-PAIyER:Contrato)))$
$Reservacioin \sqsubseteq \forall esCreadaPor.(Event \forall esCreadaPor.Empleado)$	$\Rightarrow SubClassOf(domain-PAIyER:Reservacion$ $ObjectAllValuesFrom(domain-PAIyER: esCreadaPor$ $ObjectIntersectionOf(ObjectAllValuesFrom(domain-$ $PAIyER: esCreadaPor domain-PAIyER:Empleado)))$
-- Propiedades de objeto temporales -- <i>dynamicOP</i>	$\Rightarrow Declaration(ObjectProperty(domain-$ $PAIyER:dynamicOP))$
<i>during</i> $\sqsubseteq dynamicOP$	$\Rightarrow Declaration(ObjectProperty(time:during))$ $SubObjectPropertyOf(time:during domain-$ $PAIyER:dynamicOP)$
<i>hasBeginning</i> $\sqsubseteq dynamicOP$	$\Rightarrow Declaration(ObjectProperty(time:hasBeginning))$ $SubObjectPropertyOf(time:hasBeginning domain-$ $PAIyER:dynamicOP)$
<i>hasEnd</i> $\sqsubseteq dynamicOP$	$\Rightarrow Declaration(ObjectProperty(time:hasEnd))$ $SubObjectPropertyOf(time:hasEnd domain-$ $PAIyER:dynamicOP)$
<i>participatesIn</i> $\sqsubseteq dynamicOP$	$\Rightarrow Declaration(ObjectProperty(timeEvents:participatesIn)$ $)$ $SubObjectPropertyOf(timeEvents:participatesIn$ $domain-PAIyER:dynamicOP)$
<i>actualiza</i> $\sqsubseteq participatesIn$	$\Rightarrow SubObjectPropertyOf(domain-PAIyER:actualiza$ $timeEvents:participatesIn)$
<i>adquiere</i> $\sqsubseteq participatesIn$	$\Rightarrow SubObjectPropertyOf(domain-PAIyER:adquiere$ $timeEvents:participatesIn)$
<i>buscaExistenciasDe</i> $\sqsubseteq participatesIn$	$\Rightarrow SubObjectPropertyOf(domain-PAIyER:$ $buscaExistenciasDe timeEvents:participatesIn)$
<i>cancela</i> $\sqsubseteq participatesIn$	$\Rightarrow SubObjectPropertyOf(domain-PAIyER:cancela$ $timeEvents:participatesIn)$
<i>crea</i> $\sqsubseteq participatesIn$	$\Rightarrow SubObjectPropertyOf(domain-PAIyER:crea$ $timeEvents:participatesIn)$
<i>daDeBaja</i> $\sqsubseteq participatesIn$	$\Rightarrow SubObjectPropertyOf(domain-PAIyER:daDeBaja$ $timeEvents:participatesIn)$
<i>devuelve</i> $\sqsubseteq participatesIn$	$\Rightarrow SubObjectPropertyOf(domain-PAIyER:devuelve$ $timeEvents:participatesIn)$
<i>seActualizaPor</i> $\sqsubseteq participatesIn$	$\Rightarrow SubObjectPropertyOf(domain-PAIyER: seActualizaPor$ $timeEvents:participatesIn)$
<i>esAdquiridoPor</i> $\sqsubseteq participatesIn$	$\Rightarrow SubObjectPropertyOf(domain-PAIyER:esAdquiridoPor$

Tabla D.4 TBox de la KB del patrón de análisis PAIyER con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
<i>esCanceladaPor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>timeEvents:participatesIn</i> <i>SubObjectPropertyOf(domain-PAIyER:esCanceladaPor timeEvents:participatesIn)</i>
<i>esCreadaPor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER:esCreadaPor timeEvents:participatesIn)</i>
<i>esDadaDeBajaPor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER:esDadaDeBajaPor timeEvents:participatesIn)</i>
<i>esDevueltaPor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER:esDevueltaPor timeEvents:participatesIn)</i>
<i>esExpedidoPor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER:esExpedidoPor timeEvents:participatesIn)</i>
<i>esModificadaPor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER:esModificadaPor timeEvents:participatesIn)</i>
<i>especificaTipoExistenciasDe</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER:especificaTipoExistenciasDe timeEvents:participatesIn)</i>
<i>seRegistraPor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER: seRegistraPor timeEvents:participatesIn)</i>
<i>esReservadoPor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER: esReservadoPor timeEvents:participatesIn)</i>
<i>seSeparaPor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER: seSeparaPor timeEvents:participatesIn)</i>
<i>seUbicaPor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER: seUbicaPor timeEvents:participatesIn)</i>
<i>esUsadaPor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER: esUsadaPor timeEvents:participatesIn)</i>
<i>esVerificadaPor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER: esVerificadaPor timeEvents:participatesIn)</i>
<i>existenciasEsVerificadaPor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER: existenciasEsVerificadaPor timeEvents:participatesIn)</i>
<i>existenciasSeBuscaPor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER: existenciasSeBuscaPor timeEvents:participatesIn)</i>
<i>existenciasSeObtienePor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER: existenciasSeObtienePor timeEvents:participatesIn)</i>
<i>expide</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER: expide timeEvents:participatesIn)</i>
<i>modifica</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER: modifica timeEvents:participatesIn)</i>
<i>obtieneExistenciasDe</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER: obtieneExistenciasDe timeEvents:participatesIn)</i>
<i>obtieneUbicacionDe</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER: obtieneUbicacionDe timeEvents:participatesIn)</i>
<i>registra</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER: registra timeEvents:participatesIn)</i>
<i>registraUsoDe</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER: registraUsoDe timeEvents:participatesIn)</i>
<i>reserva</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER: reserva timeEvents:participatesIn)</i>
<i>separa</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER: separa timeEvents:participatesIn)</i>
<i>solicitaUbicacionDe</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER: solicitaUbicacionDe timeEvents:participatesIn)</i>
<i>suUsoEsRegistradaPor</i> \sqsubseteq <i>participatesIn</i>	\Rightarrow <i>SubObjectPropertyOf(domain-PAIyER: suUsoEsRegistradaPor timeEvents:participatesIn)</i>

Tabla D.4 TBox de la KB del patrón de análisis PAIyER con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
$tipoExistenciasSeEspecificaPor \sqsubseteq participatesIn$	$\Rightarrow SubObjectPropertyOf(domain-PAIyER: tipoExistenciasSeEspecificaPor timeEvents: participatesIn)$
$ubica \sqsubseteq participatesIn$	$\Rightarrow SubObjectPropertyOf(domain-PAIyER: ubica timeEvents: participatesIn)$
$ubicacionSeObtienePor \sqsubseteq participatesIn$	$\Rightarrow SubObjectPropertyOf(domain-PAIyER: ubicacionSeObtienePor timeEvents: participatesIn)$
$ubicacionSeSolicitaPor \sqsubseteq participatesIn$	$\Rightarrow SubObjectPropertyOf(domain-PAIyER: ubicacionSeSolicitaPor timeEvents: participatesIn)$
$usa \sqsubseteq participatesIn$	$\Rightarrow SubObjectPropertyOf(domain-PAIyER: usa timeEvents: participatesIn)$
$verificaDisponibilidadDe \sqsubseteq participatesIn$	$\Rightarrow SubObjectPropertyOf(domain-PAIyER: verificaDisponibilidadDe timeEvents: participatesIn)$
$verificaExistenciaDe \sqsubseteq participatesIn$	$\Rightarrow SubObjectPropertyOf(domain-PAIyER: verificaExistenciaDe timeEvents: participatesIn)$
-- Restricciones y características de propiedades de objeto temporales --	
$\exists actualiza.T \sqsubseteq (Event \sqcup Empleado)$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER: actualiza ObjectUnionOf(domain-PAIyER: Empleado timeEvents: Event))$
$T \sqsubseteq \forall actualiza.(Event \sqcup Inventario \sqcup Entidad \sqcup Reservacion)$	$\Rightarrow ObjectPropertyRange(domain-PAIyER: actualiza ObjectUnionOf(domain-PAIyER: Inventario domain-PAIyER: Entidad domain-PAIyER: Reservacion timeEvents: Event))$
$T \sqsubseteq \leq_1 actualiza^-$	$\Rightarrow InverseFunctionalObjectProperty(domain-PAIyER: seActualizaPor)$
$\exists adquiere.T \sqsubseteq (Event \sqcup Cliente)$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER: adquiere ObjectUnionOf(domain-PAIyER: Cliente timeEvents: Event))$
$T \sqsubseteq \forall adquiere.(Event \sqcup Contrato)$	$\Rightarrow ObjectPropertyRange(domain-PAIyER: adquiere ObjectUnionOf(domain-PAIyER: Contrato timeEvents: Event))$
$T \sqsubseteq \leq_1 adquiere^-$	$\Rightarrow InverseFunctionalObjectProperty(domain-PAIyER: adquiere)$
$\exists buscaExistenciasDe.T \sqsubseteq (Event \sqcup Empleado)$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER: buscaExistenciasDe ObjectUnionOf(domain-PAIyER: Empleado timeEvents: Event))$
$T \sqsubseteq \forall buscaExistenciasDe.(Event \sqcup Entidad)$	$\Rightarrow ObjectPropertyRange(domain-PAIyER: buscaExistenciasDe ObjectUnionOf(domain-PAIyER: Entidad timeEvents: Event))$
$T \sqsubseteq \leq_1 buscaExistenciasDe^-$	$\Rightarrow InverseFunctionalObjectProperty(domain-PAIyER: buscaExistenciasDe)$
$\exists cancela.T \sqsubseteq (Event \sqcup Empleado)$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER: cancela ObjectUnionOf(domain-PAIyER: Empleado timeEvents: Event))$
$T \sqsubseteq \forall cancela.(Event \sqcup Reservacion)$	$\Rightarrow ObjectPropertyRange(domain-PAIyER: cancela ObjectUnionOf(domain-PAIyER: Reservacion timeEvents: Event))$
$T \sqsubseteq \leq_1 cancela^-$	$\Rightarrow InverseFunctionalObjectProperty(domain-PAIyER: cancela)$
$\exists crea.T \sqsubseteq (Event \sqcup Empleado)$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER: crea ObjectUnionOf(domain-PAIyER: Empleado timeEvents: Event))$
$T \sqsubseteq \forall crea.(Event \sqcup Reservacion)$	$\Rightarrow ObjectPropertyRange(domain-PAIyER: crea ObjectUnionOf(domain-PAIyER: Reservacion$

Tabla D.4 TBox de la KB del patrón de análisis PAIyER con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
$T \sqsubseteq \leq_1 \text{ crea}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER: crea})$
$\exists \text{ daDeBaja.T} \sqsubseteq (\text{Event} \sqcup \text{Cliente})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: daDeBaja}) \text{ ObjectUnionOf}(\text{domain-PAIyER: Cliente} \text{ timeEvents:Event})$
$T \sqsubseteq \forall \text{ daDeBaja.}(\text{Event} \sqcup \text{Reservacion})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: daDeBaja}) \text{ ObjectUnionOf}(\text{domain-PAIyER: Reservacion} \text{ timeEvents:Event})$
$T \sqsubseteq \leq_1 \text{ daDeBaja}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER: daDeBaja})$
$\exists \text{ devuelve.T} \sqsubseteq (\text{Event} \sqcup \text{Cliente})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: devuelve}) \text{ ObjectUnionOf}(\text{domain-PAIyER: Cliente} \text{ timeEvents:Event})$
$T \sqsubseteq \forall \text{ devuelve.}(\text{Event} \sqcup \text{Entidad})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: devuelve}) \text{ ObjectUnionOf}(\text{domain-PAIyER: Entidad} \text{ timeEvents:Event})$
$T \sqsubseteq \leq_1 \text{ devuelve}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER: devuelve})$
$\exists \text{ seActualizaPor.T} \sqsubseteq (\text{Event} \sqcup \text{Inventario} \sqcup \text{Entidad} \sqcup \text{Reservacion})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: seActualizaPor}) \text{ ObjectUnionOf}(\text{domain-PAIyER: Inventario} \text{ domain-PAIyER: Entidad} \text{ domain-PAIyER: Reservacion} \text{ timeEvents:Event})$
$T \sqsubseteq \forall \text{ seActualizaPor.}(\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: seActualizaPor}) \text{ ObjectUnionOf}(\text{domain-PAIyER: Empleado} \text{ timeEvents:Event})$
$T \sqsubseteq \leq_1 \text{ seActualizaPor}$	$\Rightarrow \text{FunctionalObjectProperty}(\text{domain-PAIyER: seActualizaPor})$
$\exists \text{ esAdquiridoPor.T} \sqsubseteq (\text{Event} \sqcup \text{Contrato})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: esAdquiridoPor}) \text{ ObjectUnionOf}(\text{domain-PAIyER: Contrato} \text{ timeEvents:Event})$
$T \sqsubseteq \forall \text{ esAdquiridoPor.}(\text{Event} \sqcup \text{Cliente})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: esAdquiridoPor}) \text{ ObjectUnionOf}(\text{domain-PAIyER: Cliente} \text{ timeEvents:Event})$
$T \sqsubseteq \leq_1 \text{ esAdquiridoPor}$	$\Rightarrow \text{FunctionalObjectProperty}(\text{domain-PAIyER: esAdquiridoPor})$
$\exists \text{ esCanceladaPor.T} \sqsubseteq (\text{Event} \sqcup \text{Reservacion})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: esCanceladaPor}) \text{ ObjectUnionOf}(\text{domain-PAIyER: Reservacion} \text{ timeEvents:Event})$
$T \sqsubseteq \forall \text{ esCanceladaPor.}(\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: esCanceladaPor}) \text{ ObjectUnionOf}(\text{domain-PAIyER: Empleado} \text{ timeEvents:Event})$
$T \sqsubseteq \leq_1 \text{ esCanceladaPor}$	$\Rightarrow \text{FunctionalObjectProperty}(\text{domain-PAIyER: esCanceladaPor})$
$\exists \text{ esCreadaPor.T} \sqsubseteq (\text{Event} \sqcup \text{Reservacion})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: esCreadaPor}) \text{ ObjectUnionOf}(\text{domain-PAIyER: Reservacion} \text{ timeEvents:Event})$
$T \sqsubseteq \forall \text{ esCreadaPor.}(\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: esCreadaPor}) \text{ ObjectUnionOf}(\text{domain-PAIyER: Empleado} \text{ timeEvents:Event})$
$T \sqsubseteq \leq_1 \text{ esCreadaPor}$	$\Rightarrow \text{FunctionalObjectProperty}(\text{domain-PAIyER: esCreadaPor})$
$\exists \text{ esDadaDeBajaPor.T} \sqsubseteq (\text{Event} \sqcup \text{Reservacion})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: esDadaDeBajaPor}) \text{ ObjectUnionOf}(\text{domain-PAIyER: Reservacion} \text{ timeEvents:Event})$

Tabla D.4 TBox de la KB del patrón de análisis PAIyER con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
$T \sqsubseteq \forall esDadaDeBajaPor.(Event \sqcup Cliente)$	$\Rightarrow PAIyER:Reservacion \ timeEvents:Event)) \Rightarrow ObjectPropertyRange(domain-PAIyER: esDadaDeBajaPor \ ObjectUnionOf(domain-PAIyER:Cliente \ timeEvents:Event))$
$T \sqsubseteq \leq I \ esDadaDeBajaPor$	$\Rightarrow FunctionalObjectProperty(domain-PAIyER: esDadaDeBajaPor)$
$\exists esDevueltaPor.T \sqsubseteq (Event \sqcup Entidad)$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER: esDevueltaPor \ ObjectUnionOf(domain-PAIyER:Entidad \ timeEvents:Event))$
$T \sqsubseteq \forall esDevueltaPor.(Event \sqcup Cliente)$	$\Rightarrow ObjectPropertyRange(domain-PAIyER: esDevueltaPor \ ObjectUnionOf(domain-PAIyER:Cliente \ timeEvents:Event))$
$T \sqsubseteq \leq I \ esDevueltaPor$	$\Rightarrow FunctionalObjectProperty(domain-PAIyER: esDevueltaPor)$
$\exists esExpedidoPor.T \sqsubseteq (Event \sqcup Contrato)$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER: esExpedidoPor \ ObjectUnionOf(domain-PAIyER:Contrato \ timeEvents:Event))$
$T \sqsubseteq \forall esExpedidoPor.(Event \sqcup Empleado)$	$\Rightarrow ObjectPropertyRange(domain-PAIyER: esExpedidoPor \ ObjectUnionOf(domain-PAIyER:Empleado \ timeEvents:Event))$
$T \sqsubseteq \leq I \ esExpedidoPor$	$\Rightarrow FunctionalObjectProperty(domain-PAIyER: esExpedidoPor)$
$\exists esModificadaPor.T \sqsubseteq (Event \sqcup Reservacion)$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER: esModificadaPor \ ObjectUnionOf(domain-PAIyER:Reservacion \ timeEvents:Event))$
$T \sqsubseteq \forall esModificadaPor.(Event \sqcup Cliente)$	$\Rightarrow ObjectPropertyRange(domain-PAIyER: esModificadaPor \ ObjectUnionOf(domain-PAIyER:Cliente \ timeEvents:Event))$
$T \sqsubseteq \leq I \ esModificadaPor$	$\Rightarrow FunctionalObjectProperty(domain-PAIyER: esModificadaPor)$
$\exists especificaTipoExistenciasDe.T \sqsubseteq (Event \sqcup Empleado)$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER: especificaTipoExistenciasDe \ ObjectUnionOf(domain-PAIyER:Empleado \ timeEvents:Event))$
$T \sqsubseteq \forall especificaTipoExistenciasDe.(Event \sqcup Entidad)$	$\Rightarrow ObjectPropertyRange(domain-PAIyER: especificaTipoExistenciasDe \ ObjectUnionOf(domain-PAIyER:Entidad \ timeEvents:Event))$
$T \sqsubseteq \leq I \ especificaTipoExistenciasDe^-$	$\Rightarrow InverseFunctionalObjectProperty(domain-PAIyER: especificaTipoExistenciasDe)$
$\exists seRegistraPor.T \sqsubseteq (Event \sqcup Entidad)$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER: seRegistraPor \ ObjectUnionOf(domain-PAIyER:Entidad \ timeEvents:Event))$
$T \sqsubseteq \forall seRegistraPor.(Event \sqcup Empleado)$	$\Rightarrow ObjectPropertyRange(domain-PAIyER: seRegistraPor \ ObjectUnionOf(domain-PAIyER:Empleado \ timeEvents:Event))$
$T \sqsubseteq \leq I \ seRegistraPor$	$\Rightarrow FunctionalObjectProperty(domain-PAIyER: seRegistraPor)$
$\exists esReservadaPor.T \sqsubseteq (Event \sqcup Entidad)$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER: esReservadaPor \ ObjectUnionOf(domain-PAIyER:Entidad \ timeEvents:Event))$
$T \sqsubseteq \forall esReservadaPor.(Event \sqcup Cliente)$	$\Rightarrow ObjectPropertyRange(domain-PAIyER: esReservadaPor \ ObjectUnionOf(domain-PAIyER:Cliente \ timeEvents:Event))$
$T \sqsubseteq \leq I \ esReservadaPor$	$\Rightarrow FunctionalObjectProperty(domain-PAIyER: esReservadaPor)$

Tabla D.4 TBox de la KB del patrón de análisis PAIyER con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
$\exists seSeparaPor.T \sqsubseteq (Event \sqcup Entidad)$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER: seSeparaPor ObjectUnionOf(domain-PAIyER:Entidad timeEvents:Event))$
$T \sqsubseteq \forall seSeparaPor.(Event \sqcup Empleado)$	$\Rightarrow ObjectPropertyRange(domain-PAIyER: seSeparaPor ObjectUnionOf(domain-PAIyER:Empleado timeEvents:Event))$
$T \sqsubseteq \leq I seSeparaPor$	$\Rightarrow FunctionalObjectProperty(domain-PAIyER: seSeparaPor)$
$\exists seUbicaPor.T \sqsubseteq (Event \sqcup Entidad)$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER: seUbicaPor ObjectUnionOf(domain-PAIyER:Entidad timeEvents:Event))$
$T \sqsubseteq \forall seUbicaPor.(Event \sqcup Empleado)$	$\Rightarrow ObjectPropertyRange(domain-PAIyER: seUbicaPor ObjectUnionOf(domain-PAIyER:Empleado timeEvents:Event))$
$T \sqsubseteq \leq I seUbicaPor$	$\Rightarrow FunctionalObjectProperty(domain-PAIyER: seUbicaPor)$
$\exists esUsadaPor.T \sqsubseteq (Event \sqcup Entidad)$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER: esUsadaPor ObjectUnionOf(domain-PAIyER:Entidad timeEvents:Event))$
$T \sqsubseteq \forall esUsadaPor.(Event \sqcup Cliente)$	$\Rightarrow ObjectPropertyRange(domain-PAIyER: esUsadaPor ObjectUnionOf(domain-PAIyER:Cliente timeEvents:Event))$
$T \sqsubseteq \leq I esUsadaPor$	$\Rightarrow FunctionalObjectProperty(domain-PAIyER: esUsadaPor)$
$\exists esVerificadaPor.T \sqsubseteq (Event \sqcup Entidad)$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER: esVerificadaPor ObjectUnionOf(domain-PAIyER:Entidad timeEvents:Event))$
$T \sqsubseteq \forall esVerificadaPor.(Event \sqcup Empleado)$	$\Rightarrow ObjectPropertyRange(domain-PAIyER: esVerificadaPor ObjectUnionOf(domain-PAIyER:Empleado timeEvents:Event))$
$T \sqsubseteq \leq I esVerificadaPor$	$\Rightarrow FunctionalObjectProperty(domain-PAIyER: esVerificadaPor)$
$\exists existenciaSeVerificaPor.T \sqsubseteq (Event \sqcup Entidad)$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER: existenciaSeVerificaPor ObjectUnionOf(domain-PAIyER:Entidad timeEvents:Event))$
$T \sqsubseteq \forall existenciaSeVerificaPor.(Event \sqcup Empleado)$	$\Rightarrow ObjectPropertyRange(domain-PAIyER: existenciaSeVerificaPor ObjectUnionOf(domain-PAIyER:Empleado timeEvents:Event))$
$T \sqsubseteq \leq I existenciaSeVerificaPor$	$\Rightarrow FunctionalObjectProperty(domain-PAIyER: existenciaSeVerificaPor)$
$\exists existenciasSeBuscaPor.T \sqsubseteq (Event \sqcup Entidad)$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER: existenciasSeBuscaPor ObjectUnionOf(domain-PAIyER:Entidad timeEvents:Event))$
$T \sqsubseteq \forall existenciasSeBuscaPor.(Event \sqcup Empleado)$	$\Rightarrow ObjectPropertyRange(domain-PAIyER: existenciasSeBuscaPor ObjectUnionOf(domain-PAIyER:Empleado timeEvents:Event))$
$T \sqsubseteq \leq I existenciasSeBuscaPor$	$\Rightarrow FunctionalObjectProperty(domain-PAIyER: existenciasSeBuscaPor)$
$\exists existenciasSeObtienePor.T \sqsubseteq (Event \sqcup Entidad)$	$\Rightarrow ObjectPropertyDomain(domain-PAIyER: existenciasSeObtienePor ObjectUnionOf(domain-PAIyER:Entidad timeEvents:Event))$
$T \sqsubseteq \forall existenciasSeObtienePor.(Event \sqcup Empleado)$	$\Rightarrow ObjectPropertyRange(domain-PAIyER: existenciasSeObtienePor ObjectUnionOf(domain-PAIyER:Empleado timeEvents:Event))$

Tabla D.4 TBox de la KB del patrón de análisis PAIyER con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
$T \sqsubseteq \leq l \text{ existenciasSeObtienePor}$	$\Rightarrow \text{FunctionalObjectProperty}(\text{domain-PAIyER: existenciasSeObtienePor})$
$\exists \text{ expide.} T \sqsubseteq (\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: expide} \text{ ObjectUnionOf}(\text{domain-PAIyER: Empleado} \text{ timeEvents: Event}))$
$T \sqsubseteq \forall \text{ expide.} (\text{Event} \sqcup \text{Contrato})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: expide} \text{ ObjectUnionOf}(\text{domain-PAIyER: Contrato} \text{ timeEvents: Event}))$
$T \sqsubseteq \leq l \text{ expide}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER: expide})$
$\exists \text{ modifica.} T \sqsubseteq (\text{Event} \sqcup \text{Cliente})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: modifica} \text{ ObjectUnionOf}(\text{domain-PAIyER: Cliente} \text{ timeEvents: Event}))$
$T \sqsubseteq \forall \text{ modifica.} (\text{Event} \sqcup \text{Reservacion})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: modifica} \text{ ObjectUnionOf}(\text{domain-PAIyER: Reservacion} \text{ timeEvents: Event}))$
$T \sqsubseteq \leq l \text{ modifica}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER: modifica})$
$\exists \text{ obtieneExistenciasDe.} T \sqsubseteq (\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: obtieneExistenciasDe} \text{ ObjectUnionOf}(\text{domain-PAIyER: Empleado} \text{ timeEvents: Event}))$
$T \sqsubseteq \forall \text{ obtieneExistenciasDe.} (\text{Event} \sqcup \text{Entidad})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: obtieneExistenciasDe} \text{ ObjectUnionOf}(\text{Domain-PAIyER: Entidad} \text{ timeEvents: Event}))$
$T \sqsubseteq \leq l \text{ obtieneExistenciasDe}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER: obtieneExistenciasDe})$
$\exists \text{ obtieneUbicacionDe.} T \sqsubseteq (\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: obtieneUbicacionDe} \text{ ObjectUnionOf}(\text{domain-PAIyER: Empleado} \text{ timeEvents: Event}))$
$T \sqsubseteq \forall \text{ obtieneUbicacionDe.} (\text{Event} \sqcup \text{Entidad})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: obtieneUbicacionDe} \text{ ObjectUnionOf}(\text{Domain-PAIyER: Entidad} \text{ timeEvents: Event}))$
$T \sqsubseteq \leq l \text{ obtieneUbicacionDe}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER: obtieneUbicacionDe})$
$\exists \text{ registra.} T \sqsubseteq (\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: registra} \text{ ObjectUnionOf}(\text{domain-PAIyER: Empleado} \text{ timeEvents: Event}))$
$T \sqsubseteq \forall \text{ registra.} (\text{Event} \sqcup \text{Entidad})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: registra} \text{ ObjectUnionOf}(\text{Domain-PAIyER: Entidad} \text{ timeEvents: Event}))$
$T \sqsubseteq \leq l \text{ registra}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER: registra})$
$\exists \text{ registraUsoDe.} T \sqsubseteq (\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: registraUsoDe} \text{ ObjectUnionOf}(\text{domain-PAIyER: Empleado} \text{ timeEvents: Event}))$
$T \sqsubseteq \forall \text{ registraUsoDe.} (\text{Event} \sqcup \text{Entidad})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: registraUsoDe} \text{ ObjectUnionOf}(\text{Domain-PAIyER: Entidad} \text{ timeEvents: Event}))$
$T \sqsubseteq \leq l \text{ registraUsoDe}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER: registraUsoDe})$
$\exists \text{ reserva.} T \sqsubseteq (\text{Event} \sqcup \text{Cliente})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: reserva} \text{ ObjectUnionOf}(\text{domain-PAIyER: Cliente} \text{ timeEvents: Event}))$
$T \sqsubseteq \forall \text{ reserva.} (\text{Event} \sqcup \text{Entidad})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: reserva})$

Tabla D.4 TBox de la KB del patrón de análisis PAIyER con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
$T \sqsubseteq \leq I \text{ reserva}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER: reserva})$
$\exists \text{separa.} T \sqsubseteq (\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: separa} \text{ ObjectUnionOf}(\text{domain-PAIyER: Empleado} \text{ timeEvents: Event}))$
$T \sqsubseteq \forall \text{separa.} (\text{Event} \sqcup \text{Entidad})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: separa} \text{ ObjectUnionOf}(\text{Domain-PAIyER: Entidad} \text{ timeEvents: Event}))$
$T \sqsubseteq \leq I \text{ separa}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER: separa})$
$\exists \text{solicitaUbicacionDe.} T \sqsubseteq (\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: solicitaUbicacionDe} \text{ ObjectUnionOf}(\text{domain-PAIyER: Empleado} \text{ timeEvents: Event}))$
$T \sqsubseteq \forall \text{solicitaUbicacionDe.} (\text{Event} \sqcup \text{Entidad})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: solicitaUbicacionDe} \text{ ObjectUnionOf}(\text{Domain-PAIyER: Entidad} \text{ timeEvents: Event}))$
$T \sqsubseteq \leq I \text{ solicitaUbicacionDe}$	$\Rightarrow \text{FunctionalObjectProperty}(\text{domain-PAIyER: solicitaUbicacionDe})$
$\exists \text{suUsoEsRegistradaPor.} T \sqsubseteq (\text{Event} \sqcup \text{Entidad})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: suUsoEsRegistradaPor} \text{ ObjectUnionOf}(\text{domain-PAIyER: Entidad} \text{ timeEvents: Event}))$
$T \sqsubseteq \forall \text{suUsoEsRegistradaPor.} (\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: suUsoEsRegistradaPor} \text{ ObjectUnionOf}(\text{Domain-PAIyER: Empleado} \text{ timeEvents: Event}))$
$T \sqsubseteq \leq I \text{ suUsoEsRegistradaPor}$	$\Rightarrow \text{FunctionalObjectProperty}(\text{domain-PAIyER: suUsoEsRegistradaPor})$
$\exists \text{tipoExistenciasSeEspecificaPor.} T \sqsubseteq (\text{Event} \sqcup \text{Entidad})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: tipoExistenciasSeEspecificaPor} \text{ ObjectUnionOf}(\text{domain-PAIyER: Entidad} \text{ timeEvents: Event}))$
$T \sqsubseteq \forall \text{tipoExistenciasSeEspecificaPor.} (\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: tipoExistenciasSeEspecificaPor} \text{ ObjectUnionOf}(\text{Domain-PAIyER: Empleado} \text{ timeEvents: Event}))$
$T \sqsubseteq \leq I \text{ tipoExistenciasSeEspecificaPor}$	$\Rightarrow \text{FunctionalObjectProperty}(\text{domain-PAIyER: tipoExistenciasSeEspecificaPor})$
$\exists \text{ubica.} T \sqsubseteq (\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: ubica} \text{ ObjectUnionOf}(\text{domain-PAIyER: Empleado} \text{ timeEvents: Event}))$
$T \sqsubseteq \forall \text{ubica.} (\text{Event} \sqcup \text{Entidad})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: ubica} \text{ ObjectUnionOf}(\text{Domain-PAIyER: Entidad} \text{ timeEvents: Event}))$
$T \sqsubseteq \leq I \text{ ubica}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER: ubica})$
$\exists \text{ubicacionSeObtienePor.} T \sqsubseteq (\text{Event} \sqcup \text{Entidad})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: ubicacionSeObtienePor} \text{ ObjectUnionOf}(\text{domain-PAIyER: Entidad} \text{ timeEvents: Event}))$
$T \sqsubseteq \forall \text{ubicacionSeObtienePor.} (\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: ubicacionSeObtienePor} \text{ ObjectUnionOf}(\text{Domain-PAIyER: Empleado} \text{ timeEvents: Event}))$
$T \sqsubseteq \leq I \text{ ubicacionSeObtienePor}$	$\Rightarrow \text{FunctionalObjectProperty}(\text{domain-PAIyER: ubicacionSeObtienePor})$

Tabla D.4 TBox de la KB del patrón de análisis PAIyER con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
$\exists \text{ubicacionSeSolicitaPor}.T \sqsubseteq (\text{Event} \sqcup \text{Entidad})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: ubicacionSeSolicitaPor} \text{ ObjectUnionOf}(\text{domain-PAIyER:Entidad} \text{ timeEvents:Event}))$
$T \sqsubseteq \forall \text{ubicacionSeSolicitaPor}.(\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: ubicacionSeSolicitaPor} \text{ ObjectUnionOf}(\text{Domain-PAIyER:Empleado} \text{ timeEvents:Event}))$
$T \sqsubseteq \leq 1 \text{ ubicacionSeSolicitaPor}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER: ubicacionSeSolicitaPor})$
$\exists \text{usa}.T \sqsubseteq (\text{Event} \sqcup \text{Cliente})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: usa} \text{ ObjectUnionOf}(\text{domain-PAIyER:Cliente} \text{ timeEvents:Event}))$
$T \sqsubseteq \forall \text{usa}.(\text{Event} \sqcup \text{Entidad})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: usa} \text{ ObjectUnionOf}(\text{Domain-PAIyER:Entidad} \text{ timeEvents:Event}))$
$T \sqsubseteq \leq 1 \text{ usa}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER: usa})$
$\exists \text{verificaDisponibilidadDe}.T \sqsubseteq (\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: verificaDisponibilidadDe} \text{ ObjectUnionOf}(\text{domain-PAIyER:Empleado} \text{ timeEvents:Event}))$
$T \sqsubseteq \forall \text{verificaDisponibilidadDe}.(\text{Event} \sqcup \text{Entidad})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: verificaDisponibilidadDe} \text{ ObjectUnionOf}(\text{Domain-PAIyER:Entidad} \text{ timeEvents:Event}))$
$T \sqsubseteq \leq 1 \text{ verificaDisponibilidadDe}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER: verificaDisponibilidadDe})$
$\exists \text{verificaExistenciaDe}.T \sqsubseteq (\text{Event} \sqcup \text{Empleado})$	$\Rightarrow \text{ObjectPropertyDomain}(\text{domain-PAIyER: verificaExistenciaDe} \text{ ObjectUnionOf}(\text{domain-PAIyER:Empleado} \text{ timeEvents:Event}))$
$T \sqsubseteq \forall \text{verificaExistenciaDe}.(\text{Event} \sqcup \text{Entidad})$	$\Rightarrow \text{ObjectPropertyRange}(\text{domain-PAIyER: verificaExistenciaDe} \text{ ObjectUnionOf}(\text{Domain-PAIyER:Entidad} \text{ timeEvents:Event}))$
$T \sqsubseteq \leq 1 \text{ verificaExistenciaDe}^-$	$\Rightarrow \text{InverseFunctionalObjectProperty}(\text{domain-PAIyER: verificaExistenciaDe})$
$\text{actualiza} \equiv \text{seActualizaPor}^-$	$\Rightarrow \text{InverseObjectProperties}(\text{domain-PAIyER:actualiza} \text{ domain-PAIyER:seActualizaPor})$
$\text{obtieneExistenciasDe} \equiv \text{existenciasSeObtienePor}^-$	$\Rightarrow \text{InverseObjectProperties}(\text{domain-PAIyER: obtieneExistenciasDe} \text{ domain-PAIyER: existenciasSeObtienePor})$
$\text{buscaExistenciasDe} \equiv \text{existenciasSeBuscaPor}^-$	$\Rightarrow \text{InverseObjectProperties}(\text{domain-PAIyER: buscaExistenciasDe} \text{ domain-PAIyER: existenciasSeBuscaPor})$
$\text{especificaTipoExistenciasDe} \equiv \text{tipoExistenciasSeEspecificaPor}^-$	$\Rightarrow \text{InverseObjectProperties}(\text{domain-PAIyER: especificaTipoExistenciasDe} \text{ domain-PAIyER: tipoExistenciasSeEspecificaPor})$
$\text{registra} \equiv \text{seRegistraPor}^-$	$\Rightarrow \text{InverseObjectProperties}(\text{domain-PAIyER: registra} \text{ domain-PAIyER: seRegistraPor})$
$\text{separa} \equiv \text{seSeparaPor}^-$	$\Rightarrow \text{InverseObjectProperties}(\text{domain-PAIyER: separa} \text{ domain-PAIyER: seSeparaPor})$
$\text{ubica} \equiv \text{seUbicaPor}^-$	$\Rightarrow \text{InverseObjectProperties}(\text{domain-PAIyER: ubica} \text{ domain-PAIyER: seUbicaPor})$
$\text{verificaExistenciaDe} \equiv \text{existenciaSeVerificaPor}^-$	$\Rightarrow \text{InverseObjectProperties}(\text{domain-PAIyER: verificaExistenciaDe} \text{ domain-PAIyER: existenciaSeVerificaPor})$
$\text{obtieneUbicacionDe} \equiv \text{ubicacionSeObtienePor}^-$	$\Rightarrow \text{InverseObjectProperties}(\text{domain-PAIyER: obtieneUbicacionDe} \text{ domain-PAIyER: ubicacionSeObtienePor})$

Tabla D.4 TBox de la KB del patrón de análisis PAIyER con aspectos temporales (cont.)

Sintaxis DL	Sintaxis Funcional
$solicitaUbicacionDe \equiv ubicacionSeSolicitaPor^-$	$ubicacionSeObtienePor)$ $\Rightarrow InverseObjectProperties(domain-PAIyER: solicitaUbicacionDe domain-PAIyER: ubicacionSeSolicitaPor)$
$adquiere \equiv esAdquiridoPor^-$	$\Rightarrow InverseObjectProperties(domain-PAIyER: adquiere domain-PAIyER: esAdquiridoPor)$
$cancela \equiv esCanceladaPor^-$	$\Rightarrow InverseObjectProperties(domain-PAIyER: cancela domain-PAIyER: esCanceladaPor)$
$crea \equiv esCreadaPor^-$	$\Rightarrow InverseObjectProperties(domain-PAIyER: crea domain-PAIyER: esCreadaPor)$
$daDeBaja \equiv esDadaDeBaja^-$	$\Rightarrow InverseObjectProperties(domain-PAIyER: daDeBaja domain-PAIyER: esDadaDeBaja)$
$devuelve \equiv esDevueltaPor^-$	$\Rightarrow InverseObjectProperties(domain-PAIyER: devuelve domain-PAIyER: esDevueltaPor)$
$expide \equiv esExpedidaPor^-$	$\Rightarrow InverseObjectProperties(domain-PAIyER: expide domain-PAIyER: esExpedidaPor)$
$modifica \equiv esModificadaPor^-$	$\Rightarrow InverseObjectProperties(domain-PAIyER: modifica domain-PAIyER: esModificadaPor)$
$reserva \equiv esReservadaPor^-$	$\Rightarrow InverseObjectProperties(domain-PAIyER: reserva domain-PAIyER: esReservadaPor)$
$usa \equiv esUsadaPor^-$	$\Rightarrow InverseObjectProperties(domain-PAIyER: usa domain-PAIyER: esUsadaPor)$
$verificaDisponibilidadDe \equiv esVerificadaPor^-$	$\Rightarrow InverseObjectProperties(domain-PAIyER: verificaDisponibilidad domain-PAIyER: esVerificadaPor)$
$registraUsoDe \equiv suUsoSeRegistraPor^-$	$\Rightarrow InverseObjectProperties(domain-PAIyER: registraUsoDe domain-PAIyER: suUsoSeRegistraPor)$
}	

Para una mejor comprensión de las descripciones lógicas presentadas en la TBox de la Tabla D.4, a continuación se describen algunos ejemplos tomados de las secciones *Clases*, *Clases Disjuntas*, *Restricciones existenciales y universales* y *Restricciones y características de propiedades de objeto temporales* de la misma tabla:

Clases

T : Conjunto de todos los individuos en el dominio.

$staticClasses \sqsubseteq T$: La clase *staticClasses* es un tipo de individuos en el dominio del patrón de análisis PAIyER.

$dynamicClasses \sqsubseteq T$: La clase *dynamicClasses* es un tipo de individuos en el dominio temporal de la ontología OWL-Time.

$Persona \sqsubseteq staticClasses$: La clase *Persona* es un tipo de individuos de la clase *staticClasses*.

$Empleado \sqsubseteq Persona$: La clase *Empleado* es un tipo de la clase *Persona*.

Clases Disjuntas

$Contrato \sqcap Entidad \sqcap Empleado \sqcap Inventario \sqcap Institucion \sqcap Cliente \sqcap Propietario \sqcap Reservacion \sqsubseteq \perp$: Establece que los individuos que son miembros de una de las

clases (Contrato, Entidad, Empleado, Inventario, Institucion, Cliente, Propietario y Reservacion), no son instancias de otra clase.

Restricciones existenciales y universales

$Entidad \sqsubseteq \forall seRegistraPor.(Event \ \forall seRegistraPor.Empleado)$: Establece que una Entidad es todo aquello que es registrado por un Empleado a través de un Evento.

$Empleado \sqsubseteq \forall actualiza.(Event \ \forall actualiza.Inventario)$: Establece que un Empleado es quien al menos actualiza un Inventario a través de un Evento.

Restricciones y características de propiedades de objeto temporales

$\exists actualiza.T \sqsubseteq Empleado$: Establece a la clase Empleado y a la clase Event en el dominio de la propiedad de objeto actualiza.

$T \sqsubseteq \forall actualiza.Inventario$: Establece a la clase Inventario y a la clase Event en el rango de la propiedad de objeto actualiza.

$T \sqsubseteq \leq 1 actualiza^-$: Establece que la propiedad de objeto temporal *actualiza* es una propiedad de objeto inversa funcional.

En la Figura D.1 se presenta la jerarquía de clases estáticas y temporales que conforman la KB del patrón de análisis PAIyER.

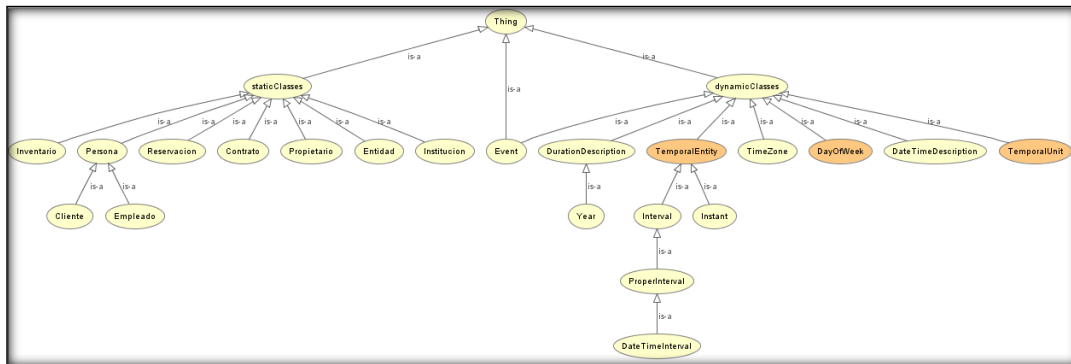


Figura D.1 Jerarquía de clases estáticas y temporales en la KB del patrón de análisis PAIyER.

1.2. Componente extensional

Para obtener el componente *ABox de casos de éxito* y el componente *ABox de casos de éxito y fracaso*, se utilizaron individuos (nombres de individuos abstractos N_{IA} y nombres de individuos concretos N_{IC} en las DLs) como:

{Valentino, Heidi, Juan_Carlos, Santiago, Violeta, Libro1, Libro2, Libro3, Libro4, Inventario_CENIDET, CENIDET, Biblioteca_CENIDET, Contrato_Heidi, Contrato_JuanCarlos, Contrato_Santiago, Contrato_Violeta, Reservacion_prestamo_Heidi, Reservacion_prestamo_JuanCarlos, Reservacion_prestamo_Santiago, Reservacion_prestamo_Violeta}

y los valores de datos (dominios concretos en las DLs y valores de datos o datatypes en OWL-DL) como:

{ string, Boolean, integer, double, dateTime, long }

Generalmente, una *ABox* se integra de secuencias de interpretaciones (*ABox1*, *ABox2*... *ABoxn*). En las secciones subsecuentes se describe un fragmento de las aserciones más relevantes sobre secuencias de interpretaciones de casos de éxito y casos de éxito y fracaso que representa comportamientos basados en los requerimientos funcionales del patrón de análisis PAIyER. Se ha omitido el detalle acerca de las aserciones de propiedades de datos que hacen uso de los individuos concretos como string, Boolean, integer, double, dateTime y long en la *ABox*. Estas y otras características pueden consultarse en la versión completa de la KB con aspectos temporales (OTPAE) del patrón de análisis PAIyER. Al igual que en la ontología OPAE descrita en el *Anexo C. Ontología OPAE del patrón de análisis PAIyER*, algunas de estas aserciones contemplan propiedades de datos basadas en el modelo SSM como son *Comportamiento*, *Especificidad*, *Nivel de estabilidad* y *Percepción*), entre otras.

1.2.1 RF1. Hacer una reservación

Las secuencias de interpretaciones descritas en esta sección satisfacen/cumplen con el comportamiento dinámico del RF1. Hacer una reservación representado formalmente con la fórmula de la Tabla D.5.

Tabla D.5 Fórmula ALC_{LTL} del RF1. Hacer una reservación.

$$\mathcal{G}(\mathcal{X}(\text{Cliente} \sqsubseteq \exists \text{reserva. Entidad}) \rightarrow (\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \exists \text{verificaDisponibilidadDe. Entidad}) \wedge \mathcal{F}(\neg \mathcal{X}\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \forall \text{crea. Reservacion} \wedge \text{Empleado} \sqsubseteq \exists \text{actualiza. Entidad}) \wedge \neg \mathcal{X}\mathcal{X}\mathcal{X}\mathcal{X}(\text{Empleado} \sqsubseteq \forall \text{expide. Contrato})))$$

1.2.1.1. Casos de éxito

En la Tabla D.6 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la *ABox*) basadas en casos de éxito, que conforman una secuencia que representa el comportamiento dinámico para hacer una reservación. Una representación gráfica simplificada de la secuencia se muestra en la Figura D.2. A manera de ejemplo, en la Figura D.3 se muestra el detalle de las aserciones A_1 , A_2 , A_3 y A_4 en el caso de éxito número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de razonamiento

automatizado, los demás casos se definen de manera similar. Sin embargo, difieren en los individuos involucrados.

Tabla D.6 ABox de casos de éxito en el RF1. Hacer una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
Caso de éxito 1 (A₁, A₂, A₃, A₄):	
A ₁ : [X(Cliente \sqsubseteq \exists reserva.Entidad)]: El cliente Heidi reserva la Entidad Libro1.	
A ₁ = {	
-- Aserciones de clase --	
Cliente(Heidi)	⇒ ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Heidi)
Entidad(Libro1)	⇒ ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro1)
Event(Req1-Ev1)	⇒ ClassAssertion(timeEvents:Event domain-PAIyER:Req1-Ev1)
DateTimeInterval(Req1-DT11)	⇒ ClassAssertion(time:DateTimeInterval domain-PAIyER:Req1-DT11)
Instant(Req1-DT11-Sins)	⇒ ClassAssertion(time:Instant domain-PAIyER:Req1-DT11-Sins)
Instant(Req1-DT11-Eins)	⇒ ClassAssertion(time:Instant domain-PAIyER:Req1-DT11-Eins)
-- Aserciones de propiedades de objeto --	
reserva(Heidi,Req1-Ev1)	⇒ ObjectPropertyAssertion(domain-PAIyER:reserva domain-PAIyER:Heidi domain-PAIyER:Req1-Ev1)
reserva(Req1-Ev1,Libro1)	⇒ ObjectPropertyAssertion(domain-PAIyER:reserva domain-PAIyER:Req1-Ev1 domain-PAIyER:Libro1)
during(Req1-Ev1,Req1-DT11)	⇒ ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req1-Ev1 domain-PAIyER:Req1-DT11)
hasBeginning(Req1-DT11,Req1-DT11-Sins)	⇒ ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req1-DT11 domain-PAIyER:Req1-DT11-Sins)
hasEnd(Req1-DT11,Req1-DT11-Eins)	⇒ ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req1-DT11 domain-PAIyER:Req1-DT11-Eins)
}	⇒
A ₂ : [XX (Empleado \sqsubseteq \exists verificaDisponibilidadDe.Entidad)]: El empleado Valentino verifica la disponibilidad de la Entidad Libro1	
A ₂ = {	
-- Aserciones de clase --	
Empleado(Valentino)	⇒ ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)
Entidad(Libro1)	⇒ ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro1)
Event(Req1-Ev2)	⇒ ClassAssertion(timeEvents:Event domain-PAIyER:Req1-Ev2)
DateTimeInterval(Req1-DT12)	⇒ ClassAssertion(time:DateTimeInterval domain-PAIyER:Req1-DT12)
Instant(Req1-DT12-Sins)	⇒ ClassAssertion(time:Instant domain-PAIyER:Req1-DT12-Sins)
Instant(Req1-DT12-Eins)	⇒ ClassAssertion(time:Instant domain-

Tabla D.6 ABox de casos de éxito en el RF1. Hacer una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
	PAIyER:Req1-DTI2-Eins)
-- Aserciones de propiedades de objeto --	
<i>verificaDisponibilidadDe</i> (Valentino,Req1-Ev2)	⇒ <i>ObjectPropertyAssertion</i> (domain-PAIyER: <i>verificaDisponibilidadDe</i> domain- PAIyER:Valentino domain-PAIyER:Req1-Ev2)
<i>verificaDisponibilidadDe</i> (Req1-Ev2,Libro1)	⇒ <i>ObjectPropertyAssertion</i> (domain-PAIyER: <i>verificaDisponibilidadDe</i> domain-PAIyER:Req1- Ev2 domain-PAIyER:Libro1)
<i>during</i> (Req1-Ev2,Req1-DTI2)	⇒ <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req1-Ev2 domain- PAIyER:Req1-DTI2)
<i>hasBeginning</i> (Req1-DTI2,Req1-DTI2-Sins)	⇒ <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req1-DTI2 domain- PAIyER:Req1-DTI2-Sins)
<i>hasEnd</i> (Req1-DTI2,Req1-DTI2-Eins)	⇒ <i>ObjectPropertyAssertion</i> (time:hasEnd domain- PAIyER:Req1-DTI2 domain-PAIyER:Req1- DTI2-Eins)
}	
A₃: [XXX(Empleado \sqsubseteq \forall crea.Reservacion \wedge Empleado \sqsubseteq \exists actualiza.Entidad)]: El empleado Valentino crea la Reservacion Reservacion_prestamo_Heidi y actualiza la Entidad Libro1.	
A ₃ = {	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	⇒ <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Entidad</i> (Libro1)	⇒ <i>ClassAssertion</i> (domain-PAIyER:Entidad domain- PAIyER:Libro1)
<i>Reservacion</i> (Reservacion_prestamo_Heidi)	⇒ <i>ClassAssertion</i> (domain-PAIyER:Resrvacion domain-PAIyER:Reservacion_prestamo_Heidi)
<i>Event</i> (Req1-Ev3_1)	⇒ <i>ClassAssertion</i> (timeEvents:Event domain- PAIyER:Req1-Ev3_1)
<i>DateTimeInterval</i> (Req1-DTI3_1)	⇒ <i>ClassAssertion</i> (time:DateTimeInterval domain- PAIyER:Req1-DTI3_1)
<i>Instant</i> (Req1-DTI3_1-Sins)	⇒ <i>ClassAssertion</i> (time:Instant domain- PAIyER:Req1-DTI3_1-Sins)
<i>Instant</i> (Req1-DTI3_1-Eins)	⇒ <i>ClassAssertion</i> (time:Instant domain- PAIyER:Req1-DTI3_1-Eins)
<i>Event</i> (Req1-Ev3_2)	⇒ <i>ClassAssertion</i> (timeEvents:Event domain- PAIyER:Req1-Ev3_2)
<i>DateTimeInterval</i> (Req1-DTI3_2)	⇒ <i>ClassAssertion</i> (time:DateTimeInterval domain- PAIyER:Req1-DTI3_2)
<i>Instant</i> (Req1-DTI3_2-Sins)	⇒ <i>ClassAssertion</i> (time:Instant domain- PAIyER:Req1-DTI3_2-Sins)
<i>Instant</i> (Req1-DTI3_2-Eins)	⇒ <i>ClassAssertion</i> (time:Instant domain- PAIyER:Req1-DTI3_2-Eins)
-- Aserciones de propiedades de objeto --	
<i>crea</i> (Valentino,Req1-Ev3_1)	⇒ <i>ObjectPropertyAssertion</i> (domain-PAIyER:crea domain-PAIyER:Valentino domain- PAIyER:Req1-Ev3_1)
<i>crea</i> (Req1-Ev3_1,Reservacion_prestamo_Heidi)	⇒ <i>ObjectPropertyAssertion</i> (domain-PAIyER:crea domain-PAIyER:Req1-Ev3_1 domain- PAIyER:Reservacion_prestamo_Heidi)
<i>during</i> (Req1-Ev3_1,Req1-DTI3_1)	⇒ <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req1-Ev3_1 domain- PAIyER:Req1-DTI3_1)

Tabla D.6 ABox de casos de éxito en el RF1. Hacer una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>hasBeginning(Req1-DTI3_1,Req1-DTI3_1-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req1-DTI3_1 domain-PAIyER:Req1-DTI3_1-Sins)</i>
<i>hasEnd(Req1-DTI3_1,Req1-DTI3_1-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req1-DTI3_1 domain-PAIyER:Req1-DTI3_1-Eins)</i>
<i>actualiza(Valentino,Req1-Ev3_2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Valentino domain-PAIyER:Req1-Ev3_2)</i>
<i>actualiza(Req1-Ev3_2,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Req1-Ev3_2 domain-PAIyER:Libro1)</i>
<i>during(Req1-Ev3_2,Req1-DTI3_2)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req1-Ev3_2 domain-PAIyER:Req1-DTI3_2)</i>
<i>hasBeginning(Req1-DTI3_2,Req1-DTI3_2-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req1-DTI3_2 domain-PAIyER:Req1-DTI3_2-Sins)</i>
<i>hasEnd(Req1-DTI3_2,Req1-DTI3_2-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req1-DTI3_2 domain-PAIyER:Req1-DTI3_2-Eins)</i>
}	
A₄: [XXXX (Empleado \sqsubseteq \existsexpide. Contrato)]: El empleado Valentino expide el contrato Contrato_Heidi.	
A ₄ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Contrato(Contrato_Heidi)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Contrato domain-PAIyER:Contrato_Heidi)</i>
<i>Event(Req1-Ev4)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req1-Ev4)</i>
<i>DateTimeInterval(Req1-DTI4)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req1-DTI4)</i>
<i>Instant(Req1-DTI4-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req1-DTI4-Sins)</i>
<i>Instant(Req1-DTI4-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req1-DTI4-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>expide(Valentino,Req1-Ev4)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:expide domain-PAIyER:Valentino domain-PAIyER:Req1-Ev4)</i>
<i>expide(Req1-Ev4, Contrato_Heidi)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:expide domain-PAIyER:Req1-Ev4 domain-PAIyER:Contrato_Heidi)</i>
<i>during(Req1-Ev4,Req1-DTI4)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req1-Ev4 domain-PAIyER:Req1-DTI4)</i>
<i>hasBeginning(Req1-DTI4,Req1-DTI4-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req1-DTI4 domain-PAIyER:Req1-DTI4-Sins)</i>
<i>hasEnd(Req1-DTI4,Req1-DTI4-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req1-DTI4 domain-PAIyER:Req1-DTI4-Eins)</i>
}	

Tabla D.6 ABox de casos de éxito en el RF1. Hacer una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
Caso de éxito 2 (A₅, A₆, A₇, A₈):	
A ₅ : [X(Cliente \sqsubseteq \exists reserva.Entidad)]: El cliente Juan_Carlos reserva la Entidad Libro2.	
A ₅ = {	
-- Aserciones de clase --	
Cliente(Juan_Carlos)	⇒ ClassAssertion(domain-PAIyER:Cliente domain-PAIyER: Juan_Carlos)
Entidad(Libro2)	⇒ ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)
Event(Req1-Ev5)	⇒ ClassAssertion(timeEvents:Event domain-PAIyER:Req1-Ev5)
DateTimeInterval(Req1-DTI5)	⇒ ClassAssertion(time:DateTimeInterval domain-PAIyER:Req1-DTI5)
Instant(Req1-DTI5-Sins)	⇒ ClassAssertion(time:Instant domain-PAIyER:Req1-DTI5-Sins)
Instant(Req1-DTI5-Eins)	⇒ ClassAssertion(time:Instant domain-PAIyER:Req1-DTI5-Eins)
-- Aserciones de propiedades de objeto --	
reserva(Juan_Carlos,Req1-Ev5)	⇒ ObjectPropertyAssertion(domain-PAIyER:reserva domain-PAIyER: Juan_Carlos domain-PAIyER:Req1-Ev5)
reserva(Req1-Ev5,Libro2)	⇒ ObjectPropertyAssertion(domain-PAIyER:reserva domain-PAIyER:Req1-Ev5 domain-PAIyER:Libro2)
during(Req1-Ev5,Req1-DTI5)	⇒ ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req1-Ev5 domain-PAIyER:Req1-DTI5)
hasBeginning(Req1-DTI5,Req1-DTI5-Sins)	⇒ ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req1-DTI5 domain-PAIyER:Req1-DTI5-Sins)
hasEnd(Req1-DTI5,Req1-DTI5-Eins)	⇒ ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req1-DTI5 domain-PAIyER:Req1-DTI5-Eins)
}	⇒
A ₆ : [XX (Empleado \sqsubseteq \exists verificaDisponibilidadDe.Entidad)]: El empleado Valentino verifica la disponibilidad de la Entidad Libro2	
A ₆ = {	
-- Aserciones de clase --	
Empleado(Valentino)	⇒ ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)
Entidad(Libro2)	⇒ ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)
Event(Req1-Ev6)	⇒ ClassAssertion(timeEvents:Event domain-PAIyER:Req1-Ev6)
DateTimeInterval(Req1-DTI6)	⇒ ClassAssertion(time:DateTimeInterval domain-PAIyER:Req1-DTI6)
Instant(Req1-DTI6-Sins)	⇒ ClassAssertion(time:Instant domain-PAIyER:Req1-DTI6-Sins)
Instant(Req1-DTI6-Eins)	⇒ ClassAssertion(time:Instant domain-PAIyER:Req1-DTI6-Eins)
-- Aserciones de propiedades de objeto --	
verificaDisponibilidadDe(Valentino,Req1-Ev6)	⇒ ObjectPropertyAssertion(domain-PAIyER:verificaDisponibilidadDe domain-PAIyER:Valentino domain-PAIyER:Req1-Ev6)
verificaDisponibilidadDe (Req1-Ev6,Libro2)	⇒ ObjectPropertyAssertion(domain-PAIyER:

Tabla D.6 ABox de casos de éxito en el RF1. Hacer una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>during</i> (Req1-Ev6,Req1-DTI6)	\Rightarrow <i>verificaDisponibilidadDe</i> domain-PAIyER:Req1-Ev6 domain-PAIyER:Libro2)
<i>hasBeginning</i> (Req1-DTI6,Req1-DTI6-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req1-Ev6 domain-PAIyER:Req1-DTI6)
<i>hasEnd</i> (Req1-DTI6,Req1-DTI6-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req1-DTI6 domain-PAIyER:Req1-DTI6-Sins)
	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req1-DTI6 domain-PAIyER:Req1-DTI6-Eins)
}	
A7: [XXX(Empleado \sqsubseteq \forall crea.Reservacion \wedge Empleado \sqsubseteq \exists actualiza.Entidad)]: El empleado Valentino crea la reservación Reservacion_prestamo_JuanCarlos y actualiza la Entidad Libro2.	
A7 = {	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Entidad</i> (Libro2)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Entidad domain-PAIyER:Libro2)
<i>Reservacion</i> (Reservacion_prestamo_JuanCarlos)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Resrvacion domain-PAIyER:Reservacion_prestamo_JuanCarlos)
<i>Event</i> (Req1-Ev7_1)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req1-Ev7_1)
<i>DateTimeInterval</i> (Req1-DTI7_1)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req1-DTI7_1)
<i>Instant</i> (Req1-DTI7_1-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req1-DTI7_1-Sins)
<i>Instant</i> (Req1-DTI7_1-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req1-DTI7_1-Eins)
<i>Event</i> (Req1-Ev7_2)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req1-Ev7_2)
<i>DateTimeInterval</i> (Req1-DTI7_2)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req1-DTI7_2)
<i>Instant</i> (Req1-DTI7_2-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req1-DTI7_2-Sins)
<i>Instant</i> (Req1-DTI7_2-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req1-DTI7_2-Eins)
-- Aserciones de propiedades de objeto --	
<i>crea</i> (Valentino,Req1-Ev7_1)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:crea domain-PAIyER:Valentino domain-PAIyER:Req1-Ev7_1)
<i>crea</i> (Req1-Ev7_1,Reservacion_prestamo_JuanCarlos)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:crea domain-PAIyER:Req1-Ev7_1 domain-PAIyER:Reservacion_prestamo_JuanCarlos)
<i>during</i> (Req1-Ev7_1,Req1-DTI7_1)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req1-Ev7_1 domain-PAIyER:Req1-DTI7_1)
<i>hasBeginning</i> (Req1-DTI7_1,Req1-DTI7_1-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req1-DTI7_1 domain-PAIyER:Req1-DTI7_1-Sins)
<i>hasEnd</i> (Req1-DTI7_1,Req1-DTI7_1-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req1-DTI7_1 domain-PAIyER:Req1-

Tabla D.6 ABox de casos de éxito en el RF1. Hacer una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>actualiza(Valentino,Req1-Ev7_2)</i>	<i>DTI7_1-Eins)</i> ⇒ <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Valentino domain-PAIyER:Req1-Ev7_2)</i>
<i>actualiza(Req1-Ev7_2,Libro2)</i>	⇒ <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Req1-Ev7_2 domain-PAIyER:Libro2)</i>
<i>during(Req1-Ev7_2,Req1-DTI7_2)</i>	⇒ <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req1-Ev7_2 domain-PAIyER:Req1-DTI7_2)</i>
<i>hasBeginning(Req1-DTI7_2,Req1-DTI7_2-Sins)</i>	⇒ <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req1-DTI7_2 domain-PAIyER:Req1-DTI7_2-Sins)</i>
<i>hasEnd(Req1-DTI7_2,Req1-DTI7_2-Eins)</i>	⇒ <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req1-DTI7_2 domain-PAIyER:Req1-DTI7_2-Eins)</i>
<i>}</i>	
<i>A₈: [XXXX (Empleado \sqsubseteq \existsexpide.Contrato)]: El empleado Valentino expide el contrato Contrato_Juan_Carlos.</i>	
<i>A₈ = {</i>	
<i>-- Aserciones de clase --</i>	
<i>Empleado(Valentino)</i>	⇒ <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Contrato(Contrato_Juan_Carlos)</i>	⇒ <i>ClassAssertion(domain-PAIyER:Contrato domain-PAIyER:Contrato_Juan_Carlos)</i>
<i>Event(Req1-Ev8)</i>	⇒ <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req1-Ev8)</i>
<i>DateTimeInterval(Req1-DTI8)</i>	⇒ <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req1-DTI8)</i>
<i>Instant(Req1-DTI8-Sins)</i>	⇒ <i>ClassAssertion(time:Instant domain-PAIyER:Req1-DTI8-Sins)</i>
<i>Instant(Req1-DTI8-Eins)</i>	⇒ <i>ClassAssertion(time:Instant domain-PAIyER:Req1-DTI8-Eins)</i>
<i>-- Aserciones de propiedades de objeto --</i>	
<i>expide(Valentino,Req1-Ev8)</i>	⇒ <i>ObjectPropertyAssertion(domain-PAIyER:expide domain-PAIyER:Valentino domain-PAIyER:Req1-Ev8)</i>
<i>expide(Req1-Ev8, Contrato_Juan_Carlos)</i>	⇒ <i>ObjectPropertyAssertion(domain-PAIyER:expide domain-PAIyER:Req1-Ev8 domain-PAIyER:Contrato_Juan_Carlos)</i>
<i>during(Req1-Ev8,Req1-DTI8)</i>	⇒ <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req1-Ev8 domain-PAIyER:Req1-DTI8)</i>
<i>hasBeginning(Req1-DTI8,Req1-DTI8-Sins)</i>	⇒ <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req1-DTI8 domain-PAIyER:Req1-DTI8-Sins)</i>
<i>hasEnd(Req1-DTI8,Req1-DTI8-Eins)</i>	⇒ <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req1-DTI8 domain-PAIyER:Req1-DTI8-Eins)</i>
<i>}</i>	
Caso de éxito 3 (A₉, A₁₀, A₁₁, A₁₂):	
<i>A₉: [X(Cliente \sqsubseteq \existsreserva.Entidad)]: El cliente Santiago separa el Entidad Libro3.</i>	
...	
<i>A₁₀: [XX (Empleado \sqsubseteq \existsverificaDisponibilidadDe.Entidad)]: El empleado Valentino verifica la</i>	

Tabla D.6 ABox de casos de éxito en el RF1. Hacer una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<p>disponibilidad de la entidad Libro3.</p> <p>...</p> <p>A_{11}: [XXX($Empleado \sqsubseteq \forall crea.Reservacion \wedge Empleado \sqsubseteq \exists actualiza.Entidad$)]: El empleado Valentino crea reservación Reservacion_prestamo_Santiago y actualiza la entidad Libro3.</p> <p>...</p> <p>A_{12}: [XXXX ($Empleado \sqsubseteq \exists expide.Contrato$)]: El empleado Valentino expide el contrato Contrato_Santiago.</p> <p>...</p>	
<p>Caso de éxito 4 ($A_{13}, A_{14}, A_{15}, A_{16}$):</p> <p>$A_{13}$: [X($Cliente \sqsubseteq \exists reserva.Entidad$)]: El cliente Violeta separa el Entidad Libro4.</p> <p>...</p> <p>A_{14}: [XX ($Empleado \sqsubseteq \exists verificaDisponibilidadDe.Entidad$)]: El empleado Valentino verifica la disponibilidad de la entidad Libro4.</p> <p>...</p> <p>A_{15}: [XXX($Empleado \sqsubseteq \forall crea.Reservacion \wedge Empleado \sqsubseteq \exists actualiza.Entidad$)]: El empleado Valentino crea reservación Reservacion_prestamo_Violeta y actualiza la entidad Libro4.</p> <p>...</p> <p>A_{16}: [XXXX ($Empleado \sqsubseteq \exists expide.Contrato$)]: El empleado Valentino expide el contrato Contrato_Violeta.</p> <p>...</p>	
<p>Caso de éxito n</p> <p>A_n</p> <p>...</p>	

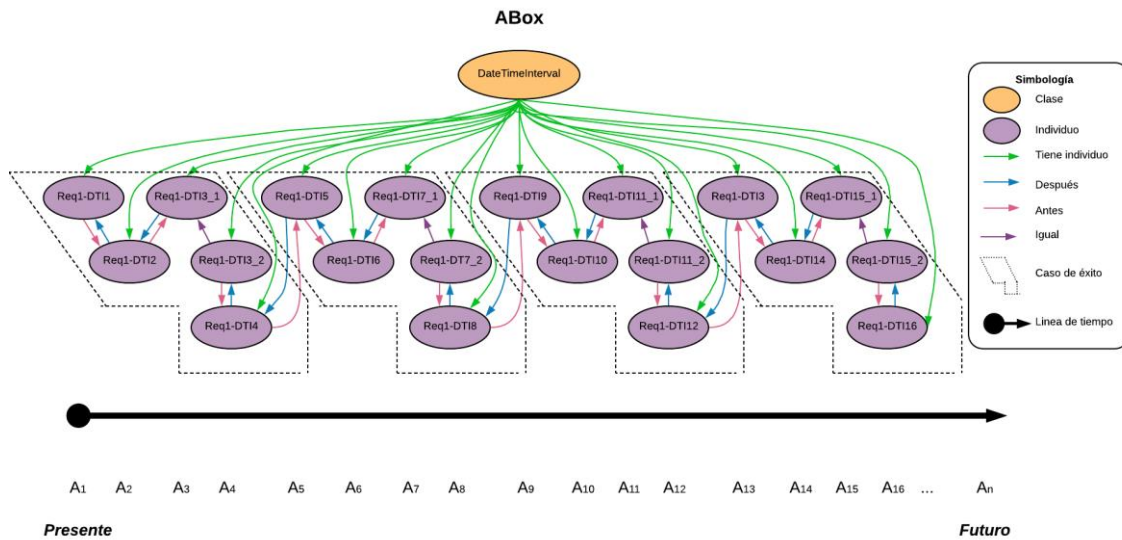


Figura D.2 SoIs en la ABox de casos de éxito de acuerdo al RF1. Hacer una reservación.

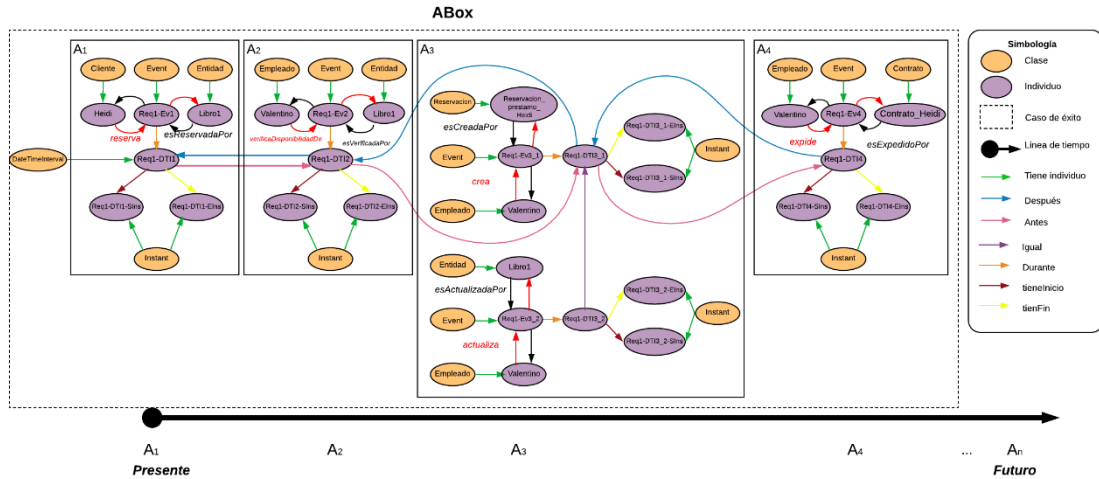


Figura D.3 Caso de éxito 1 en la ABox de casos de éxito de acuerdo al RF1. Hacer una reservación.

1.2.1.2. Casos de éxito y fracaso

En la Tabla D.7 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la ABox) basadas en casos de éxito y fracaso, que conforman una secuencia que representa el comportamiento dinámico para hacer una reservación. Una representación gráfica simplificada de la secuencia se muestra en la Figura D.4. A manera de ejemplo, en la Figura D.5 se muestra el detalle de las aserciones A_1 , A_2 , A_3 y A_4 en el caso de fracaso número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de razonamiento automatizado, los demás casos se definen de manera similar. Sin embargo, difieren en los individuos involucrados.

En la Tabla D.8 se describen las reglas SWRL utilizadas para restringir la existencia de afirmaciones de propiedades de objeto en el intervalo de tiempo correspondiente.

Tabla D.7 ABox de casos de éxito y fracaso en el RF1. Hacer una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
Caso de fracaso 1 (A_1, A_2, A_3, A_4):	
A_1 : $[X(\text{Cliente} \sqsubseteq \exists \text{reserva. Entidad})]$: El cliente Heidi reserva la Entidad Libro1.	
$A_1 = \{$	
-- Aserciones de clase --	
$\text{Cliente}(\text{Heidi})$	$\Rightarrow \text{ClassAssertion}(\text{domain-PAIyER:Cliente domain-PAIyER:Heidi})$
$\text{Entidad}(\text{Libro1})$	$\Rightarrow \text{ClassAssertion}(\text{domain-PAIyER:Entidad domain-PAIyER:Libro1})$
$\text{Event}(\text{Req1-Ev1})$	$\Rightarrow \text{ClassAssertion}(\text{timeEvents:Event domain-PAIyER:Req1-Ev1})$
$\text{DateTimeInterval}(\text{Req1-DT11})$	$\Rightarrow \text{ClassAssertion}(\text{time:DateTimeInterval domain-PAIyER:Req1-DT11})$
$\text{Instant}(\text{Req1-DT11-Sins})$	$\Rightarrow \text{ClassAssertion}(\text{time:Instant domain-PAIyER:Req1-DT11-Sins})$

Tabla D.7 ABox de casos de éxito y fracaso en el RF1. Hacer una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Instant(Req1-DT11-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req1-DT11-Eins)</i>
-- Aserciones de propiedades de objeto --	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:reserva domain-PAIyER:Heidi domain-PAIyER:Req1-Ev1)</i>
<i>reserva(Heidi,Req1-Ev1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:reserva domain-PAIyER:Req1-Ev1 domain-PAIyER:Libro1)</i>
<i>reserva(Req1-Ev1,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req1-Ev1 domain-PAIyER:Req1-DT11)</i>
<i>during(Req1-Ev1,Req1-DT11)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req1-DT11 domain-PAIyER:Req1-DT11-Sins)</i>
<i>hasBeginning(Req1-DT11,Req1-DT11-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req1-DT11 domain-PAIyER:Req1-DT11-Eins)</i>
<i>hasEnd(Req1-DT11,Req1-DT11-Eins)</i>	\Rightarrow
}	
<i>A₂: [XX (Empleado \sqsubseteq \existsverificaDisponibilidadDe.Entidad)]: El empleado Valentino verifica la disponibilidad de la Entidad Libro1</i>	
<i>A₂ = {</i>	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro1)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro1)</i>
<i>Event(Req1-Ev2)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req1-Ev2)</i>
<i>DateTimeInterval(Req1-DT12)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req1-DT12)</i>
<i>Instant(Req1-DT12-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req1-DT12-Sins)</i>
<i>Instant(Req1-DT12-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req1-DT12-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>verificaDisponibilidadDe(Valentino,Req1-Ev2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaDisponibilidadDe domain-PAIyER:Valentino domain-PAIyER:Req1-Ev2)</i>
<i>verificaDisponibilidadDe (Req1-Ev2,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaDisponibilidadDe domain-PAIyER:Req1-Ev2 domain-PAIyER:Libro1)</i>
<i>during(Req1-Ev2,Req1-DT12)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req1-Ev2 domain-PAIyER:Req1-DT12)</i>
<i>hasBeginning(Req1-DT12,Req1-DT12-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req1-DT12 domain-PAIyER:Req1-DT12-Sins)</i>
<i>hasEnd(Req1-DT12,Req1-DT12-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req1-DT12 domain-PAIyER:Req1-DT12-Eins)</i>
}	
<i>A₃: [\negXXX (Empleado \sqsubseteq \forallcrea.Reservacion \wedge Empleado \sqsubseteq \existsactualiza.Entidad)]: El empleado Valentino no crea la Reservacion Reservacion_prestamo_Heidi ni actualiza la Entidad Libro1.</i>	

Tabla D.7 ABox de casos de éxito y fracaso en el RF1. Hacer una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
$A_3 = \{$	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Entidad</i> (Libro1)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Entidad domain-PAIyER:Libro1)
<i>Reservacion</i> (Reservacion_prestamo_Heidi)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Reservacion domain-PAIyER:Reservacion_prestamo_Heidi)
<i>Event</i> (Req1-Ev3_1)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req1-Ev3_1)
<i>DateTimeInterval</i> (Req1-DTI3_1)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req1-DTI3_1)
<i>Instant</i> (Req1-DTI3_1-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req1-DTI3_1-Sins)
<i>Instant</i> (Req1-DTI3_1-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req1-DTI3_1-Eins)
<i>Event</i> (Req1-Ev3_2)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req1-Ev3_2)
<i>DateTimeInterval</i> (Req1-DTI3_2)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req1-DTI3_2)
<i>Instant</i> (Req1-DTI3_2-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req1-DTI3_2-Sins)
<i>Instant</i> (Req1-DTI3_2-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req1-DTI3_2-Eins)
-- Aserciones de propiedades de objeto --	
<i>crea</i> (Valentino,Req1-Ev3_1)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:crea domain-PAIyER:Valentino domain-PAIyER:Req1-Ev3_1)
<i>crea</i> (Req1-Ev3_1,Reservacion_prestamo_Heidi)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:crea domain-PAIyER:Req1-Ev3_1 domain-PAIyER:Reservacion_prestamo_Heidi)
<i>during</i> (Req1-Ev3_1,Req1-DTI3_1)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req1-Ev3_1 domain-PAIyER:Req1-DTI3_1)
<i>hasBeginning</i> (Req1-DTI3_1,Req1-DTI3_1-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req1-DTI3_1 domain-PAIyER:Req1-DTI3_1-Sins)
<i>hasEnd</i> (Req1-DTI3_1,Req1-DTI3_1-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req1-DTI3_1 domain-PAIyER:Req1-DTI3_1-Eins)
<i>actualiza</i> (Valentino,Req1-Ev3_2)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:actualiza domain-PAIyER:Valentino domain-PAIyER:Req1-Ev3_2)
<i>actualiza</i> (Req1-Ev3_2,Libro1)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:actualiza domain-PAIyER:Req1-Ev3_2 domain-PAIyER:Libro1)
<i>during</i> (Req1-Ev3_2,Req1-DTI3_2)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req1-Ev3_2 domain-PAIyER:Req1-DTI3_2)
<i>hasBeginning</i> (Req1-DTI3_2,Req1-DTI3_2-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req1-DTI3_2 domain-PAIyER:Req1-DTI3_2-Sins)
<i>hasEnd</i> (Req1-DTI3_2,Req1-DTI3_2-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req1-DTI3_2 domain-PAIyER:Req1-DTI3_2-Eins)

Tabla D.7 ABox de casos de éxito y fracaso en el RF1. Hacer una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<p style="text-align: right;">DTI3_2-Eins)</p> <p>}</p> <p>A4: [\negXXXX (Empleado \sqsubseteq \existsexpide.Contrato)]: El empleado Valentino no expide el contrato Contrato_Heidi.</p> <p>A4 = {</p> <p>-- Aserciones de clase --</p> <p>Empleado(Valentino)</p> <p>Contrato(Contrato_Heidi)</p> <p>Event(Req1-Ev4)</p> <p>DateTimeInterval(Req1-DTI4)</p> <p>Instant(Req1-DTI4-Sins)</p> <p>Instant(Req1-DTI4-Eins)</p> <p>-- Aserciones de propiedades de objeto --</p> <p>expide(Valentino,Req1-Ev4)</p> <p>expide(Req1-Ev4, Contrato_Heidi)</p> <p>during(Req1-Ev4,Req1-DTI4)</p> <p>hasBeginning(Req1-DTI4,Req1-DTI4-Sins)</p> <p>hasEnd(Req1-DTI4,Req1-DTI4-Eins)</p> <p>}</p>	<p>\Rightarrow ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</p> <p>\Rightarrow ClassAssertion(domain-PAIyER:Contrato domain-PAIyER:Contrato_Heidi)</p> <p>\Rightarrow ClassAssertion(timeEvents:Event domain-PAIyER:Req1-Ev4)</p> <p>\Rightarrow ClassAssertion(time:DateTimeInterval domain-PAIyER:Req1-DTI4)</p> <p>\Rightarrow ClassAssertion(time:Instant domain-PAIyER:Req1-DTI4-Sins)</p> <p>\Rightarrow ClassAssertion(time:Instant domain-PAIyER:Req1-DTI4-Eins)</p> <p>\Rightarrow ObjectPropertyAssertion(domain-PAIyER:expide domain-PAIyER:Valentino domain-PAIyER:Req1-Ev4)</p> <p>\Rightarrow ObjectPropertyAssertion(domain-PAIyER:expide domain-PAIyER:Req1-Ev4 domain-PAIyER:Contrato_Heidi)</p> <p>\Rightarrow ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req1-Ev4 domain-PAIyER:Req1-DTI4)</p> <p>\Rightarrow ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req1-DTI4 domain-PAIyER:Req1-DTI4-Sins)</p> <p>\Rightarrow ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req1-DTI4 domain-PAIyER:Req1-DTI4-Eins)</p>
<p>Caso de éxito 1 (A5, A6, A7, A8):</p> <p>A5: [X(Cliente \sqsubseteq \existsreserva.Entidad)]: El cliente Juan_Carlos reserva la Entidad Libro2.</p> <p>A5 = {</p> <p>-- Aserciones de clase --</p> <p>Cliente(Juan_Carlos)</p> <p>Entidad(Libro2)</p> <p>Event(Req1-Ev5)</p> <p>DateTimeInterval(Req1-DTI5)</p> <p>Instant(Req1-DTI5-Sins)</p> <p>Instant(Req1-DTI5-Eins)</p> <p>-- Aserciones de propiedades de objeto --</p> <p>reserva(Juan_Carlos,Req1-Ev5)</p>	<p>\Rightarrow ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Juan_Carlos)</p> <p>\Rightarrow ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</p> <p>\Rightarrow ClassAssertion(timeEvents:Event domain-PAIyER:Req1-Ev5)</p> <p>\Rightarrow ClassAssertion(time:DateTimeInterval domain-PAIyER:Req1-DTI5)</p> <p>\Rightarrow ClassAssertion(time:Instant domain-PAIyER:Req1-DTI5-Sins)</p> <p>\Rightarrow ClassAssertion(time:Instant domain-PAIyER:Req1-DTI5-Eins)</p> <p>\Rightarrow ObjectPropertyAssertion(domain-PAIyER:reserva domain-PAIyER:Juan_Carlos)</p>

Tabla D.7 ABox de casos de éxito y fracaso en el RF1. Hacer una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>reserva</i> (Req1-Ev5,Libro2)	\Rightarrow <i>domain-PAIyER:Req1-Ev5</i> \Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:reserva domain-PAIyER:Req1-Ev5 domain-PAIyER:Libro2)</i>
<i>during</i> (Req1-Ev5,Req1-DTI5)	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req1-Ev5 domain-PAIyER:Req1-DTI5)</i>
<i>hasBeginning</i> (Req1-DTI5,Req1-DTI5-Sins)	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req1-DTI5 domain-PAIyER:Req1-DTI5-Sins)</i>
<i>hasEnd</i> (Req1-DTI5,Req1-DTI5-Eins)	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req1-DTI5 domain-PAIyER:Req1-DTI5-Eins)</i>
}	\Rightarrow
A₆: [XX (Empleado \sqsubseteq \exists verificaDisponibilidadDe.Entidad)]: El empleado Valentino verifica la disponibilidad de la Entidad Libro2	
A ₆ = {	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad</i> (Libro2)	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event</i> (Req1-Ev6)	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req1-Ev6)</i>
<i>DateTimeInterval</i> (Req1-DTI6)	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req1-DTI6)</i>
<i>Instant</i> (Req1-DTI6-Sins)	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req1-DTI6-Sins)</i>
<i>Instant</i> (Req1-DTI6-Eins)	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req1-DTI6-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>verificaDisponibilidadDe</i> (Valentino,Req1-Ev6)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaDisponibilidadDe domain-PAIyER:Valentino domain-PAIyER:Req1-Ev6)</i>
<i>verificaDisponibilidadDe</i> (Req1-Ev6,Libro2)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaDisponibilidadDe domain-PAIyER:Req1-Ev6 domain-PAIyER:Libro2)</i>
<i>during</i> (Req1-Ev6,Req1-DTI6)	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req1-Ev6 domain-PAIyER:Req1-DTI6)</i>
<i>hasBeginning</i> (Req1-DTI6,Req1-DTI6-Sins)	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req1-DTI6 domain-PAIyER:Req1-DTI6-Sins)</i>
<i>hasEnd</i> (Req1-DTI6,Req1-DTI6-Eins)	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req1-DTI6 domain-PAIyER:Req1-DTI6-Eins)</i>
}	
A₇: [XXX(Empleado \sqsubseteq \forall crea.Reservacion \wedge Empleado \sqsubseteq \exists actualiza.Entidad)]: El empleado Valentino crea la reservación Reservacion_prestamo_JuanCarlos y actualiza la Entidad Libro2.	
A ₇ = {	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>

Tabla D.7 ABox de casos de éxito y fracaso en el RF1. Hacer una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Entidad(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Reservacion(Reservacion_prestamo_JuanCarlos)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Resrvacion domain-PAIyER:Reservacion_prestamo_JuanCarlos)</i>
<i>Event(Req1-Ev7_1)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req1-Ev7_1)</i>
<i>DateTimeInterval(Req1-DTI7_1)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req1-DTI7_1)</i>
<i>Instant(Req1-DTI7_1-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req1-DTI7_1-Sins)</i>
<i>Instant(Req1-DTI7_1-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req1-DTI7_1-Eins)</i>
<i>Event(Req1-Ev7_2)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req1-Ev7_2)</i>
<i>DateTimeInterval(Req1-DTI7_2)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req1-DTI7_2)</i>
<i>Instant(Req1-DTI7_2-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req1-DTI7_2-Sins)</i>
<i>Instant(Req1-DTI7_2-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req1-DTI7_2-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>crea(Valentino,Req1-Ev7_1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:crea domain-PAIyER:Valentino domain-PAIyER:Req1-Ev7_1)</i>
<i>crea(Req1-Ev7_1,Reservacion_prestamo_JuanCarlos)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:crea domain-PAIyER:Req1-Ev7_1 domain-PAIyER:Reservacion_prestamo_JuanCarlos)</i>
<i>during(Req1-Ev7_1,Req1-DTI7_1)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req1-Ev7_1 domain-PAIyER:Req1-DTI7_1)</i>
<i>hasBeginning(Req1-DTI7_1,Req1-DTI7_1-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req1-DTI7_1 domain-PAIyER:Req1-DTI7_1-Sins)</i>
<i>hasEnd(Req1-DTI7_1,Req1-DTI7_1-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req1-DTI7_1 domain-PAIyER:Req1-DTI7_1-Eins)</i>
<i>actualiza(Valentino,Req1-Ev7_2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Valentino domain-PAIyER:Req1-Ev7_2)</i>
<i>actualiza(Req1-Ev7_2,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Req1-Ev7_2 domain-PAIyER:Libro2)</i>
<i>during(Req1-Ev7_2,Req1-DTI7_2)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req1-Ev7_2 domain-PAIyER:Req1-DTI7_2)</i>
<i>hasBeginning(Req1-DTI7_2,Req1-DTI7_2-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req1-DTI7_2 domain-PAIyER:Req1-DTI7_2-Sins)</i>
<i>hasEnd(Req1-DTI7_2,Req1-DTI7_2-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req1-DTI7_2 domain-PAIyER:Req1-DTI7_2-Eins)</i>
}	
A₈: [XXXXX (Empleado \sqsubseteq \existsexpide.Contrato)]: El empleado Valentino expide el contrato	

Tabla D.7 ABox de casos de éxito y fracaso en el RF1. Hacer una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<p><i>Contrato_Juan_Carlos.</i></p> <p>$A_8 = \{$</p> <p>-- Aserciones de clase --</p> <p><i>Empleado(Valentino)</i></p> <p><i>Contrato(Contrato_Juan_Carlos)</i></p> <p><i>Event(Req1-Ev8)</i></p> <p><i>DateTimeInterval(Req1-DT18)</i></p> <p><i>Instant(Req1-DT18-Sins)</i></p> <p><i>Instant(Req1-DT18-Eins)</i></p> <p>-- Aserciones de propiedades de objeto --</p> <p><i>expide(Valentino,Req1-Ev8)</i></p> <p><i>expide(Req1-Ev8, Contrato_Juan_Carlos)</i></p> <p><i>during(Req1-Ev8,Req1-DT18)</i></p> <p><i>hasBeginning(Req1-DT18,Req1-DT18-Sins)</i></p> <p><i>hasEnd(Req1-DT18,Req1-DT18-Eins)</i></p> <p>$\}$</p>	<p>\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i></p> <p>\Rightarrow <i>ClassAssertion(domain-PAIyER:Contrato domain-PAIyER:Contrato_Juan_Carlos)</i></p> <p>\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req1-Ev8)</i></p> <p>\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req1-DT18)</i></p> <p>\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req1-DT18-Sins)</i></p> <p>\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req1-DT18-Eins)</i></p> <p>\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:expide domain-PAIyER:Valentino domain-PAIyER:Req1-Ev8)</i></p> <p>\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:expide domain-PAIyER:Req1-Ev8 domain-PAIyER:Contrato_Juan_Carlos)</i></p> <p>\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req1-Ev8 domain-PAIyER:Req1-DT18)</i></p> <p>\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req1-DT18 domain-PAIyER:Req1-DT18-Sins)</i></p> <p>\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req1-DT18 domain-PAIyER:Req1-DT18-Eins)</i></p>
<p>Caso de fracaso 2 ($A_9, A_{10}, A_{11}, A_{12}$):</p> <p>$A_9$: [X(<i>Cliente \sqsubseteq \existsreserva.Entidad</i>)]: El cliente Santiago separa el Entidad Libro3.</p> <p>...</p> <p>A_{10}: [XX (<i>Empleado \sqsubseteq \existsverificaDisponibilidadDe.Entidad</i>)]: El empleado Valentino verifica la disponibilidad de la entidad Libro3.</p> <p>...</p> <p>A_{11}: [XXX(<i>Empleado \sqsubseteq \forallcrea.Reservacion \wedge Empleado \sqsubseteq \existsactualiza.Entidad</i>)]: El empleado Valentino no crea reservación <i>Reservacion_prestamo_Santiago</i> ni actualiza la entidad Libro3.</p> <p>...</p> <p>A_{12}: [XXXX (<i>Empleado \sqsubseteq \existsexpide.Contrato</i>)]: El empleado Valentino no expide el contrato <i>Contrato_Santiago</i>.</p> <p>...</p>	
<p>Caso de éxito 2 ($A_{13}, A_{14}, A_{15}, A_{16}$):</p> <p>$A_{13}$: [X(<i>Cliente \sqsubseteq \existsreserva.Entidad</i>)]: El cliente Violeta separa el Entidad Libro4.</p> <p>...</p> <p>A_{14}: [XX (<i>Empleado \sqsubseteq \existsverificaDisponibilidadDe.Entidad</i>)]: El empleado Valentino verifica la disponibilidad de la entidad Libro4.</p> <p>...</p> <p>A_{15}: [XXX(<i>Empleado \sqsubseteq \forallcrea.Reservacion \wedge Empleado \sqsubseteq \existsactualiza.Entidad</i>)]: El empleado Valentino crea reservación <i>Reservacion_prestamo_Violeta</i> y actualiza la entidad Libro4.</p> <p>...</p>	

Tabla D.7 ABox de casos de éxito y fracaso en el RF1. Hacer una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
A_{16} : $[XXXX (Empleado \sqsubseteq \exists \text{expide. Contrato})]$: El empleado Valentino expide el contrato Contrato_Violeta.	
...	
Caso de éxito n	
A_n	
...	

Tabla D.8 Reglas SWRL para la SoIs de casos de éxito y fracaso de acuerdo al RF Hacer una reservación. (cont.)

ABox	Regla SWRL
A_3	$Event(?e), during(?e, ?i), overlaps(?i, Req1-DT13_1), crea(?e, Reservacion_prestamo_Heidi), crea(Valentino, ?e) \rightarrow Nothing(Valentino)$
	$Event(?e), during(?e, ?i), overlaps(?i, Req1-DT13_2), actualiza(?e, Libro1), actualiza(Valentino, ?e) \rightarrow Nothing(Valentino)$
A_4	$Event(?e), during(?e, ?i), overlaps(?i, Req1-DT14), expide(?e, Contrato_Heidi), expide(Valentino, ?e) \rightarrow Nothing(Valentino)$
A_{11}	$Event(?e), during(?e, ?i), overlaps(?i, Req1-DT11_1), crea(?e, Reservacion_prestamo_Santiago), crea(Valentino, ?e) \rightarrow Nothing(Valentino)$
	$Event(?e), during(?e, ?i), overlaps(?i, Req1-DT11_2), actualiza(?e, Libro3), actualiza(Valentino, ?e) \rightarrow Nothing(Valentino)$
A_{12}	$Event(?e), during(?e, ?i), overlaps(?i, Req1-DT12), expide(?e, Contrato_Santiago), expide(Valentino, ?e) \rightarrow Nothing(Valentino)$

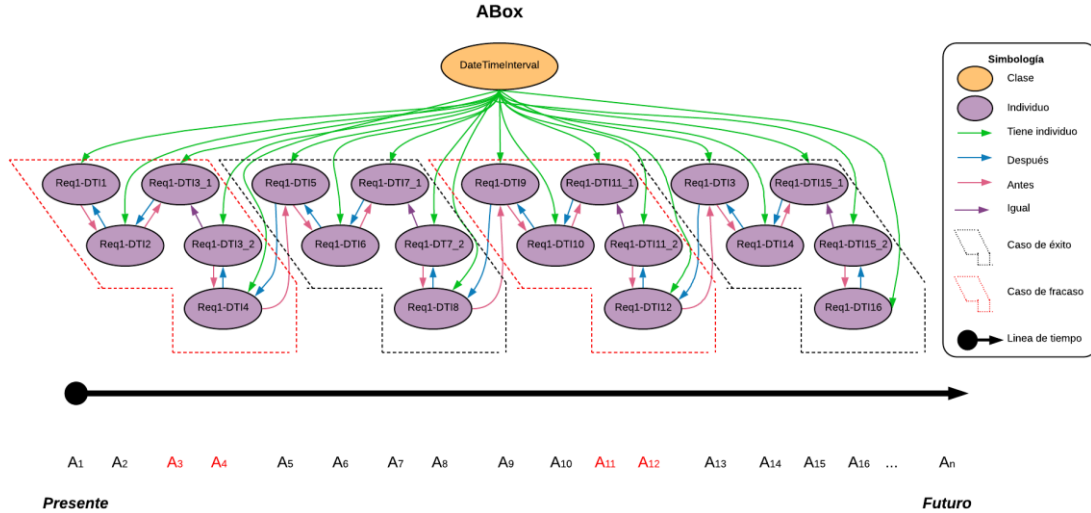


Figura D.4 SoIs en la ABox de casos de éxito y fracaso de acuerdo al RF1. Hacer una reservación.

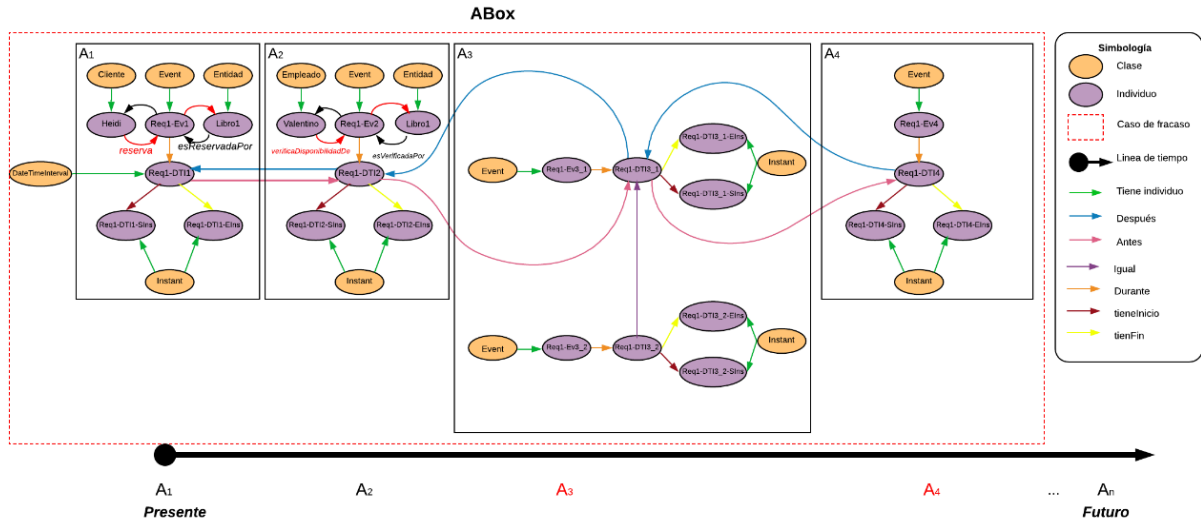


Figura D.5 Caso de fracaso 1 en la ABox de casos de éxito y fracaso de acuerdo al RF1. Hacer una reservación.

1.2.2 RF2. Usar una entidad reservada

Las secuencias de interpretaciones descritas en esta sección satisfacen/cumplen con el comportamiento dinámico del RF2. Usar una entidad reservada representado formalmente con la fórmula de la Tabla D.9.

Tabla D.9 Fórmula ALC_{LTL} del RF2. Usar una entidad reservada.

$$\begin{aligned}
 & \mathcal{G} (\\
 & \quad \mathcal{X} (\text{Empleado} \sqsubseteq \exists \text{registraUsoDe. Entidad}) \rightarrow \mathcal{F} \\
 & \quad \neg \mathcal{X} (\text{Cliente} \sqsubseteq \exists \text{usa. Entidad}) \wedge \\
 & \quad \neg \mathcal{X} \mathcal{X} (\text{Cliente} \sqsubseteq \exists \text{devuelve. Entidad}) \\
 &) \\
 &)
 \end{aligned}$$

1.2.2.1. Casos de éxito

En la Tabla D.10 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la *ABox*) basadas en casos de éxito, que conforman una secuencia que representa el comportamiento dinámico para usar una entidad reservada. Una representación gráfica simplificada de la secuencia se muestra en la Figura D.6. A manera de ejemplo, en la Figura D.7 se muestra el detalle de las aserciones A_1 , A_2 , A_3 y A_4 en el caso de éxito número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de razonamiento automatizado, los demás casos se definen de manera similar, sin embargo, difieren en los individuos involucrados.

Tabla D.10 ABox de casos de éxito en el RF2. Usar una entidad reservada (cont.)

Sintaxis DL	Sintaxis Funcional
Caso de éxito 1 (A_1, A_2, A_3):	
A_1 : [X (Empleado $\sqsubseteq \exists$ registraUsoDe.Entidad)]: El empleado Valentino registra el uso de la entidad Libro1.	
$A_1 = \{$	
-- Aserciones de clase --	
Empleado(Valentino)	\Rightarrow ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)
Entidad(Libro1)	\Rightarrow ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro1)
Event(Req2-Ev1)	\Rightarrow ClassAssertion(timeEvents:Event domain-PAIyER:Req2-Ev1)
DateTimeInterval(Req2-DT11)	\Rightarrow ClassAssertion(time:DateTimeInterval domain-PAIyER:Req2-DT11)
Instant(Req2-DT11-Sins)	\Rightarrow ClassAssertion(time:Instant domain-PAIyER:Req2-DT11-Sins)
Instant(Req2-DT11-Eins)	\Rightarrow ClassAssertion(time:Instant domain-PAIyER:Req2-DT11-Eins)
-- Aserciones de propiedades de objeto --	
registraUsoDe(Valentino,Req2-Ev1)	\Rightarrow ObjectPropertyAssertion(domain-PAIyER:registraUsoDe domain-PAIyER:Valentino domain-PAIyER:Req2-Ev1)
registraUsoDe(Req2-Ev1,Libro1)	\Rightarrow ObjectPropertyAssertion(domain-PAIyER:registraUsoDe domain-PAIyER:Req2-Ev1 domain-PAIyER:Libro1)
during(Req2-Ev1,Req2-DT11)	\Rightarrow ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req2-Ev1 domain-PAIyER:Req2-DT11)
hasBeginning(Req2-DT11,Req2-DT11-Sins)	\Rightarrow ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req2-DT11 domain-PAIyER:Req2-DT11-Sins)
hasEnd(Req2-DT11,Req2-DT11-Eins)	\Rightarrow ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req2-DT11 domain-PAIyER:Req2-DT11-Eins)
}	\Rightarrow
A_2 : [XX (Cliente $\sqsubseteq \exists$ usa.Entidad)]: El cliente Heidi usa la entidad Libro1.	
$A_2 = \{$	
-- Aserciones de clase --	
Cliente(Heidi)	\Rightarrow ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Heidi)
Entidad(Libro1)	\Rightarrow ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro1)
Event(Req2-Ev2)	\Rightarrow ClassAssertion(timeEvents:Event domain-PAIyER:Req2-Ev2)
DateTimeInterval(Req2-DT12)	\Rightarrow ClassAssertion(time:DateTimeInterval domain-PAIyER:Req2-DT12)
Instant(Req2-DT12-Sins)	\Rightarrow ClassAssertion(time:Instant domain-PAIyER:Req2-DT12-Sins)
Instant(Req2-DT12-Eins)	\Rightarrow ClassAssertion(time:Instant domain-PAIyER:Req2-DT12-Eins)
-- Aserciones de propiedades de objeto --	
usa(Heidi,Req2-Ev2)	\Rightarrow ObjectPropertyAssertion(domain-PAIyER:usa domain-PAIyER:Heidi domain-PAIyER:Req2-Ev2)
usa (Req2-Ev2,Libro1)	\Rightarrow ObjectPropertyAssertion(domain-PAIyER:usa domain-PAIyER:Req2-Ev2 domain-PAIyER:Libro1)
during(Req2-Ev2,Req2-DT12)	\Rightarrow ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req2-Ev2 domain-PAIyER:Req2-DT12)
hasBeginning(Req2-DT12,Req2-DT12-Sins)	\Rightarrow ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req2-DT12 domain-PAIyER:Req2-DT12-Sins)
hasEnd(Req2-DT12,Req2-DT12-Eins)	\Rightarrow ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req2-DT12 domain-PAIyER:Req2-DT12-Eins)

Tabla D.10 ABox de casos de éxito en el RF2. Usar una entidad reservada (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Eins</i>)	<i>PAIyER:Req2-DTI2 domain-PAIyER:Req2-DTI2-Eins</i>)
}	
A3: [XXXX (Cliente \sqsubseteq \exists devuelve.Entidad)]: El cliente Heidi devuelve la entidad Libro1.	
A3 = {	
-- Aserciones de clase --	
<i>Cliente</i> (Heidi)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Cliente domain-PAIyER:Heidi)
<i>Entidad</i> (Libro1)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Entidad domain-PAIyER: Libro1)
<i>Event</i> (Req2-Ev3)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req2-Ev3)
<i>DateTimeInterval</i> (Req2-DTI3)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req2-DTI3)
<i>Instant</i> (Req2-DTI3-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req2-DTI3-Sins)
<i>Instant</i> (Req2-DTI3-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req2-DTI3-Eins)
-- Aserciones de propiedades de objeto --	
<i>devuelve</i> (Heidi,Req2-Ev3)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: devuelve domain-PAIyER:Heidi domain-PAIyER:Req2-Ev3)
<i>devuelve</i> (Req2-Ev3, LLibro1)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: devuelve domain-PAIyER:Req2-Ev3 domain-PAIyER: Libro1)
<i>during</i> (Req2-Ev3,Req2-DTI3)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req2-Ev3 domain-PAIyER:Req2-DTI3)
<i>hasBeginning</i> (Req2-DTI3,Req2-DTI3-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req2-DTI3 domain-PAIyER:Req2-DTI3-Sins)
<i>hasEnd</i> (Req2-DTI3,Req2-DTI3-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req2-DTI3 domain-PAIyER:Req2-DTI3-Eins)
}	
Caso de éxito 2 (A4, A5, A6):	
A4: [X(Empleado \sqsubseteq \exists registraUsoDe.Entidad)]: El empleado Valentino registra el uso de la entidad Libro2.	
A4 = {	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Entidad</i> (Libro2)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Entidad domain-PAIyER:Libro2)
<i>Event</i> (Req2-Ev4)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req2-Ev4)
<i>DateTimeInterval</i> (Req2-DTI4)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req2-DTI4)
<i>Instant</i> (Req2-DTI4-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req2-DTI4-Sins)
<i>Instant</i> (Req2-DTI4-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req2-DTI4-Eins)
-- Aserciones de propiedades de objeto --	
<i>registraUsoDe</i> (Valentino,Req2-Ev4)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: registraUsoDe domain-PAIyER:Valentino domain-PAIyER:Req2-Ev4)
<i>registraUsoDe</i> (Req2-Ev4,Libro2)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: registraUsoDe domain-PAIyER:Req2-Ev4 domain-PAIyER:Libro2)
<i>during</i> (Req2-Ev4,Req2-DTI4)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req2-Ev4 domain-PAIyER:Req2-DTI4)
<i>hasBeginning</i> (Req2-DTI4,Req2-	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-

Tabla D.10 ABox de casos de éxito en el RF2. Usar una entidad reservada (cont.)

Sintaxis DL	Sintaxis Funcional
<i>DTI4-Sins</i>	<i>PAIyER:Req2-DTI4 domain-PAIyER:Req2-DTI4-Sins</i>
<i>hasEnd(Req2-DTI4,Req2-DTI4-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req2-DTI4 domain-PAIyER:Req2-DTI4-Eins)</i>
}	\Rightarrow
A5 = {	A5: [XX (<i>Cliente</i> \sqsubseteq \exists <i>usa.Entidad</i>)]: <i>El cliente Juan_Carlos usa la entidad Libro2.</i>
-- Aserciones de clase --	
<i>Cliente(Juan_Carlos)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Juan_Carlos)</i>
<i>Entidad(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event(Req2-Ev5)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req2-Ev5)</i>
<i>DateTimeInterval(Req2-DTI5)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req2-DTI5)</i>
<i>Instant(Req2-DTI5-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req2-DTI5-Sins)</i>
<i>Instant(Req2-DTI5-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req2-DTI5-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>usa(Juan_Carlos,Req2-Ev5)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER: usa domain-PAIyER:Juan_Carlos domain-PAIyER:Req2-Ev5)</i>
<i>usa (Req2-Ev5,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER: usa domain-PAIyER:Req2-Ev5 domain-PAIyER:Libro2)</i>
<i>during(Req2-Ev5,Req2-DTI5)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req2-Ev5 domain-PAIyER:Req2-DTI5)</i>
<i>hasBeginning(Req2-DTI5,Req2-DTI5-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req2-DTI5 domain-PAIyER:Req2-DTI5-Sins)</i>
<i>hasEnd(Req2-DTI5,Req2-DTI5-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req2-DTI5 domain-PAIyER:Req2-DTI5-Eins)</i>
}	
A6 = {	A6: [XXXX (<i>Cliente</i> \sqsubseteq \exists <i>devuelve.Entidad</i>)]: <i>El cliente Juan_Carlos devuelve la entidad Libro2.</i>
-- Aserciones de clase --	
<i>Cliente(Juan_Carlos)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Juan_Carlos)</i>
<i>Entidad(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER: Libro2)</i>
<i>Event(Req2-Ev6)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req2-Ev6)</i>
<i>DateTimeInterval(Req2-DTI6)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req2-DTI6)</i>
<i>Instant(Req2-DTI6-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req2-DTI6-Sins)</i>
<i>Instant(Req2-DTI6-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req2-DTI6-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>devuelve(Juan_Carlos,Req2-Ev6)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:devuelve domain-PAIyER:Juan_Carlos domain-PAIyER:Req2-Ev6)</i>
<i>devuelve(Req2-Ev6, Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:devuelve domain-PAIyER:Req2-Ev6 domain-PAIyER: Libro2)</i>
<i>during(Req2-Ev6,Req2-DTI6)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req2-Ev6 domain-PAIyER:Req2-DTI6)</i>

Tabla D.10 ABox de casos de éxito en el RF2. Usar una entidad reservada (cont.)

Sintaxis DL	Sintaxis Funcional
$hasBeginning(Req2-DTI6, Req2-DTI6-Sins)$ $hasEnd(Req2-DTI6, Req2-DTI6-Eins)$	$\Rightarrow ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req2-DTI6 domain-PAIyER:Req2-DTI6-Sins)$ $\Rightarrow ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req2-DTI6 domain-PAIyER:Req2-DTI6-Eins)$
<p>Caso de éxito 3 (A₇, A₈, A₉): A₇: [X(Empleado \sqsubseteq $\exists registraUsode.Entidad$)]: El empleado Valentino registra el uso de la entidad Libro3. ... A₈: [XX (Cliente \sqsubseteq $\exists usa.Entidad$)]: El cliente Santiago usa la entidad Libro3. ... A₉: [XXXX (Cliente \sqsubseteq $\exists devuelve.Entidad$)]: El cliente Santiago devuelve la entidad Libro3. ...</p>	
<p>Caso de éxito 4 (A₁₀, A₁₁, A₁₂): A₁₀: [X(Empleado \sqsubseteq $\exists registraUsode.Entidad$)]: El empleado Valentino registra el uso de la entidad Libro4. ... A₁₁: [XX (Cliente \sqsubseteq $\exists usa.Entidad$)]: El cliente Violeta usa la entidad Libro4. ... A₁₂: [XXXX (Cliente \sqsubseteq $\exists devuelve.Entidad$)]: El cliente Violeta devuelve la entidad Libro4. ...</p>	
<p>Caso de éxito n A_n ...</p>	

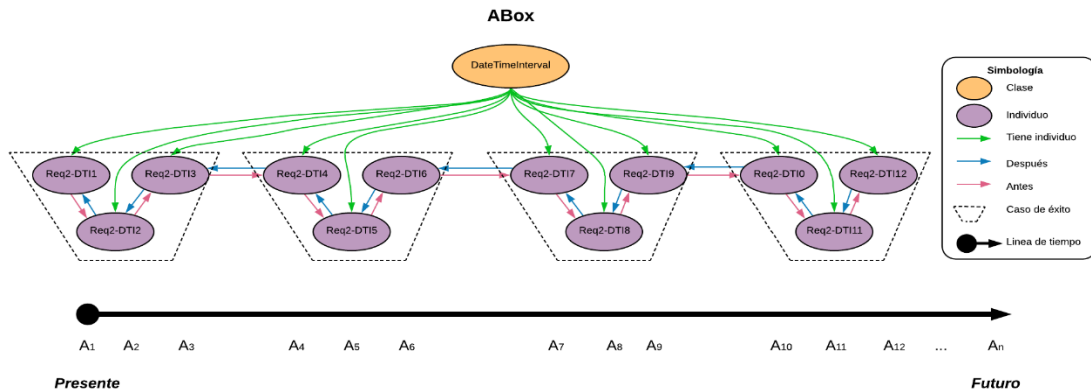


Figura D.6 SoIs en la ABox de casos de éxito de acuerdo al RF2. Usar una entidad reservada.

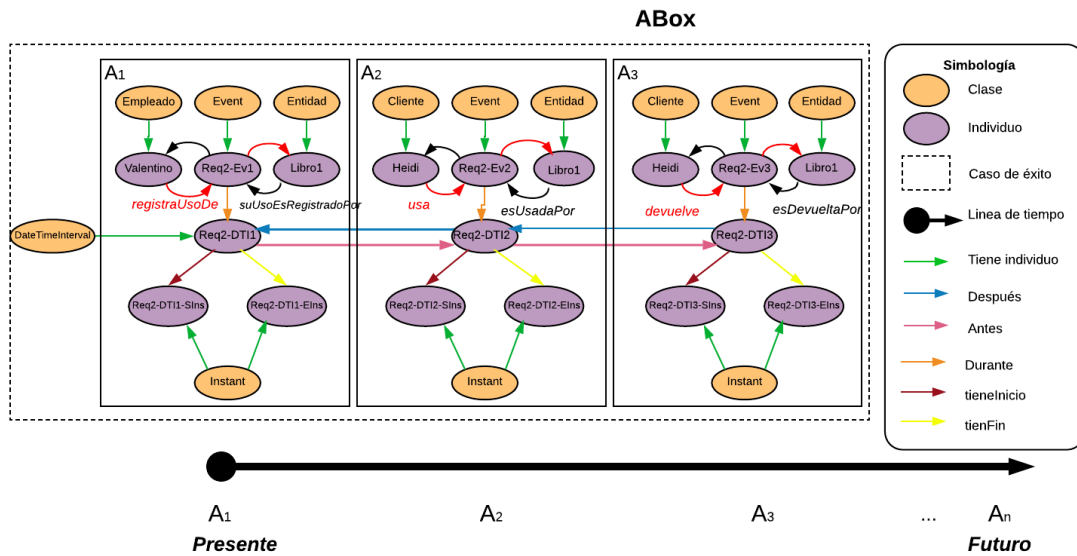


Figura D.7 Caso de éxito 1 en la ABox de casos de éxito de acuerdo al RF2. Usar una entidad reservada.

1.2.2.2. Casos de éxito y fracaso

En la Tabla D.11 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la *ABox*) basadas en casos de éxito y fracaso, que conforman una secuencia que representa el comportamiento dinámico para usar una entidad reservada. Una representación gráfica simplificada de la secuencia se muestra en la Figura D.8. A manera de ejemplo, en la Figura D.9 se muestra el detalle de las aserciones A_1, A_2, A_3 y A_4 en el caso de fracaso número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de razonamiento automatizado. Los demás casos se definen de manera similar, sin embargo, difieren en los individuos involucrados.

En la Tabla D.12 se describen las reglas SWRL utilizadas para restringir la existencia de afirmaciones de propiedades de objeto en el intervalo de tiempo correspondiente.

Tabla D.11 ABox de casos de éxito y fracaso en el RF2. Usar una entidad reservada (cont.)

Sintaxis DL	Sintaxis Funcional
Caso de fracaso 1 (A_1, A_2, A_3):	
$A_1: [X(Empleado \sqsubseteq \exists registraUsoDe.Entidad)]:$ El empleado Valentino registra el uso de la entidad Libro1.	
$A_1 = \{$	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Entidad</i> (Libro1)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Entidad domain-PAIyER:Libro1)
<i>Event</i> (Req2-Ev1)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req2-Ev1)

Tabla D.11 ABox de casos de éxito y fracaso en el RF2. Usar una entidad reservada (cont.)

Sintaxis DL	Sintaxis Funcional
<i>DateTimeInterval(Req2-DT11)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req2-DT11)</i>
<i>Instant(Req2-DT11-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req2-DT11-Sins)</i>
<i>Instant(Req2-DT11-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req2-DT11-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>registraUsode(Valentino,Req2-Ev1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:registraUsode domain-PAIyER:Valentino domain-PAIyER:Req2-Ev1)</i>
<i>registraUsode(Req2-Ev1,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:registraUsode domain-PAIyER:Req2-Ev1 domain-PAIyER:Libro1)</i>
<i>during(Req2-Ev1,Req2-DT11)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req2-Ev1 domain-PAIyER:Req2-DT11)</i>
<i>hasBeginning(Req2-DT11,Req2-DT11-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req2-DT11 domain-PAIyER:Req2-DT11-Sins)</i>
<i>hasEnd(Req2-DT11,Req2-DT11-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req2-DT11 domain-PAIyER:Req2-DT11-Eins)</i>
}	\Rightarrow
$A_2 = \{$	A₂: [\negXX (Cliente \sqsubseteq \existsusa.Entidad)]: El cliente Heidi no usa la entidad Libro1.
-- Aserciones de clase --	
<i>Cliente(Heidi)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Heidi)</i>
<i>Entidad(Libro1)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro1)</i>
<i>Event(Req2-Ev2)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req2-Ev2)</i>
<i>DateTimeInterval(Req2-DT12)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req2-DT12)</i>
<i>Instant(Req2-DT12-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req2-DT12-Sins)</i>
<i>Instant(Req2-DT12-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req2-DT12-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>usa(Heidi,Req2-Ev2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:usa domain-PAIyER:Heidi domain-PAIyER:Req2-Ev2)</i>
<i>usa(Req2-Ev2,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:usa domain-PAIyER:Req2-Ev2 domain-PAIyER:Libro1)</i>
<i>during(Req2-Ev2,Req2-DT12)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req2-Ev2 domain-PAIyER:Req2-DT12)</i>
<i>hasBeginning(Req2-DT12,Req2-DT12-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req2-DT12 domain-PAIyER:Req2-DT12-Sins)</i>
<i>hasEnd(Req2-DT12,Req2-DT12-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req2-DT12 domain-PAIyER:Req2-DT12-Eins)</i>
}	
$A_3 = \{$	A₃: [\negXXXX (Cliente \sqsubseteq \existsdevuelve.Entidad)]: El cliente Heidi no devuelve la entidad Libro1.
-- Aserciones de clase --	
<i>Cliente(Heidi)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Heidi)</i>
<i>Entidad(Libro1)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro1)</i>
<i>Event(Req2-Ev3)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req2-Ev3)</i>

Tabla D.11 ABox de casos de éxito y fracaso en el RF2. Usar una entidad reservada (cont.)

Sintaxis DL	Sintaxis Funcional
<i>DateTimeInterval(Req2-DTI3)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req2-DTI3)</i>
<i>Instant(Req2-DTI3-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req2-DTI3-Sins)</i>
<i>Instant(Req2-DTI3-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req2-DTI3-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>devuelve(Heidi,Req2-Ev3)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:devuelve domain-PAIyER:Heidi domain-PAIyER:Req2-Ev3)</i>
<i>devuelve(Req2-Ev3, LLibro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:devuelve domain-PAIyER:Req2-Ev3 domain-PAIyER: Libro1)</i>
<i>during(Req2-Ev3,Req2-DTI3)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req2-Ev3 domain-PAIyER:Req2-DTI3)</i>
<i>hasBeginning(Req2-DTI3,Req2-DTI3-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req2-DTI3 domain-PAIyER:Req2-DTI3-Sins)</i>
<i>hasEnd(Req2-DTI3,Req2-DTI3-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req2-DTI3 domain-PAIyER:Req2-DTI3-Eins)</i>
}	
Caso de éxito 1 (A₄, A₅, A₆):	
A₄: [X(Empleado \sqsubseteq \existsregistraUsoDe.Entidad)]: El empleado Valentino registra el uso de la entidad Libro2.	
A ₄ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event(Req2-Ev4)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req2-Ev4)</i>
<i>DateTimeInterval(Req2-DTI4)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req2-DTI4)</i>
<i>Instant(Req2-DTI4-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req2-DTI4-Sins)</i>
<i>Instant(Req2-DTI4-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req2-DTI4-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>registraUsoDe(Valentino,Req2-Ev4)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:registraUsoDe domain-PAIyER:Valentino domain-PAIyER:Req2-Ev4)</i>
<i>registraUsoDe(Req2-Ev4,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:registraUsoDe domain-PAIyER:Req2-Ev4 domain-PAIyER:Libro2)</i>
<i>during(Req2-Ev4,Req2-DTI4)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req2-Ev4 domain-PAIyER:Req2-DTI4)</i>
<i>hasBeginning(Req2-DTI4,Req2-DTI4-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req2-DTI4 domain-PAIyER:Req2-DTI4-Sins)</i>
<i>hasEnd(Req2-DTI4,Req2-DTI4-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req2-DTI4 domain-PAIyER:Req2-DTI4-Eins)</i>
}	\Rightarrow
A₅: [XX (Cliente \sqsubseteq \existsusa.Entidad)]: El cliente Juan_Carlos usa la entidad Libro2.	
A ₅ = {	
-- Aserciones de clase --	
<i>Cliente(Juan_Carlos)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Juan_Carlos)</i>
<i>Entidad(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>

Tabla D.11 ABox de casos de éxito y fracaso en el RF2. Usar una entidad reservada (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Event</i> (Req2-Ev5)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req2-Ev5)
<i>DateTimeInterval</i> (Req2-DTI5)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req2-DTI5)
<i>Instant</i> (Req2-DTI5-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req2-DTI5-Sins)
<i>Instant</i> (Req2-DTI5-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req2-DTI5-Eins)
-- Aserciones de propiedades de objeto --	
<i>usa</i> (Juan_Carlos,Req2-Ev5)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: usa domain-PAIyER:Juan_Carlos domain-PAIyER:Req2-Ev5)
<i>usa</i> (Req2-Ev5,Libro2)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: usa domain-PAIyER:Req2-Ev5 domain-PAIyER:Libro2)
<i>during</i> (Req2-Ev5,Req2-DTI5)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req2-Ev5 domain-PAIyER:Req2-DTI5)
<i>hasBeginning</i> (Req2-DTI5,Req2-DTI5-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req2-DTI5 domain-PAIyER:Req2-DTI5-Sins)
<i>hasEnd</i> (Req2-DTI5,Req2-DTI5-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req2-DTI5 domain-PAIyER:Req2-DTI5-Eins)
}	
A6: [XXXX (Cliente \sqsubseteq \existsdevuelve.Entidad)]: El cliente Juan_Carlos devuelve la entidad Libro2.	
A6 = {	
-- Aserciones de clase --	
<i>Cliente</i> (Juan_Carlos)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Cliente domain-PAIyER:Juan_Carlos)
<i>Entidad</i> (Libro2)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Entidad domain-PAIyER: Libro2)
<i>Event</i> (Req2-Ev6)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req2-Ev6)
<i>DateTimeInterval</i> (Req2-DTI6)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req2-DTI6)
<i>Instant</i> (Req2-DTI6-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req2-DTI6-Sins)
<i>Instant</i> (Req2-DTI6-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req2-DTI6-Eins)
-- Aserciones de propiedades de objeto --	
<i>devuelve</i> (Juan_Carlos,Req2-Ev6)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:devuelve domain-PAIyER:Juan_Carlos domain-PAIyER:Req2-Ev6)
<i>devuelve</i> (Req2-Ev6, Libro2)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:devuelve domain-PAIyER:Req2-Ev6 domain-PAIyER: Libro2)
<i>during</i> (Req2-Ev6,Req2-DTI6)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req2-Ev6 domain-PAIyER:Req2-DTI6)
<i>hasBeginning</i> (Req2-DTI6,Req2-DTI6-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req2-DTI6 domain-PAIyER:Req2-DTI6-Sins)
<i>hasEnd</i> (Req2-DTI6,Req2-DTI6-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req2-DTI6 domain-PAIyER:Req2-DTI6-Eins)
}	
Caso de fracaso 2 (A7, A8, A9):	
A7: [X(Empleado \sqsubseteq \existsregistraUsode.Entidad)]: El empleado Valentino registra el uso de la entidad Libro3.	
...	
A8: [\negXX (Cliente \sqsubseteq \existsusa.Entidad)]: El cliente Santiago no usa la entidad Libro3.	
...	

Tabla D.11 ABox de casos de éxito y fracaso en el RF2. Usar una entidad reservada (cont.)

Sintaxis DL	Sintaxis Funcional
$A_9: [\neg\text{XXXX} (\text{Cliente} \sqsubseteq \exists \text{devuelve. Entidad})]$: El cliente Santiago no devuelve la entidad Libro3. ...	
Caso de éxito 2 (A_{10}, A_{11}, A_{12}):	
$A_{10}: [\text{X}(\text{Empleado} \sqsubseteq \exists \text{registraUsode. Entidad})]$: El empleado Valentino registra el uso de la entidad Libro4. ...	
$A_{11}: [\text{XX} (\text{Cliente} \sqsubseteq \exists \text{usa. Entidad})]$: El cliente Violeta usa la entidad Libro4. ...	
$A_{12}: [\text{XXXX} (\text{Cliente} \sqsubseteq \exists \text{devuelve. Entidad})]$: El cliente Violeta devuelve la entidad Libro4. ...	
Caso de éxito n	
A_n ...	

Tabla D.12 Reglas SWRL para la SoIs de casos de éxito y fracaso de acuerdo al RF2. Usar una entidad reservada (cont.)

ABox	Regla SWRL
A_2	$Event(?e), \text{during}(?e, ?i), \text{overlaps}(?i, \text{Req2-DT12}), \text{usa}(?e, \text{Libro1}), \text{usa}(\text{Heidi}, ?e) \rightarrow \text{Nothing}(\text{Heidi})$
A_3	$Event(?e), \text{during}(?e, ?i), \text{overlaps}(?i, \text{Req2-DT13}), \text{devuelve}(?e, \text{Libro1}), \text{devuelve}(\text{Heidi}, ?e) \rightarrow \text{Nothing}(\text{Heidi})$
A_8	$Event(?e), \text{during}(?e, ?i), \text{overlaps}(?i, \text{Req2-DT18}), \text{usa}(?e, \text{Libro3}), \text{usa}(\text{Santiago}, ?e) \rightarrow \text{Nothing}(\text{Santiago})$
A_9	$Event(?e), \text{during}(?e, ?i), \text{overlaps}(?i, \text{Req2-DT19}), \text{devuelve}(?e, \text{Libro3}), \text{devuelve}(\text{Santiago}, ?e) \rightarrow \text{Nothing}(\text{Santiago})$

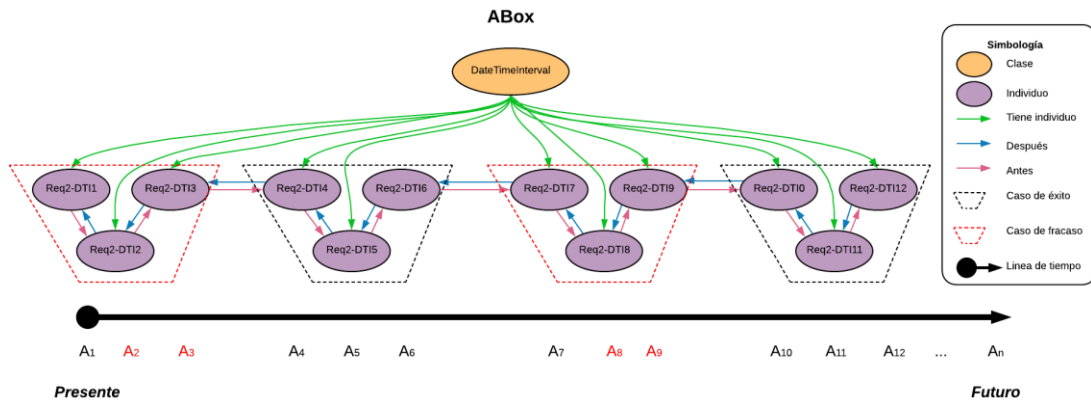


Figura D.8 SoIs en la ABox de casos de éxito y fracaso de acuerdo al RF2. Usar una entidad reservada.

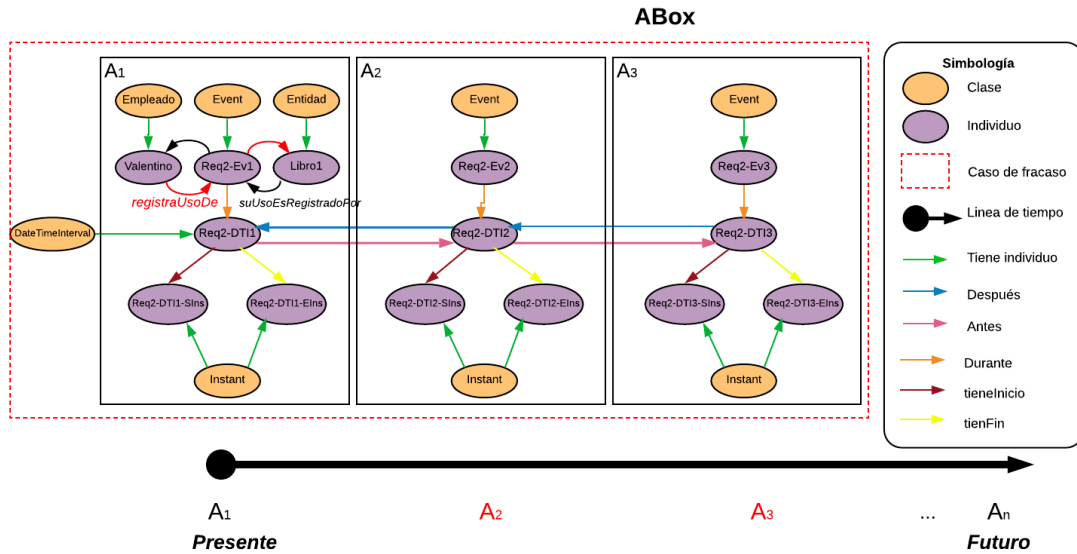


Figura D.9 Caso de fracaso 1 en la ABox de casos de éxito y fracaso de acuerdo al RF2. Usar una entidad reservada.

1.2.3 RF3. Modificar una reservación

Las secuencias de interpretaciones descritas en esta sección satisfacen/cumplen con el comportamiento dinámico del RF3. Modificar una reservación representado formalmente con la fórmula de la Tabla D.13.

Tabla D.13 Fórmula ALC_{LTL} del RF3. Modificar una reservación.

$$\begin{aligned}
 & \mathcal{G} (\\
 & \quad \mathcal{X} (\text{Cliente} \sqsubseteq \exists \text{modifica.Reservacion}) \rightarrow (\\
 & \quad \quad \mathcal{XX} (\text{Empleado} \sqsubseteq \exists \text{verificaDisponibilidadDe.Entidad}) \wedge \mathcal{F} (\\
 & \quad \quad \quad \neg \mathcal{XXX} (\text{Empleado} \sqsubseteq \forall \text{cancela.Reservacion} \wedge \text{Empleado} \sqsubseteq \forall \text{actualiza.Reservacion}) \\
 & \quad \quad \quad \neg \mathcal{XXX} (\text{Empleado} \sqsubseteq \exists \text{actualiza.Entidad}) \\
 & \quad \quad) \\
 & \quad) \\
 &)
 \end{aligned}$$

1.2.3.1. Casos de éxito

En la Tabla D.14 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la ABox) basadas en casos de éxito, que conforman una secuencia que representa el comportamiento dinámico para modificar una reservación. Una representación gráfica simplificada de la secuencia se muestra en la Figura D.10. A manera de ejemplo, en la Figura D.11 se muestra el detalle de las aserciones A_1 , A_2 , A_3 y A_4 en el caso de éxito número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de

razonamiento automatizado, los demás casos se definen de manera similar, sin embargo, difieren en los individuos involucrados.

Tabla D.14 ABox de casos de éxito en el RF3. Modificar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
Caso de éxito 1 (A₁, A₂, A₃, A₄):	
A ₁ : [X (Cliente \sqsubseteq \exists modifica.Reservacion)]: El cliente Heidi modifica la reservación Reservacion_prestamo_Heidi .	
A ₁ = {	
-- Aserciones de clase --	
Cliente(Heidi)	⇒ ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Heidi)
Reservacion(Reservacion_prestamo_Heidi)	⇒ ClassAssertion(domain-PAIyER:Reservacion domain-PAIyER: Reservacion_prestamo_Heidi)
Event(Req3-Ev1)	⇒ ClassAssertion(timeEvents:Event domain-PAIyER:Req3-Ev1)
DateTimeInterval(Req3-DT11)	⇒ ClassAssertion(time:DateTimeInterval domain-PAIyER:Req3-DT11)
Instant(Req3-DT11-Sins)	⇒ ClassAssertion(time:Instant domain-PAIyER:Req3-DT11-Sins)
Instant(Req3-DT11-Eins)	⇒ ClassAssertion(time:Instant domain-PAIyER:Req3-DT11-Eins)
-- Aserciones de propiedades de objeto --	
modifica(Heidi,Req3-Ev1)	⇒ ObjectPropertyAssertion(domain-PAIyER:modifica domain-PAIyER:Heidi domain-PAIyER:Req3-Ev1)
modifica(Req3-Ev1, Reservacion_prestamo_Heidi)	⇒ ObjectPropertyAssertion(domain-PAIyER:modifica domain-PAIyER:Req3-Ev1 domain-PAIyER: Reservacion_prestamo_Heidi)
during(Req3-Ev1,Req3-DT11)	⇒ ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req3-Ev1 domain-PAIyER:Req3-DT11)
hasBeginning(Req3-DT11,Req3-DT11-Sins)	⇒ ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req3-DT11 domain-PAIyER:Req3-DT11-Sins)
hasEnd(Req3-DT11,Req3-DT11-Eins)	⇒ ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req3-DT11 domain-PAIyER:Req3-DT11-Eins)
}	⇒
A ₂ : [XX (Empleado \sqsubseteq \exists verificaDisponibilidadDe.Entidad)]: El empleado Valentino verifica la disponibilidad de la Entidad Libro1	
A ₂ = {	
-- Aserciones de clase --	
Empleado(Valentino)	⇒ ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)
Entidad(Libro1)	⇒ ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro1)
Event(Req3-Ev2)	⇒ ClassAssertion(timeEvents:Event domain-PAIyER:Req3-Ev2)
DateTimeInterval(Req3-DT12)	⇒ ClassAssertion(time:DateTimeInterval domain-PAIyER:Req3-DT12)
Instant(Req3-DT12-Sins)	⇒ ClassAssertion(time:Instant domain-PAIyER:Req3-DT12-Sins)
Instant(Req3-DT12-Eins)	⇒ ClassAssertion(time:Instant domain-PAIyER:Req3-DT12-Eins)

Tabla D.14 ABox de casos de éxito en el RF3. Modificar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>-- Aserciones de propiedades de objeto --</i>	
<i>verificaDisponibilidadDe(Valentino,Req3-Ev2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaDisponibilidadDe domain-PAIyER:Valentino domain-PAIyER:Req3-Ev2)</i>
<i>verificaDisponibilidadDe (Req3-Ev2,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaDisponibilidadDe domain-PAIyER:Req3-Ev2 domain-PAIyER:Libro1)</i>
<i>during(Req3-Ev2,Req3-DTI2)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req3-Ev2 domain-PAIyER:Req3-DTI2)</i>
<i>hasBeginning(Req3-DTI2,Req3-DTI2-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req3-DTI2 domain-PAIyER:Req3-DTI2-Sins)</i>
<i>hasEnd(Req3-DTI2,Req3-DTI2-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req3-DTI2 domain-PAIyER:Req3-DTI2-Eins)</i>
}	
<i>A₃: [XXX(Empleado \sqsubseteq \forallcancela.Reservacion \wedge Empleado \sqsubseteq \existsactualiza.Reservacion)]: El empleado Valentino cancela la reservación Reservacion_prestamo_Heidi y actualiza la reservación Reservacion_prestamo_Heidi.</i>	
A ₃ = {	
<i>-- Aserciones de clase --</i>	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Reservacion(Reservacion_prestamo_Heidi)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Reservacion domain-PAIyER:Reservacion_prestamo_Heidi)</i>
<i>Event(Req3-Ev3_1)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req3-Ev3_1)</i>
<i>DateTimeInterval(Req3-DTI3_1)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req3-DTI3_1)</i>
<i>Instant(Req3-DTI3_1-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req3-DTI3_1-Sins)</i>
<i>Instant(Req3-DTI3_1-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req3-DTI3_1-Eins)</i>
<i>Event(Req3-Ev3_2)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req3-Ev3_2)</i>
<i>DateTimeInterval(Req3-DTI3_2)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req3-DTI3_2)</i>
<i>Instant(Req3-DTI3_2-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req3-DTI3_2-Sins)</i>
<i>Instant(Req3-DTI3_2-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req3-DTI3_2-Eins)</i>
<i>-- Aserciones de propiedades de objeto --</i>	
<i>cancela(Valentino,Req3-Ev3_1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:cancela domain-PAIyER:Valentino domain-PAIyER:Req3-Ev3_1)</i>
<i>cancela(Req3-Ev3_1,Reservacion_prestamo_Heidi)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:cancela domain-PAIyER:Req3-Ev3_1 domain-PAIyER:Reservacion_prestamo_Heidi)</i>
<i>during(Req3-Ev3_1,Req3-DTI3_1)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req3-Ev3_1 domain-PAIyER:Req3-DTI3_1)</i>
<i>hasBeginning(Req3-DTI3_1,Req3-DTI3_1-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req3-DTI3_1 domain-</i>

Tabla D.14 ABox de casos de éxito en el RF3. Modificar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>hasEnd(Req3-DTI3_1,Req3-DTI3_1-Eins)</i>	\Rightarrow <i>PAIyER:Req3-DTI3_1-Sins</i> <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req3-DTI3_1 domain-PAIyER:Req3-DTI3_1-Eins)</i>
<i>actualiza(Valentino,Req3-Ev3_2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Valentino domain-PAIyER:Req3-Ev3_2)</i>
<i>actualiza(Req3-Ev3_2,Reservacion_prestamo_Heidi)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Req3-Ev3_2 domain-PAIyER:Reservacion_prestamo_Heidi)</i>
<i>during(Req3-Ev3_2,Req3-DTI3_2)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req3-Ev3_2 domain-PAIyER:Req3-DTI3_2)</i>
<i>hasBeginning(Req3-DTI3_2,Req3-DTI3_2-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req3-DTI3_2 domain-PAIyER:Req3-DTI3_2-Sins)</i>
<i>hasEnd(Req3-DTI3_2,Req3-DTI3_2-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req3-DTI3_2 domain-PAIyER:Req3-DTI3_2-Eins)</i>
}	
A4: [XXXX (Empleado \sqsubseteq \exists actualiza.Entidad)]: El empleado Valentino actualiza la entidad Libro1.	
A4 = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro1)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro1)</i>
<i>Event(Req3-Ev4)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req3-Ev4)</i>
<i>DateTimeInterval(Req3-DTI4)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req3-DTI4)</i>
<i>Instant(Req3-DTI4-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req3-DTI4-Sins)</i>
<i>Instant(Req3-DTI4-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req3-DTI4-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>actualiza(Valentino,Req3-Ev4)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Valentino domain-PAIyER:Req3-Ev4)</i>
<i>actualiza(Req3-Ev4, Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Req3-Ev4 domain-PAIyER:Libro1)</i>
<i>during(Req3-Ev4,Req3-DTI4)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req3-Ev4 domain-PAIyER:Req3-DTI4)</i>
<i>hasBeginning(Req3-DTI4,Req3-DTI4-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req3-DTI4 domain-PAIyER:Req3-DTI4-Sins)</i>
<i>hasEnd(Req3-DTI4,Req3-DTI4-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req3-DTI4 domain-PAIyER:Req3-DTI4-Eins)</i>
}	
Caso de éxito 2 (A5, A6, A7, A8):	
A5: [X(Cliente \sqsubseteq \exists modifica.Reservacion)]: El cliente Juan_Carlos modifica la reservacion	

Tabla D.14 ABox de casos de éxito en el RF3. Modificar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Reservacion_prestamo_JuanCarlos.</i>	
$A_5 = \{$	
-- Aserciones de clase --	
<i>Cliente(Juan_Carlos)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Cliente domain-PAIyER: Juan_Carlos)</i>
<i>Reservacion(Reservacion_prestamo_JuanCarlos)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Reservacion domain-PAIyER: Reservacion_prestamo_JuanCarlos)</i>
<i>Event(Req3-Ev5)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req3-Ev5)</i>
<i>DateTimeInterval(Req3-DTI5)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req3-DTI5)</i>
<i>Instant(Req3-DTI5-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req3-DTI5-Sins)</i>
<i>Instant(Req3-DTI5-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req3-DTI5-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>modifica(Juan_Carlos,Req3-Ev5)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:modifica domain-PAIyER: Juan_Carlos domain-PAIyER:Req3-Ev5)</i>
<i>modifica(Req3-Ev5, Reservacion_prestamo_JuanCarlos)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:modifica domain-PAIyER:Req3-Ev5 domain-PAIyER: Reservacion_prestamo_JuanCarlos)</i>
<i>during(Req3-Ev5,Req3-DTI5)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req3-Ev5 domain-PAIyER:Req3-DTI5)</i>
<i>hasBeginning(Req3-DTI5,Req3-DTI5-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req3-DTI5 domain-PAIyER:Req3-DTI5-Sins)</i>
<i>hasEnd(Req3-DTI5,Req3-DTI5-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req3-DTI5 domain-PAIyER:Req3-DTI5-Eins)</i>
$\}$	\Rightarrow
A₆: [XX (Empleado \sqsubseteq \existsverificaDisponibilidadDe.Entidad)]: El empleado Valentino verifica la disponibilidad de la Entidad Libro2	
$A_6 = \{$	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event(Req3-Ev6)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req3-Ev6)</i>
<i>DateTimeInterval(Req3-DTI6)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req3-DTI6)</i>
<i>Instant(Req3-DTI6-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req3-DTI6-Sins)</i>
<i>Instant(Req3-DTI6-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req3-DTI6-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>verificaDisponibilidadDe(Valentino,Req3-Ev6)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaDisponibilidadDe domain-PAIyER:Valentino domain-PAIyER:Req3-Ev6)</i>

Tabla D.14 ABox de casos de éxito en el RF3. Modificar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>verificaDisponibilidadDe (Req3-Ev6,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaDisponibilidadDe domain-PAIyER:Req3-Ev6 domain-PAIyER:Libro2)</i>
<i>during(Req3-Ev6,Req3-DTI6)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req3-Ev6 domain-PAIyER:Req3-DTI6)</i>
<i>hasBeginning(Req3-DTI6,Req3-DTI6-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req3-DTI6 domain-PAIyER:Req3-DTI6-Sins)</i>
<i>hasEnd(Req3-DTI6,Req3-DTI6-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req3-DTI6 domain-PAIyER:Req3-DTI6-Eins)</i>
}	
A7: [XXX(Empleado \sqsubseteq \exists cancela.Reservacion \wedge Empleado \sqsubseteq \exists \forall actualiza.Reservacion)]: El empleado Valentino cancela la reservación Reservacion_prestamo_JuanCarlos y actualiza la reservación Reservacion_prestamo_JuanCarlos.	
A7 = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Reservacion(Reservacion_prestamo_JuanCarlos)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Resrvacion domain-PAIyER:Reservacion_prestamo_JuanCarlos)</i>
<i>Event(Req3-Ev7_1)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req3-Ev7_1)</i>
<i>DateTimeInterval(Req3-DTI7_1)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req3-DTI7_1)</i>
<i>Instant(Req3-DTI7_1-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req3-DTI7_1-Sins)</i>
<i>Instant(Req3-DTI7_1-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req3-DTI7_1-Eins)</i>
<i>Event(Req3-Ev7_2)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req3-Ev7_2)</i>
<i>DateTimeInterval(Req3-DTI7_2)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req3-DTI7_2)</i>
<i>Instant(Req3-DTI7_2-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req3-DTI7_2-Sins)</i>
<i>Instant(Req3-DTI7_2-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req3-DTI7_2-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>cancela(Valentino,Req3-Ev7_1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:cancela domain-PAIyER:Valentino domain-PAIyER:Req3-Ev7_1)</i>
<i>c(Req3-Ev7_1,Reservacion_prestamo_JuanCarlos)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:crea domain-PAIyER:Req3-Ev7_1 domain-PAIyER:Reservacion_prestamo_JuanCarlos)</i>
<i>during(Req3-Ev7_1,Req3-DTI7_1)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req3-Ev7_1 domain-PAIyER:Req3-DTI7_1)</i>
<i>hasBeginning(Req3-DTI7_1,Req3-DTI7_1-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req3-DTI7_1 domain-PAIyER:Req3-DTI7_1-Sins)</i>
<i>hasEnd(Req3-DTI7_1,Req3-DTI7_1-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req3-DTI7_1 domain-PAIyER:Req3-</i>

Tabla D.14 ABox de casos de éxito en el RF3. Modificar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>actualiza(Valentino,Req3-Ev7_2)</i>	<i>DTI7_1-Eins)</i> ⇒ <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Valentino domain-PAIyER:Req3-Ev7_2)</i>
<i>actualiza(Req3-Ev7_2,Reservacion_prestamo_JuanCarlos)</i>	⇒ <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Req3-Ev7_2 domain-PAIyER:Reservacion_prestamo_JuanCarlos)</i>
<i>during(Req3-Ev7_2,Req3-DTI7_2)</i>	⇒ <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req3-Ev7_2 domain-PAIyER:Req3-DTI7_2)</i>
<i>hasBeginning(Req3-DTI7_2,Req3-DTI7_2-Sins)</i>	⇒ <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req3-DTI7_2 domain-PAIyER:Req3-DTI7_2-Sins)</i>
<i>hasEnd(Req3-DTI7_2,Req3-DTI7_2-Eins)</i>	⇒ <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req3-DTI7_2 domain-PAIyER:Req3-DTI7_2-Eins)</i>
<i>}</i>	
<i>A₈: [XXXX (Empleado \sqsubseteq \existsactualiza.Entidad)]: El empleado Valentino actualiza la entidad Libro2.</i>	
<i>A₈ = {</i>	
<i>-- Aserciones de clase --</i>	
<i>Empleado(Valentino)</i>	⇒ <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro2)</i>	⇒ <i>ClassAssertion(domain-PAIyER:Contrato domain-PAIyER: Libro2)</i>
<i>Event(Req3-Ev8)</i>	⇒ <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req3-Ev8)</i>
<i>DateTimeInterval(Req3-DTI8)</i>	⇒ <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req3-DTI8)</i>
<i>Instant(Req3-DTI8-Sins)</i>	⇒ <i>ClassAssertion(time:Instant domain-PAIyER:Req3-DTI8-Sins)</i>
<i>Instant(Req3-DTI8-Eins)</i>	⇒ <i>ClassAssertion(time:Instant domain-PAIyER:Req3-DTI8-Eins)</i>
<i>-- Aserciones de propiedades de objeto --</i>	
<i>actualiza(Valentino,Req3-Ev8)</i>	⇒ <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Valentino domain-PAIyER:Req3-Ev8)</i>
<i>actualiza(Req3-Ev8, Libro2)</i>	⇒ <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Req3-Ev8 domain-PAIyER: Libro2)</i>
<i>during(Req3-Ev8,Req3-DTI8)</i>	⇒ <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req3-Ev8 domain-PAIyER:Req3-DTI8)</i>
<i>hasBeginning(Req3-DTI8,Req3-DTI8-Sins)</i>	⇒ <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req3-DTI8 domain-PAIyER:Req3-DTI8-Sins)</i>
<i>hasEnd(Req3-DTI8,Req3-DTI8-Eins)</i>	⇒ <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req3-DTI8 domain-PAIyER:Req3-DTI8-Eins)</i>
<i>}</i>	
Caso de éxito 3 (A₉, A₁₀, A₁₁, A₁₂):	
<i>A₉: [X(Cliente \sqsubseteq \existsmodifica.Reservacion)]: El cliente Santiago modifica la reservacion Reservacion_prestamo_Santiago.</i>	
...	

Tabla D.14 ABox de casos de éxito en el RF3. Modificar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<p>A_{10}: [XX (Empleado $\sqsubseteq \exists$verificaDisponibilidadDe.Entidad)]: El empleado Valentino verifica la disponibilidad de la Entidad Libro3.</p> <p>...</p> <p>A_{11}: [XXX(Empleado $\sqsubseteq \exists$cancela.Reservacion \wedge Empleado $\sqsubseteq \exists \forall$actualiza.Reservacion)]: El empleado Valentino cancela la reservación Reservacion_prestamo_Santiago y actualiza la reservación Reservacion_prestamo_Santiago.</p> <p>...</p> <p>A_{12}: [XXXX (Empleado $\sqsubseteq \exists$actualiza.Entidad)]: El empleado Valentino actualiza la entidad Libro3.</p> <p>...</p>	
<p>Caso de éxito 4 (A_{13}, A_{14}, A_{15}, A_{16}):</p> <p>A_{13}: [X(Cliente $\sqsubseteq \exists$modifica.Reservacion)]: El cliente Violeta modifica la reservacion Reservacion_prestamo_Violeta.</p> <p>...</p> <p>A_{14}: [XX (Empleado $\sqsubseteq \exists$verificaDisponibilidadDe.Entidad)]: El empleado Valentino verifica la disponibilidad de la Entidad Libro4.</p> <p>...</p> <p>A_{15}: A_{11}: [XXX(Empleado $\sqsubseteq \exists$cancela.Reservacion \wedge Empleado $\sqsubseteq \exists \forall$actualiza.Reservacion)]: El empleado Valentino cancela la reservación Reservacion_prestamo_Violeta y actualiza la reservación Reservacion_prestamo_Violeta.</p> <p>...</p> <p>A_{16}: [XXXX (Empleado $\sqsubseteq \exists$actualiza.Entidad)]: El empleado Valentino actualiza la entidad Libro4.</p> <p>...</p>	
<p>Caso de éxito n</p> <p>A_n</p> <p>...</p>	

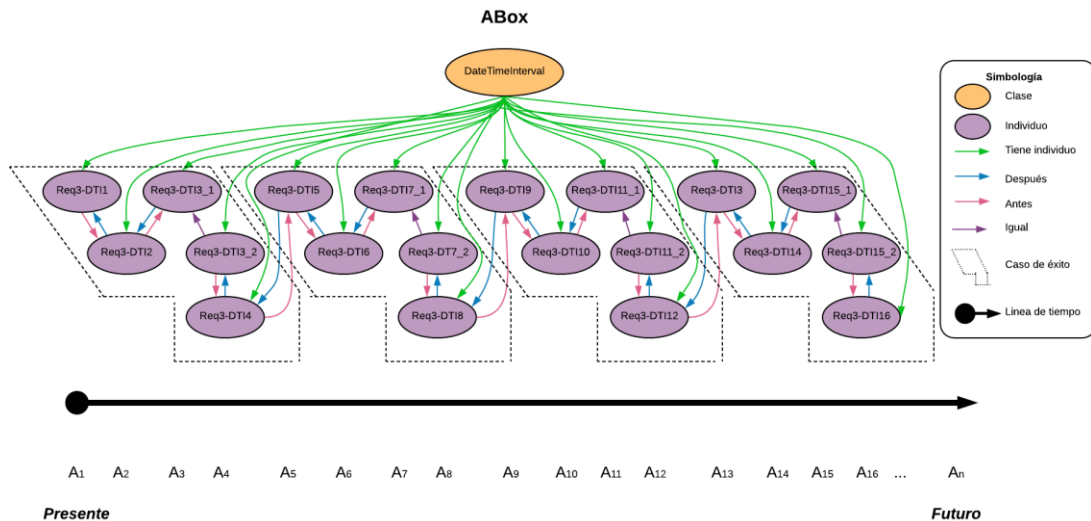


Figura D.10 SoIs en la ABox de casos de éxito de acuerdo al RF3. Modificar una reservación.

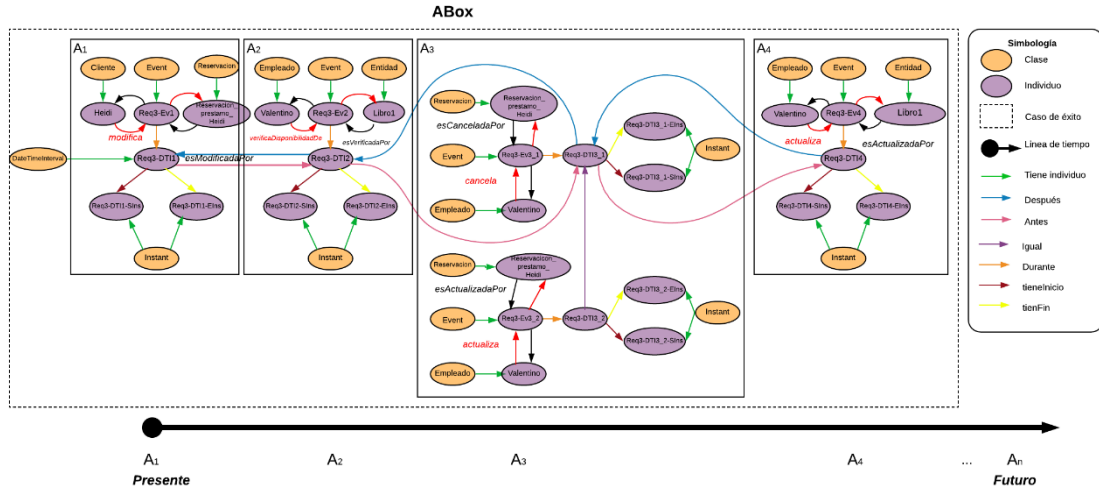


Figura D.11 Caso de éxito 1 en la ABox de casos de éxito de acuerdo al RF3. Modificar una reservación.

1.2.3.2. Casos de éxito y fracaso

En la Tabla D.15 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la ABox) basadas en casos de éxito y fracaso, que conforman una secuencia que representa el comportamiento dinámico para modificar una reservación. Una representación gráfica simplificada de la secuencia se muestra en la Figura D.12. A manera de ejemplo, en la Figura D.13 se muestra el detalle de las aserciones A_1 , A_2 , A_3 y A_4 en el caso de fracaso número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de razonamiento automatizado. Los demás casos se definen de manera similar, sin embargo, difieren en los individuos involucrados.

En la Tabla D.16 se describen las reglas SWRL utilizadas para restringir la existencia de afirmaciones de propiedades de objeto en el intervalo de tiempo correspondiente.

Tabla D.15 ABox de casos de éxito y fracaso en el RF3. Modificar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
Caso de fracaso 1 (A_1, A_2, A_3, A_4):	
A_1 : $[X(\text{Cliente} \sqsubseteq \exists \text{modifica.Reservacion})]$: El cliente Heidi modifica la reservación <i>Reservacion_prestamo_Heidi</i> .	
$A_1 = \{$	
-- Aserciones de clase --	
<i>Cliente</i> (Heidi)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Cliente domain-PAIyER:Heidi)
<i>Reservacion</i> (<i>Reservacion_prestamo_Heidi</i>)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Reservacion domain-PAIyER:Reservacion_prestamo_Heidi)
<i>Event</i> (<i>Req3-Ev1</i>)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req3-Ev1)
<i>DateTimeInterval</i> (<i>Req3-DT11</i>)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req3-DT11)
<i>Instant</i> (<i>Req3-DT11-Sins</i>)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req3-DT11-Sins)

Tabla D.15 ABox de casos de éxito y fracaso en el RF3. Modificar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Instant(Req3-DT11-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req3-DT11-Eins)</i>
-- Aserciones de propiedades de objeto --	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:modifica domain-PAIyER:Heidi domain-PAIyER:Req3-Ev1)</i>
<i>modifica(Heidi,Req3-Ev1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:modifica domain-PAIyER:Req3-Ev1 domain-PAIyER:Reservacion_prestamo_Heidi)</i>
<i>modifica(Req3-Ev1,Reservacion_prestamo_Heidi)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req3-Ev1 domain-PAIyER:Req3-DT11)</i>
<i>during(Req3-Ev1,Req3-DT11)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req3-DT11 domain-PAIyER:Req3-DT11-Sins)</i>
<i>hasBeginning(Req3-DT11,Req3-DT11-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req3-DT11 domain-PAIyER:Req3-DT11-Eins)</i>
<i>hasEnd(Req3-DT11,Req3-DT11-Eins)</i>	\Rightarrow
}	
<i>A₂: [XX (Empleado \sqsubseteq \existsverificaDisponibilidadDe.Entidad)]: El empleado Valentino verifica la disponibilidad de la Entidad Libro1</i>	
<i>A₂ = {</i>	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro1)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro1)</i>
<i>Event(Req3-Ev2)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req3-Ev2)</i>
<i>DateTimeInterval(Req3-DT12)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req3-DT12)</i>
<i>Instant(Req3-DT12-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req3-DT12-Sins)</i>
<i>Instant(Req3-DT12-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req3-DT12-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>verificaDisponibilidadDe(Valentino,Req3-Ev2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaDisponibilidadDe domain-PAIyER:Valentino domain-PAIyER:Req3-Ev2)</i>
<i>verificaDisponibilidadDe (Req3-Ev2,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaDisponibilidadDe domain-PAIyER:Req3-Ev2 domain-PAIyER:Libro1)</i>
<i>during(Req3-Ev2,Req3-DT12)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req3-Ev2 domain-PAIyER:Req3-DT12)</i>
<i>hasBeginning(Req3-DT12,Req3-DT12-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req3-DT12 domain-PAIyER:Req3-DT12-Sins)</i>
<i>hasEnd(Req3-DT12,Req3-DT12-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req3-DT12 domain-PAIyER:Req3-DT12-Eins)</i>
}	
<i>A₃: [\negXXX (Empleado \sqsubseteq \forallcancela.Reservacion \wedge Empleado \sqsubseteq \existsactualiza.Reservacion)]: El empleado Valentino no cancela la reservación Reservacion_prestamo_Heidi ni actualiza la reservación</i>	

Tabla D.15 ABox de casos de éxito y fracaso en el RF3. Modificar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Reservacion_prestamo_Heidi.</i>	
$A_3 = \{$	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Reservacion</i> (<i>Reservacion_prestamo_Heidi</i>)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Reservacion domain-PAIyER:Reservacion_prestamo_Heidi)
<i>Event</i> (Req3-Ev3_1)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req3-Ev3_1)
<i>DateTimeInterval</i> (Req3-DTI3_1)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req3-DTI3_1)
<i>Instant</i> (Req3-DTI3_1-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req3-DTI3_1-Sins)
<i>Instant</i> (Req3-DTI3_1-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req3-DTI3_1-Eins)
<i>Event</i> (Req3-Ev3_2)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req3-Ev3_2)
<i>DateTimeInterval</i> (Req3-DTI3_2)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req3-DTI3_2)
<i>Instant</i> (Req3-DTI3_2-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req3-DTI3_2-Sins)
<i>Instant</i> (Req3-DTI3_2-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req3-DTI3_2-Eins)
-- Aserciones de propiedades de objeto --	
<i>cancela</i> (Valentino,Req3-Ev3_1)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:cancela domain-PAIyER:Valentino domain-PAIyER:Req3-Ev3_1)
<i>cancela</i> (Req3-Ev3_1,Reservacion_prestamo_Heidi)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:cancela domain-PAIyER:Req3-Ev3_1 domain-PAIyER:Reservacion_prestamo_Heidi)
<i>during</i> (Req3-Ev3_1,Req3-DTI3_1)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req3-Ev3_1 domain-PAIyER:Req3-DTI3_1)
<i>hasBeginning</i> (Req3-DTI3_1,Req3-DTI3_1-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req3-DTI3_1 domain-PAIyER:Req3-DTI3_1-Sins)
<i>hasEnd</i> (Req3-DTI3_1,Req3-DTI3_1-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req3-DTI3_1 domain-PAIyER:Req3-DTI3_1-Eins)
<i>actualiza</i> (Valentino,Req3-Ev3_2)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:actualiza domain-PAIyER:Valentino domain-PAIyER:Req3-Ev3_2)
<i>actualiza</i> (Req3-Ev3_2,Reservacion_prestamo_Heidi)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:actualiza domain-PAIyER:Req3-Ev3_2 domain-PAIyER:Reservacion_prestamo_Heidi)
<i>during</i> (Req3-Ev3_2,Req3-DTI3_2)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req3-Ev3_2 domain-PAIyER:Req3-DTI3_2)
<i>hasBeginning</i> (Req3-DTI3_2,Req3-DTI3_2-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req3-DTI3_2 domain-PAIyER:Req3-DTI3_2-Sins)
<i>hasEnd</i> (Req3-DTI3_2,Req3-DTI3_2-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req3-DTI3_2 domain-PAIyER:Req3-DTI3_2-Eins)

Tabla D.15 ABox de casos de éxito y fracaso en el RF3. Modificar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<p>} A₄: [XXXX (Empleado \sqsubseteq \existsactualiza.Entidad)]: El empleado Valentino no actualiza la entidad Libro1. A₄ = { -- Aserciones de clase -- Empleado(Valentino) Entidad(Libro1) Event(Req3-Ev4) DateTimeInterval(Req3-DTI4) Instant(Req3-DTI4-Sins) Instant(Req3-DTI4-Eins) -- Aserciones de propiedades de objeto -- actualiza(Valentino,Req3-Ev4) actualiza(Req3-Ev4, Libro1) during(Req3-Ev4,Req3-DTI4) hasBeginning(Req3-DTI4,Req3-DTI4-Sins) hasEnd(Req3-DTI4,Req3-DTI4-Eins) }</p>	<p>\Rightarrow ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino) \Rightarrow ClassAssertion(domain-PAIyER:Entidad domain-PAIyER: Libro1) \Rightarrow ClassAssertion(timeEvents:Event domain-PAIyER:Req3-Ev4) \Rightarrow ClassAssertion(time:DateTimeInterval domain-PAIyER:Req3-DTI4) \Rightarrow ClassAssertion(time:Instant domain-PAIyER:Req3-DTI4-Sins) \Rightarrow ClassAssertion(time:Instant domain-PAIyER:Req3-DTI4-Eins) \Rightarrow ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Valentino domain-PAIyER:Req3-Ev4) \Rightarrow ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Req3-Ev4 domain-PAIyER: Libro1) \Rightarrow ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req3-Ev4 domain-PAIyER:Req3-DTI4) \Rightarrow ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req3-DTI4 domain-PAIyER:Req3-DTI4-Sins) \Rightarrow ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req3-DTI4 domain-PAIyER:Req3-DTI4-Eins)</p>
<p>Caso de éxito 2 (A₅, A₆, A₇, A₈): A₅: [X(Cliente \sqsubseteq \existsmodifica.Reservacion)]: El cliente Juan_Carlos modifica la reservacion Reservacion_prestamo_JuanCarlos. A₅ = { -- Aserciones de clase -- Cliente(Juan_Carlos) Reservacion(Reservacion_prestamo_JuanCarlos) Event(Req3-Ev5) DateTimeInterval(Req3-DTI5) Instant(Req3-DTI5-Sins) Instant(Req3-DTI5-Eins) -- Aserciones de propiedades de objeto -- modifica(Juan_Carlos,Req3-Ev5)</p>	<p>\Rightarrow ClassAssertion(domain-PAIyER:Cliente domain-PAIyER: Juan_Carlos) \Rightarrow ClassAssertion(domain-PAIyER:Reservacion domain-PAIyER:Reservacion_prestamo_JuanCarlos) \Rightarrow ClassAssertion(timeEvents:Event domain-PAIyER:Req3-Ev5) \Rightarrow ClassAssertion(time:DateTimeInterval domain-PAIyER:Req3-DTI5) \Rightarrow ClassAssertion(time:Instant domain-PAIyER:Req3-DTI5-Sins) \Rightarrow ClassAssertion(time:Instant domain-PAIyER:Req3-DTI5-Eins) \Rightarrow ObjectPropertyAssertion(domain-PAIyER:modifica domain-PAIyER: Juan_Carlos)</p>

Tabla D.15 ABox de casos de éxito y fracaso en el RF3. Modificar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>modifica</i> (Req3-Ev5, <i>Reservacion_prestamo_JuanCarlos</i>)	\Rightarrow <i>domain-PAIyER:Req3-Ev5</i> \Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:modifica domain-PAIyER:Req3-Ev5 domain-PAIyER:Reservacion_prestamo_JuanCarlos)</i>
<i>during</i> (Req3-Ev5,Req3-DTI5)	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req3-Ev5 domain-PAIyER:Req3-DTI5)</i>
<i>hasBeginning</i> (Req3-DTI5,Req3-DTI5-Sins)	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req3-DTI5 domain-PAIyER:Req3-DTI5-Sins)</i>
<i>hasEnd</i> (Req3-DTI5,Req3-DTI5-Eins)	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req3-DTI5 domain-PAIyER:Req3-DTI5-Eins)</i>
}	\Rightarrow
A₆ : [XX (Empleado \sqsubseteq \exists verificaDisponibilidadDe.Entidad)]: El empleado Valentino verifica la disponibilidad de la Entidad Libro2	
A ₆ = {	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad</i> (Libro2)	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event</i> (Req3-Ev6)	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req3-Ev6)</i>
<i>DateTimeInterval</i> (Req3-DTI6)	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req3-DTI6)</i>
<i>Instant</i> (Req3-DTI6-Sins)	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req3-DTI6-Sins)</i>
<i>Instant</i> (Req3-DTI6-Eins)	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req3-DTI6-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>verificaDisponibilidadDe</i> (Valentino,Req3-Ev6)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaDisponibilidadDe domain-PAIyER:Valentino domain-PAIyER:Req3-Ev6)</i>
<i>verificaDisponibilidadDe</i> (Req3-Ev6,Libro2)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaDisponibilidadDe domain-PAIyER:Req3-Ev6 domain-PAIyER:Libro2)</i>
<i>during</i> (Req3-Ev6,Req3-DTI6)	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req3-Ev6 domain-PAIyER:Req3-DTI6)</i>
<i>hasBeginning</i> (Req3-DTI6,Req3-DTI6-Sins)	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req3-DTI6 domain-PAIyER:Req3-DTI6-Sins)</i>
<i>hasEnd</i> (Req3-DTI6,Req3-DTI6-Eins)	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req3-DTI6 domain-PAIyER:Req3-DTI6-Eins)</i>
}	
A₇ : [XXX(Empleado \sqsubseteq \exists cancela.Reservacion \wedge Empleado \sqsubseteq \exists actualiza.Reservacion)]: El empleado Valentino cancela la reservación <i>Reservacion_prestamo_JuanCarlos</i> y actualiza la reservación <i>Reservacion_prestamo_JuanCarlos</i> .	
A ₇ = {	
-- Aserciones de clase --	

Tabla D.15 ABox de casos de éxito y fracaso en el RF3. Modificar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Empleado</i> (Valentino)	⇒ <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Reservacion</i> (Reservacion_prestamo_JuanCarlos)	⇒ <i>ClassAssertion</i> (domain-PAIyER:Reservacion domain-PAIyER:Reservacion_prestamo_JuanCarlos)
<i>Event</i> (Req3-Ev7_1)	⇒ <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req3-Ev7_1)
<i>DateTimeInterval</i> (Req3-DTI7_1)	⇒ <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req3-DTI7_1)
<i>Instant</i> (Req3-DTI7_1-Sins)	⇒ <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req3-DTI7_1-Sins)
<i>Instant</i> (Req3-DTI7_1-Eins)	⇒ <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req3-DTI7_1-Eins)
<i>Event</i> (Req3-Ev7_2)	⇒ <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req3-Ev7_2)
<i>DateTimeInterval</i> (Req3-DTI7_2)	⇒ <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req3-DTI7_2)
<i>Instant</i> (Req3-DTI7_2-Sins)	⇒ <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req3-DTI7_2-Sins)
<i>Instant</i> (Req3-DTI7_2-Eins)	⇒ <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req3-DTI7_2-Eins)
-- Aserciones de propiedades de objeto --	
<i>cancela</i> (Valentino,Req3-Ev7_1)	⇒ <i>ObjectPropertyAssertion</i> (domain-PAIyER:cancela domain-PAIyER:Valentino domain-PAIyER:Req3-Ev7_1)
<i>c</i> (Req3-Ev7_1,Reservacion_prestamo_JuanCarlos)	⇒ <i>ObjectPropertyAssertion</i> (domain-PAIyER:crea domain-PAIyER:Req3-Ev7_1 domain-PAIyER:Reservacion_prestamo_JuanCarlos)
<i>during</i> (Req3-Ev7_1,Req3-DTI7_1)	⇒ <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req3-Ev7_1 domain-PAIyER:Req3-DTI7_1)
<i>hasBeginning</i> (Req3-DTI7_1,Req3-DTI7_1-Sins)	⇒ <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req3-DTI7_1 domain-PAIyER:Req3-DTI7_1-Sins)
<i>hasEnd</i> (Req3-DTI7_1,Req3-DTI7_1-Eins)	⇒ <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req3-DTI7_1 domain-PAIyER:Req3-DTI7_1-Eins)
<i>actualiza</i> (Valentino,Req3-Ev7_2)	⇒ <i>ObjectPropertyAssertion</i> (domain-PAIyER:actualiza domain-PAIyER:Valentino domain-PAIyER:Req3-Ev7_2)
<i>actualiza</i> (Req3-Ev7_2,Reservacion_prestamo_JuanCarlos)	⇒ <i>ObjectPropertyAssertion</i> (domain-PAIyER:actualiza domain-PAIyER:Req3-Ev7_2 domain-PAIyER:Reservacion_prestamo_JuanCarlos)
<i>during</i> (Req3-Ev7_2,Req3-DTI7_2)	⇒ <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req3-Ev7_2 domain-PAIyER:Req3-DTI7_2)
<i>hasBeginning</i> (Req3-DTI7_2,Req3-DTI7_2-Sins)	⇒ <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req3-DTI7_2 domain-PAIyER:Req3-DTI7_2-Sins)
<i>hasEnd</i> (Req3-DTI7_2,Req3-DTI7_2-Eins)	⇒ <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req3-DTI7_2 domain-PAIyER:Req3-DTI7_2-Eins)

}

Tabla D.15 ABox de casos de éxito y fracaso en el RF3. Modificar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<p>A_8: [XXXX (Empleado $\sqsubseteq \exists$actualiza.Entidad)]: El empleado Valentino actualiza la entidad Libro2.</p> <p>$A_8 = \{$</p>	
<p>-- Aserciones de clase --</p> <p>Empleado(Valentino)</p>	<p>\Rightarrow ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</p>
<p>Entidad(Libro2)</p>	<p>\Rightarrow ClassAssertion(domain-PAIyER:Contrato domain-PAIyER: Libro2)</p>
<p>Event(Req3-Ev8)</p>	<p>\Rightarrow ClassAssertion(timeEvents:Event domain-PAIyER:Req3-Ev8)</p>
<p>DateTimeInterval(Req3-DTI8)</p>	<p>\Rightarrow ClassAssertion(time:DateTimeInterval domain-PAIyER:Req3-DTI8)</p>
<p>Instant(Req3-DTI8-Sins)</p>	<p>\Rightarrow ClassAssertion(time:Instant domain-PAIyER:Req3-DTI8-Sins)</p>
<p>Instant(Req3-DTI8-Eins)</p>	<p>\Rightarrow ClassAssertion(time:Instant domain-PAIyER:Req3-DTI8-Eins)</p>
<p>-- Aserciones de propiedades de objeto --</p> <p>actualiza(Valentino,Req3-Ev8)</p>	<p>\Rightarrow ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Valentino domain-PAIyER:Req3-Ev8)</p>
<p>actualiza(Req3-Ev8, Libro2)</p>	<p>\Rightarrow ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Req3-Ev8 domain-PAIyER: Libro2)</p>
<p>during(Req3-Ev8,Req3-DTI8)</p>	<p>\Rightarrow ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req3-Ev8 domain-PAIyER:Req3-DTI8)</p>
<p>hasBeginning(Req3-DTI8,Req3-DTI8-Sins)</p>	<p>\Rightarrow ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req3-DTI8 domain-PAIyER:Req3-DTI8-Sins)</p>
<p>hasEnd(Req3-DTI8,Req3-DTI8-Eins)</p>	<p>\Rightarrow ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req3-DTI8 domain-PAIyER:Req3-DTI8-Eins)</p>
<p>}</p>	
<p>Caso de éxito 3 (A₉, A₁₀, A₁₁, A₁₂):</p> <p>A_9: [X(Cliente $\sqsubseteq \exists$modifica.Reservacion)]: El cliente Santiago modifica la reservacion Reservacion_prestamo_Santiago.</p> <p>...</p> <p>A_{10}: [XX (Empleado $\sqsubseteq \exists$verificaDisponibilidadDe.Entidad)]: El empleado Valentino verifica la disponibilidad de la Entidad Libro3.</p> <p>...</p> <p>A_{11}: [\negXXX(Empleado $\sqsubseteq \exists$cancela.Reservacion \wedge Empleado $\sqsubseteq \exists \forall$actualiza.Reservacion)]: El empleado Valentino no cancela la reservación Reservacion_prestamo_Santiago ni actualiza la reservación Reservacion_prestamo_Santiago.</p> <p>...</p> <p>A_{12}: [\negXXXX (Empleado $\sqsubseteq \exists$actualiza.Entidad)]: El empleado Valentino no actualiza la entidad Libro3.</p> <p>...</p>	
<p>Caso de éxito 4 (A₁₃, A₁₄, A₁₅, A₁₆):</p> <p>A_{13}: [X(Cliente $\sqsubseteq \exists$modifica.Reservacion)]: El cliente Violeta modifica la reservacion Reservacion_prestamo_Violeta.</p> <p>...</p> <p>A_{14}: [XX (Empleado $\sqsubseteq \exists$verificaDisponibilidadDe.Entidad)]: El empleado Valentino verifica la disponibilidad de la Entidad Libro4.</p> <p>...</p> <p>A_{15}: A_{11}: [XXX(Empleado $\sqsubseteq \exists$cancela.Reservacion \wedge Empleado $\sqsubseteq \exists \forall$actualiza.Reservacion)]: El</p>	

Tabla D.15 ABox de casos de éxito y fracaso en el RF3. Modificar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<p><i>empleado Valentino cancela la reservación Reservacion_prestamo_Violeta y actualiza la reservación Reservacion_prestamo_Violeta.</i></p> <p>...</p> <p>A₁₆: [XXXX (Empleado \sqsubseteq \existsactualiza.Entidad)]: <i>El empleado Valentino actualiza la entidad Libro4.</i></p> <p>...</p>	
<p>Caso de éxito n</p> <p>A_n</p> <p>...</p>	

Tabla D.16 Reglas SWRL para la SoIs de casos de éxito y fracaso de acuerdo al RF3. Modificar una reservación. (cont.)

ABox	Regla SWRL
A₃	<i>Event(?e, during(?e, ?i), overlaps(?i, Req3-DTI3_1), cancela(?e, Reservacion_prestamo_Heidi), cancela(Valentino, ?e) -> Nothing(Valentino)</i>
	<i>Event(?e, during(?e, ?i), overlaps(?i, Req3-DTI3_2), actualiza(?e, Reservacion_prestamo_Heidi), actualiza(Valentino, ?e) -> Nothing(Valentino)</i>
A₄	<i>Event(?e, during(?e, ?i), overlaps(?i, Req3- DTI4), actualiza(?e, LLibro1), actualiza(Valentino, ?e) -> Nothing(Valentino)</i>
A₁₁	<i>Event(?e, during(?e, ?i), overlaps(?i, Req3-DTI11_1), cancela(?e, Reservacion_prestamo_Santiago), cancela(Valentino, ?e) -> Nothing(Valentino)</i>
	<i>Event(?e, during(?e, ?i), overlaps(?i, Req3-DTI11_2), actualiza(?e, Reservacion_prestamo_Santiago), actualiza(Valentino, ?e) -> Nothing(Valentino)</i>
A₁₂	<i>Event(?e, during(?e, ?i), overlaps(?i, Req3- DTI12), actualiza(?e, Libro3), actualiza(Valentino, ?e) -> Nothing(Valentino)</i>

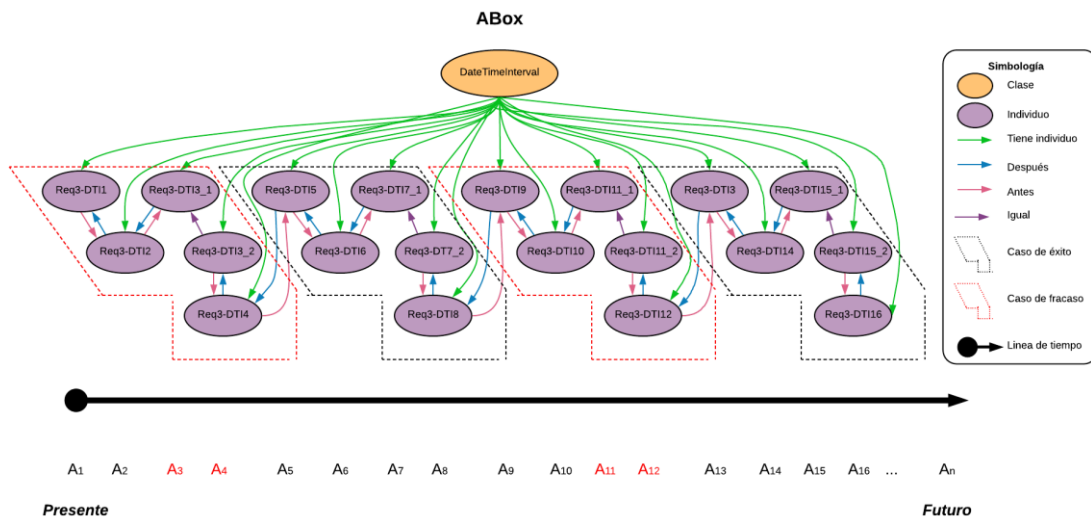


Figura D.12 SoIs en la ABox de casos de éxito y fracaso de acuerdo al RF3. Modificar una reservación.

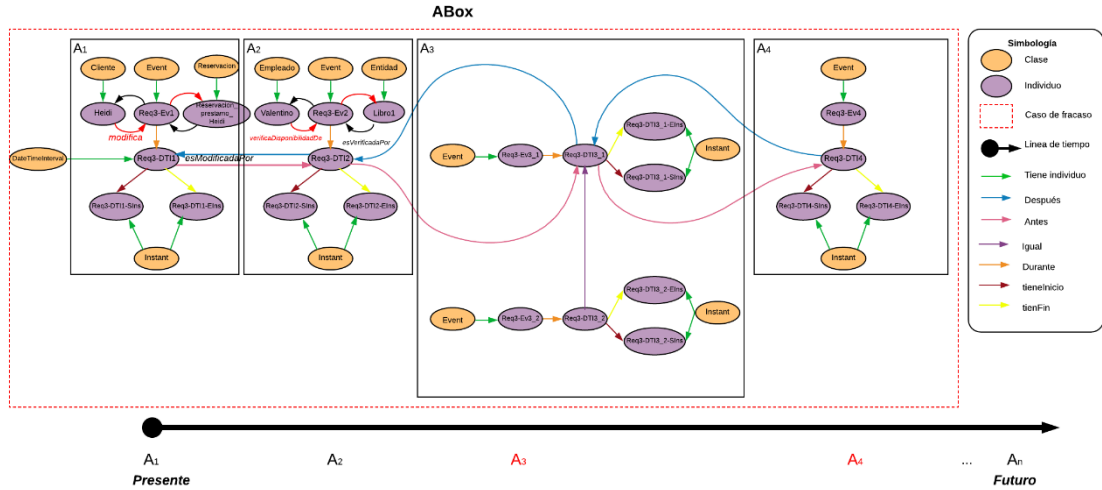


Figura D.13 Caso de fracaso 1 en la ABox de casos de éxito y fracaso de acuerdo al RF3. Modificar una reservación.

1.2.4 RF4. Cancelar una reservación

Las secuencias de interpretaciones descritas en esta sección satisfacen/cumplen con el comportamiento dinámico del RF4. Cancelar una reservación representado formalmente con la fórmula de la Tabla D.17.

Tabla D.17 Fórmula ALC_{LTL} del RF4. Cancelar una reservación.

$$\mathcal{G} (
 \begin{aligned}
 & \mathcal{X}(\text{Cliente} \sqsubseteq \exists daDeBaja.Reservacion) \rightarrow (\\
 & \quad \neg \mathcal{X} \mathcal{X} \text{Empleado} \sqsubseteq \exists aplicaPoliticlas.(Reservacion \sqcup \text{Cliente}) \wedge \mathcal{F}(\\
 & \quad \quad \neg \mathcal{X} \mathcal{X} \mathcal{X} \text{empleado} \sqsubseteq \exists cancela.Reservacion \wedge empleado \sqsubseteq \exists actualiza.Entidad) \\
 & \quad) \\
 &)
 \end{aligned}$$

1.2.4.1. Casos de éxito

En la Tabla D.18 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la ABox) basadas en casos de éxito, que conforman una secuencia que representa el comportamiento dinámico para cancelar una reservación. Una representación gráfica simplificada de la secuencia se muestra en la Figura D.14. A manera de ejemplo, en la Figura D.15 se muestra el detalle de las aserciones A_1 , A_2 , A_3 y A_4 en el caso de éxito número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de razonamiento automatizado, los demás casos se definen de manera similar, sin embargo, difieren en los individuos involucrados.

Tabla D.18 ABox de casos de éxito en el RF4. Cancelar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
--------------------	---------------------------

Tabla D.18 ABox de casos de éxito en el RF4. Cancelar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
Caso de éxito 1 (A_1, A_2, A_3):	
A_1 : [\mathbf{X} ($\text{Cliente} \sqsubseteq \exists \text{daDeBaja.Reservacion}$)]: El cliente Heidi da de baja la reservación <i>Reservacion_prestamo_Heidi</i> .	
$A_1 = \{$	
-- Aserciones de clase --	
<i>Cliente</i> (Heidi)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Cliente domain-PAIyER:Heidi)
<i>Reservacion</i> (<i>Reservacion_prestamo_Heidi</i>)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Reservacion domain-PAIyER: Reservacion_prestamo_Heidi)
<i>Event</i> (Req4-Ev1)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req4-Ev1)
<i>DateTimeInterval</i> (Req4-DT11)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req4-DT11)
<i>Instant</i> (Req4-DT11-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req4-DT11-Sins)
<i>Instant</i> (Req4-DT11-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req4-DT11-Eins)
-- Aserciones de propiedades de objeto --	
<i>daDeBaja</i> (Heidi,Req4-Ev1)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:daDeBaja domain-PAIyER:Heidi domain-PAIyER:Req4-Ev1)
<i>daDeBaja</i> (Req4-Ev1, <i>Reservacion_prestamo_Heidi</i>)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:daDeBaja domain-PAIyER:Req4-Ev1 domain-PAIyER: Reservacion_prestamo_Heidi)
<i>during</i> (Req4-Ev1,Req4-DT11)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req4-Ev1 domain-PAIyER:Req4-DT11)
<i>hasBeginning</i> (Req4-DT11,Req4-DT11-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req4-DT11 domain-PAIyER:Req4-DT11-Sins)
<i>hasEnd</i> (Req4-DT11,Req4-DT11-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req4-DT11 domain-PAIyER:Req4-DT11-Eins)
$\}$	\Rightarrow
A_2 : [\mathbf{XX} ($\text{Empleado} \sqsubseteq \exists \text{aplicaPoliticlas} . (\text{Reservacion} \sqcup \text{Cliente})$)]: El empleado Valentino aplica las políticas correspondientes a la reservación <i>Reservacion_prestamo_Heidi</i> y al cliente Heidi.	
$A_2 = \{$	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Reservacion</i> (<i>Reservacion_prestamo_Heidi</i>)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Reservacion domain-PAIyER: Reservacion_prestamo_Heidi)
<i>Event</i> (Req4-Ev2_1)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req4-Ev2_1)
<i>DateTimeInterval</i> (Req4-DT12_1)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req4-DT12_1)
<i>Instant</i> (Req4-DT12_1-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req4-DT12_1-Sins)
<i>Instant</i> (Req4-DT12_1-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req4-DT12_1-Eins)
<i>Cliente</i> (Heidi)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Cliente domain-PAIyER: Heidi)
<i>Event</i> (Req4-Ev2_2)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req4-Ev2_2)

Tabla D.18 ABox de casos de éxito en el RF4. Cancelar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>DateTimeInterval(Req4-DTI2_2)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req4-DTI2_2)</i>
<i>Instant(Req4-DTI2_2-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DTI2_2-Sins)</i>
<i>Instant(Req4-DTI2_2-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DTI2_2-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>aplicaPoliticass(Valentino,Req4-Ev2_1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:aplicaPoliticass domain-PAIyER:Valentino domain-PAIyER:Req4-Ev2_1)</i>
<i>aplicaPoliticass (Req4-Ev2_1,Reservacion_prestamo_Heidi)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:aplicaPoliticass domain-PAIyER:Req4-Ev2_1 domain-PAIyER:Reservacion_prestamo_Heidi)</i>
<i>during(Req4-Ev2_1,Req4-DTI2_1)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req4-Ev2_1 domain-PAIyER:Req4-DTI2_1)</i>
<i>hasBeginning(Req4-DTI2_1,Req4-DTI2_1-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req4-DTI2_1 domain-PAIyER:Req4-DTI2_1-Sins)</i>
<i>hasEnd(Req4-DTI2_1,Req4-DTI2_1-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req4-DTI2_1 domain-PAIyER:Req4-DTI2_1-Eins)</i>
<i>aplicaPoliticass(Valentino,Req4-Ev2_2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:aplicaPoliticass domain-PAIyER:Valentino domain-PAIyER:Req4-Ev2_2)</i>
<i>aplicaPoliticass (Req4-Ev2_2, Heidi)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:aplicaPoliticass domain-PAIyER:Req4-Ev2_2 domain-PAIyER:Heidi)</i>
<i>during(Req4-Ev2_2,Req4-DTI2_2)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req4-Ev2_2 domain-PAIyER:Req4-DTI2_2)</i>
<i>hasBeginning(Req4-DTI2_2,Req4-DTI2_2-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req4-DTI2_2 domain-PAIyER:Req4-DTI2_2-Sins)</i>
<i>hasEnd(Req4-DTI2_2,Req4-DTI2_2-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req4-DTI2_2 domain-PAIyER:Req4-DTI2_2-Eins)</i>
}	
A₃: [XXXX (Empleado \sqsubseteq \existscancela.Reservacion \wedge Empleado \sqsubseteq \existsactualiza.Entidad)]: El empleado Valentino cancela la reservación Reservacion_prestamo_Heidi y actualiza la entidad Libro1.	
A₃ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Reservacion(Reservacion_prestamo_Heidi)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Reservacion domain-PAIyER:Reservacion_prestamo_Heidi)</i>
<i>Event(Req4-Ev3_1)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req4-Ev3_1)</i>
<i>DateTimeInterval(Req4-DTI3_1)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req4-DTI3_1)</i>
<i>Instant(Req4-DTI3_1-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DTI3_1-Sins)</i>
<i>Instant(Req4-DTI3_1-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DTI3_1-Eins)</i>

Tabla D.18 ABox de casos de éxito en el RF4. Cancelar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Entidad(Libro1)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER: Libro1)</i>
<i>Event(Req4-Ev3_2)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req4-Ev3_2)</i>
<i>DateTimeInterval(Req4-DTI3_2)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req4-DTI3_2)</i>
<i>Instant(Req4-DTI3_2-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DTI3_2-Sins)</i>
<i>Instant(Req4-DTI3_2-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DTI3_2-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>cancela(Valentino,Req4-Ev3_1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:cancela domain-PAIyER:Valentino domain-PAIyER:Req4-Ev3_1)</i>
<i>cancela (Req4-Ev3_1,Reservacion_prestamo_Heidi)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:cancela domain-PAIyER:Req4-Ev3_1 domain-PAIyER:Reservacion_prestamo_Heidi)</i>
<i>during(Req4-Ev3_1,Req4-DTI3_1)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req4-Ev3_1 domain-PAIyER:Req4-DTI3_1)</i>
<i>hasBeginning(Req4-DTI3_1,Req4-DTI3_1-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req4-DTI3_1 domain-PAIyER:Req4-DTI3_1-Sins)</i>
<i>hasEnd(Req4-DTI3_1,Req4-DTI3_1-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req4-DTI3_1 domain-PAIyER:Req4-DTI3_1-Eins)</i>
<i>actualiza(Valentino,Req4-Ev3_2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Valentino domain-PAIyER:Req4-Ev3_2)</i>
<i>actualiza (Req4-Ev3_2, Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Req4-Ev3_2 domain-PAIyER: Libro1)</i>
<i>during(Req4-Ev3_2,Req4-DTI3_2)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req4-Ev3_2 domain-PAIyER:Req4-DTI3_2)</i>
<i>hasBeginning(Req4-DTI3_2,Req4-DTI3_2-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req4-DTI3_2 domain-PAIyER:Req4-DTI3_2-Sins)</i>
<i>hasEnd(Req4-DTI3_2,Req4-DTI3_2-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req4-DTI3_2 domain-PAIyER:Req4-DTI3_2-Eins)</i>
}	
Caso de éxito 2 (A₄, A₅, A₆):	
A ₄ : [X (Cliente \sqsubseteq \exists daDeBaja.Reservacion)]: El cliente Juan_Carlos da de baja la reservación Reservacion_prestamo_JuanCarlos.	
A ₄ = {	
-- Aserciones de clase --	
<i>Cliente(Juan_Carlos)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Juan_Carlos)</i>
<i>Reservacion(Reservacion_prestamo_JuanCarlos)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Reservacion domain-PAIyER:Reservacion_prestamo_JuanCarlos)</i>
<i>Event(Req4-Ev4)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req4-Ev4)</i>

Tabla D.18 ABox de casos de éxito en el RF4. Cancelar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>DateTimeInterval(Req4-DTI4)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req4-DTI4)</i>
<i>Instant(Req4-DTI4-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DTI4-Sins)</i>
<i>Instant(Req4-DTI4-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DTI4-Eins)</i>
-- Aserciones de propiedades de objeto --	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:daDeBaja domain-PAIyER:Juan_Carlos domain-PAIyER:Req4-Ev4)</i>
<i>daDeBaja(Juan_Carlos,Req4-Ev4)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:daDeBaja domain-PAIyER:Req4-Ev4 domain-PAIyER:Reservacion_prestamo_JuanCarlos)</i>
<i>daDeBaja(Req4-Ev4, Reservacion_prestamo_JuanCarlos)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req4-Ev4 domain-PAIyER:Req4-DTI4)</i>
<i>during(Req4-Ev4,Req4-DTI4)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req4-DTI4 domain-PAIyER:Req4-DTI4-Sins)</i>
<i>hasBeginning(Req4-DTI4,Req4-DTI4-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req4-DTI4 domain-PAIyER:Req4-DTI4-Eins)</i>
<i>hasEnd(Req4-DTI4,Req4-DTI4-Eins)</i>	\Rightarrow
}	
<i>A5: [XX (Empleado \sqsubseteq \existsaplicaPolíticas.(Reservacion \sqcup Cliente))]: El empleado Valentino aplica las políticas correspondientes a la reservación Reservacion_prestamo_JuanCarlos y al cliente Juan_Carlos.</i>	
<i>A5 = {</i>	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Reservacion(Reservacion_prestamo_JuanCarlos)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Reservacion domain-PAIyER:Reservacion_prestamo_JuanCarlos)</i>
<i>Event(Req4-Ev5_1)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req4-Ev5_1)</i>
<i>DateTimeInterval(Req4-DTI5_1)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req4-DTI5_1)</i>
<i>Instant(Req4-DTI5_1-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DTI5_1-Sins)</i>
<i>Instant(Req4-DTI5_1-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DTI5_1-Eins)</i>
<i>Cliente(Juan_Carlos)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Juan_Carlos)</i>
<i>Event(Req4-Ev5_2)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req4-Ev5_2)</i>
<i>DateTimeInterval(Req4-DTI5_2)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req4-DTI5_2)</i>
<i>Instant(Req4-DTI5_2-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DTI5_2-Sins)</i>
<i>Instant(Req4-DTI5_2-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DTI5_2-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>aplicaPolíticas(Valentino,Req4-Ev5_1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:aplicaPolíticas domain-PAIyER:Valentino domain-PAIyER:Req4-Ev5_1)</i>

Tabla D.18 ABox de casos de éxito en el RF4. Cancelar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>aplicaPolíticas</i> (Req4-Ev5_1,Reservacion_prestamo_JuanCarlos)	⇒ <i>ObjectPropertyAssertion</i> (domain-PAIyER: <i>aplicaPolíticas</i> domain-PAIyER:Req4-Ev5_1 domain-PAIyER: <i>Reservacion_prestamo_JuanCarlos</i>)
<i>during</i> (Req4-Ev5_1,Req4-DTI5_1)	⇒ <i>ObjectPropertyAssertion</i> (timeEvents: <i>during</i> domain-PAIyER:Req4-Ev5_1 domain-PAIyER:Req4-DTI5_1)
<i>hasBeginning</i> (Req4-DTI5_1,Req4-DTI5_1-Sins)	⇒ <i>ObjectPropertyAssertion</i> (time: <i>hasBeginning</i> domain-PAIyER:Req4-DTI5_1 domain-PAIyER:Req4-DTI5_1-Sins)
<i>hasEnd</i> (Req4-DTI5_1,Req4-DTI5_1-Eins)	⇒ <i>ObjectPropertyAssertion</i> (time: <i>hasEnd</i> domain-PAIyER:Req4-DTI5_1 domain-PAIyER:Req4-DTI5_1-Eins)
<i>aplicaPolíticas</i> (Valentino,Req4-Ev5_2)	⇒ <i>ObjectPropertyAssertion</i> (domain-PAIyER: <i>aplicaPolíticas</i> domain-PAIyER:Valentino domain-PAIyER:Req4-Ev5_2)
<i>aplicaPolíticas</i> (Req4-Ev5_2, Juan_Carlos)	⇒ <i>ObjectPropertyAssertion</i> (domain-PAIyER: <i>aplicaPolíticas</i> domain-PAIyER:Req4-Ev5_2 domain-PAIyER: Juan_Carlos)
<i>during</i> (Req4-Ev5_2,Req4-DTI5_2)	⇒ <i>ObjectPropertyAssertion</i> (timeEvents: <i>during</i> domain-PAIyER:Req4-Ev5_2 domain-PAIyER:Req4-DTI5_2)
<i>hasBeginning</i> (Req4-DTI5_2,Req4-DTI5_2-Sins)	⇒ <i>ObjectPropertyAssertion</i> (time: <i>hasBeginning</i> domain-PAIyER:Req4-DTI5_2 domain-PAIyER:Req4-DTI5_2-Sins)
<i>hasEnd</i> (Req4-DTI5_2,Req4-DTI5_2-Eins)	⇒ <i>ObjectPropertyAssertion</i> (time: <i>hasEnd</i> domain-PAIyER:Req4-DTI5_2 domain-PAIyER:Req4-DTI5_2-Eins)
}	
<i>A₆</i> : [XXXX (Empleado ⊑ ∃cancela.Reservacion ∧ Empleado ⊑ ∃actualiza.Entidad)]: El empleado Valentino cancela la reservación <i>Reservacion_prestamo_JuanCarlos</i> y actualiza la entidad <i>Libro2</i> .	
<i>A₆</i> = {	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	⇒ <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Reservacion</i> (Reservacion_prestamo_JuanCarlos)	⇒ <i>ClassAssertion</i> (domain-PAIyER:Reservacion domain-PAIyER:Reservacion_prestamo_JuanCarlos)
<i>Event</i> (Req4-Ev6_1)	⇒ <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req4-Ev6_1)
<i>DateTimeInterval</i> (Req4-DTI6_1)	⇒ <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req4-DTI6_1)
<i>Instant</i> (Req4-DTI6_1-Sins)	⇒ <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req4-DTI6_1-Sins)
<i>Instant</i> (Req4-DTI6_1-Eins)	⇒ <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req4-DTI6_1-Eins)
<i>Entidad</i> (Libro)	⇒ <i>ClassAssertion</i> (domain-PAIyER:Entidad domain-PAIyER:Libro2)
<i>Event</i> (Req4-Ev6_2)	⇒ <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req4-Ev6_2)
<i>DateTimeInterval</i> (Req4-DTI6_2)	⇒ <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req4-DTI6_2)
<i>Instant</i> (Req4-DTI6_2-Sins)	⇒ <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req4-DTI6_2-Sins)

Tabla D.18 ABox de casos de éxito en el RF4. Cancelar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Instant(Req4-DTI6_2-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DTI6_2-Eins)</i>
<i>-- Aserciones de propiedades de objeto --</i>	
<i>cancela(Valentino,Req4-Ev6_1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:cancela domain-PAIyER:Valentino domain-PAIyER:Req4-Ev6_1)</i>
<i>cancela (Req4-Ev6_1,Reservacion_prestamo_JuanCarlos)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:cancela domain-PAIyER:Req4-Ev6_1 domain-PAIyER:Reservacion_prestamo_JuanCarlos)</i>
<i>during(Req4-Ev6_1,Req4-DTI6_1)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req4-Ev6_1 domain-PAIyER:Req4-DTI6_1)</i>
<i>hasBeginning(Req4-DTI6_1,Req4-DTI6_1-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req4-DTI6_1 domain-PAIyER:Req4-DTI6_1-Sins)</i>
<i>hasEnd(Req4-DTI6_1,Req4-DTI6_1-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req4-DTI6_1 domain-PAIyER:Req4-DTI6_1-Eins)</i>
<i>actualiza(Valentino,Req4-Ev6_2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Valentino domain-PAIyER:Req4-Ev6_2)</i>
<i>actualiza (Req4-Ev6_2, Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Req4-Ev6_2 domain-PAIyER:Libro2)</i>
<i>during(Req4-Ev6_2,Req4-DTI6_2)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req4-Ev6_2 domain-PAIyER:Req4-DTI6_2)</i>
<i>hasBeginning(Req4-DTI6_2,Req4-DTI6_2-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req4-DTI6_2 domain-PAIyER:Req4-DTI6_2-Sins)</i>
<i>hasEnd(Req4-DTI6_2,Req4-DTI6_2-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req4-DTI6_2 domain-PAIyER:Req4-DTI6_2-Eins)</i>
}	
Caso de éxito 3 (A7, A8, A9):	
<i>A7: [X(Cliente \sqsubseteq \existsdaDeBaja.Reservacion)]: El cliente Santiago da de baja la reservación Reservacion_prestamo_Santiago.</i>	
...	
<i>A8: [XX (Empleado \sqsubseteq \existsaplicaPolíticas.(Reservacion \sqcup Cliente))]: El empleado Valentino aplica las políticas correspondientes a la reservación Reservacion_prestamo_Santiago y al cliente Santiago.</i>	
...	
<i>A9: [XXXX (Empleado \sqsubseteq \existscancela.Reservacion \wedge Empleado \sqsubseteq \existsactualiza.Entidad)]: El empleado Valentino cancela la reservación Reservacion_prestamo_Santiago y actualiza la entidad Libro3.</i>	
...	
Caso de éxito 4 (A10, A11, A12):	
<i>A10: [X(Cliente \sqsubseteq \existsdaDeBaja.Reservacion)]: El cliente Violeta da de baja la reservación Reservacion_prestamo_Violeta.</i>	
...	
<i>A11: [XX (Empleado \sqsubseteq \existsaplicaPolíticas.(Reservacion \sqcup Cliente))]: El empleado Valentino aplica las políticas correspondientes a la reservación Reservacion_prestamo_Violeta y al cliente Violeta.</i>	
...	
<i>A12: [XXXX (Empleado \sqsubseteq \existscancela.Reservacion \wedge Empleado \sqsubseteq \existsactualiza.Entidad)]: El empleado Valentino cancela la reservación Reservacion_prestamo_Violeta y actualiza la entidad Libro4.</i>	
...	

Tabla D.18 ABox de casos de éxito en el RF4. Cancelar una reservación (cont.)

	Sintaxis DL	Sintaxis Funcional
Caso de éxito n		
A_n		
...		

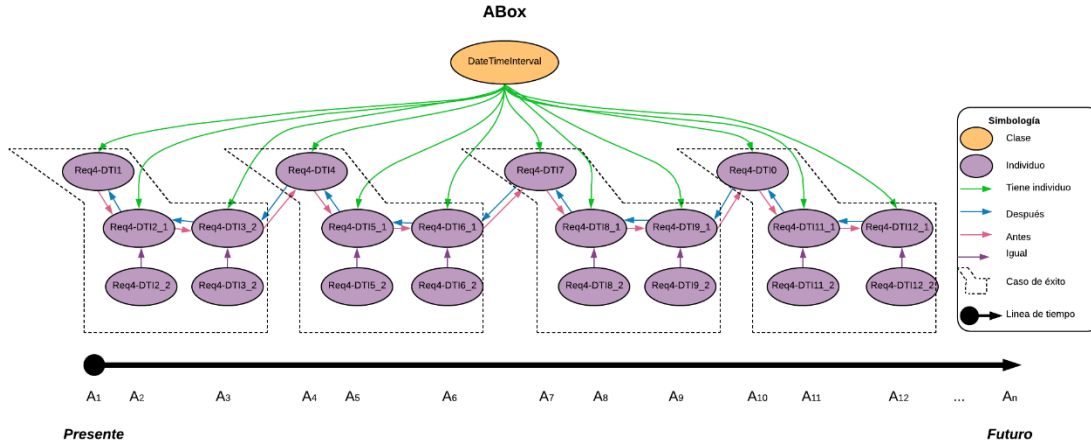


Figura D.14 SoIs en la ABox de casos de éxito de acuerdo al RF4. Cancelar una reservación.

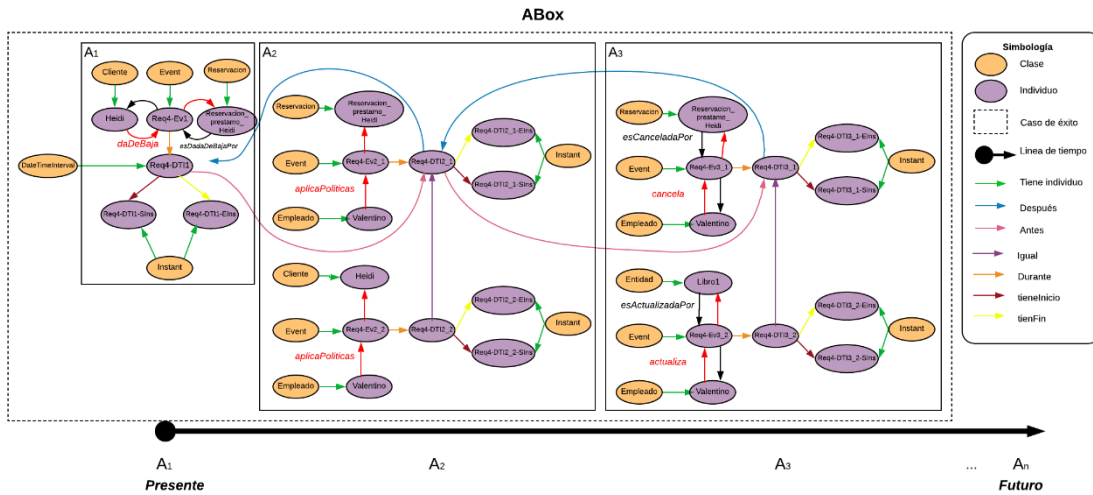


Figura D.15 Caso de éxito 1 en la ABox de casos de éxito de acuerdo al RF4. Cancelar una reservación.

1.2.4.2. Casos de éxito y fracaso

En la Tabla D.19 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la ABox) basadas en casos de éxito y fracaso, que conforman una secuencia que representa el comportamiento dinámico para cancelar una reservación. Una representación gráfica simplificada de la secuencia se muestra en la Figura D.16. A manera de

ejemplo, en la Figura D.17 se muestra el detalle de las aserciones A_1 , A_2 , A_3 y A_4 en el caso de fracaso número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de razonamiento automatizado. Los demás casos se definen de manera similar, sin embargo, difieren en los individuos involucrados.

En la Tabla D.20 se describen las reglas SWRL utilizadas para restringir la existencia de afirmaciones de propiedades de objeto en el intervalo de tiempo correspondiente.

Tabla D.19 ABox de casos de éxito y fracaso en el RF4. Cancelar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
Caso de fracaso 1 (A_1, A_2, A_3):	
A_1 : $[X(\text{Cliente} \sqsubseteq \exists \text{daDeBaja.Reservacion})]$: El cliente Heidi da de baja la reservación <i>Reservacion_prestamo_Heidi</i> .	
$A_1 = \{$	
-- Aserciones de clase --	
<i>Cliente(Heidi)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Heidi)</i>
<i>Reservacion(Reservacion_prestamo_Heidi)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Reservacion domain-PAIyER:Reservacion_prestamo_Heidi)</i>
<i>Event(Req4-Ev1)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req4-Ev1)</i>
<i>DateTimeInterval(Req4-DT11)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req4-DT11)</i>
<i>Instant(Req4-DT11-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DT11-Sins)</i>
<i>Instant(Req4-DT11-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DT11-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>daDeBaja(Heidi,Req4-Ev1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER: daDeBaja domain-PAIyER:Heidi domain-PAIyER:Req4-Ev1)</i>
<i>daDeBaja(Req4-Ev1, Reservacion_prestamo_Heidi)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER: daDeBaja domain-PAIyER:Req4-Ev1 domain-PAIyER:Reservacion_prestamo_Heidi)</i>
<i>during(Req4-Ev1,Req4-DT11)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req4-Ev1 domain-PAIyER:Req4-DT11)</i>
<i>hasBeginning(Req4-DT11,Req4-DT11-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req4-DT11 domain-PAIyER:Req4-DT11-Sins)</i>
<i>hasEnd(Req4-DT11,Req4-DT11-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req4-DT11 domain-PAIyER:Req4-DT11-Eins)</i>
$\}$	\Rightarrow
A_2 : $[XX(\text{Empleado} \sqsubseteq \exists \text{aplicaPolitic}(\text{Reservacion} \sqcup \text{Cliente}))]$: El empleado Valentino aplica las políticas correspondientes a la reservación <i>Reservacion_prestamo_Heidi</i> y al cliente Heidi.	
$A_2 = \{$	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Reservacion(Reservacion_prestamo_Heidi)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Reservacion domain-PAIyER:Reservacion_prestamo_Heidi)</i>
<i>Event(Req4-Ev2_1)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req4-Ev2_1)</i>
<i>DateTimeInterval(Req4-DT12_1)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req4-DT12_1)</i>
<i>Instant(Req4-DT12_1-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-</i>

Tabla D.19 ABox de casos de éxito y fracaso en el RF4. Cancelar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Instant</i> (Req4-DTI2_1-Eins)	\Rightarrow <i>ClassAssertion</i> (time: <i>Instant domain-PAIyER:Req4-DTI2_1-Eins</i>)
<i>Cliente</i> (Heidi)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER: <i>Cliente domain-PAIyER: Heidi</i>)
<i>Event</i> (Req4-Ev2_2)	\Rightarrow <i>ClassAssertion</i> (timeEvents: <i>Event domain-PAIyER:Req4-Ev2_2</i>)
<i>DateTimeInterval</i> (Req4-DTI2_2)	\Rightarrow <i>ClassAssertion</i> (time: <i>DateTimeInterval domain-PAIyER:Req4-DTI2_2</i>)
<i>Instant</i> (Req4-DTI2_2-Sins)	\Rightarrow <i>ClassAssertion</i> (time: <i>Instant domain-PAIyER:Req4-DTI2_2-Sins</i>)
<i>Instant</i> (Req4-DTI2_2-Eins)	\Rightarrow <i>ClassAssertion</i> (time: <i>Instant domain-PAIyER:Req4-DTI2_2-Eins</i>)
-- Aserciones de propiedades de objeto --	
<i>aplicaPolitic</i> s(Valentino,Req4-Ev2_1)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: <i>aplicaPolitic</i> s domain-PAIyER:Valentino domain-PAIyER:Req4-Ev2_1)
<i>aplicaPolitic</i> s (Req4-Ev2_1,Reservacion_prestamo_Heidi)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: <i>aplicaPolitic</i> s domain-PAIyER:Req4-Ev2_1 domain-PAIyER: Reservacion_prestamo_Heidi)
<i>during</i> (Req4-Ev2_1,Req4-DTI2_1)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents: <i>during domain-PAIyER:Req4-Ev2_1 domain-PAIyER:Req4-DTI2_1</i>)
<i>hasBeginning</i> (Req4-DTI2_1,Req4-DTI2_1-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time: <i>hasBeginning domain-PAIyER:Req4-DTI2_1 domain-PAIyER:Req4-DTI2_1-Sins</i>)
<i>hasEnd</i> (Req4-DTI2_1,Req4-DTI2_1-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time: <i>hasEnd domain-PAIyER:Req4-DTI2_1 domain-PAIyER:Req4-DTI2_1-Eins</i>)
<i>aplicaPolitic</i> s(Valentino,Req4-Ev2_2)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: <i>aplicaPolitic</i> s domain-PAIyER:Valentino domain-PAIyER:Req4-Ev2_2)
<i>aplicaPolitic</i> s (Req4-Ev2_2,Heidi)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: <i>aplicaPolitic</i> s domain-PAIyER:Req4-Ev2_2 domain-PAIyER: Heidi)
<i>during</i> (Req4-Ev2_2,Req4-DTI2_2)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents: <i>during domain-PAIyER:Req4-Ev2_2 domain-PAIyER:Req4-DTI2_2</i>)
<i>hasBeginning</i> (Req4-DTI2_2,Req4-DTI2_2-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time: <i>hasBeginning domain-PAIyER:Req4-DTI2_2 domain-PAIyER:Req4-DTI2_2-Sins</i>)
<i>hasEnd</i> (Req4-DTI2_2,Req4-DTI2_2-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time: <i>hasEnd domain-PAIyER:Req4-DTI2_2 domain-PAIyER:Req4-DTI2_2-Eins</i>)
}	
A₃: [\negXXXX (Empleado \sqsubseteq \existscancela.Reservacion \wedge Empleado \sqsubseteq \existsactualiza.Entidad)]: El empleado Valentino no cancela la reservación Reservacion_prestamo_Heidi ni actualiza la entidad Libro1.	
A ₃ = {	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER: <i>Empleado domain-PAIyER:Valentino</i>)
<i>Reservacion</i> (Reservacion_prestamo_Heidi)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER: <i>Reservacion domain-PAIyER: Reservacion_prestamo_Heidi</i>)
<i>Event</i> (Req4-Ev3_1)	\Rightarrow <i>ClassAssertion</i> (timeEvents: <i>Event domain-PAIyER:Req4-Ev3_1</i>)

Tabla D.19 ABox de casos de éxito y fracaso en el RF4. Cancelar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>DateTimeInterval(Req4-DTI3_1)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req4-DTI3_1)</i>
<i>Instant(Req4-DTI3_1-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DTI3_1-Sins)</i>
<i>Instant(Req4-DTI3_1-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DTI3_1-Eins)</i>
<i>Entidad(Libro1)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER: Libro1)</i>
<i>Event(Req4-Ev3_2)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req4-Ev3_2)</i>
<i>DateTimeInterval(Req4-DTI3_2)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req4-DTI3_2)</i>
<i>Instant(Req4-DTI3_2-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DTI3_2-Sins)</i>
<i>Instant(Req4-DTI3_2-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DTI3_2-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>cancela(Valentino,Req4-Ev3_1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER: cancela domain-PAIyER:Valentino domain-PAIyER:Req4-Ev3_1)</i>
<i>cancela (Req4-Ev3_1,Reservacion_prestamo_Heidi)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER: cancela domain-PAIyER:Req4-Ev3_1 domain-PAIyER:Reservacion_prestamo_Heidi)</i>
<i>during(Req4-Ev3_1,Req4-DTI3_1)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req4-Ev3_1 domain-PAIyER:Req4-DTI3_1)</i>
<i>hasBeginning(Req4-DTI3_1,Req4-DTI3_1-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req4-DTI3_1 domain-PAIyER:Req4-DTI3_1-Sins)</i>
<i>hasEnd(Req4-DTI3_1,Req4-DTI3_1-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req4-DTI3_1 domain-PAIyER:Req4-DTI3_1-Eins)</i>
<i>actualiza(Valentino,Req4-Ev3_2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER: actualiza domain-PAIyER:Valentino domain-PAIyER:Req4-Ev3_2)</i>
<i>actualiza (Req4-Ev3_2, Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER: actualiza domain-PAIyER:Req4-Ev3_2 domain-PAIyER: Libro1)</i>
<i>during(Req4-Ev3_2,Req4-DTI3_2)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req4-Ev3_2 domain-PAIyER:Req4-DTI3_2)</i>
<i>hasBeginning(Req4-DTI3_2,Req4-DTI3_2-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req4-DTI3_2 domain-PAIyER:Req4-DTI3_2-Sins)</i>
<i>hasEnd(Req4-DTI3_2,Req4-DTI3_2-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req4-DTI3_2 domain-PAIyER:Req4-DTI3_2-Eins)</i>
}	
Caso de éxito 1 (A₄, A₅, A₆):	
A₄: [X(Cliente \sqsubseteq \exists daDeBaja.Reservacion)]: El cliente Juan_Carlos da de baja la reservación Reservacion_prestamo_JuanCarlos.	
A ₄ = {	
-- Aserciones de clase --	
<i>Cliente(Juan_Carlos)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Juan_Carlos)</i>
<i>Reservacion(Reservacion_prestamo_JuanCarlos)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Reservacion domain-PAIyER: Reservacion_prestamo_JuanCarlos)</i>

Tabla D.19 ABox de casos de éxito y fracaso en el RF4. Cancelar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Event</i> (Req4-Ev4)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req4-Ev4)
<i>DateTimeInterval</i> (Req4-DTI4)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req4-DTI4)
<i>Instant</i> (Req4-DTI4-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req4-DTI4-Sins)
<i>Instant</i> (Req4-DTI4-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req4-DTI4-Eins)
-- Aserciones de propiedades de objeto --	
<i>daDeBaja</i> (Juan_Carlos,Req4-Ev4)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: daDeBaja domain-PAIyER:Juan_Carlos domain-PAIyER:Req4-Ev4)
<i>daDeBaja</i> (Req4-Ev4,Reservacion_prestamo_JuanCarlos)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: daDeBaja domain-PAIyER:Req4-Ev4 domain-PAIyER:Reservacion_prestamo_JuanCarlos)
<i>during</i> (Req4-Ev4,Req4-DTI4)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req4-Ev4 domain-PAIyER:Req4-DTI4)
<i>hasBeginning</i> (Req4-DTI4,Req4-DTI4-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req4-DTI4 domain-PAIyER:Req4-DTI4-Sins)
<i>hasEnd</i> (Req4-DTI4,Req4-DTI4-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req4-DTI4 domain-PAIyER:Req4-DTI4-Eins)
}	\Rightarrow
As: [XX (Empleado \sqsubseteq \exists aplicaPolíticas.(Reservacion \sqcup Cliente))]: El empleado Valentino aplica las políticas correspondientes a la reservación Reservacion_prestamo_JuanCarlos y al cliente Juan_Carlos.	
A ₅ = {	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Reservacion</i> (Reservacion_prestamo_JuanCarlos)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Reservacion domain-PAIyER:Reservacion_prestamo_JuanCarlos)
<i>Event</i> (Req4-Ev5_1)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req4-Ev5_1)
<i>DateTimeInterval</i> (Req4-DTI5_1)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req4-DTI5_1)
<i>Instant</i> (Req4-DTI5_1-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req4-DTI5_1-Sins)
<i>Instant</i> (Req4-DTI5_1-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req4-DTI5_1-Eins)
<i>Cliente</i> (Juan_Carlos)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Cliente domain-PAIyER:Juan_Carlos)
<i>Event</i> (Req4-Ev5_2)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req4-Ev5_2)
<i>DateTimeInterval</i> (Req4-DTI5_2)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req4-DTI5_2)
<i>Instant</i> (Req4-DTI5_2-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req4-DTI5_2-Sins)
<i>Instant</i> (Req4-DTI5_2-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req4-DTI5_2-Eins)
-- Aserciones de propiedades de objeto --	
<i>aplicaPolíticas</i> (Valentino,Req4-Ev5_1)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: aplicaPolíticas domain-PAIyER:Valentino domain-PAIyER:Req4-Ev5_1)
<i>aplicaPolíticas</i> (Req4-	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:

Tabla D.19 ABox de casos de éxito y fracaso en el RF4. Cancelar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Ev5_1,Reservacion_prestamo_JuanCarlos</i>	<i>aplicaPoliticadomain-PAIyER:Req4-Ev5_1 domain-PAIyER:Reservacion_prestamo_JuanCarlos</i>
<i>during(Req4-Ev5_1,Req4-DTI5_1)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req4-Ev5_1 domain-PAIyER:Req4-DTI5_1)</i>
<i>hasBeginning(Req4-DTI5_1,Req4-DTI5_1-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req4-DTI5_1 domain-PAIyER:Req4-DTI5_1-Sins)</i>
<i>hasEnd(Req4-DTI5_1,Req4-DTI5_1-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req4-DTI5_1 domain-PAIyER:Req4-DTI5_1-Eins)</i>
<i>aplicaPoliticadomain-PAIyER:Valentino,Req4-Ev5_2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:aplicaPoliticadomain-PAIyER:Valentino domain-PAIyER:Req4-Ev5_2)</i>
<i>aplicaPoliticadomain-PAIyER:Req4-Ev5_2, Juan_Carlos)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:aplicaPoliticadomain-PAIyER:Req4-Ev5_2 domain-PAIyER:Juan_Carlos)</i>
<i>during(Req4-Ev5_2,Req4-DTI5_2)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req4-Ev5_2 domain-PAIyER:Req4-DTI5_2)</i>
<i>hasBeginning(Req4-DTI5_2,Req4-DTI5_2-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req4-DTI5_2 domain-PAIyER:Req4-DTI5_2-Sins)</i>
<i>hasEnd(Req4-DTI5_2,Req4-DTI5_2-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req4-DTI5_2 domain-PAIyER:Req4-DTI5_2-Eins)</i>
}	
A₆: [XXXX (Empleado \sqsubseteq \exists cancela.Reservacion \wedge Empleado \sqsubseteq \exists actualiza.Entidad)]: El empleado Valentino cancela la reservación Reservacion_prestamo_JuanCarlos y actualiza la entidad Libro2.	
A ₆ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Reservacion(Reservacion_prestamo_JuanCarlos)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Reservacion domain-PAIyER:Reservacion_prestamo_JuanCarlos)</i>
<i>Event(Req4-Ev6_1)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req4-Ev6_1)</i>
<i>DateTimeInterval(Req4-DTI6_1)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req4-DTI6_1)</i>
<i>Instant(Req4-DTI6_1-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DTI6_1-Sins)</i>
<i>Instant(Req4-DTI6_1-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DTI6_1-Eins)</i>
<i>Entidad(Libro)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event(Req4-Ev6_2)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req4-Ev6_2)</i>
<i>DateTimeInterval(Req4-DTI6_2)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req4-DTI6_2)</i>
<i>Instant(Req4-DTI6_2-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DTI6_2-Sins)</i>
<i>Instant(Req4-DTI6_2-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req4-DTI6_2-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>cancela(Valentino,Req4-Ev6_1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:cancela</i>

Tabla D.19 ABox de casos de éxito y fracaso en el RF4. Cancelar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
<i>cancela (Req4-Ev6_1,Reservacion_prestamo_JuanCarlos)</i>	<i>domain-PAIyER:Valentino domain-PAIyER:Req4-Ev6_1)</i>
<i>during(Req4-Ev6_1,Req4-DTI6_1)</i>	<i>ObjectPropertyAssertion(domain-PAIyER: cancela domain-PAIyER:Req4-Ev6_1 domain-PAIyER:Reservacion_prestamo_JuanCarlos)</i>
<i>hasBeginning(Req4-DTI6_1,Req4-DTI6_1-Sins)</i>	<i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req4-Ev6_1 domain-PAIyER:Req4-DTI6_1)</i>
<i>hasEnd(Req4-DTI6_1,Req4-DTI6_1-Eins)</i>	<i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req4-DTI6_1 domain-PAIyER:Req4-DTI6_1-Sins)</i>
<i>actualiza(Valentino,Req4-Ev6_2)</i>	<i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req4-DTI6_1 domain-PAIyER:Req4-DTI6_1-Eins)</i>
<i>actualiza (Req4-Ev6_2, Libro2)</i>	<i>ObjectPropertyAssertion(domain-PAIyER: actualiza domain-PAIyER:Valentino domain-PAIyER:Req4-Ev6_2)</i>
<i>during(Req4-Ev6_2,Req4-DTI6_2)</i>	<i>ObjectPropertyAssertion(domain-PAIyER: actualiza domain-PAIyER:Req4-Ev6_2 domain-PAIyER:Libro2)</i>
<i>hasBeginning(Req4-DTI6_2,Req4-DTI6_2-Sins)</i>	<i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req4-Ev6_2 domain-PAIyER:Req4-DTI6_2)</i>
<i>hasEnd(Req4-DTI6_2,Req4-DTI6_2-Eins)</i>	<i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req4-DTI6_2 domain-PAIyER:Req4-DTI6_2-Sins)</i>
<i>}</i>	<i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req4-DTI6_2 domain-PAIyER:Req4-DTI6_2-Eins)</i>
Caso de fracaso 2 (A7, A8, A9):	
<i>A7: [X(Cliente \sqsubseteq \existsdaDeBaja.Reservacion)]: El cliente Santiago da de baja la reservación Reservacion_prestamo_Santiago.</i>	
...	
<i>A8: [XX (Empleado \sqsubseteq \existsaplicaPolíticas.(Reservacion \sqcup Cliente))]: El empleado Valentino aplica las políticas correspondientes a la reservación Reservacion_prestamo_Santiago y al cliente Santiago.</i>	
...	
<i>A9: [\negXXXX (Empleado \sqsubseteq \existscancela.Reservacion \wedge Empleado \sqsubseteq \existsactualiza.Entidad)]: El empleado Valentino no cancela la reservación Reservacion_prestamo_Santiago ni actualiza la entidad Libro3.</i>	
...	
Caso de éxito 2 (A10, A11, A12):	
<i>A10: [X(Cliente \sqsubseteq \existsdaDeBaja.Reservacion)]: El cliente Violeta da de baja la reservación Reservacion_prestamo_Violeta.</i>	
...	
<i>A11: [XX (Empleado \sqsubseteq \existsaplicaPolíticas.(Reservacion \sqcup Cliente))]: El empleado Valentino aplica las políticas correspondientes a la reservación Reservacion_prestamo_Violeta y al cliente Violeta.</i>	
...	
<i>A12: [XXXX (Empleado \sqsubseteq \existscancela.Reservacion \wedge Empleado \sqsubseteq \existsactualiza.Entidad)]: El empleado Valentino cancela la reservación Reservacion_prestamo_Violeta y actualiza la entidad Libro4.</i>	
...	
Caso de fracaso y de éxito n	
A_n	

Tabla D.19 ABox de casos de éxito y fracaso en el RF4. Cancelar una reservación (cont.)

Sintaxis DL	Sintaxis Funcional
...	

Tabla D.20 Reglas SWRL para la SoIs de casos de éxito y fracaso de acuerdo al RF4. Cancelar una reservación (cont.)

ABox	Regla SWRL
A3	<i>Event(?e), during(?e, ?i), overlaps(?i, Req4- DTI3_1), cancela(?e, Reservacion_prestamo_Heidi), cancela(Valentino, ?e) -> Nothing(Valentino)</i>
	<i>Event(?e), during(?e, ?i), overlaps(?i, Req4- DTI3_2), actualiza(?e, Libro1), actualiza(Valentino, ?e) -> Nothing(Valentino)</i>
A9	<i>Event(?e), during(?e, ?i), overlaps(?i, Req4- DTI9_1), cancela(?e, Reservacion_prestamo_Santiago), cancela(Valentino, ?e) -> Nothing(Valentino)</i>
	<i>Event(?e), during(?e, ?i), overlaps(?i, Req4- DTI9_2), actualiza(?e, Libro3), actualiza(Valentino, ?e) -> Nothing(Valentino)</i>

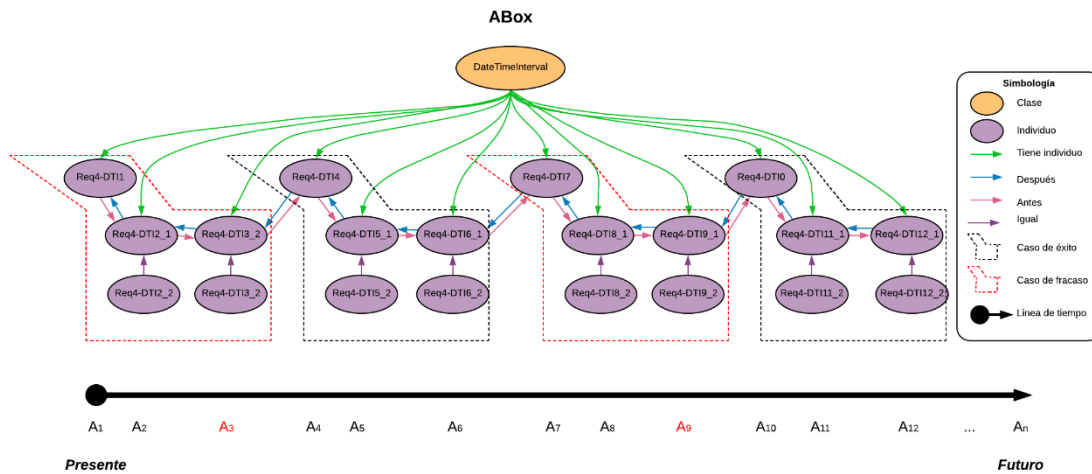


Figura D.16 SoIs en la ABox de casos de éxito y fracaso de acuerdo al RF4. Cancelar una reservación.

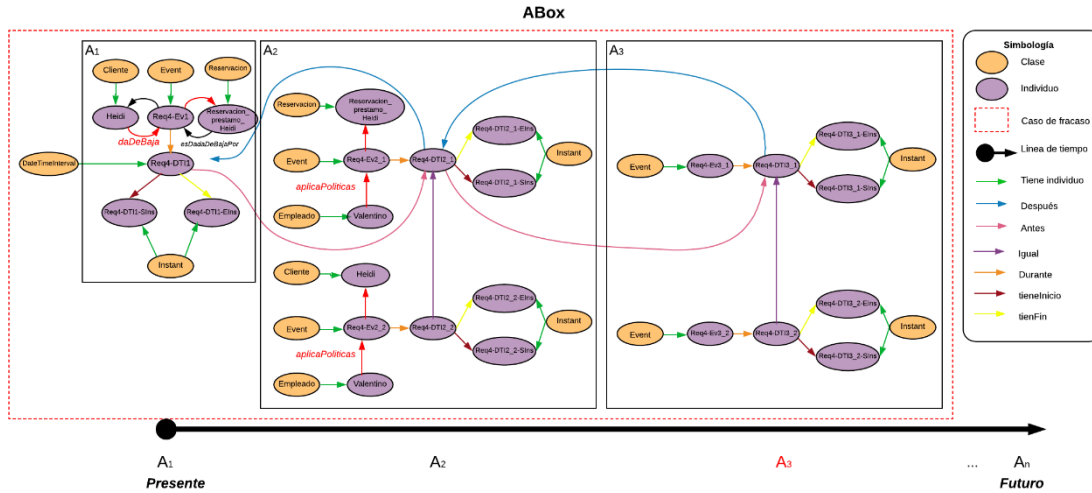


Figura D.17 Caso de fracaso 1 en la ABox de casos de éxito y fracaso de acuerdo al RF4. Cancelar una reservación.

1.2.5 RF5. Crear un contrato

Las secuencias de interpretaciones descritas en esta sección satisfacen/cumplen con el comportamiento dinámico del RF5. Crear un contrato representado formalmente con la fórmula de la Tabla D.21.

Tabla D.21 Fórmula ALC_{LTL} del RF5. Crear un contrato.

$$\mathcal{G} (\mathcal{X} (\text{Cliente} \sqsubseteq \exists \text{reserva. Entidad}) \rightarrow \mathcal{F} (\neg \mathcal{X} \mathcal{X} (\text{Empleado} \sqsubseteq \exists \text{crea. Reservacion}) \wedge \neg \mathcal{X} \mathcal{X} \mathcal{X} (\text{Empleado} \sqsubseteq \exists \text{expide. Contrato}) \wedge \neg \mathcal{X} \mathcal{X} \mathcal{X} \mathcal{X} (\text{Contrato} \sqsubseteq \exists \text{esAdquiridoPor. Cliente}))))$$

1.2.5.1. Casos de éxito

En la Tabla D.22 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la ABox) basadas en casos de éxito, que conforman una secuencia que representa el comportamiento dinámico para crear un contrato. Una representación gráfica simplificada de la secuencia se muestra en la Figura D.18. A manera de ejemplo, en la Figura D.19 se muestra el detalle de las aserciones A_1 , A_2 , A_3 y A_4 en el caso de éxito número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de razonamiento automatizado, los demás casos se definen de manera similar. Sin embargo, difieren en los individuos involucrados.

Tabla D.22 ABox de casos de éxito en el RF5. Crear un contrato. (cont.)

Sintaxis DL	Sintaxis Funcional
Caso de éxito 1 (A₁, A₂, A₃, A₄):	
A₁: [X (Cliente \sqsubseteq \exists reserva.Entidad)]: El cliente Heidi reserva la entidad Libro1.	
A ₁ = {	
-- Aserciones de clase --	
<i>Cliente(Heidi)</i>	⇒ <i>ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Heidi)</i>
<i>Entidad(Libro1)</i>	⇒ <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro1)</i>
<i>Event(Req5-Ev1)</i>	⇒ <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req5-Ev1)</i>
<i>DateTimeInterval(Req5-DT11)</i>	⇒ <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req5-DT11)</i>
<i>Instant(Req5-DT11-Sins)</i>	⇒ <i>ClassAssertion(time:Instant domain-PAIyER:Req5-DT11-Sins)</i>
<i>Instant(Req5-DT11-Eins)</i>	⇒ <i>ClassAssertion(time:Instant domain-PAIyER:Req5-DT11-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>reserva(Heidi,Req5-Ev1)</i>	⇒ <i>ObjectPropertyAssertion(domain-PAIyER: reserva domain-PAIyER:Heidi domain-PAIyER:Req5-Ev1)</i>
<i>reserva(Req5-Ev1,Libro1)</i>	⇒ <i>ObjectPropertyAssertion(domain-PAIyER: reserva domain-PAIyER:Req5-Ev1 domain-PAIyER:Libro1)</i>
<i>during(Req5-Ev1,Req5-DT11)</i>	⇒ <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req5-Ev1 domain-PAIyER:Req5-DT11)</i>
<i>hasBeginning(Req5-DT11,Req5-DT11-Sins)</i>	⇒ <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req5-DT11 domain-PAIyER:Req5-DT11-Sins)</i>
<i>hasEnd(Req5-DT11,Req5-DT11-Eins)</i>	⇒ <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req5-DT11 domain-PAIyER:Req5-DT11-Eins)</i>
}	⇒
A₂: [XX (Empleado \sqsubseteq \exists crea.Reservacion)]: El empleado Valentino crea la reservación Reservacion_prestamo_Heidi.	
A ₂ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	⇒ <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Reservacion(Reservacion_prestamo_Heidi)</i>	⇒ <i>ClassAssertion(domain-PAIyER:Reservacion domain-PAIyER:Reservacion_prestamo_Heidi)</i>
<i>Event(Req5-Ev2)</i>	⇒ <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req5-Ev2)</i>
<i>DateTimeInterval(Req5-DT12)</i>	⇒ <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req5-DT12)</i>
<i>Instant(Req5-DT12-Sins)</i>	⇒ <i>ClassAssertion(time:Instant domain-PAIyER:Req5-DT12-Sins)</i>
<i>Instant(Req5-DT12-Eins)</i>	⇒ <i>ClassAssertion(time:Instant domain-PAIyER:Req5-DT12-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>crea(Valentino,Req5-Ev2)</i>	⇒ <i>ObjectPropertyAssertion(domain-PAIyER: crea domain-PAIyER:Valentino domain-PAIyER:Req5-Ev2)</i>
<i>crea(Req5-Ev2, Reservacion_prestamo_Heidi)</i>	⇒ <i>ObjectPropertyAssertion(domain-PAIyER: crea domain-PAIyER:Req5-Ev2 domain-PAIyER:Reservacion_prestamo_Heidi)</i>
<i>during(Req5-Ev2,Req5-DT12)</i>	⇒ <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req5-Ev2 domain-PAIyER:Req5-DT12)</i>
<i>hasBeginning(Req5-DT12,Req5-DT12-Sins)</i>	⇒ <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req5-DT12 domain-PAIyER:Req5-DT12-Sins)</i>

Tabla D.22 ABox de casos de éxito en el RF5. Crear un contrato. (cont.)

Sintaxis DL	Sintaxis Funcional
<i>hasEnd(Req5-DTI2,Req5-DTI2-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req5-DTI2 domain-PAIyER:Req5-DTI2-Eins)</i>
}	
A₃: [XXX (Empleado \sqsubseteq \existsexpide.Contrato)]: El empleado Valentino expide el contrato Contrato_Heidi.	
A ₃ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Contrato(Contrato_Heidi)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Contrato domain-PAIyER:Contrato_Heidi)</i>
<i>Event(Req5-Ev3)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req5-Ev3)</i>
<i>DateTimeInterval(Req5-DTI3)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req5-DTI3)</i>
<i>Instant(Req5-DTI3-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req5-DTI3-Sins)</i>
<i>Instant(Req5-DTI3-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req5-DTI3-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>expide(Valentino,Req5-Ev3)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER: expide domain-PAIyER:Valentino domain-PAIyER:Req5-Ev3)</i>
<i>expide (Req5-Ev3, Contrato_Heidi)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER: expide domain-PAIyER:Req5-Ev3 domain-PAIyER:Contrato_Heidi)</i>
<i>during(Req5-Ev3,Req5-DTI3)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req5-Ev3 domain-PAIyER:Req5-DTI3)</i>
<i>hasBeginning(Req5-DTI3,Req5-DTI3-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req5-DTI3 domain-PAIyER:Req5-DTI3-Sins)</i>
<i>hasEnd(Req5-DTI3,Req5-DTI3-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req5-DTI3 domain-PAIyER:Req5-DTI3-Eins)</i>
}	
A₄: [XXXX (Contrato \sqsubseteq \existsesAdquiridoPor.Cliente)]: El contrato Contrato_Heidi es adquirido por el cliente Heidi.	
A ₄ = {	
-- Aserciones de clase --	
<i>Contrato(Contrato_Heidi)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Contrato domain-PAIyER:Contrato_Heidi)</i>
<i>Cliente(Heidi)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Heidi)</i>
<i>Event(Req5-Ev4)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req5-Ev4)</i>
<i>DateTimeInterval(Req5-DTI4)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req5-DTI4)</i>
<i>Instant(Req5-DTI4-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req5-DTI4-Sins)</i>
<i>Instant(Req5-DTI4-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req5-DTI4-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>esAdquiridoPor(Contrato_Heidi,Req5-Ev4)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:esAdquiridoPor domain-PAIyER:Contrato_Heidi domain-PAIyER:Req5-Ev4)</i>
<i>esAdquiridoPor(Req5-Ev4,Heidi)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:esAdquiridoPor domain-PAIyER:Req5-Ev4)</i>

Tabla D.22 ABox de casos de éxito en el RF5. Crear un contrato. (cont.)

Sintaxis DL	Sintaxis Funcional
<i>during</i> (Req5-Ev4,Req5-DTI4)	\Rightarrow <i>domain-PAIyER:Heidi</i> <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req5-Ev4 domain-PAIyER:Req5-DTI4)
<i>hasBeginning</i> (Req5-DTI4,Req5-DTI4-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req5-DTI4 domain-PAIyER:Req5-DTI4-Sins)
<i>hasEnd</i> (Req5-DTI4,Req5-DTI4-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req5-DTI4 domain-PAIyER:Req5-DTI4-Eins)
}	
Caso de éxito 2 (A₅, A₆, A₇, A₈):	
As: [X (Cliente \sqsubseteq \exists reserva.Entidad)]: El cliente Juan_Carlos reserva la entidad Libro2.	
A ₅ = {	
-- Aserciones de clase --	
<i>Cliente</i> (Juan_Carlos)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Cliente domain-PAIyER:Juan_Carlos)
<i>Entidad</i> (Libro2)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Entidad domain-PAIyER:Libro2)
<i>Event</i> (Req5-Ev5)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req5-Ev5)
<i>DateTimeInterval</i> (Req5-DTI5)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req5-DTI5)
<i>Instant</i> (Req5-DTI5-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req5-DTI5-Sins)
<i>Instant</i> (Req5-DTI5-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req5-DTI5-Eins)
-- Aserciones de propiedades de objeto --	
<i>reserva</i> (Juan_Carlos,Req5-Ev5)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: reserva domain-PAIyER:Juan_Carlos domain-PAIyER:Req5-Ev5)
<i>reserva</i> (Req5-Ev5,Libro2)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: reserva domain-PAIyER:Req5-Ev5 domain-PAIyER:Libro2)
<i>during</i> (Req5-Ev5,Req5-DTI5)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req5-Ev5 domain-PAIyER:Req5-DTI5)
<i>hasBeginning</i> (Req5-DTI5,Req5-DTI5-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req5-DTI5 domain-PAIyER:Req5-DTI5-Sins)
<i>hasEnd</i> (Req5-DTI5,Req5-DTI5-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req5-DTI5 domain-PAIyER:Req5-DTI5-Eins)
}	\Rightarrow
A ₆ : [XX (Empleado \sqsubseteq \exists crea.Reservacion)]: El empleado Valentino crea la reservación Reservacion_prestamo_JuanCarlos.	
A ₆ = {	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Reservacion</i> (Reservacion_prestamo_JuanCarlos)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Reservacion domain-PAIyER: Reservacion_prestamo_JuanCarlos)
<i>Event</i> (Req5-Ev6)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req5-Ev6)
<i>DateTimeInterval</i> (Req5-DTI6)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req5-DTI6)
<i>Instant</i> (Req5-DTI6-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req5-DTI6-Sins)
<i>Instant</i> (Req5-DTI6-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req5-DTI6-Eins)
-- Aserciones de propiedades de objeto --	

Tabla D.22 ABox de casos de éxito en el RF5. Crear un contrato. (cont.)

Sintaxis DL	Sintaxis Funcional
<i>crea</i> (Valentino,Req5-Ev6)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: <i>crea</i> domain-PAIyER:Valentino domain-PAIyER:Req5-Ev6)
<i>crea</i> (Req5-Ev6, <i>Reservacion_prestamo_JuanCarlos</i>)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: <i>crea</i> domain-PAIyER:Req5-Ev6 domain-PAIyER: <i>Reservacion_prestamo_JuanCarlos</i>)
<i>during</i> (Req5-Ev6,Req5-DTI6)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents: <i>during</i> domain-PAIyER:Req5-Ev6 domain-PAIyER:Req5-DTI6)
<i>hasBeginning</i> (Req5-DTI6,Req5-DTI6-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time: <i>hasBeginning</i> domain-PAIyER:Req5-DTI6 domain-PAIyER:Req5-DTI6-Sins)
<i>hasEnd</i> (Req5-DTI6,Req5-DTI6-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time: <i>hasEnd</i> domain-PAIyER:Req5-DTI6 domain-PAIyER:Req5-DTI6-Eins)
}	
A7: [XXX (Empleado \sqsubseteq \exists expide.Contrato)]: El empleado Valentino expide el contrato Contrato_JuanCarlos.	
A7 = {	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Contrato</i> (Contrato_JuanCarlos)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Contrato domain-PAIyER: Contrato_JuanCarlos)
<i>Event</i> (Req5-Ev7)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req5-Ev7)
<i>DateTimeInterval</i> (Req5-DTI7)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req5-DTI7)
<i>Instant</i> (Req5-DTI7-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req5-DTI7-Sins)
<i>Instant</i> (Req5-DTI7-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req5-DTI7-Eins)
-- Aserciones de propiedades de objeto --	
<i>expide</i> (Valentino,Req5-Ev7)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: <i>expide</i> domain-PAIyER:Valentino domain-PAIyER:Req5-Ev7)
<i>expide</i> (Req5-Ev7, <i>Contrato_JuanCarlos</i>)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: <i>expide</i> domain-PAIyER:Req5-Ev7 domain-PAIyER: <i>Contrato_JuanCarlos</i>)
<i>during</i> (Req5-Ev7,Req5-DTI7)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents: <i>during</i> domain-PAIyER:Req5-Ev7 domain-PAIyER:Req5-DTI7)
<i>hasBeginning</i> (Req5-DTI7,Req5-DTI7-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time: <i>hasBeginning</i> domain-PAIyER:Req5-DTI7 domain-PAIyER:Req5-DTI7-Sins)
<i>hasEnd</i> (Req5-DTI7,Req5-DTI7-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time: <i>hasEnd</i> domain-PAIyER:Req5-DTI7 domain-PAIyER:Req5-DTI7-Eins)
}	
A8: [XXXX (Contrato \sqsubseteq \exists esAdquiridoPor.Cliente)]: El contrato Contrato_JuanCarlos es adquirido por el cliente Juan_Carlos.	
A8 = {	
-- Aserciones de clase --	
<i>Contrato</i> (Contrato_JuanCarlos)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Contrato domain-PAIyER:Contrato_JuanCarlos)
<i>Cliente</i> (Juan_Carlos)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Cliente domain-PAIyER:Juan_Carlos)
<i>Event</i> (Req5-Ev8)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req5-Ev8)
<i>DateTimeInterval</i> (Req5-DTI8)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req5-DTI8)
<i>Instant</i> (Req5-DTI8-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req5-

Tabla D.22 ABox de casos de éxito en el RF5. Crear un contrato. (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Instant</i> (Req5-DTI8-Eins)	<i>DTI8-Sins</i> ⇒ <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req5-DTI8-Eins)
-- Aserciones de propiedades de objeto --	
<i>esAdquiridoPor</i> (Contrato_JuanCarlos,Req5-Ev8)	⇒ <i>ObjectPropertyAssertion</i> (domain-PAIyER:esAdquiridoPor domain-PAIyER:Contrato_JuanCarlos domain-PAIyER:Req5-Ev8)
<i>esAdquiridoPor</i> (Req5-Ev8,Juan_Carlos)	⇒ <i>ObjectPropertyAssertion</i> (domain-PAIyER:esAdquiridoPor domain-PAIyER:Req5-Ev8 domain-PAIyER:Juan_Carlos)
<i>during</i> (Req5-Ev8,Req5-DTI8)	⇒ <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req5-Ev8 domain-PAIyER:Req5-DTI8)
<i>hasBeginning</i> (Req5-DTI8,Req5-DTI8-Sins)	⇒ <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req5-DTI8 domain-PAIyER:Req5-DTI8-Sins)
<i>hasEnd</i> (Req5-DTI8,Req5-DTI8-Eins)	⇒ <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req5-DTI8 domain-PAIyER:Req5-DTI8-Eins)
}	
Caso de éxito 3 (A₉, A₁₀, A₁₁, A₁₂):	
A ₉ : [X (Cliente ⊆ ∃reserva.Entidad)]: El cliente Santiago reserva la entidad Libro3.	
...	
A ₁₀ : [XX (Empleado ⊆ ∃crea.Reservacion)]: El empleado Valentino crea la reservación Reservacion_prestamo_Santiago.	
...	
A ₁₁ : [XXX (Empleado ⊆ ∃expide.Contrato)]: El empleado Valentino expide el contrato Contrato_Santiago.	
...	
A ₁₂ : [XXXX (Contrato ⊆ ∃esAdquiridoPor.Cliente)]: El contrato Contrato_Santiago es adquirido por el cliente Santiago.	
...	
Caso de éxito 4 (A₁₃, A₁₄, A₁₅, A₁₆):	
A ₁₃ : [X (Cliente ⊆ ∃reserva.Entidad)]: El cliente Violeta reserva la entidad Libro4.	
...	
A ₁₄ : [XX (Empleado ⊆ ∃crea.Reservacion)]: El empleado Valentino crea la reservación Reservacion_prestamo_Violeta.	
...	
A ₁₅ : [XXX (Empleado ⊆ ∃expide.Contrato)]: El empleado Valentino expide el contrato Contrato_Violeta.	
...	
A ₁₆ : [XXXX (Contrato ⊆ ∃esAdquiridoPor.Cliente)]: El contrato Contrato_Violeta es adquirido por el cliente Violeta.	
...	
Caso de éxito n	
A _n	
...	

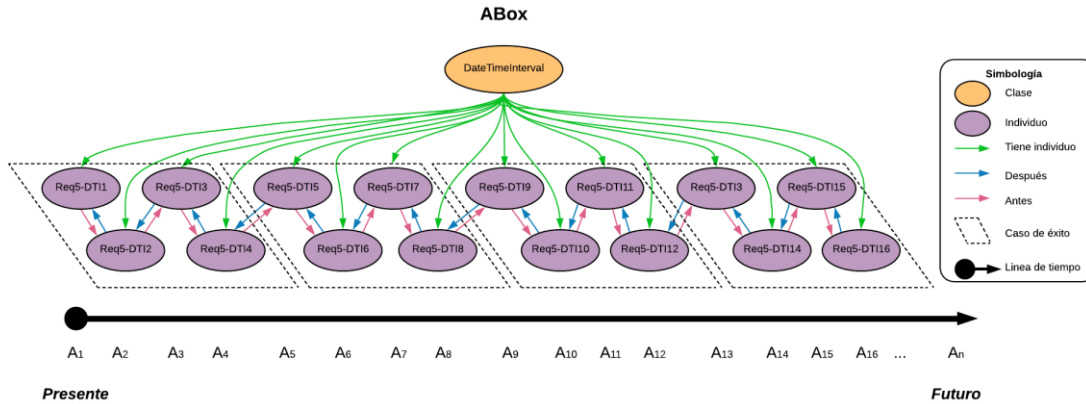


Figura D.18 SoIs en la ABox de casos de éxito de acuerdo al RF5. Crear un contrato.

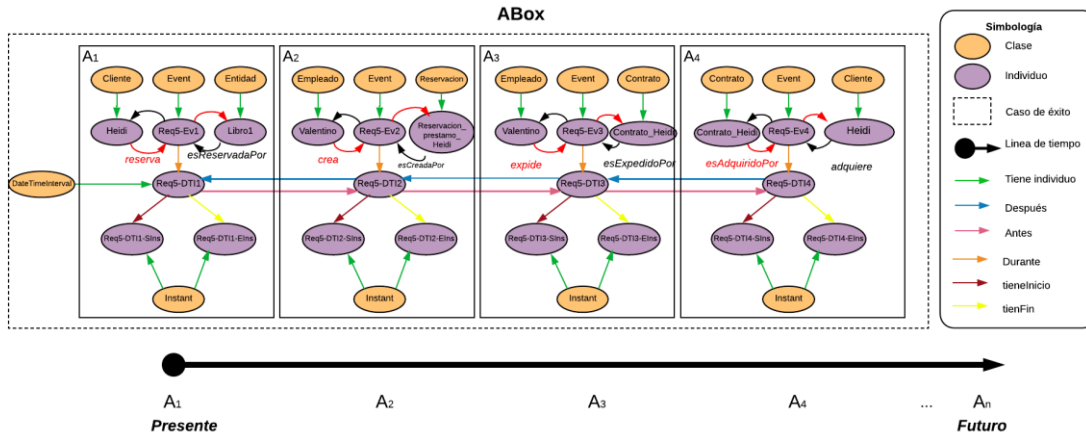


Figura D.19 Caso de éxito 1 en la ABox de casos de éxito de acuerdo al RF5. Crear un contrato.

1.2.5.2. Casos de éxito y fracaso

En la Tabla D.23 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la *ABox*) basadas en casos de éxito y fracaso, que conforman una secuencia que representa el comportamiento dinámico para crear un contrato. Una representación gráfica simplificada de la secuencia se muestra en la Figura D.20. A manera de ejemplo, en la Figura D.21 se muestra el detalle de las aserciones A_1 , A_2 , A_3 y A_4 en el caso de fracaso número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de razonamiento automatizado los demás casos se definen de manera similar, sin embargo, difieren en los individuos involucrados.

En la Tabla D.24 se describen las reglas SWRL utilizadas para restringir la existencia de afirmaciones de propiedades de objeto en el intervalo de tiempo correspondiente.

Tabla D.23 ABox de casos de éxito y fracaso en el RF5. Crear un contrato. (cont.)

Sintaxis DL	Sintaxis Funcional
Caso de fracaso 1 (A₁, A₂, A₃, A₄):	
A₁: [X (Cliente \sqsubseteq \existsreserva.Entidad)]: El cliente Heidi reserva la entidad Libro1.	
A ₁ = {	
-- Aserciones de clase --	
Cliente(Heidi)	⇒ ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Heidi)
Entidad(Libro1)	⇒ ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro1)
Event(Req5-Ev1)	⇒ ClassAssertion(timeEvents:Event domain-PAIyER:Req5-Ev1)
DateTimeInterval(Req5-DTI1)	⇒ ClassAssertion(time:DateTimeInterval domain-PAIyER:Req5-DTI1)
Instant(Req5-DTI1-Sins)	⇒ ClassAssertion(time:Instant domain-PAIyER:Req5-DTI1-Sins)
Instant(Req5-DTI1-Eins)	⇒ ClassAssertion(time:Instant domain-PAIyER:Req5-DTI1-Eins)
-- Aserciones de propiedades de objeto --	
reserva(Heidi,Req5-Ev1)	⇒ ObjectPropertyAssertion(domain-PAIyER: reserva domain-PAIyER:Heidi domain-PAIyER:Req5-Ev1)
reserva(Req5-Ev1,Libro1)	⇒ ObjectPropertyAssertion(domain-PAIyER: reserva domain-PAIyER:Req5-Ev1 domain-PAIyER:Libro1)
during(Req5-Ev1,Req5-DTI1)	⇒ ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req5-Ev1 domain-PAIyER:Req5-DTI1)
hasBeginning(Req5-DTI1,Req5-DTI1-Sins)	⇒ ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req5-DTI1 domain-PAIyER:Req5-DTI1-Sins)
hasEnd(Req5-DTI1,Req5-DTI1-Eins)	⇒ ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req5-DTI1 domain-PAIyER:Req5-DTI1-Eins)
}	⇒
A₂: [\negXX (Empleado \sqsubseteq \existscrea.Reservacion)]: El empleado Valentino no crea la reservación Reservacion_prestamo_Heidi.	
A ₂ = {	
-- Aserciones de clase --	
Empleado(Valentino)	⇒ ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)
Reservacion(Reservacion_prestamo_Heidi)	⇒ ClassAssertion(domain-PAIyER:Reservacion domain-PAIyER:Reservacion_prestamo_Heidi)
Event(Req5-Ev2)	⇒ ClassAssertion(timeEvents:Event domain-PAIyER:Req5-Ev2)
DateTimeInterval(Req5-DTI2)	⇒ ClassAssertion(time:DateTimeInterval domain-PAIyER:Req5-DTI2)
Instant(Req5-DTI2-Sins)	⇒ ClassAssertion(time:Instant domain-PAIyER:Req5-DTI2-Sins)
Instant(Req5-DTI2-Eins)	⇒ ClassAssertion(time:Instant domain-PAIyER:Req5-DTI2-Eins)
-- Aserciones de propiedades de objeto --	
crea(Valentino,Req5-Ev2)	⇒ ObjectPropertyAssertion(domain-PAIyER: crea domain-PAIyER:Valentino domain-PAIyER:Req5-Ev2)
crea(Req5-Ev2,Reservacion_prestamo_Heidi)	⇒ ObjectPropertyAssertion(domain-PAIyER: crea domain-PAIyER:Req5-Ev2 domain-PAIyER:Reservacion_prestamo_Heidi)
during(Req5-Ev2,Req5-DTI2)	⇒ ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req5-Ev2 domain-PAIyER:Req5-DTI2)

Tabla D.23 ABox de casos de éxito y fracaso en el RF5. Crear un contrato. (cont.)

Sintaxis DL	Sintaxis Funcional
<i>hasBeginning(Req5-DTI2,Req5-DTI2-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req5-DTI2 domain-PAIyER:Req5-DTI2-Sins)</i>
<i>hasEnd(Req5-DTI2,Req5-DTI2-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req5-DTI2 domain-PAIyER:Req5-DTI2-Eins)</i>
}	
A₃: [\negXXX (Empleado \sqsubseteq \existsexpide.Contrato)]: El empleado Valentino no expide el contrato Contrato_Heidi.	
A ₃ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Contrato(Contrato_Heidi)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Contrato domain-PAIyER:Contrato_Heidi)</i>
<i>Event(Req5-Ev3)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req5-Ev3)</i>
<i>DateTimeInterval(Req5-DTI3)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req5-DTI3)</i>
<i>Instant(Req5-DTI3-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req5-DTI3-Sins)</i>
<i>Instant(Req5-DTI3-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req5-DTI3-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>expide(Valentino,Req5-Ev3)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER: expide domain-PAIyER:Valentino domain-PAIyER:Req5-Ev3)</i>
<i>expide(Req5-Ev3, Contrato_Heidi)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER: expide domain-PAIyER:Req5-Ev3 domain-PAIyER: Contrato_Heidi)</i>
<i>during(Req5-Ev3,Req5-DTI3)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req5-Ev3 domain-PAIyER:Req5-DTI3)</i>
<i>hasBeginning(Req5-DTI3,Req5-DTI3-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req5-DTI3 domain-PAIyER:Req5-DTI3-Sins)</i>
<i>hasEnd(Req5-DTI3,Req5-DTI3-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req5-DTI3 domain-PAIyER:Req5-DTI3-Eins)</i>
}	
A₄: [\negXXXX (Contrato \sqsubseteq \existsesAdquiridoPor.Cliente)]: El contrato Contrato_Heidi no es adquirido por el cliente Heidi.	
A ₄ = {	
-- Aserciones de clase --	
<i>Contrato(Contrato_Heidi)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Contrato domain-PAIyER:Contrato_Heidi)</i>
<i>Cliente(Heidi)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Heidi)</i>
<i>Event(Req5-Ev4)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req5-Ev4)</i>
<i>DateTimeInterval(Req5-DTI4)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req5-DTI4)</i>
<i>Instant(Req5-DTI4-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req5-DTI4-Sins)</i>
<i>Instant(Req5-DTI4-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req5-DTI4-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>esAdquiridoPor(Contrato_Heidi, Req5-Ev4)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:esAdquiridoPor domain-PAIyER: Contrato_Heidi domain-PAIyER:Req5-Ev4)</i>

Tabla D.23 ABox de casos de éxito y fracaso en el RF5. Crear un contrato. (cont.)

Sintaxis DL	Sintaxis Funcional
<i>esAdquiridoPor</i> (Req5-Ev4,Heidi)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:esAdquiridoPor domain-PAIyER:Req5-Ev4 domain-PAIyER:Heidi)
<i>during</i> (Req5-Ev4,Req5-DTI4)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req5-Ev4 domain-PAIyER:Req5-DTI4)
<i>hasBeginning</i> (Req5-DTI4,Req5-DTI4-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req5-DTI4 domain-PAIyER:Req5-DTI4-Sins)
<i>hasEnd</i> (Req5-DTI4,Req5-DTI4-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req5-DTI4 domain-PAIyER:Req5-DTI4-Eins)
}	
Caso de éxito 1 (A₅, A₆, A₇, A₈):	
A₅: [X (Cliente \sqsubseteq \exists reserva.Entidad)]: El cliente Juan_Carlos reserva la entidad Libro2.	
A ₅ = {	
-- Aserciones de clase --	
<i>Cliente</i> (Juan_Carlos)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Cliente domain-PAIyER:Juan_Carlos)
<i>Entidad</i> (Libro2)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Entidad domain-PAIyER:Libro2)
<i>Event</i> (Req5-Ev5)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req5-Ev5)
<i>DateTimeInterval</i> (Req5-DTI5)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req5-DTI5)
<i>Instant</i> (Req5-DTI5-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req5-DTI5-Sins)
<i>Instant</i> (Req5-DTI5-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req5-DTI5-Eins)
-- Aserciones de propiedades de objeto --	
<i>reserva</i> (Juan_Carlos,Req5-Ev5)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: reserva domain-PAIyER:Juan_Carlos domain-PAIyER:Req5-Ev5)
<i>reserva</i> (Req5-Ev5,Libro2)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: reserva domain-PAIyER:Req5-Ev5 domain-PAIyER:Libro2)
<i>during</i> (Req5-Ev5,Req5-DTI5)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req5-Ev5 domain-PAIyER:Req5-DTI5)
<i>hasBeginning</i> (Req5-DTI5,Req5-DTI5-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req5-DTI5 domain-PAIyER:Req5-DTI5-Sins)
<i>hasEnd</i> (Req5-DTI5,Req5-DTI5-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req5-DTI5 domain-PAIyER:Req5-DTI5-Eins)
}	\Rightarrow
A₆: [XX (Empleado \sqsubseteq \exists crea.Reservacion)]: El empleado Valentino crea la reservación Reservacion_prestamo_JuanCarlos.	
A ₆ = {	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Reservacion</i> (Reservacion_prestamo_JuanCarlos)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Reservacion domain-PAIyER: Reservacion_prestamo_JuanCarlos)
<i>Event</i> (Req5-Ev6)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req5-Ev6)
<i>DateTimeInterval</i> (Req5-DTI6)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req5-DTI6)
<i>Instant</i> (Req5-DTI6-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req5-DTI6-Sins)

Tabla D.23 ABox de casos de éxito y fracaso en el RF5. Crear un contrato. (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Instant(Req5-DTI6-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req5-DTI6-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>crea(Valentino,Req5-Ev6)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER: crea domain-PAIyER:Valentino domain-PAIyER:Req5-Ev6)</i>
<i>crea(Req5-Ev6, Reservacion_prestamo_JuanCarlos)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER: crea domain-PAIyER:Req5-Ev6 domain-PAIyER: Reservacion_prestamo_JuanCarlos)</i>
<i>during(Req5-Ev6,Req5-DTI6)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req5-Ev6 domain-PAIyER:Req5-DTI6)</i>
<i>hasBeginning(Req5-DTI6,Req5-DTI6-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req5-DTI6 domain-PAIyER:Req5-DTI6-Sins)</i>
<i>hasEnd(Req5-DTI6,Req5-DTI6-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req5-DTI6 domain-PAIyER:Req5-DTI6-Eins)</i>
}	
A₇: [XXX (Empleado \sqsubseteq \existsexpide.Contrato)]: El empleado Valentino expide el contrato Contrato_JuanCarlos.	
A ₇ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Contrato(Contrato_JuanCarlos)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Contrato domain-PAIyER: Contrato_JuanCarlos)</i>
<i>Event(Req5-Ev7)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req5-Ev7)</i>
<i>DateTimeInterval(Req5-DTI7)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req5-DTI7)</i>
<i>Instant(Req5-DTI7-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req5-DTI7-Sins)</i>
<i>Instant(Req5-DTI7-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req5-DTI7-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>expide(Valentino,Req5-Ev7)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER: expide domain-PAIyER:Valentino domain-PAIyER:Req5-Ev7)</i>
<i>expide (Req5-Ev7, Contrato_JuanCarlos)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER: expide domain-PAIyER:Req5-Ev7 domain-PAIyER: Contrato_JuanCarlos)</i>
<i>during(Req5-Ev7,Req5-DTI7)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req5-Ev7 domain-PAIyER:Req5-DTI7)</i>
<i>hasBeginning(Req5-DTI7,Req5-DTI7-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req5-DTI7 domain-PAIyER:Req5-DTI7-Sins)</i>
<i>hasEnd(Req5-DTI7,Req5-DTI7-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req5-DTI7 domain-PAIyER:Req5-DTI7-Eins)</i>
}	
A₈: [XXXX (Contrato \sqsubseteq \existsesAdquiridoPor.Cliente)]: El contrato Contrato_JuanCarlos es adquirido por el cliente Juan_Carlos.	
A ₈ = {	
-- Aserciones de clase --	
<i>Contrato(Contrato_JuanCarlos)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Contrato domain-PAIyER: Contrato_JuanCarlos)</i>
<i>Cliente(Juan_Carlos)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Juan_Carlos)</i>
<i>Event(Req5-Ev8)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req5-Ev8)</i>

Tabla D.23 ABox de casos de éxito y fracaso en el RF5. Crear un contrato. (cont.)

Sintaxis DL	Sintaxis Funcional
<i>DateTimeInterval(Req5-DTI8)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req5-DTI8)</i>
<i>Instant(Req5-DTI8-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req5-DTI8-Sins)</i>
<i>Instant(Req5-DTI8-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req5-DTI8-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>esAdquiridoPor(Contrato_JuanCarlos,Req5-Ev8)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:esAdquiridoPor domain-PAIyER:Contrato_JuanCarlos domain-PAIyER:Req5-Ev8)</i>
<i>esAdquiridoPor(Req5-Ev8,Juan_Carlos)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:esAdquiridoPor domain-PAIyER:Req5-Ev8 domain-PAIyER:Juan_Carlos)</i>
<i>during(Req5-Ev8,Req5-DTI8)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req5-Ev8 domain-PAIyER:Req5-DTI8)</i>
<i>hasBeginning(Req5-DTI8,Req5-DTI8-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req5-DTI8 domain-PAIyER:Req5-DTI8-Sins)</i>
<i>hasEnd(Req5-DTI8,Req5-DTI8-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req5-DTI8 domain-PAIyER:Req5-DTI8-Eins)</i>
}	
Caso de fracaso 2 (A9, A10, A11, A12):	
<i>A9: [X (Cliente \sqsubseteq \existsreserva.Entidad)]: El cliente Santiago reserva la entidad Libro3.</i>	
...	
<i>A10: [\negXX (Empleado \sqsubseteq \existscrea.Reservacion)]: El empleado Valentino no crea la reservación Reservacion_prestamo_Santiago.</i>	
...	
<i>A11: [\negXXX (Empleado \sqsubseteq \existsexpide.Contrato)]: El empleado Valentino no expide el contrato Contrato_Santiago.</i>	
...	
<i>A12: [\negXXXX (Contrato \sqsubseteq \existsesAdquiridoPor.Cliente)]: El contrato Contrato_Santiago no es adquirido por el cliente Santiago.</i>	
...	
Caso de éxito 2 (A13, A14, A15, A16):	
<i>A13: [X (Cliente \sqsubseteq \existsreserva.Entidad)]: El cliente Violeta reserva la entidad Libro4.</i>	
...	
<i>A14: [XX (Empleado \sqsubseteq \existscrea.Reservacion)]: El empleado Valentino crea la reservación Reservacion_prestamo_Violeta.</i>	
...	
<i>A15: [XXX (Empleado \sqsubseteq \existsexpide.Contrato)]: El empleado Valentino expide el contrato Contrato_Violeta.</i>	
...	
<i>A16: [XXXX (Contrato \sqsubseteq \existsesAdquiridoPor.Cliente)]: El contrato Contrato_Violeta es adquirido por el cliente Violeta.</i>	
...	
Caso de fracaso y de éxito n	
<i>A_n</i>	
...	

Tabla D.24 Reglas SWRL para la SoIs de casos de éxito y fracaso de acuerdo al RF5. Crear un contrato (cont.)

ABox	Regla SWRL
A ₂	<i>Event(?e), during(?e, ?i), overlaps(?i, Req5-DT12), crea(?e, Reservacion_prestamo_Heidi), crea(Valentino, ?e) -> Nothing(Valentino)</i>
A ₃	<i>Event(?e), during(?e, ?i), overlaps(?i, Req5-DT13), expide(?e, Contrato_Heidi), expide(Valentino, ?e) -> Nothing(Valentino)</i>
A ₄	<i>Event(?e), during(?e, ?i), overlaps(?i, Req5-DT14), esAdquiridoPor(?e, Heidi), esAdquiridoPor(Contrato_Heidi, ?e) -> Nothing(Contrato_Heidi)</i>
A ₁₀	<i>Event(?e), during(?e, ?i), overlaps(?i, Req5-DT110), crea(?e, Reservacion_prestamo_Santiago), crea(Valentino, ?e) -> Nothing(Valentino)</i>
A ₁₁	<i>Event(?e), during(?e, ?i), overlaps(?i, Req5-DT111), expide(?e, Contrato_Santiago), expide(Valentino, ?e) -> Nothing(Valentino)</i>
A ₁₂	<i>Event(?e), during(?e, ?i), overlaps(?i, Req5-DT112), esAdquiridoPor(?e, Santiago), esAdquiridoPor(Contrato_Santiago, ?e) -> Nothing(Contrato_Santiago)</i>

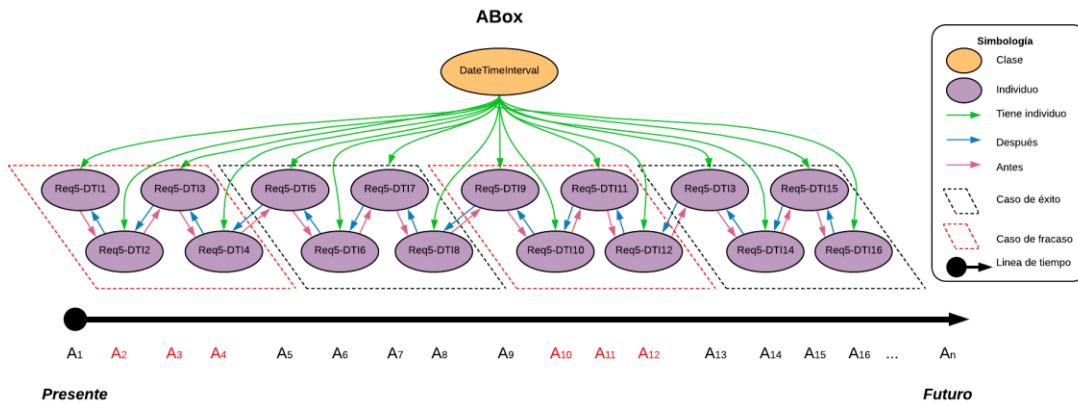


Figura D.20 SoIs en la ABox de casos de éxito y fracaso de acuerdo al RF5. Crear un contrato.

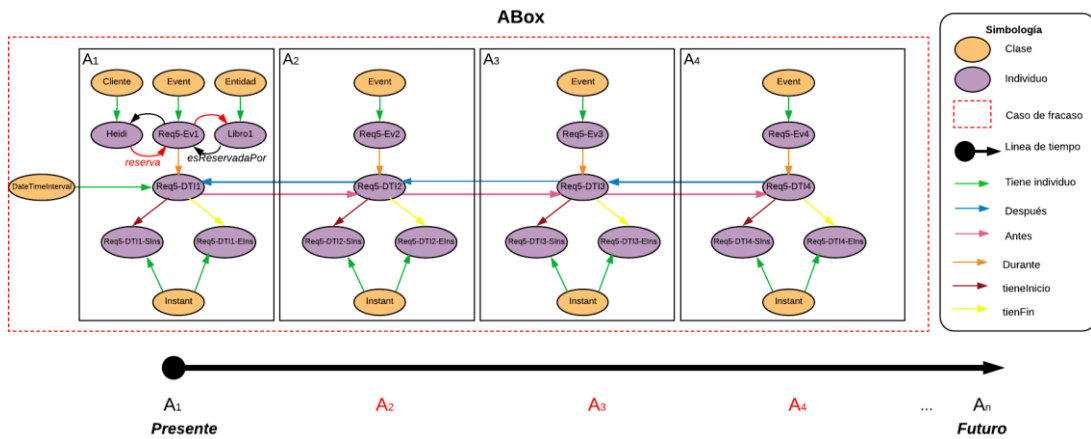


Figura D.21 Caso de fracaso 1 en la ABox de casos de éxito y fracaso de acuerdo al RF5. Crear un contrato.

1.2.6 RF6. Separar diferentes tipos de existencias

Las secuencias de interpretaciones descritas en esta sección satisfacen/cumplen con el comportamiento dinámico del RF6. Separar diferentes tipos de existencias representado formalmente con la fórmula de la Tabla D.25.

Tabla D.25 Fórmula ALC_{LTL} del RF6. Separar diferentes tipos de existencias.

$\mathcal{G} ($ $\mathcal{X} (Empleado \sqsubseteq \exists \text{separa. Entidad}) \rightarrow ($ $\mathcal{X}\mathcal{X} (Empleado \sqsubseteq \exists \text{verificaExistenciaDe. Entidad}) \wedge \mathcal{F} ($ $\neg \mathcal{X}\mathcal{X}\mathcal{X} (Empleado \sqsubseteq \exists \text{actualiza. Entidad} \vee (Empleado \sqsubseteq \exists \text{registra. Entidad})) \wedge$ $\neg \mathcal{X}\mathcal{X}\mathcal{X}\mathcal{X} (Empleado \sqsubseteq \exists \text{actualiza. Inventario})$ $)$ $)$ $)$

1.2.6.1. Casos de éxito

En la Tabla D.26 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la *ABox*) basadas en casos de éxito, que conforman una secuencia que representa el comportamiento dinámico en la separación de diferentes tipos de entidades para un inventario. Una representación gráfica simplificada de la secuencia se muestra en la Figura D.22. A manera de ejemplo, en la Figura D.23 se muestra el detalle de las aserciones A_1 , A_2 , A_3 y A_4 en el caso de éxito número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de razonamiento automatizado, los demás casos se definen de manera similar. Sin embargo, difieren en los individuos involucrados.

Tabla D.26 *ABox* de casos de éxito en el RF6. Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
Caso de éxito 1 (A_1, A_2, A_3, A_4):	
$A_1: [\mathcal{X}(Empleado \sqsubseteq \exists \text{separa. Entidad})]:$ El empleado Valentino separa el Entidad Libro1.	
$A_1 = \{$	
-- Aserciones de clase --	
$Empleado(Valentino)$	\Rightarrow $ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Valentino)$
$Entidad(Libro1)$	\Rightarrow $ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro1)$
$Event(Req6-Ev1)$	\Rightarrow $ClassAssertion(timeEvents:Event domain-PAIyER:Req6-Ev1)$
$DateTimeInterval(Req6-DT11)$	\Rightarrow $ClassAssertion(time:DateTimeInterval domain-PAIyER:Req6-DT11)$
$Instant(Req6-DT11-Sins)$	\Rightarrow $ClassAssertion(time:Instant domain-PAIyER:Req6-DT11-Sins)$
$Instant(Req6-DT11-Eins)$	\Rightarrow $ClassAssertion(time:Instant domain-PAIyER:Req6-DT11-Eins)$
-- Aserciones de propiedades de objeto --	
$\text{separa}(Valentino, Req6-Ev1)$	\Rightarrow $ObjectPropertyAssertion(domain-PAIyER:separa domain-PAIyER:Valentino domain-PAIyER:Req6-Ev1)$

Tabla D.26 ABox de casos de éxito en el RF6. Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
<i>separa</i> (Req6-Ev1, Libro1)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: <i>separa</i> domain-PAIyER:Req6-Ev1 domain-PAIyER:Libro1)
<i>during</i> (Req6-Ev1, Req6-DTI1)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents: <i>during</i> domain-PAIyER:Req6-Ev1 domain-PAIyER:Req6-DTI1)
<i>hasBeginning</i> (Req6-DTI1, Req6-DTI1-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time: <i>hasBeginning</i> domain-PAIyER:Req6-DTI1 domain-PAIyER:Req6-DTI1-Sins)
<i>hasEnd</i> (Req6-DTI1, Req6-DTI1-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time: <i>hasEnd</i> domain-PAIyER:Req6-DTI1 domain-PAIyER:Req6-DTI1-Eins)
}	\Rightarrow
$A_2 = \{$	A₂: [XX (Empleado \sqsubseteq \exists verificaExistenciaDe.Entidad)]: El empleado Valentino verifica la existencia de la entidad Libro1
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Entidad</i> (Libro1)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Entidad domain-PAIyER:Libro1)
<i>Event</i> (Req6-Ev2)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req6-Ev2)
<i>DateTimeInterval</i> (Req6-DTI2)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req6-DTI2)
<i>Instant</i> (Req6-DTI2-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req6-DTI2-Sins)
<i>Instant</i> (Req6-DTI2-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req6-DTI2-Eins)
-- Aserciones de propiedades de objeto --	
<i>verificaExistenciaDe</i> (Valentino, Req6-Ev2)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: <i>verificaExistenciaDe</i> domain-PAIyER:Valentino domain-PAIyER:Req6-Ev2)
<i>verificaExistenciaDe</i> (Req6-Ev2, Libro1)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: <i>verificaExistenciaDe</i> domain-PAIyER:Req6-Ev2 domain-PAIyER:Libro1)
<i>during</i> (Req6-Ev2, Req6-DTI2)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents: <i>during</i> domain-PAIyER:Req6-Ev2 domain-PAIyER:Req6-DTI2)
<i>hasBeginning</i> (Req6-DTI2, Req6-DTI2-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time: <i>hasBeginning</i> domain-PAIyER:Req6-DTI2 domain-PAIyER:Req6-DTI2-Sins)
<i>hasEnd</i> (Req6-DTI2, Req6-DTI2-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time: <i>hasEnd</i> domain-PAIyER:Req6-DTI2 domain-PAIyER:Req6-DTI2-Eins)
}	
$A_3 = \{$	A₃: [XXX((Empleado \sqsubseteq \exists actualiza.Entidad) \vee (Empleado \sqsubseteq \exists registra.Entidad))]: El empleado Valentino actualiza la entidad Libro1 o registra la entidad Libro1.
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Entidad</i> (Libro1)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Entidad domain-PAIyER:Libro1)
<i>Event</i> (Req6-Ev3)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req6-Ev3)
<i>DateTimeInterval</i> (Req6-DTI3)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req6-DTI3)
<i>Instant</i> (Req6-DTI3-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req6-DTI3-Sins)

Tabla D.26 ABox de casos de éxito en el RF6. Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Instant(Req6-DTI3-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI3-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>registra(Valentino,Req6-Ev3)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:registra domain-PAIyER:Valentino domain-PAIyER:Req6-Ev3)</i>
<i>registra(Req6-Ev3,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:registra domain-PAIyER:Req6-Ev3 domain-PAIyER:Libro1)</i>
<i>during(Req6-Ev3,Req6-DTI3)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req6-Ev3 domain-PAIyER:Req6-DTI3)</i>
<i>hasBeginning(Req6-DTI3,Req6-DTI3-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req6-DTI3 domain-PAIyER:Req6-DTI3-Sins)</i>
<i>hasEnd(Req6-DTI3,Req6-DTI3-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req6-DTI3 domain-PAIyER:Req6-DTI3-Eins)</i>
}	
A₄: [XXXX (Empleado \sqsubseteq \existsactualiza.Inventario)]: El empleado Valentino actualiza el inventario Inventario_CENIDET.	
A ₄ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Inventario(Inventario_CENIDET)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Inventario domain-PAIyER:Inventario_CENIDET)</i>
<i>Event(Req6-Ev4)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req6-Ev4)</i>
<i>DateTimeInterval(Req6-DTI4)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req6-DTI4)</i>
<i>Instant(Req6-DTI4-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI4-Sins)</i>
<i>Instant(Req6-DTI4-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI4-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>actualiza(Valentino,Req6-Ev4)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Valentino domain-PAIyER:Req6-Ev4)</i>
<i>actualiza(Req6-Ev4,Inventario_CENIDET)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Req6-Ev4 domain-PAIyER:cInventario_CENIDET)</i>
<i>during(Req6-Ev4,Req6-DTI4)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req6-Ev4 domain-PAIyER:Req6-DTI4)</i>
<i>hasBeginning(Req6-DTI4,Req6-DTI4-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req6-DTI4 domain-PAIyER:Req6-DTI4-Sins)</i>
<i>hasEnd(Req6-DTI4,Req6-DTI4-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req6-DTI4 domain-PAIyER:Req6-DTI4-Eins)</i>
}	
Caso de éxito 2 (A₅, A₆, A₇, A₈):	
A₅: [X(Empleado \sqsubseteq \existssepara.Entidad)]: El empleado Valentino separa el Entidad Libro2.	
A ₅ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event(Req6-Ev5)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req6-Ev5)</i>

Tabla D.26 ABox de casos de éxito en el RF6. Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
<i>DateTimeInterval(Req6-DTI5)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req6-DTI5)</i>
<i>Instant(Req6-DTI5-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI5-Sins)</i>
<i>Instant(Req6-DTI5-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI5-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>separa(Valentino,Req6-Ev5)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:separa domain-PAIyER:Valentino domain-PAIyER:Req6-Ev5)</i>
<i>separa(Req6-Ev5,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:separa domain-PAIyER:Req6-Ev5 domain-PAIyER:Libro2)</i>
<i>during(Req6-Ev5,Req6-DTI5)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req6-Ev5 domain-PAIyER:Req6-DTI5)</i>
<i>hasBeginning(Req6-DTI5,Req6-DTI5-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req6-DTI5 domain-PAIyER:Req6-DTI5-Sins)</i>
<i>hasEnd(Req6-DTI5,Req6-DTI5-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req6-DTI5 domain-PAIyER:Req6-DTI5-Eins)</i>
}	\Rightarrow
A₆: [XX (Empleado \sqsubseteq \existsverificaExistenciaDe.Entidad)]: El empleado Valentino verifica la existencia de la entidad Libro2	
A ₆ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event(Req6-Ev6)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req6-Ev6)</i>
<i>DateTimeInterval(Req6-DTI6)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req6-DTI6)</i>
<i>Instant(Req6-DTI6-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI6-Sins)</i>
<i>Instant(Req6-DTI6-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI6-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>verificaExistenciaDe(Valentino,Req6-Ev6)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaExistenciaDe domain-PAIyER:Valentino domain-PAIyER:Req6-Ev6)</i>
<i>verificaExistenciaDe(Req6-Ev6,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaExistenciaDe domain-PAIyER:Req6-Ev6 domain-PAIyER:Libro2)</i>
<i>during(Req6-Ev6,Req6-DTI6)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req6-Ev6 domain-PAIyER:Req6-DTI6)</i>
<i>hasBeginning(Req6-DTI6,Req6-DTI6-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req6-DTI6 domain-PAIyER:Req6-DTI6-Sins)</i>
<i>hasEnd(Req6-DTI6,Req6-DTI6-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req6-DTI6 domain-PAIyER:Req6-DTI6-Eins)</i>
}	
A₇: [XXX((Empleado \sqsubseteq \existsactualiza.Entidad) \vee (Empleado \sqsubseteq \existsregistra.Entidad))]: El empleado Valentino actualiza la entidad Libro2 o registra la entidad Libro2.	
A ₇ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>

Tabla D.26 ABox de casos de éxito en el RF6. Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Entidad(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event(Req6-Ev7)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req6-Ev7)</i>
<i>DateTimeInterval(Req6-DTI7)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req6-DTI7)</i>
<i>Instant(Req6-DTI7-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI7-Sins)</i>
<i>Instant(Req6-DTI7-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI7-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>actualiza(Valentino,Req6-Ev7)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Valentino domain-PAIyER:Req6-Ev7)</i>
<i>actualiza(Req6-Ev7,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Req6-Ev7 domain-PAIyER:Libro2)</i>
<i>during(Req6-Ev7,Req6-DTI7)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req6-Ev7 domain-PAIyER:Req6-DTI7)</i>
<i>hasBeginning(Req6-DTI7,Req6-DTI7-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req6-DTI7 domain-PAIyER:Req6-DTI7-Sins)</i>
<i>hasEnd(Req6-DTI7,Req6-DTI7-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req6-DTI7 domain-PAIyER:Req6-DTI7-Eins)</i>
}	
A₈: [XXXX (Empleado \sqsubseteq \existsactualiza.Inventario)]: El empleado Valentino actualiza el inventario Inventario_CENIDET.	
A ₈ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Inventario(Inventario_CENIDET)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Inventario domain-PAIyER:Inventario_CENIDET)</i>
<i>Event(Req6-Ev8)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req6-Ev8)</i>
<i>DateTimeInterval(Req6-DTI8)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req6-DTI8)</i>
<i>Instant(Req6-DTI8-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI8-Sins)</i>
<i>Instant(Req6-DTI8-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI8-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>actualiza(Valentino,Req6-Ev8)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Valentino domain-PAIyER:Req6-Ev8)</i>
<i>actualiza(Req6-Ev8,Inventario_CENIDET)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Req6-Ev8 domain-PAIyER:cInventario_CENIDET)</i>
<i>during(Req6-Ev8,Req6-DTI8)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req6-Ev8 domain-PAIyER:Req6-DTI8)</i>
<i>hasBeginning(Req6-DTI8,Req6-DTI8-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req6-DTI8 domain-PAIyER:Req6-DTI8-Sins)</i>
<i>hasEnd(Req6-DTI8,Req6-DTI8-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req6-DTI8 domain-PAIyER:Req6-DTI8-Eins)</i>
}	
Caso de éxito 3 (A₉, A₁₀, A₁₁, A₁₂):	
A₉: [X(Empleado \sqsubseteq \existssepara.Entidad)]: El empleado Valentino separa el Entidad Libro3.	
...	

Tabla D.26 ABox de casos de éxito en el RF6. Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
<p>A_{10}: $[XX (Empleado \sqsubseteq \exists verificaExistenciaDe.Entidad)]$: El empleado Valentino verifica la existencia de la entidad Libro3</p> <p>...</p> <p>A_{11}: $[XXX (Empleado \sqsubseteq \exists actualiza.Entidad) \vee (Empleado \sqsubseteq \exists registra.Entidad)]$: El empleado Valentino actualiza la entidad Libro3 o registra la entidad Libro3.</p> <p>...</p> <p>A_{12}: $[XXXX (Empleado \sqsubseteq \exists actualiza.Inventario)]$: El empleado Valentino actualiza el inventario Inventario_CENIDET.</p> <p>...</p>	
<p>Caso de éxito 4 ($A_{13}, A_{14}, A_{15}, A_{16}$):</p> <p>$A_{13}$: $[X(Empleado \sqsubseteq \exists separa.Entidad)]$: El empleado Valentino separa el Entidad Libro4.</p> <p>...</p> <p>A_{14}: $[XX (Empleado \sqsubseteq \exists verificaExistenciaDe.Entidad)]$: El empleado Valentino verifica la existencia de la entidad Libro4.</p> <p>...</p> <p>A_{15}: $[XXX (Empleado \sqsubseteq \exists actualiza.Entidad) \vee (Empleado \sqsubseteq \exists registra.Entidad)]$: El empleado Valentino actualiza la entidad Libro3 o registra la entidad Libro4.</p> <p>...</p> <p>A_{16}: $[XXXX (Empleado \sqsubseteq \exists actualiza.Inventario)]$: El empleado Valentino actualiza el inventario Inventario_CENIDET.</p> <p>...</p>	
<p>Caso de éxito n</p> <p>A_n</p> <p>...</p>	

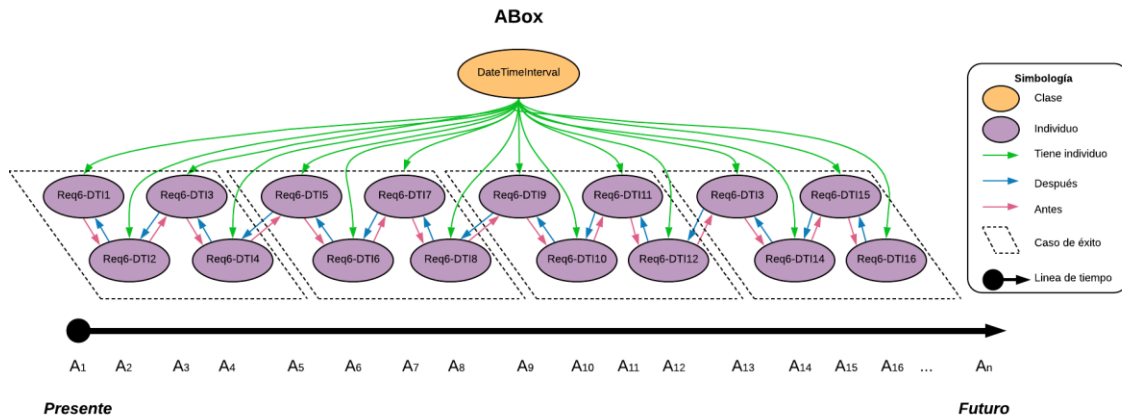


Figura D.22 SoIs en la ABox de casos de éxito de acuerdo al RF6. Separar diferentes tipos de existencias.

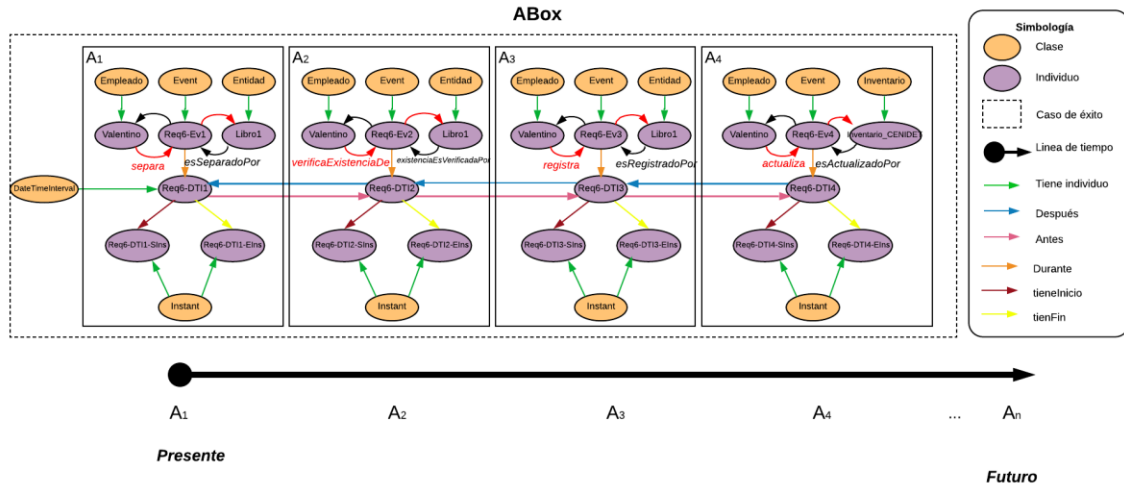


Figura D.23 Caso de éxito 1 en la ABox de casos de éxito de acuerdo al RF6. Separar diferentes tipos de existencias.

1.2.6.2. Casos de éxito y fracaso

En la Tabla D.27 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la ABox) basadas en casos de éxito y fracaso, que conforman una secuencia que representa el comportamiento dinámico en la separación de diferentes tipos de entidades para un inventario. Una representación gráfica simplificada de la secuencia se muestra en la Figura D.24. A manera de ejemplo, en la Figura D.25 se muestra el detalle de las aserciones A1, A2, A3 y A4 en el caso de fracaso número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de razonamiento automatizado. Los demás casos se definen de manera similar, sin embargo, difieren en los individuos involucrados.

En la Tabla D.28 se describen las reglas SWRL utilizadas para restringir la existencia de afirmaciones de propiedades de objeto en el intervalo de tiempo correspondiente.

Tabla D.27 ABox de casos de éxito y fracaso en el RF6. Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
Caso de fracaso 1 (A1, A2, A3, A4):	
A1: $[X(\text{Empleado} \sqsubseteq \exists \text{separa}.\text{Entidad})]$: El empleado Valentino separa el Entidad Libro1.	
A1 = {	
-- Aserciones de clase --	
Empleado(Valentino)	\Rightarrow ClassAssertion(domain-PAIyER:Cliente domain-PAIyER:Valentino)
Entidad(Libro1)	\Rightarrow ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro1)
Event(Req6-Ev1)	\Rightarrow ClassAssertion(timeEvents:Event domain-PAIyER:Req6-Ev1)
DateTimeInterval(Req6-DT11)	\Rightarrow ClassAssertion(time:DateTimeInterval domain-PAIyER:Req6-DT11)
Instant(Req6-DT11-Sins)	\Rightarrow ClassAssertion(time:Instant domain-PAIyER:Req6-DT11-Sins)
Instant(Req6-DT11-Eins)	\Rightarrow ClassAssertion(time:Instant domain-PAIyER:Req6-

Tabla D.27 ABox de casos de éxito y fracaso en el RF6. Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
	<i>DTI1-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>separa(Valentino,Req6-Ev1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:separa domain-PAIyER:Valentino domain-PAIyER:Req6-Ev1)</i>
<i>separa(Req6-Ev1,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:separa domain-PAIyER:Req6-Ev1 domain-PAIyER:Libro1)</i>
<i>during(Req6-Ev1,Req6-DTI1)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req6-Ev1 domain-PAIyER:Req6-DTI1)</i>
<i>hasBeginning(Req6-DTI1,Req6-DTI1-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req6-DTI1 domain-PAIyER:Req6-DTI1-Sins)</i>
<i>hasEnd(Req6-DTI1,Req6-DTI1-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req6-DTI1 domain-PAIyER:Req6-DTI1-Eins)</i>
}	\Rightarrow
$A_2 = \{$	$A_2: [XX (Empleado \sqsubseteq \exists verificaExistenciaDe.Entidad)]:$ El empleado Valentino verifica la existencia de la entidad Libro1.
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro1)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro1)</i>
<i>Event(Req6-Ev2)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req6-Ev2)</i>
<i>DateTimeInterval(Req6-DTI2)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req6-DTI2)</i>
<i>Instant(Req6-DTI2-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI2-Sins)</i>
<i>Instant(Req6-DTI2-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI2-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>verificaExistenciaDe(Valentino,Req6-Ev2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaExistenciaDe domain-PAIyER:Valentino domain-PAIyER:Req6-Ev2)</i>
<i>verificaExistenciaDe(Req6-Ev2,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaExistenciaDe domain-PAIyER:Req6-Ev2 domain-PAIyER:Libro1)</i>
<i>during(Req6-Ev2,Req6-DTI2)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req6-Ev2 domain-PAIyER:Req6-DTI2)</i>
<i>hasBeginning(Req6-DTI2,Req6-DTI2-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req6-DTI2 domain-PAIyER:Req6-DTI2-Sins)</i>
<i>hasEnd(Req6-DTI2,Req6-DTI2-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req6-DTI2 domain-PAIyER:Req6-DTI2-Eins)</i>
}	
$A_3 = \{$	$A_3: [\neg XXX (Empleado \sqsubseteq \exists actualiza.Entidad) \vee (Empleado \sqsubseteq \exists registra.Entidad)]:$ El empleado Valentino no actualiza la entidad Libro1 o no registra la entidad Libro1.
-- Aserciones de clase --	
<i>Event(Req6-Ev3)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req6-Ev3)</i>
<i>DateTimeInterval(Req6-DTI3)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req6-DTI3)</i>
<i>Instant(Req6-DTI3-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI3-Sins)</i>

Tabla D.27 ABox de casos de éxito y fracaso en el RF6. Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Instant(Req6-DTI3-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI3-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>during(Req6-Ev3,Req6-DTI3)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req6-Ev3 domain-PAIyER:Req6-DTI3)</i>
<i>hasBeginning(Req6-DTI3,Req6-DTI3-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req6-DTI3 domain-PAIyER:Req6-DTI3-Sins)</i>
<i>hasEnd(Req6-DTI3,Req6-DTI3-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req6-DTI3 domain-PAIyER:Req6-DTI3-Eins)</i>
}	
A₄: [\negXXXX (Empleado \sqsubseteq \existsactualiza.Inventario)]: El empleado Valentino no actualiza el inventario Inventario_CENIDET.	
A ₄ = {	
-- Aserciones de clase --	
<i>Event(Req6-Ev4)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req6-Ev4)</i>
<i>DateTimeInterval(Req6-DTI4)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req6-DTI4)</i>
<i>Instant(Req6-DTI4-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI4-Sins)</i>
<i>Instant(Req6-DTI4-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI4-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>during(Req6-Ev4,Req6-DTI4)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req6-Ev4 domain-PAIyER:Req6-DTI4)</i>
<i>hasBeginning(Req6-DTI4,Req6-DTI4-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req6-DTI4 domain-PAIyER:Req6-DTI4-Sins)</i>
<i>hasEnd(Req6-DTI4,Req6-DTI4-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req6-DTI4 domain-PAIyER:Req6-DTI4-Eins)</i>
}	
Caso de éxito I (A₅, A₆, A₇, A₈):	
A₅: [X(Empleado \sqsubseteq \existssepara.Entidad)]: El empleado Valentino separa el Entidad Libro2.	
A ₅ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event(Req6-Ev5)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req6-Ev5)</i>
<i>DateTimeInterval(Req6-DTI5)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req6-DTI5)</i>
<i>Instant(Req6-DTI5-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI5-Sins)</i>
<i>Instant(Req6-DTI5-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI5-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>separa(Valentino,Req6-Ev5)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:separa domain-PAIyER:Valentino domain-PAIyER:Req6-Ev5)</i>
<i>separa(Req6-Ev5,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:separa domain-PAIyER:Req6-Ev5 domain-PAIyER:Libro2)</i>
<i>during(Req6-Ev5,Req6-DTI5)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req6-Ev5 domain-PAIyER:Req6-DTI5)</i>
<i>hasBeginning(Req6-DTI5,Req6-</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-</i>

Tabla D.27 ABox de casos de éxito y fracaso en el RF6. Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
<i>DTI5-Sins</i>	<i>PAIyER:Req6-DTI5 domain-PAIyER:Req6-DTI5-Sins</i>
<i>hasEnd(Req6-DTI5,Req6-DTI5-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req6-DTI5 domain-PAIyER:Req6-DTI5-Eins)</i>
}	\Rightarrow
$A_6 = \{$	A₆: [XX (Empleado \sqsubseteq \existsverificaExistenciaDe.Entidad)]: El empleado Valentino verifica la existencia de la entidad Libro2
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event(Req6-Ev6)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req6-Ev6)</i>
<i>DateTimeInterval(Req6-DTI6)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req6-DTI6)</i>
<i>Instant(Req6-DTI6-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI6-Sins)</i>
<i>Instant(Req6-DTI6-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI6-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>verificaExistenciaDe(Valentino,R</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaExistenciaDe domain-PAIyER:Valentino domain-PAIyER:Req6-Ev6)</i>
<i>ev6)</i>	
<i>verificaExistenciaDe(Req6-Ev6,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaExistenciaDe domain-PAIyER:Req6-Ev6 domain-PAIyER:Libro2)</i>
<i>during(Req6-Ev6,Req6-DTI6)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req6-Ev6 domain-PAIyER:Req6-DTI6)</i>
<i>hasBeginning(Req6-DTI6,Req6-DTI6-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req6-DTI6 domain-PAIyER:Req6-DTI6-Sins)</i>
<i>hasEnd(Req6-DTI6,Req6-DTI6-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req6-DTI6 domain-PAIyER:Req6-DTI6-Eins)</i>
}	
$A_7 = \{$	A₇: [XXX((Empleado \sqsubseteq \existsactualiza.Entidad) \vee (Empleado \sqsubseteq \existsregistra.Entidad))]: El empleado Valentino actualiza la entidad Libro2 o registra la entidad Libro2.
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event(Req6-Ev7)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req6-Ev7)</i>
<i>DateTimeInterval(Req6-DTI7)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req6-DTI7)</i>
<i>Instant(Req6-DTI7-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI7-Sins)</i>
<i>Instant(Req6-DTI7-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI7-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>actualiza(Valentino,Req6-Ev7)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Valentino domain-PAIyER:Req6-Ev7)</i>
<i>actualiza(Req6-Ev7,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza</i>

Tabla D.27 ABox de casos de éxito y fracaso en el RF6. Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
<i>during</i> (Req6-Ev7,Req6-DTI7)	\Rightarrow <i>domain-PAIyER:Req6-Ev7 domain-PAIyER:Libro2</i> <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req6-Ev7 domain-PAIyER:Req6-DTI7)</i>
<i>hasBeginning</i> (Req6-DTI7,Req6-DTI7-Sins)	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req6-DTI7 domain-PAIyER:Req6-DTI7-Sins)</i>
<i>hasEnd</i> (Req6-DTI7,Req6-DTI7-Eins)	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req6-DTI7 domain-PAIyER:Req6-DTI7-Eins)</i>
}	
<i>As: [XXXX (Empleado \sqsubseteq \existsactualiza.Inventario)]: El empleado Valentino actualiza el inventario Inventario_CENIDET.</i>	
<i>A_s = {</i>	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Inventario</i> (Inventario_CENIDET)	\Rightarrow <i>ClassAssertion(domain-PAIyER:Inventario domain-PAIyER:Inventario_CENIDET)</i>
<i>Event</i> (Req6-Ev8)	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req6-Ev8)</i>
<i>DateTimeInterval</i> (Req6-DTI8)	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req6-DTI8)</i>
<i>Instant</i> (Req6-DTI8-Sins)	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI8-Sins)</i>
<i>Instant</i> (Req6-DTI8-Eins)	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req6-DTI8-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>actualiza</i> (Valentino,Req6-Ev8)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Valentino domain-PAIyER:Req6-Ev8)</i>
<i>actualiza</i> (Req6-Ev8,Inventario_CENIDET)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:actualiza domain-PAIyER:Req6-Ev8 domain-PAIyER:cInventario_CENIDET)</i>
<i>during</i> (Req6-Ev8,Req6-DTI8)	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req6-Ev8 domain-PAIyER:Req6-DTI8)</i>
<i>hasBeginning</i> (Req6-DTI8,Req6-DTI8-Sins)	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req6-DTI8 domain-PAIyER:Req6-DTI8-Sins)</i>
<i>hasEnd</i> (Req6-DTI8,Req6-DTI8-Eins)	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req6-DTI8 domain-PAIyER:Req6-DTI8-Eins)</i>
}	
Caso de fracaso 2 (A₉, A₁₀, A₁₁, A₁₂):	
<i>A₉: [X(Empleado \sqsubseteq \existssepara.Entidad)]: El empleado Valentino separa el Entidad Libro3.</i>	
...	
<i>A₁₀: [XX (Empleado \sqsubseteq \existsverificaExistenciaDe.Entidad)]: El empleado Valentino verifica la existencia de la entidad Libro3.</i>	
...	
<i>A₁₁: [\negXXX((Empleado \sqsubseteq \existsactualiza.Entidad) \vee (Empleado \sqsubseteq \existsregistra.Entidad))]: El empleado Valentino no actualiza la entidad Libro3 o no registra la entidad Libro3.</i>	
...	
<i>A₁₂: [\negXXXX (Empleado \sqsubseteq \existsactualiza.Inventario)]: El empleado Valentino no actualiza el inventario Inventario_CENIDET.</i>	
Caso de éxito 2 (A₁₃, A₁₄, A₁₅, A₁₆):	
<i>A₁₃: [X(Empleado \sqsubseteq \existssepara.Entidad)]: El empleado Valentino separa el Entidad Libro4.</i>	
...	
<i>A₁₄: [XX (Empleado \sqsubseteq \existsverificaExistenciaDe.Entidad)]: El empleado Valentino verifica la existencia de la entidad Libro4</i>	

Tabla D.27 ABox de casos de éxito y fracaso en el RF6. Separar diferentes tipos de existencias (cont.)

Sintaxis DL	Sintaxis Funcional
...	
A_{15} : [XXX(Empleado \sqsubseteq \exists actualiza.Entidad) \vee (Empleado \sqsubseteq \exists registra.Entidad)]: El empleado Valentino actualiza la entidad Libro4 o registra la entidad Libro4.	
...	
A_{16} : [XXXX(Empleado \sqsubseteq \exists actualiza.Inventario)]: El empleado Valentino actualiza el inventario Inventario_CENIDET.	
Caso de éxito o fracaso n	
A_n	
...	

Tabla D.28 Reglas SWRL para la SoIs de casos de éxito y fracaso de acuerdo al RF6. Separar diferentes tipos de existencias. (cont.)

ABox	Regla SWRL
A_3	Event(?e), during(?e, ?i), overlaps(?i, Req6-DTI3), registra(?e, Libro1), registra(Valentino, ?e) -> Nothing(Valentino)
A_4	Event(?e), during(?e, ?i), overlaps(?i, Req6-DTI4), actualiza(?e, Inventario_CENIDET), actualiza(Valentino, ?e) -> Nothing(Valentino)
A_{11}	Event(?e), during(?e, ?i), overlaps(?i, Req6-DTI11), registra(?e, Libro3), registra(Valentino, ?e) -> Nothing(Valentino)
A_{12}	Event(?e), during(?e, ?i), overlaps(?i, Req6-DTI12), actualiza(?e, Inventario_CENIDET), actualiza(Valentino, ?e) -> Nothing(Valentino)

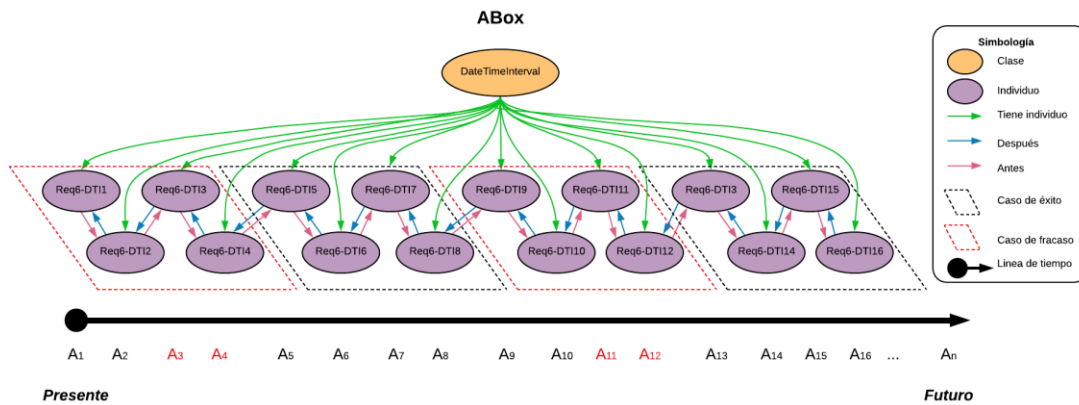


Figura D.24 SoIs en la ABox de casos de éxito y fracaso de acuerdo al RF6. Separar diferentes tipos de existencias.

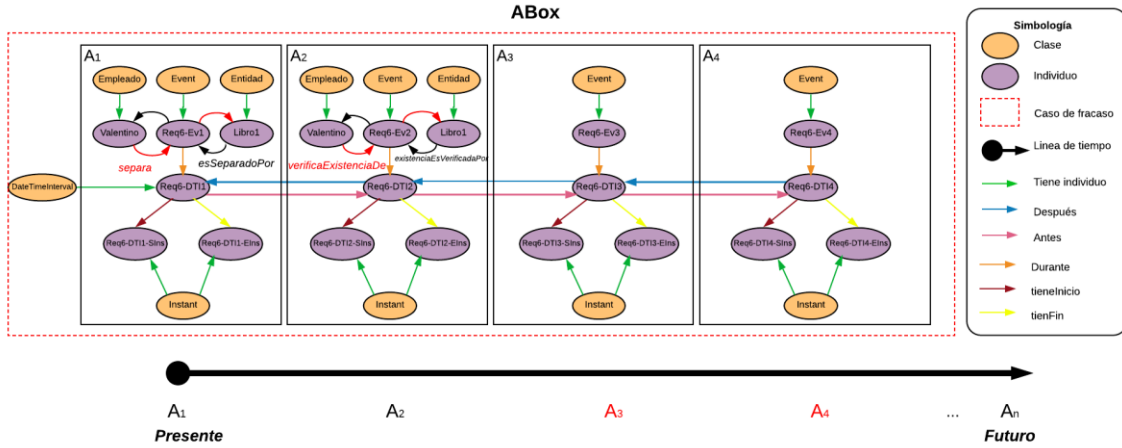


Figura D.25 Caso de fracaso 1 en la ABox de casos de éxito y fracaso de acuerdo al RF6. Separar diferentes tipos de existencias.

1.2.7 RF7. Realizar seguimiento de cantidades de entidades en existencias

Las secuencias de interpretaciones descritas en esta sección satisfacen/cumplen con el comportamiento dinámico del RF7. Realizar seguimiento de cantidades de entidades en existencias representado formalmente con la fórmula de la Tabla D.29.

Tabla D.29 Fórmula ALC_{LTL} del RF7. Realizar seguimiento de cantidades de entidades en existencias.

$$\mathcal{G} (\mathcal{X} (Empleado \sqsubseteq \exists buscaExistenciasDe.Entidad) \rightarrow (\mathcal{X}\mathcal{X} (Empleado \sqsubseteq \exists verificaExistenciaDe.Entidad) \wedge \mathcal{F} (\neg \mathcal{X}\mathcal{X}\mathcal{X} (Empleado \sqsubseteq \exists especificaTipoExistenciasDe.Entidad) \wedge \neg \mathcal{X}\mathcal{X}\mathcal{X} (Empleado \sqsubseteq \exists obtieneExistenciasDe.Entidad)))))$$

1.2.7.1. Casos de éxito

En la Tabla D.30 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la ABox) basadas en casos de éxito, que conforman una secuencia que representa el comportamiento dinámico durante el seguimiento de la cantidad de existencias de la entidad Libro1 de un inventario. Una representación gráfica simplificada de la secuencia se muestra en la Figura D.26. A manera de ejemplo, en la Figura D.27 se muestra el detalle de las aserciones A_1 , A_2 , A_3 y A_4 en el caso de éxito número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de razonamiento automatizado, los demás casos se definen de manera similar. Sin embargo, difieren en los individuos involucrados.

Tabla D.30 ABox de casos de éxito en el RF7. Realizar seguimiento de cantidades de entidades en existencias.
(cont.)

Sintaxis DL	Sintaxis Funcional
Caso de éxito 1 (A_1, A_2, A_3, A_4):	
A_1 : [X (<i>Empleado</i> $\sqsubseteq \exists$ buscaExistenciasDe.Entidad)]: El empleado Valentino busca existencias de la entidad Libro1.	
$A_1 = \{$	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Entidad</i> (Libro1)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Entidad domain-PAIyER:Libro1)
<i>Event</i> (Req7-Ev1)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req7-Ev1)
<i>DateTimeInterval</i> (Req7-DT11)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req7-DT11)
<i>Instant</i> (Req7-DT11-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req7-DT11-Sins)
<i>Instant</i> (Req7-DT11-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req7-DT11-Eins)
-- Aserciones de propiedades de objeto --	
<i>buscaExistenciasDe</i> (Valentino,Req7-Ev1)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:buscaExistenciasDe domain-PAIyER:Valentino domain-PAIyER:Req7-Ev1)
<i>buscaExistenciasDe</i> (Req7-Ev1,Libro1)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:buscaExistenciasDe domain-PAIyER:Req7-Ev1 domain-PAIyER:Libro1)
<i>during</i> (Req7-Ev1,Req7-DT11)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req7-Ev1 domain-PAIyER:Req7-DT11)
<i>hasBeginning</i> (Req7-DT11,Req7-DT11-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req7-DT11 domain-PAIyER:Req7-DT11-Sins)
<i>hasEnd</i> (Req7-DT11,Req7-DT11-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req7-DT11 domain-PAIyER:Req7-DT11-Eins)
$\}$	\Rightarrow
A_2 : [XX (<i>Empleado</i> $\sqsubseteq \exists$ verificaExistenciaDe.Entidad)]: El empleado Valentino verifica la existencia de la entidad Libro1	
$A_2 = \{$	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Entidad</i> (Libro1)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Entidad domain-PAIyER:Libro1)
<i>Event</i> (Req7-Ev2)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req7-Ev2)
<i>DateTimeInterval</i> (Req7-DT12)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req7-DT12)
<i>Instant</i> (Req7-DT12-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req7-DT12-Sins)
<i>Instant</i> (Req7-DT12-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req7-DT12-Eins)
-- Aserciones de propiedades de objeto --	
<i>verificaExistenciaDe</i> (Valentino,Req7-Ev2)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:verificaExistenciaDe domain-PAIyER:Valentino)

Tabla D.30 ABox de casos de éxito en el RF7. Realizar seguimiento de cantidades de entidades en existencias. (cont.)

Sintaxis DL	Sintaxis Funcional
<i>verificaExistenciaDe(Req7-Ev2,Libro1)</i>	\Rightarrow <i>domain-PAIyER:Req7-Ev2</i> <i>ObjectPropertyAssertion(domain-PAIyER:verificaExistenciaDe domain-PAIyER:Req7-Ev2 domain-PAIyER:Libro1)</i>
<i>during(Req7-Ev2,Req7-DTI2)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req7-Ev2 domain-PAIyER:Req7-DTI2)</i>
<i>hasBeginning(Req7-DTI2,Req7-DTI2-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req7-DTI2 domain-PAIyER:Req7-DTI2-Sins)</i>
<i>hasEnd(Req7-DTI2,Req7-DTI2-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req7-DTI2 domain-PAIyER:Req7-DTI2-Eins)</i>
}	
A₃: [XXX (Empleado \sqsubseteq \existsespecificaTipoExistenciasDe.Entidad)]: El empleado Valentino especifica el tipo de existencias (enMano, enOrden, reservadas) del Entidad Libro1.	
A ₃ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro1)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro1)</i>
<i>Event(Req7-Ev3)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req7-Ev3)</i>
<i>DateTimeInterval(Req7-DTI3)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req7-DTI3)</i>
<i>Instant(Req7-DTI3-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req7-DTI3-Sins)</i>
<i>Instant(Req7-DTI3-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req7-DTI3-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>especificaTipoExistenciasDe(Valentino,Req7-Ev3)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:especificaTipoExistenciasDe domain-PAIyER:Valentino domain-PAIyER:Req7-Ev3)</i>
<i>especificaTipoExistenciasDe (Req7-Ev3,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:especificaTipoExistenciasDe domain-PAIyER:Req7-Ev3 domain-PAIyER:Libro1)</i>
<i>during(Req7-Ev3,Req7-DTI3)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req7-Ev3 domain-PAIyER:Req7-DTI3)</i>
<i>hasBeginning(Req7-DTI3,Req7-DTI3-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req7-DTI3 domain-PAIyER:Req7-DTI3-Sins)</i>
<i>hasEnd(Req7-DTI3,Req7-DTI3-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req7-DTI3 domain-PAIyER:Req7-DTI3-Eins)</i>
}	
A₄: [XXXX (Empleado \sqsubseteq \existsobtieneExistenciasDe.Entidad)]: El empleado Valentino obtiene la cantidad de existencias especificada del Entidad Libro1.	
A ₄ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>

Tabla D.30 ABox de casos de éxito en el RF7. Realizar seguimiento de cantidades de entidades en existencias.
(cont.)

Sintaxis DL	Sintaxis Funcional
<i>Entidad(Libro1)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro1)</i>
<i>Event(Req7-Ev4)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req7-Ev4)</i>
<i>DateTimeInterval(Req7-DTI4)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req7-DTI4)</i>
<i>Instant(Req7-DTI4-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req7-DTI4-Sins)</i>
<i>Instant(Req7-DTI4-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req7-DTI4-Eins)</i>
-- Aserciones de propiedades de objeto -- <i>obtieneExistenciasDe(Valentino,Req7-Ev4)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:obtieneExistenciasDe domain-PAIyER:Valentino domain-PAIyER:Req7-Ev4)</i>
<i>obtieneExistenciasDe(Req7-Ev4,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:obtieneExistenciasDe domain-PAIyER:Req7-Ev4 domain-PAIyER:Libro1)</i>
<i>during(Req7-Ev4,Req7-DTI4)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req7-Ev4 domain-PAIyER:Req7-DTI4)</i>
<i>hasBeginning(Req7-DTI4,Req7-DTI4-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req7-DTI4 domain-PAIyER:Req7-DTI4-Sins)</i>
<i>hasEnd(Req7-DTI4,Req7-DTI4-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req7-DTI4 domain-PAIyER:Req7-DTI4-Eins)</i>
}	
Caso de éxito 2 (A5, A6, A7, A8):	
<i>A5: [X (Empleado \sqsubseteq buscaExistenciasDe.Entidad)]: El empleado Valentino busca existencias del Entidad Libro2.</i>	
<i>A5 = {</i>	
-- Aserciones de clase -- <i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event(Req7-Ev5)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req7-Ev5)</i>
<i>DateTimeInterval(Req7-DTI5)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req7-DTI5)</i>
<i>Instant(Req7-DTI5-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req7-DTI5-Sins)</i>
<i>Instant(Req7-DTI5-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req7-DTI5-Eins)</i>
-- Aserciones de propiedades de objeto --	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:buscaExistenciasDe domain-PAIyER:Valentino domain-PAIyER:Req7-Ev5)</i>
<i>buscaExistenciasDe(Valentino,Req7-Ev5)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:buscaExistenciasDe domain-PAIyER:Req7-Ev5 domain-PAIyER:Libro2)</i>
<i>buscaExistenciasDe(Req7-Ev5,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req7-Ev5 domain-PAIyER:Req7-</i>
<i>during(Req7-Ev5,Req7-DTI5)</i>	

Tabla D.30 ABox de casos de éxito en el RF7. Realizar seguimiento de cantidades de entidades en existencias.
(cont.)

Sintaxis DL	Sintaxis Funcional
<i>hasBeginning</i> (Req7-DTI5,Req7-DTI5-Sins)	<i>DTI5</i> ⇒ <i>ObjectPropertyAssertion</i> (time: <i>hasBeginning</i> domain-PAIyER:Req7-DTI5 domain-PAIyER:Req7-DTI5-Sins)
<i>hasEnd</i> (Req7-DTI5,Req7-DTI5-Eins)	⇒ <i>ObjectPropertyAssertion</i> (time: <i>hasEnd</i> domain-PAIyER:Req7-DTI5 domain-PAIyER:Req7-DTI5-Eins)
}	⇒
A₆ : [XX (<i>Empleado</i> ⊆ ∃ <i>verificaExistenciaDe</i> .Entidad)]: El empleado Valentino verifica la existencia de la entidad Libro2.	
A ₆ = {	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	⇒ <i>ClassAssertion</i> (domain-PAIyER: <i>Empleado</i> domain-PAIyER:Valentino)
<i>Entidad</i> (Libro2)	⇒ <i>ClassAssertion</i> (domain-PAIyER: <i>Entidad</i> domain-PAIyER:Libro2)
<i>Event</i> (Req7-Ev6)	⇒ <i>ClassAssertion</i> (timeEvents: <i>Event</i> domain-PAIyER:Req7-Ev6)
<i>DateTimeInterval</i> (Req7-DTI6)	⇒ <i>ClassAssertion</i> (time: <i>DateTimeInterval</i> domain-PAIyER:Req7-DTI6)
<i>Instant</i> (Req7-DTI6-Sins)	⇒ <i>ClassAssertion</i> (time: <i>Instant</i> domain-PAIyER:Req7-DTI6-Sins)
<i>Instant</i> (Req7-DTI6-Eins)	⇒ <i>ClassAssertion</i> (time: <i>Instant</i> domain-PAIyER:Req7-DTI6-Eins)
-- Aserciones de propiedades de objeto --	
<i>verificaExistenciaDe</i> (Valentino,Req7-Ev6)	⇒ <i>ObjectPropertyAssertion</i> (domain-PAIyER: <i>verificaExistenciaDe</i> domain-PAIyER:Valentino domain-PAIyER:Req7-Ev6)
<i>verificaExistenciaDe</i> (Req7-Ev6,Libro2)	⇒ <i>ObjectPropertyAssertion</i> (domain-PAIyER: <i>verificaExistenciaDe</i> domain-PAIyER:Req7-Ev6 domain-PAIyER:Libro2)
<i>during</i> (Req7-Ev6,Req7-DTI6)	⇒ <i>ObjectPropertyAssertion</i> (timeEvents: <i>during</i> domain-PAIyER:Req7-Ev6 domain-PAIyER:Req7-DTI6)
<i>hasBeginning</i> (Req7-DTI6,Req7-DTI6-Sins)	⇒ <i>ObjectPropertyAssertion</i> (time: <i>hasBeginning</i> domain-PAIyER:Req7-DTI6 domain-PAIyER:Req7-DTI6-Sins)
<i>hasEnd</i> (Req7-DTI6,Req7-DTI6-Eins)	⇒ <i>ObjectPropertyAssertion</i> (time: <i>hasEnd</i> domain-PAIyER:Req7-DTI6 domain-PAIyER:Req7-DTI6-Eins)
}	
A₇ : [XXX (<i>Empleado</i> ⊆ ∃ <i>especificaTipoExistenciasDe</i> .Entidad)]: El empleado Valentino especifica el tipo de existencias (enMano, enOrden, reservadas) del Entidad Libro2.	
A ₇ = {	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	⇒ <i>ClassAssertion</i> (domain-PAIyER: <i>Empleado</i> domain-PAIyER:Valentino)
<i>Entidad</i> (Libro2)	⇒ <i>ClassAssertion</i> (domain-PAIyER: <i>Entidad</i> domain-PAIyER:Libro2)
<i>Event</i> (Req7-Ev7)	⇒ <i>ClassAssertion</i> (timeEvents: <i>Event</i> domain-PAIyER:Req7-Ev7)
<i>DateTimeInterval</i> (Req7-DTI7)	⇒ <i>ClassAssertion</i> (time: <i>DateTimeInterval</i> domain-PAIyER:Req7-DTI7)

Tabla D.30 ABox de casos de éxito en el RF7. Realizar seguimiento de cantidades de entidades en existencias. (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Instant(Req7-DTI7-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req7-DTI7-Sins)</i>
<i>Instant(Req7-DTI7-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req7-DTI7-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>especificaTipoExistenciasDe(Valentino,Req7-Ev7)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:especificaTipoExistenciasDe domain-PAIyER:Valentino domain-PAIyER:Req7-Ev7)</i>
<i>especificaTipoExistenciasDe (Req7-Ev7,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:especificaTipoExistenciasDe domain-PAIyER:Req7-Ev7 domain-PAIyER:Libro2)</i>
<i>during(Req7-Ev7,Req7-DTI7)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req7-Ev7 domain-PAIyER:Req7-DTI7)</i>
<i>hasBeginning(Req7-DTI7,Req7-DTI7-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req7-DTI7 domain-PAIyER:Req7-DTI7-Sins)</i>
<i>hasEnd(Req7-DTI7,Req7-DTI7-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req7-DTI7 domain-PAIyER:Req7-DTI7-Eins)</i>
}	
A₈: [XXXX (Empleado \sqsubseteq \existsobtieneExistenciasDe.Entidad)]: El empleado Valentino obtiene la cantidad de existencias especificada del Entidad Libro2.	
A ₈ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event(Req7-Ev8)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req7-Ev8)</i>
<i>DateTimeInterval(Req7-DTI8)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req7-DTI8)</i>
<i>Instant(Req7-DTI8-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req7-DTI8-Sins)</i>
<i>Instant(Req7-DTI8-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req7-DTI8-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>obtieneExistenciasDe(Valentino,Req7-Ev8)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:obtieneExistenciasDe domain-PAIyER:Valentino domain-PAIyER:Req7-Ev8)</i>
<i>obtieneExistenciasDe(Req7-Ev8,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:obtieneExistenciasDe domain-PAIyER:Req7-Ev8 domain-PAIyER:Libro2)</i>
<i>during(Req7-Ev8,Req7-DTI8)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req7-Ev8 domain-PAIyER:Req7-DTI8)</i>
<i>hasBeginning(Req7-DTI8,Req7-DTI8-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req7-DTI8 domain-PAIyER:Req7-DTI8-Sins)</i>
<i>hasEnd(Req7-DTI8,Req7-DTI8-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req7-DTI8 domain-PAIyER:Req7-DTI8-Eins)</i>

Tabla D.30 ABox de casos de éxito en el RF7. Realizar seguimiento de cantidades de entidades en existencias. (cont.)

Sintaxis DL	Sintaxis Funcional
}	
<p>Caso de éxito 3 (A₉, A₁₀, A₁₁, A₁₂):</p> <p>A₉: [X (Empleado \sqsubseteq \existsbuscaExistenciasDe.Entidad)]: El empleado Valentino busca existencias del Entidad Libro3.</p> <p>...</p> <p>A₁₀: [XX (Empleado \sqsubseteq \existsverificaExistenciaDe.Entidad)]: El empleado Valentino verifica la existencia del artículo Libro3.</p> <p>...</p> <p>A₁₁: [XXX (Empleado \sqsubseteq \existsespecificaTipoExistenciasDe.Entidad)]: El empleado Valentino especifica el tipo de existencias (enMano, enOrden, reservadas) del Entidad Libro3.</p> <p>...</p> <p>A₁₂: [XXXX (Empleado \sqsubseteq \existsobtieneExistenciasDe.Entidad)]: El empleado Valentino obtiene la cantidad de existencias especificada del Entidad Libro3.</p> <p>...</p>	
<p>Caso de éxito 4 (A₁₃, A₁₄, A₁₅, A₁₆):</p> <p>A₁₃: [X (Empleado \sqsubseteq \existsbuscaExistenciasDe.Entidad)]: El empleado Valentino busca existencias del Entidad Libro4.</p> <p>...</p> <p>A₁₄: [XX (Empleado \sqsubseteq \existsverificaExistenciaDe.Entidad)]: El empleado Valentino verifica la existencia de la entidad Libro4.</p> <p>...</p> <p>A₁₅: [XXX (Empleado \sqsubseteq \existsespecificaTipoExistenciasDe.Entidad)]: El empleado Valentino especifica el tipo de existencias (enMano, enOrden, reservadas) del Entidad Libro4.</p> <p>...</p> <p>A₁₆: [XXXX (Empleado \sqsubseteq \existsobtieneExistenciasDe.Entidad)]: El empleado Valentino obtiene la cantidad de existencias especificada de la entidad Libro4.</p> <p>...</p>	
<p>Caso de éxito n</p> <p>A_n</p> <p>...</p>	

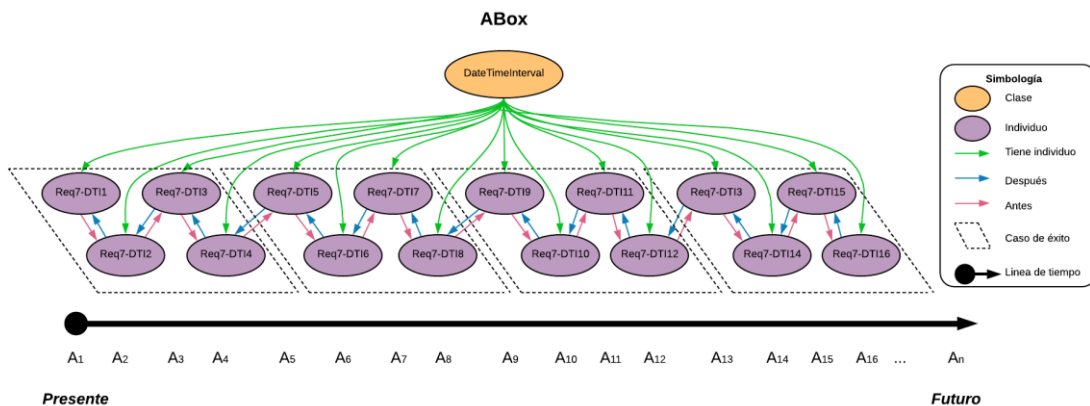


Figura D.26 SoIs en la ABox de casos de éxito de acuerdo al RF7. Realizar seguimiento de cantidades de entidades en existencias.

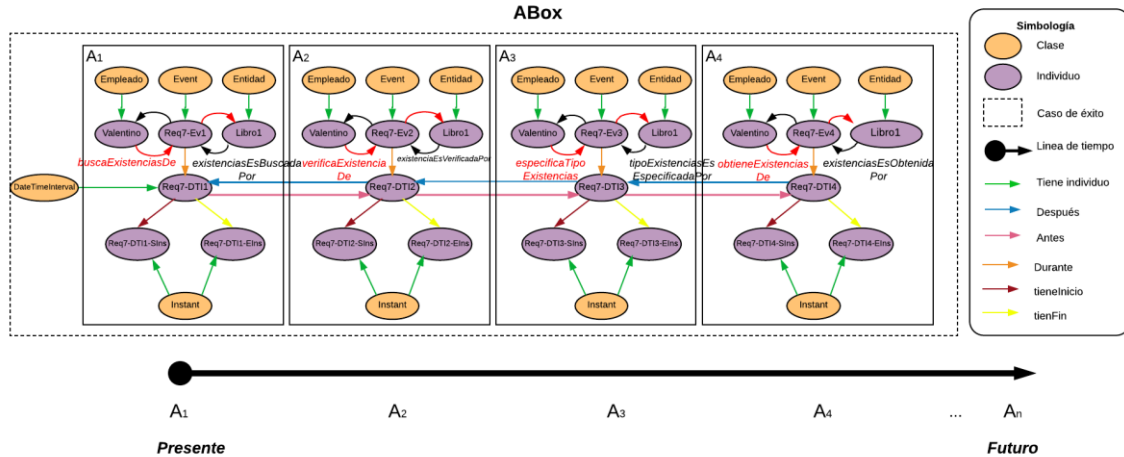


Figura D.27 Caso de éxito 1 en la ABox de casos de éxito de acuerdo al RF7. Realizar seguimiento de las cantidades de entidades en existencias.

1.2.7.2. Casos de éxito y fracaso

En la Tabla D.31 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la ABox) basadas en casos de éxito y fracaso, que conforman una secuencia que representa el comportamiento dinámico durante el seguimiento de la cantidad de existencias de la entidad Libro1 de un inventario. Una representación gráfica simplificada de la secuencia se muestra en la Figura D.28. A manera de ejemplo, en la Figura D.29 se muestra el detalle de las aserciones A_1 , A_2 , A_3 y A_4 en el caso de fracaso número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de razonamiento automatizado los demás casos se definen de manera similar, sin embargo, difieren en los individuos involucrados.

En la Tabla D.32 se describen las reglas SWRL utilizadas para restringir la existencia de afirmaciones de propiedades de objeto en el intervalo de tiempo correspondiente.

Tabla D.31 ABox de casos de éxito y fracaso en el RF7. Realizar seguimiento de cantidades de entidades en existencias (cont.)

Sintaxis DL	Sintaxis Funcional
Caso de fracaso 1 (A_1, A_2, A_3, A_4):	
A_1 : $[X(\text{Empleado} \sqsubseteq \exists \text{buscaExistenciasDe}.\text{Entidad})]$: El empleado Valentino busca existencias del Entidad Libro1.	
$A_1 = \{$	
<i>-- Aserciones de clase --</i>	
$\text{Empleado}(\text{Valentino})$	$\Rightarrow \text{ClassAssertion}(\text{domain-PAIyER:Empleado } \text{domain-PAIyER:Valentino})$
$\text{Entidad}(\text{Libro1})$	$\Rightarrow \text{ClassAssertion}(\text{domain-PAIyER:Entidad } \text{domain-PAIyER:Libro1})$
$\text{Event}(\text{Req7-Ev1})$	$\Rightarrow \text{ClassAssertion}(\text{timeEvents:Event } \text{domain-PAIyER:Req7-Ev1})$
$\text{DateTimeInterval}(\text{Req7-DT11})$	$\Rightarrow \text{ClassAssertion}(\text{time:DateTimeInterval } \text{domain-PAIyER:Req7-DT11})$

Tabla D.31 ABox de casos de éxito y fracaso en el RF7. Realizar seguimiento de cantidades de entidades en existencias (cont.)

Sintaxis DL	⇒	Sintaxis Funcional
<i>Instant(Req7-DT11-Sins)</i>	⇒	<i>ClassAssertion(time:Instant domain-PAIyER:Req7-DT11-Sins)</i>
<i>Instant(Req7-DT11-Eins)</i>	⇒	<i>ClassAssertion(time:Instant domain-PAIyER:Req7-DT11-Eins)</i>
-- Aserciones de propiedades de objeto --		
<i>buscaExistenciasDe(Valentino,Req7-Ev1)</i>	⇒	<i>ObjectPropertyAssertion(domain-PAIyER:buscaExistenciasDe domain-PAIyER:Valentino domain-PAIyER:Req7-Ev1)</i>
<i>buscaExistenciasDe(Req7-Ev1,Libro1)</i>	⇒	<i>ObjectPropertyAssertion(domain-PAIyER:buscaExistenciasDe domain-PAIyER:Req7-Ev1 domain-PAIyER:Libro1)</i>
<i>during(Req7-Ev1,Req7-DT11)</i>	⇒	<i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req7-Ev1 domain-PAIyER:Req7-DT11)</i>
<i>hasBeginning(Req7-DT11,Req7-DT11-Sins)</i>	⇒	<i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req7-DT11 domain-PAIyER:Req7-DT11-Sins)</i>
<i>hasEnd(Req7-DT11,Req7-DT11-Eins)</i>	⇒	<i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req7-DT11 domain-PAIyER:Req7-DT11-Eins)</i>
}		
A₂: [XX (Empleado ⊆ ∃verificaExistenciaDe.Entidad)]: El empleado Valentino verifica la existencia de la entidad Libro1		
A ₂ = {		
-- Aserciones de clase --		
<i>Empleado(Valentino)</i>	⇒	<i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro1)</i>	⇒	<i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro1)</i>
<i>Event(Req7-Ev2)</i>	⇒	<i>ClassAssertion(timeEvents:Event domain-PAIyER:Req7-Ev2)</i>
<i>DateTimeInterval(Req7-DT12)</i>	⇒	<i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req7-DT12)</i>
<i>Instant(Req7-DT12-Sins)</i>	⇒	<i>ClassAssertion(time:Instant domain-PAIyER:Req7-DT12-Sins)</i>
<i>Instant(Req7-DT12-Eins)</i>	⇒	<i>ClassAssertion(time:Instant domain-PAIyER:Req7-DT12-Eins)</i>
-- Aserciones de propiedades de objeto --		
<i>verificaExistenciaDe(Valentino,Req7-Ev2)</i>	⇒	<i>ObjectPropertyAssertion(domain-PAIyER:verificaExistenciaDe domain-PAIyER:Valentino domain-PAIyER:Req7-Ev2)</i>
<i>verificaExistenciaDe(Req7-Ev2,Libro1)</i>	⇒	<i>ObjectPropertyAssertion(domain-PAIyER:verificaExistenciaDe domain-PAIyER:Req7-Ev2 domain-PAIyER:Libro1)</i>
<i>during(Req7-Ev2,Req7-DT12)</i>	⇒	<i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req7-Ev2 domain-PAIyER:Req7-DT12)</i>
<i>hasBeginning(Req7-DT12,Req7-DT12-Sins)</i>	⇒	<i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req7-DT12 domain-PAIyER:Req7-DT12-Sins)</i>
<i>hasEnd(Req7-DT12,Req7-DT12-Eins)</i>	⇒	<i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req7-DT12 domain-PAIyER:Req7-DT12-Eins)</i>
}		
A₃: [¬XXX (Empleado ⊆ ∃especificaTipoExistenciasDe.Entidad)]: El empleado Valentino no especifica el tipo de existencias (enMano, enOrden, reservadas) del Entidad Libro1.		
A ₃ = {		
-- Aserciones de clase --		
<i>Event(Req7-Ev3)</i>	⇒	<i>ClassAssertion(timeEvents:Event domain-</i>

Tabla D.31 ABox de casos de éxito y fracaso en el RF7. Realizar seguimiento de cantidades de entidades en existencias (cont.)

Sintaxis DL	Sintaxis Funcional
	PAIyER:Req7-Ev3)
<i>DateTimeInterval(Req7-DTI3)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req7-DTI3)</i>
<i>Instant(Req7-DTI3-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req7-DTI3-Sins)</i>
<i>Instant(Req7-DTI3-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req7-DTI3-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>during(Req7-Ev3,Req7-DTI3)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req7-Ev3 domain-PAIyER:Req7-DTI3)</i>
<i>hasBeginning(Req7-DTI3,Req7-DTI3-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req7-DTI3 domain-PAIyER:Req7-DTI3-Sins)</i>
<i>hasEnd(Req7-DTI3,Req7-DTI3-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req7-DTI3 domain-PAIyER:Req7-DTI3-Eins)</i>
}	
A4: [\neg XXXX (Empleado \sqsubseteq \exists obtieneExistenciasDe.Entidad)]: El empleado Valentino no obtiene la cantidad de existencias especificada del Entidad Libro1.	
A4 = {	
-- Aserciones de clase --	
<i>Event(Req7-Ev4)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req7-Ev4)</i>
<i>DateTimeInterval(Req7-DTI4)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req7-DTI4)</i>
<i>Instant(Req7-DTI4-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req7-DTI4-Sins)</i>
<i>Instant(Req7-DTI4-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req7-DTI4-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>during(Req7-Ev4,Req7-DTI4)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req7-Ev4 domain-PAIyER:Req7-DTI4)</i>
<i>hasBeginning(Req7-DTI4,Req7-DTI4-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req7-DTI4 domain-PAIyER:Req7-DTI4-Sins)</i>
<i>hasEnd(Req7-DTI4,Req7-DTI4-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req7-DTI4 domain-PAIyER:Req7-DTI4-Eins)</i>
}	
Caso de éxito 1 (A5, A6, A7, A8):	
A5: [X (Empleado \sqsubseteq \exists buscaExistenciasDe.Entidad)]: El empleado Valentino busca existencias del Entidad Libro2.	
A5 = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event(Req7-Ev5)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req7-Ev5)</i>
<i>DateTimeInterval(Req7-DTI5)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req7-DTI5)</i>
<i>Instant(Req7-DTI5-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req7-DTI5-Sins)</i>
<i>Instant(Req7-DTI5-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req7-DTI5-Eins)</i>

Tabla D.31 ABox de casos de éxito y fracaso en el RF7. Realizar seguimiento de cantidades de entidades en existencias (cont.)

Sintaxis DL	Sintaxis Funcional
-- Aserciones de propiedades de objeto --	
<i>buscaExistenciasDe</i> (Valentino,Req7-Ev5)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:buscaExistenciasDe domain-PAIyER:Valentino domain-PAIyER:Req7-Ev5)
<i>buscaExistenciasDe</i> (Req7-Ev5,Libro2)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:buscaExistenciasDe domain-PAIyER:Req7-Ev5 domain-PAIyER:Libro2)
<i>during</i> (Req7-Ev5,Req7-DTI5)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req7-Ev5 domain-PAIyER:Req7-DTI5)
<i>hasBeginning</i> (Req7-DTI5,Req7-DTI5-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req7-DTI5 domain-PAIyER:Req7-DTI5-Sins)
<i>hasEnd</i> (Req7-DTI5,Req7-DTI5-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req7-DTI5 domain-PAIyER:Req7-DTI5-Eins)
}	
A6: [XX (Empleado \sqsubseteq \exists verificaExistenciaDe.Entidad)]: El empleado Valentino verifica la existencia de la entidad Libro2	
A6 = {	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Entidad</i> (Libro2)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Entidad domain-PAIyER:Libro2)
<i>Event</i> (Req7-Ev6)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req7-Ev6)
<i>DateTimeInterval</i> (Req7-DTI6)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req7-DTI6)
<i>Instant</i> (Req7-DTI6-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req7-DTI6-Sins)
<i>Instant</i> (Req7-DTI6-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req7-DTI6-Eins)
-- Aserciones de propiedades de objeto --	
<i>verificaExistenciaDe</i> (Valentino,Req7-Ev6)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:verificaExistenciaDe domain-PAIyER:Valentino domain-PAIyER:Req7-Ev6)
<i>verificaExistenciaDe</i> (Req7-Ev6,Libro2)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:verificaExistenciaDe domain-PAIyER:Req7-Ev6 domain-PAIyER:Libro2)
<i>during</i> (Req7-Ev6,Req7-DTI6)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req7-Ev6 domain-PAIyER:Req7-DTI6)
<i>hasBeginning</i> (Req7-DTI6,Req7-DTI6-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req7-DTI6 domain-PAIyER:Req7-DTI6-Sins)
<i>hasEnd</i> (Req7-DTI6,Req7-DTI6-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req7-DTI6 domain-PAIyER:Req7-DTI6-Eins)
}	
A7: [XXX (Empleado \sqsubseteq \exists especificaTipoExistenciasDe.Entidad)]: El empleado Valentino especifica el tipo de existencias (enMano, enOrden, reservadas) del Entidad Libro2.	
A7 = {	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Entidad</i> (Libro2)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Entidad domain-PAIyER:Libro2)
<i>Event</i> (Req7-Ev7)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-

Tabla D.31 ABox de casos de éxito y fracaso en el RF7. Realizar seguimiento de cantidades de entidades en existencias (cont.)

Sintaxis DL	Sintaxis Funcional
	<i>PAIyER:Req7-Ev7</i>
<i>DateTimeInterval(Req7-DTI7)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req7-DTI7)</i>
<i>Instant(Req7-DTI7-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req7-DTI7-Sins)</i>
<i>Instant(Req7-DTI7-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req7-DTI7-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>especificaTipoExistenciasDe(Valentino,Req7-Ev7)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:especificaTipoExistenciasDe domain-PAIyER:Valentino domain-PAIyER:Req7-Ev7)</i>
<i>especificaTipoExistenciasDe(Req7-Ev7,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:especificaTipoExistenciasDe domain-PAIyER:Req7-Ev7 domain-PAIyER:Libro2)</i>
<i>during(Req7-Ev7,Req7-DTI7)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req7-Ev7 domain-PAIyER:Req7-DTI7)</i>
<i>hasBeginning(Req7-DTI7,Req7-DTI7-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req7-DTI7 domain-PAIyER:Req7-DTI7-Sins)</i>
<i>hasEnd(Req7-DTI7,Req7-DTI7-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req7-DTI7 domain-PAIyER:Req7-DTI7-Eins)</i>
}	
A₈: [XXXX (Empleado \sqsubseteq \existsobtieneExistenciasDe.Entidad)]: El empleado Valentino obtiene la cantidad de existencias especificada del Entidad Libro2.	
A ₈ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event(Req7-Ev8)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req7-Ev8)</i>
<i>DateTimeInterval(Req7-DTI8)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req7-DTI8)</i>
<i>Instant(Req7-DTI8-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req7-DTI8-Sins)</i>
<i>Instant(Req7-DTI8-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req7-DTI8-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>obtieneExistenciasDe(Valentino,Req7-Ev8)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:obtieneExistenciasDe domain-PAIyER:Valentino domain-PAIyER:Req7-Ev8)</i>
<i>obtieneExistenciasDe(Req7-Ev8,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:obtieneExistenciasDe domain-PAIyER:Req7-Ev8 domain-PAIyER:Libro2)</i>
<i>during(Req7-Ev8,Req7-DTI8)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req7-Ev8 domain-PAIyER:Req7-DTI8)</i>
<i>hasBeginning(Req7-DTI8,Req7-DTI8-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req7-DTI8 domain-PAIyER:Req7-DTI8-Sins)</i>
<i>hasEnd(Req7-DTI8,Req7-DTI8-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req7-DTI8 domain-PAIyER:Req7-DTI8-Eins)</i>
}	
Caso de fracaso 2 (A₉, A₁₀, A₁₁, A₁₂):	

Tabla D.31 ABox de casos de éxito y fracaso en el RF7. Realizar seguimiento de cantidades de entidades en existencias (cont.)

Sintaxis DL	Sintaxis Funcional
<i>A₉: [X (Empleado ⊆ ∃buscaExistenciasDe.Entidad)]: El empleado Valentino busca existencias del Entidad Libro3.</i>	
...	
<i>A₁₀: [XX (Empleado ⊆ ∃verificaExistenciaDe.Entidad)]: El empleado Valentino verifica la existencia de la entidad Libro3.</i>	
...	
<i>A₁₁: [¬XXX (Empleado ⊆ ∃especificaTipoExistenciasDe.Entidad)]: El empleado Valentino no especifica el tipo de existencias (enMano, enOrden, reservadas) del Entidad Libro3.</i>	
...	
<i>A₁₂: [¬XXXX (Empleado ⊆ ∃obtieneExistenciasDe.Entidad)]: El empleado Valentino no obtiene la cantidad de existencias especificada del Entidad Libro3.</i>	
Caso de éxito 2 (A₁₃, A₁₄, A₁₅, A₁₆):	
<i>A₁₃: [X (Empleado ⊆ ∃buscaExistenciasDe.Entidad)]: El empleado Valentino busca existencias del Entidad Libro4.</i>	
...	
<i>A₁₄: [XX (Empleado ⊆ ∃verificaExistenciaDe.Entidad)]: El empleado Valentino verifica la existencia de la entidad Libro4.</i>	
...	
<i>A₁₅: [XXX (Empleado ⊆ ∃especificaTipoExistenciasDe.Entidad)]: El empleado Valentino especifica el tipo de existencias (enMano, enOrden, reservadas) del Entidad Libro4.</i>	
...	
<i>A₁₆: [XXXX (Empleado ⊆ ∃obtieneExistenciasDe.Entidad)]: El empleado Valentino obtiene la cantidad de existencias especificada de la entidad Libro4.</i>	
Caso de éxito o fracaso n	
<i>A_n</i>	
...	

Tabla D.32 Reglas SWRL para la SoIs de casos de éxito y fracaso de acuerdo al RF Realizar seguimiento de cantidades de artículos en existencias (cont.)

ABox	Regla SWRL
A₃	<i>Event(?e), during(?e, ?i), overlaps(?i, Req7-DT13), especificaTipoExistenciasDe(?e, Libro1), especificaTipoExistenciasDe(Valentino, ?e) -> Nothing(Valentino)</i>
A₄	<i>Event(?e), during(?e, ?i), overlaps(?i, Req7-DT14), obtieneExistenciasDe(?e, Libro1), obtieneExistenciasDe(Valentino, ?e) -> Nothing(Valentino)</i>
A₁₁	<i>Event(?e), during(?e, ?i), overlaps(?i, Req7-DT11), especificaTipoExistenciasDe(?e, Libro3), especificaTipoExistenciasDe(Valentino, ?e) -> Nothing(Valentino)</i>
A₁₂	<i>Event(?e), during(?e, ?i), overlaps(?i, Req7-DT12), obtieneExistenciasDe(?e, Libro3), obtieneExistenciasDe(Valentino, ?e) -> Nothing(Valentino)</i>

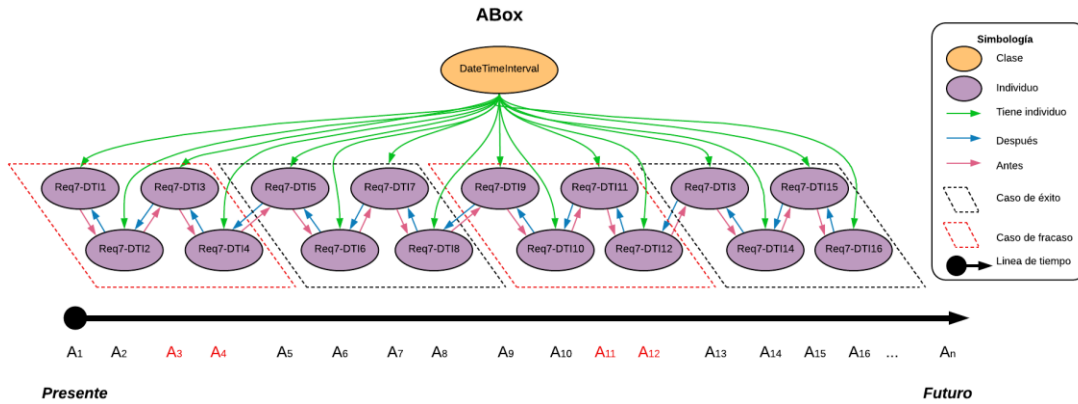


Figura D.28 SoIs en la ABox de casos de éxito y fracaso de acuerdo al RF7. Realizar seguimiento de cantidades de entidades en existencias.

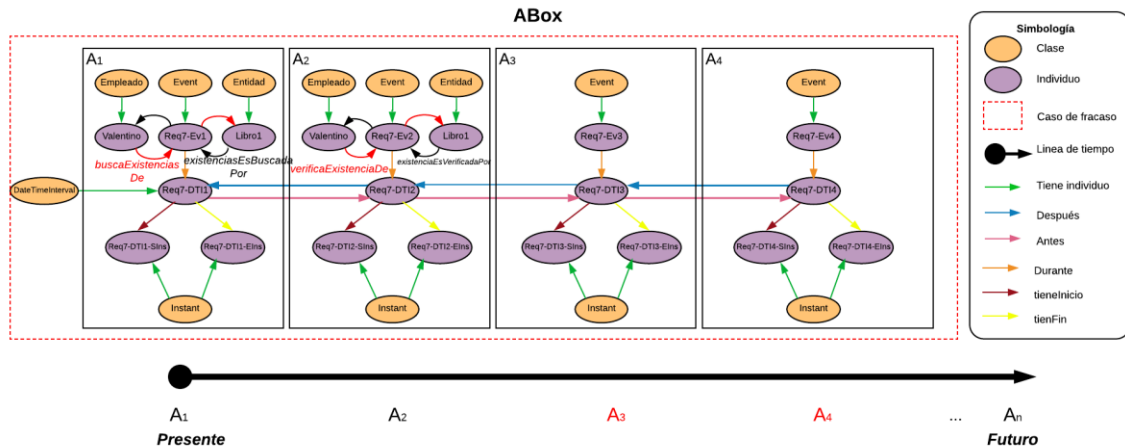


Figura D.29 Caso de fracaso 1 en la ABox de casos de éxito y fracaso de acuerdo al RF7. Realizar el seguimiento de las cantidades de entidades en existencias.

1.2.8 RF8. Realizar seguimiento de las ubicaciones de las entidades

Las secuencias de interpretaciones descritas en esta sección satisfacen/cumplen con el comportamiento dinámico del RF8. Realizar seguimiento de las ubicaciones de las entidades representado formalmente con la fórmula de la Tabla D.33.

Tabla D.33 Fórmula ALC_{LTL} del RF8. Realizar seguimiento de las ubicaciones de las entidades.

$$\mathcal{G} (\mathcal{X} (Empleado \sqsubseteq \exists ubica.Entidad) \rightarrow (\mathcal{X}\mathcal{X} (Empleado \sqsubseteq \exists verificaExistenciaDe.Entidad) \wedge \mathcal{F} (\neg \mathcal{X}\mathcal{X}\mathcal{X} (Empleado \sqsubseteq \exists solicitaUbicacionDe.Entidad) \wedge \neg \mathcal{X}\mathcal{X}\mathcal{X}\mathcal{X} (Empleado \sqsubseteq \exists obtieneUbicacionDe.Entidad)))))$$

1.2.8.1. Casos de éxito

En la Tabla D.34 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la *ABox*) basadas en casos de éxito, que conforman una secuencia que representa el comportamiento dinámico durante el seguimiento de la ubicación de la entidad Libro1 de un inventario. Una representación gráfica simplificada de la secuencia se muestra en la Figura D.30. A manera de ejemplo, en la Figura D.31 se muestra el detalle de las aserciones A_1 , A_2 , A_3 y A_4 en el caso de éxito número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de razonamiento automatizado, los demás casos se definen de manera similar. Sin embargo, difieren en los individuos involucrados.

Tabla D.34 ABox de casos de éxito en el RF8. Realizar seguimiento de las ubicaciones de las entidades.
(cont.)

Sintaxis DL	Sintaxis Funcional
Caso de éxito 1 (A_1, A_2, A_3, A_4):	
A_1 : [X (<i>Empleado</i> \sqsubseteq \exists ubica.Entidad)]: El empleado Valentino ubica el Entidad Libro1.	
$A_1 = \{$	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Entidad</i> (Libro1)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Entidad domain-PAIyER:Libro1)
<i>Event</i> (Req8-Ev1)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req8-Ev1)
<i>DateTimeInterval</i> (Req8-DT11)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req8-DT11)
<i>Instant</i> (Req8-DT11-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req8-DT11-Sins)
<i>Instant</i> (Req8-DT11-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req8-DT11-Eins)
-- Aserciones de propiedades de objeto --	
ubica(Valentino,Req8-Ev1)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: ubica domain-PAIyER:Valentino domain-PAIyER:Req8-Ev1)
ubica(Req8-Ev1,Libro1)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: ubica domain-PAIyER:Req8-Ev1 domain-PAIyER:Libro1)
during(Req8-Ev1,Req8-DT11)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req8-Ev1 domain-PAIyER:Req8-DT11)
hasBeginning(Req8-DT11,Req8-DT11-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req8-DT11 domain-PAIyER:Req8-DT11-Sins)
hasEnd(Req8-DT11,Req8-DT11-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req8-DT11 domain-PAIyER:Req8-DT11-Eins)
$\}$	
A_2 : [XX (<i>Empleado</i> \sqsubseteq \exists verificaExistenciaDe.Entidad)]: El empleado Valentino verifica la existencia de la entidad Libro1.	
$A_2 = \{$	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Entidad</i> (Libro1)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Entidad domain-PAIyER:Libro1)
<i>Event</i> (Req8-Ev2)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req8-Ev2)

Tabla D.34 ABox de casos de éxito en el RF8. Realizar seguimiento de las ubicaciones de las entidades.
(cont.)

Sintaxis DL	⇒	Sintaxis Funcional
<i>DateTimeInterval</i> (Req8-DTI2)	⇒	<i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req8-DTI2)
<i>Instant</i> (Req8-DTI2-Sins)	⇒	<i>ClassAssertion</i> (time:Instant domain-PAIyER:Req8-DTI2-Sins)
<i>Instant</i> (Req8-DTI2-Eins)	⇒	<i>ClassAssertion</i> (time:Instant domain-PAIyER:Req8-DTI2-Eins)
-- Aserciones de propiedades de objeto --		
<i>verificaExistenciaDe</i> (Valentino,R eq8-Ev2)	⇒	<i>ObjectPropertyAssertion</i> (domain-PAIyER:verificaExistenciaDe domain-PAIyER:Valentino domain-PAIyER:Req8-Ev2)
<i>verificaExistenciaDe</i> (Req8-Ev2,Libro1)	⇒	<i>ObjectPropertyAssertion</i> (domain-PAIyER:verificaExistenciaDe domain-PAIyER:Req8-Ev2 domain-PAIyER:Libro1)
<i>during</i> (Req8-Ev2,Req8-DTI2)	⇒	<i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req8-Ev2 domain-PAIyER:Req8-DTI2)
<i>hasBeginning</i> (Req8-DTI2,Req8-DTI2-Sins)	⇒	<i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req8-DTI2 domain-PAIyER:Req8-DTI2-Sins)
<i>hasEnd</i> (Req8-DTI2,Req8-DTI2-Eins)	⇒	<i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req8-DTI2 domain-PAIyER:Req8-DTI2-Eins)
}		
A₃: [XXX (Empleado ⊑ ∃solicitaUbicacionDe.Entidad)]: El empleado Valentino solicita al inventario la ubicacion del Entidad Libro1.		
A ₃ = {		
-- Aserciones de clase --		
<i>Empleado</i> (Valentino)	⇒	<i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Entidad</i> (Libro1)	⇒	<i>ClassAssertion</i> (domain-PAIyER:Entidad domain-PAIyER:Libro1)
<i>Event</i> (Req8-Ev3)	⇒	<i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req8-Ev3)
<i>DateTimeInterval</i> (Req8-DTI3)	⇒	<i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req8-DTI3)
<i>Instant</i> (Req8-DTI3-Sins)	⇒	<i>ClassAssertion</i> (time:Instant domain-PAIyER:Req8-DTI3-Sins)
<i>Instant</i> (Req8-DTI3-Eins)	⇒	<i>ClassAssertion</i> (time:Instant domain-PAIyER:Req8-DTI3-Eins)
-- Aserciones de propiedades de objeto --		
<i>solicitaUbicacionDe</i> (Valentino,R eq8-Ev3)	⇒	<i>ObjectPropertyAssertion</i> (domain-PAIyER:solicitaUbicacionDe domain-PAIyER:Valentino domain-PAIyER:Req8-Ev3)
<i>solicitaUbicacionDe</i> (Req8-Ev3,Libro1)	⇒	<i>ObjectPropertyAssertion</i> (domain-PAIyER:solicitaUbicacionDe domain-PAIyER:Req8-Ev3 domain-PAIyER:Libro1)
<i>during</i> (Req8-Ev3,Req8-DTI3)	⇒	<i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req8-Ev3 domain-PAIyER:Req8-DTI3)
<i>hasBeginning</i> (Req8-DTI3,Req8-DTI3-Sins)	⇒	<i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req8-DTI3 domain-PAIyER:Req8-DTI3-Sins)
<i>hasEnd</i> (Req8-DTI3,Req8-DTI3-Eins)	⇒	<i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req8-DTI3 domain-PAIyER:Req8-DTI3-Eins)
}		
A₄: [XXXX (Empleado ⊑ ∃obtieneUbicacionDe.Entidad)]: El empleado Valentino obtiene la ubicación del Entidad Libro1.		
A ₄ = {		

Tabla D.34 ABox de casos de éxito en el RF8. Realizar seguimiento de las ubicaciones de las entidades.
(cont.)

Sintaxis DL	Sintaxis Funcional
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Entidad</i> (Libro1)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Entidad domain-PAIyER:Libro1)
<i>Event</i> (Req8-Ev4)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req8-Ev4)
<i>DateTimeInterval</i> (Req8-DTI4)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req8-DTI4)
<i>Instant</i> (Req8-DTI4-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req8-DTI4-Sins)
<i>Instant</i> (Req8-DTI4-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req8-DTI4-Eins)
-- Aserciones de propiedades de objeto --	
<i>obtieneUbicacionDe</i> (Valentino,Req8-Ev4)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:obtieneUbicacionDe domain-PAIyER:Valentino domain-PAIyER:Req8-Ev4)
<i>obtieneUbicacionDe</i> (Req8-Ev4,Libro1)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER:obtieneUbicacionDe domain-PAIyER:Req8-Ev4 domain-PAIyER:Libro1)
<i>during</i> (Req8-Ev4,Req8-DTI4)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req8-Ev4 domain-PAIyER:Req8-DTI4)
<i>hasBeginning</i> (Req8-DTI4,Req8-DTI4-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req8-DTI4 domain-PAIyER:Req8-DTI4-Sins)
<i>hasEnd</i> (Req8-DTI4,Req8-DTI4-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-PAIyER:Req8-DTI4 domain-PAIyER:Req8-DTI4-Eins)
}	
Caso de éxito 2 (A₅, A₆, A₇, A₈):	
A ₅ : [X (Empleado \sqsubseteq \exists ubica.Entidad)]: El empleado Valentino ubica el Entidad Libro2.	
A ₅ = {	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Empleado domain-PAIyER:Valentino)
<i>Entidad</i> (Libro2)	\Rightarrow <i>ClassAssertion</i> (domain-PAIyER:Entidad domain-PAIyER:Libro2)
<i>Event</i> (Req8-Ev5)	\Rightarrow <i>ClassAssertion</i> (timeEvents:Event domain-PAIyER:Req8-Ev5)
<i>DateTimeInterval</i> (Req8-DTI5)	\Rightarrow <i>ClassAssertion</i> (time:DateTimeInterval domain-PAIyER:Req8-DTI5)
<i>Instant</i> (Req8-DTI5-Sins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req8-DTI5-Sins)
<i>Instant</i> (Req8-DTI5-Eins)	\Rightarrow <i>ClassAssertion</i> (time:Instant domain-PAIyER:Req8-DTI5-Eins)
-- Aserciones de propiedades de objeto --	
<i>ubica</i> (Valentino,Req8-Ev5)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: ubica domain-PAIyER:Valentino domain-PAIyER:Req8-Ev5)
<i>ubica</i> (Req8-Ev5,Libro2)	\Rightarrow <i>ObjectPropertyAssertion</i> (domain-PAIyER: ubica domain-PAIyER:Req8-Ev5 domain-PAIyER:Libro2)
<i>during</i> (Req8-Ev5,Req8-DTI5)	\Rightarrow <i>ObjectPropertyAssertion</i> (timeEvents:during domain-PAIyER:Req8-Ev5 domain-PAIyER:Req8-DTI5)
<i>hasBeginning</i> (Req8-DTI5,Req8-DTI5-Sins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasBeginning domain-PAIyER:Req8-DTI5 domain-PAIyER:Req8-DTI5-Sins)
<i>hasEnd</i> (Req8-DTI5,Req8-DTI5-Eins)	\Rightarrow <i>ObjectPropertyAssertion</i> (time:hasEnd domain-

Tabla D.34 ABox de casos de éxito en el RF8. Realizar seguimiento de las ubicaciones de las entidades.
(cont.)

Sintaxis DL	Sintaxis Funcional
<i>Eins</i>	<i>PAIyER:Req8-DTI5 domain-PAIyER:Req8-DTI5-Eins</i>
}	
A₆ : [XX (<i>Empleado</i> \sqsubseteq \exists <i>verificaExistenciaDe.Entidad</i>)]: El empleado Valentino verifica la existencia de la entidad <i>Libro2</i> .	
A ₆ = {	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad</i> (Libro2)	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event</i> (Req8-Ev6)	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req8-Ev6)</i>
<i>DateTimeInterval</i> (Req8-DTI6)	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req8-DTI6)</i>
<i>Instant</i> (Req8-DTI6-Sins)	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req8-DTI6-Sins)</i>
<i>Instant</i> (Req8-DTI6-Eins)	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req8-DTI6-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>verificaExistenciaDe</i> (Valentino,R eq8-Ev6)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaExistenciaDe domain-PAIyER:Valentino domain-PAIyER:Req8-Ev6)</i>
<i>verificaExistenciaDe</i> (Req8-Ev6,Libro2)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaExistenciaDe domain-PAIyER:Req8-Ev6 domain-PAIyER:Libro2)</i>
<i>during</i> (Req8-Ev6,Req8-DTI6)	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req8-Ev6 domain-PAIyER:Req8-DTI6)</i>
<i>hasBeginning</i> (Req8-DTI6,Req8-DTI6-Sins)	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req8-DTI6 domain-PAIyER:Req8-DTI6-Sins)</i>
<i>hasEnd</i> (Req8-DTI6,Req8-DTI6-Eins)	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req8-DTI6 domain-PAIyER:Req8-DTI6-Eins)</i>
}	
A₇ : [XXX (<i>Empleado</i> \sqsubseteq \exists <i>solicitaUbicacionDe.Entidad</i>)]: El empleado Valentino solicita al inventario la ubicacion del Entidad <i>Libro2</i> .	
A ₇ = {	
-- Aserciones de clase --	
<i>Empleado</i> (Valentino)	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad</i> (Libro2)	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event</i> (Req8-Ev7)	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req8-Ev7)</i>
<i>DateTimeInterval</i> (Req8-DTI7)	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req8-DTI7)</i>
<i>Instant</i> (Req8-DTI7-Sins)	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req8-DTI7-Sins)</i>
<i>Instant</i> (Req8-DTI7-Eins)	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req8-DTI7-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>solicitaUbicacionDe</i> (Valentino,R eq8-Ev7)	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:solicitaUbicacionDe domain-PAIyER:Valentino domain-PAIyER:Req8-Ev7)</i>
<i>solicitaUbicacionDe</i> (Req8-	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:</i>

Tabla D.34 ABox de casos de éxito en el RF8. Realizar seguimiento de las ubicaciones de las entidades.
(cont.)

Sintaxis DL	Sintaxis Funcional
<i>Ev7, Libro2)</i>	<i>solicitaUbicacionDe domain-PAIyER:Req8-Ev7 domain-PAIyER:Libro2)</i>
<i>during(Req8-Ev7,Req8-DTI7)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req8-Ev7 domain-PAIyER:Req8-DTI7)</i>
<i>hasBeginning(Req8-DTI7,Req8-DTI7-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req8-DTI7 domain-PAIyER:Req8-DTI7-Sins)</i>
<i>hasEnd(Req8-DTI7,Req8-DTI7-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req8-DTI7 domain-PAIyER:Req8-DTI7-Eins)</i>
}	
A₈: [XXXX (Empleado \sqsubseteq \exists obtieneUbicacionDe.Entidad)]: El empleado Valentino obtiene la ubicación del Entidad Libro2.	
A ₈ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event(Req8-Ev8)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req8-Ev8)</i>
<i>DateTimeInterval(Req8-DTI8)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req8-DTI8)</i>
<i>Instant(Req8-DTI8-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req8-DTI8-Sins)</i>
<i>Instant(Req8-DTI8-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req8-DTI8-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>obtieneUbicacionDe(Valentino,Req8-Ev8)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:obtieneUbicacionDe domain-PAIyER:Valentino domain-PAIyER:Req8-Ev8)</i>
<i>obtieneUbicacionDe(Req8-Ev8,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:obtieneUbicacionDe domain-PAIyER:Req8-Ev8 domain-PAIyER:Libro2)</i>
<i>during(Req8-Ev8,Req8-DTI8)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req8-Ev8 domain-PAIyER:Req8-DTI8)</i>
<i>hasBeginning(Req8-DTI8,Req8-DTI8-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req8-DTI8 domain-PAIyER:Req8-DTI8-Sins)</i>
<i>hasEnd(Req8-DTI8,Req8-DTI8-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req8-DTI8 domain-PAIyER:Req8-DTI8-Eins)</i>
}	
Caso de éxito 3 (A₉, A₁₀, A₁₁, A₁₂):	
A₉: [X (Empleado \sqsubseteq \exists ubica.Entidad)]: El empleado Valentino ubica el Entidad Libro3.	
...	
A₁₀: [XX (Empleado \sqsubseteq \exists verificaExistenciaDe.Entidad)]: El empleado Valentino verifica la existencia de la entidad Libro3.	
...	
A₁₁: [XXX (Empleado \sqsubseteq \exists solicitaUbicacionDe.Entidad)]: El empleado Valentino solicita al inventario la ubicacion del Entidad Libro3.	
...	
A₁₂: [XXXX (Empleado \sqsubseteq \exists obtieneUbicacionDe.Entidad)]: El empleado Valentino obtiene la ubicación del Entidad Libro3.	
...	
Caso de éxito 4 (A₁₃, A₁₄, A₁₅, A₁₆):	
A₁₃: [X (Empleado \sqsubseteq \exists ubica.Entidad)]: El empleado Valentino ubica el Entidad Libro4.	

Tabla D.34 ABox de casos de éxito en el RF8. Realizar seguimiento de las ubicaciones de las entidades.
(cont.)

Sintaxis DL	Sintaxis Funcional
...	
$A_{14}: [XX (Empleado \sqsubseteq \exists verificaExistenciaDe.Entidad)]:$ El empleado Valentino verifica la existencia de la entidad Libro4.	
...	
$A_{15}: [XXX (Empleado \sqsubseteq \exists solicitaUbicacionDe.Entidad)]:$ El empleado Valentino solicita al inventario la ubicacion del Entidad Libro4.	
...	
$A_{16}: [XXXX (Empleado \sqsubseteq \exists obtieneUbicacionDe.Entidad)]:$ El empleado Valentino obtiene la ubicación del Entidad Libro4.	
...	
Caso de éxito n	
A_n	
...	

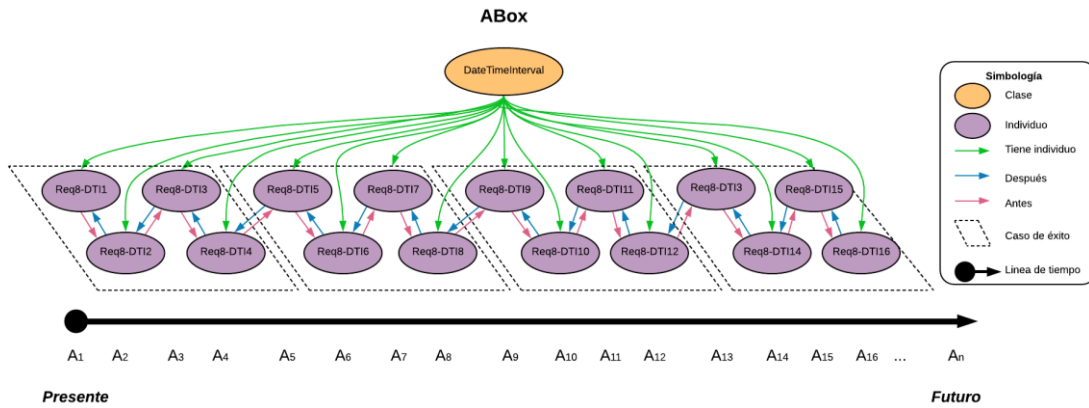


Figura D.30 SoIs en la ABox de casos de éxito de acuerdo al RF8. Realizar seguimiento de las ubicaciones de las entidades.

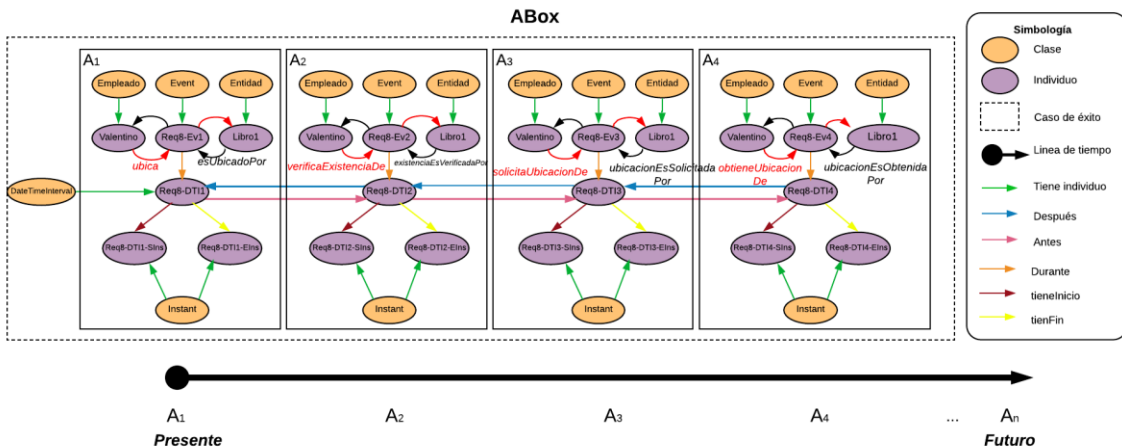


Figura D.31 Caso de éxito 1 en la ABox de casos de éxito de acuerdo al RF8. Realizar seguimiento de las ubicaciones de las entidades.

1.2.8.2. Casos de éxito y fracaso

En la Tabla D.35 se describen algunas interpretaciones (aserciones de clase y aserciones de propiedades de objeto en la *ABox*) basadas en casos de éxito y fracaso, que conforman una secuencia que representa el comportamiento dinámico durante el seguimiento de la ubicación de la entidad *Libro1* de un inventario. Una representación gráfica simplificada de la secuencia se muestra en la Figura D.32. A manera de ejemplo, en la Figura D.33 se muestra el detalle de las aserciones A_1 , A_2 , A_3 y A_4 en el caso de fracaso número 1. Se incluyen algunas propiedades de objeto inferidas en el proceso de razonamiento automatizado, los demás casos se definen de manera similar, sin embargo, difieren en los individuos involucrados.

En la Tabla D.36 se describen las reglas SWRL utilizadas para restringir la existencia de afirmaciones de propiedades de objeto en el intervalo de tiempo correspondiente.

Tabla D.35 ABox de casos de éxito y fracaso en el RF8. Realizar seguimiento de las ubicaciones de las entidades. (cont.)

Sintaxis DL	Sintaxis Funcional
Caso de fracaso 1 (A_1, A_2, A_3, A_4):	
$A_1: [X (Empleado \sqsubseteq \exists ubica.Entidad)]:$ El empleado Valentino ubica el Entidad Libro1.	
$A_1 = \{$	
-- Aserciones de clase --	
$Empleado(Valentino)$	$\Rightarrow ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)$
$Entidad(Libro1)$	$\Rightarrow ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro1)$
$Event(Req8-Ev1)$	$\Rightarrow ClassAssertion(timeEvents:Event domain-PAIyER:Req8-Ev1)$
$DateTimeInterval(Req8-DT11)$	$\Rightarrow ClassAssertion(time:DateTimeInterval domain-PAIyER:Req8-DT11)$
$Instant(Req8-DT11-Sins)$	$\Rightarrow ClassAssertion(time:Instant domain-PAIyER:Req8-DT11-Sins)$
$Instant(Req8-DT11-Eins)$	$\Rightarrow ClassAssertion(time:Instant domain-PAIyER:Req8-DT11-Eins)$
-- Aserciones de propiedades de objeto --	
$ubica(Valentino,Req8-Ev1)$	$\Rightarrow ObjectPropertyAssertion(domain-PAIyER: ubica domain-PAIyER:Valentino domain-PAIyER:Req8-Ev1)$
$ubica(Req8-Ev1,Libro1)$	$\Rightarrow ObjectPropertyAssertion(domain-PAIyER: ubica domain-PAIyER:Req8-Ev1 domain-PAIyER:Libro1)$
$during(Req8-Ev1,Req8-DT11)$	$\Rightarrow ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req8-Ev1 domain-PAIyER:Req8-DT11)$
$hasBeginning(Req8-DT11,Req8-DT11-Sins)$	$\Rightarrow ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req8-DT11 domain-PAIyER:Req8-DT11-Sins)$
$hasEnd(Req8-DT11,Req8-DT11-Eins)$	$\Rightarrow ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req8-DT11 domain-PAIyER:Req8-DT11-Eins)$
$\}$	
$A_2: [XX (Empleado \sqsubseteq \exists verificaExistenciaDe.Entidad)]:$ El empleado Valentino verifica la existencia de la entidad Libro1.	
$A_2 = \{$	
-- Aserciones de clase --	
$Empleado(Valentino)$	$\Rightarrow ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)$

Tabla D.35 ABox de casos de éxito y fracaso en el RF8. Realizar seguimiento de las ubicaciones de las entidades. (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Entidad(Libro1)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro1)</i>
<i>Event(Req8-Ev2)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req8-Ev2)</i>
<i>DateTimeInterval(Req8-DTI2)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req8-DTI2)</i>
<i>Instant(Req8-DTI2-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req8-DTI2-Sins)</i>
<i>Instant(Req8-DTI2-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req8-DTI2-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>verificaExistenciaDe(Valentino,R eq8-Ev2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaExistenciaDe domain-PAIyER:Valentino domain-PAIyER:Req8-Ev2)</i>
<i>verificaExistenciaDe(Req8-Ev2,Libro1)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaExistenciaDe domain-PAIyER:Req8-Ev2 domain-PAIyER:Libro1)</i>
<i>during(Req8-Ev2,Req8-DTI2)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req8-Ev2 domain-PAIyER:Req8-DTI2)</i>
<i>hasBeginning(Req8-DTI2,Req8-DTI2-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req8-DTI2 domain-PAIyER:Req8-DTI2-Sins)</i>
<i>hasEnd(Req8-DTI2,Req8-DTI2-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req8-DTI2 domain-PAIyER:Req8-DTI2-Eins)</i>
}	
A₃: [\negXXX (Empleado \sqsubseteq \existssolicitaUbicacionDe.Entidad)]: El empleado Valentino no solicita al inventario la ubicacion del Entidad Libro1.	
A ₃ = {	
-- Aserciones de clase --	
<i>Event(Req8-Ev3)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req8-Ev3)</i>
<i>DateTimeInterval(Req8-DTI3)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req8-DTI3)</i>
<i>Instant(Req8-DTI3-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req8-DTI3-Sins)</i>
<i>Instant(Req8-DTI3-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req8-DTI3-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>during(Req8-Ev3,Req8-DTI3)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req8-Ev3 domain-PAIyER:Req8-DTI3)</i>
<i>hasBeginning(Req8-DTI3,Req8-DTI3-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req8-DTI3 domain-PAIyER:Req8-DTI3-Sins)</i>
<i>hasEnd(Req8-DTI3,Req8-DTI3-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req8-DTI3 domain-PAIyER:Req8-DTI3-Eins)</i>
}	
A₄: [\negXXXX (Empleado \sqsubseteq \existsobtieneUbicacionDe.Entidad)]: El empleado Valentino no obtiene la ubicación del Entidad Libro1.	
A ₄ = {	
-- Aserciones de clase --	
<i>Event(Req8-Ev4)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req8-Ev4)</i>
<i>DateTimeInterval(Req8-DTI4)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req8-DTI4)</i>
<i>Instant(Req8-DTI4-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req8-</i>

Tabla D.35 ABox de casos de éxito y fracaso en el RF8. Realizar seguimiento de las ubicaciones de las entidades. (cont.)

Sintaxis DL	Sintaxis Funcional
	<i>DTI4-Sins</i>
<i>Instant(Req8-DTI4-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req8-DTI4-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>during(Req8-Ev4,Req8-DTI4)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req8-Ev4 domain-PAIyER:Req8-DTI4)</i>
<i>hasBeginning(Req8-DTI4,Req8-DTI4-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req8-DTI4 domain-PAIyER:Req8-DTI4-Sins)</i>
<i>hasEnd(Req8-DTI4,Req8-DTI4-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req8-DTI4 domain-PAIyER:Req8-DTI4-Eins)</i>
}	
Caso de éxito 1 (A₅, A₆, A₇, A₈):	
A₅: [X (Empleado \sqsubseteq \existsubica.Entidad)]: El empleado Valentino ubica el Entidad Libro2.	
A ₅ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event(Req8-Ev5)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req8-Ev5)</i>
<i>DateTimeInterval(Req8-DTI5)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req8-DTI5)</i>
<i>Instant(Req8-DTI5-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req8-DTI5-Sins)</i>
<i>Instant(Req8-DTI5-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req8-DTI5-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>ubica(Valentino,Req8-Ev5)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER: ubica domain-PAIyER:Valentino domain-PAIyER:Req8-Ev5)</i>
<i>ubica(Req8-Ev5,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER: ubica domain-PAIyER:Req8-Ev5 domain-PAIyER:Libro2)</i>
<i>during(Req8-Ev5,Req8-DTI5)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req8-Ev5 domain-PAIyER:Req8-DTI5)</i>
<i>hasBeginning(Req8-DTI5,Req8-DTI5-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req8-DTI5 domain-PAIyER:Req8-DTI5-Sins)</i>
<i>hasEnd(Req8-DTI5,Req8-DTI5-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req8-DTI5 domain-PAIyER:Req8-DTI5-Eins)</i>
}	
A₆: [XX (Empleado \sqsubseteq \existsverificaExistenciaDe.Entidad)]: El empleado Valentino verifica la existencia de la entidad Libro2.	
A ₆ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event(Req8-Ev6)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req8-Ev6)</i>
<i>DateTimeInterval(Req8-DTI6)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req8-DTI6)</i>
<i>Instant(Req8-DTI6-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req8-</i>

Tabla D.35 ABox de casos de éxito y fracaso en el RF8. Realizar seguimiento de las ubicaciones de las entidades. (cont.)

Sintaxis DL	Sintaxis Funcional
	<i>DTI6-Sins</i>
<i>Instant(Req8-DTI6-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req8-DTI6-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>verificaExistenciaDe(Valentino,R eq8-Ev6)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaExistenciaDe domain-PAIyER:Valentino domain-PAIyER:Req8-Ev6)</i>
<i>verificaExistenciaDe(Req8-Ev6,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:verificaExistenciaDe domain-PAIyER:Req8-Ev6 domain-PAIyER:Libro2)</i>
<i>during(Req8-Ev6,Req8-DTI6)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req8-Ev6 domain-PAIyER:Req8-DTI6)</i>
<i>hasBeginning(Req8-DTI6,Req8-DTI6-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req8-DTI6 domain-PAIyER:Req8-DTI6-Sins)</i>
<i>hasEnd(Req8-DTI6,Req8-DTI6-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req8-DTI6 domain-PAIyER:Req8-DTI6-Eins)</i>
}	
A₇: [XXX (Empleado \sqsubseteq \existssolicitaUbicacionDe.Entidad)]: El empleado Valentino solicita al inventario la ubicacion del Entidad Libro2.	
A ₇ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>
<i>Entidad(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event(Req8-Ev7)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req8-Ev7)</i>
<i>DateTimeInterval(Req8-DTI7)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req8-DTI7)</i>
<i>Instant(Req8-DTI7-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req8-DTI7-Sins)</i>
<i>Instant(Req8-DTI7-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req8-DTI7-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>solicitaUbicacionDe(Valentino,R eq8-Ev7)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:solicitaUbicacionDe domain-PAIyER:Valentino domain-PAIyER:Req8-Ev7)</i>
<i>solicitaUbicacionDe(Req8-Ev7,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:solicitaUbicacionDe domain-PAIyER:Req8-Ev7 domain-PAIyER:Libro2)</i>
<i>during(Req8-Ev7,Req8-DTI7)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req8-Ev7 domain-PAIyER:Req8-DTI7)</i>
<i>hasBeginning(Req8-DTI7,Req8-DTI7-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req8-DTI7 domain-PAIyER:Req8-DTI7-Sins)</i>
<i>hasEnd(Req8-DTI7,Req8-DTI7-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req8-DTI7 domain-PAIyER:Req8-DTI7-Eins)</i>
}	
A₈: [XXXX (Empleado \sqsubseteq \existsobtieneUbicacionDe.Entidad)]: El empleado Valentino obtiene la ubicación del Entidad Libro2.	
A ₈ = {	
-- Aserciones de clase --	
<i>Empleado(Valentino)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Empleado domain-PAIyER:Valentino)</i>

Tabla D.35 ABox de casos de éxito y fracaso en el RF8. Realizar seguimiento de las ubicaciones de las entidades. (cont.)

Sintaxis DL	Sintaxis Funcional
<i>Entidad(Libro2)</i>	\Rightarrow <i>ClassAssertion(domain-PAIyER:Entidad domain-PAIyER:Libro2)</i>
<i>Event(Req8-Ev8)</i>	\Rightarrow <i>ClassAssertion(timeEvents:Event domain-PAIyER:Req8-Ev8)</i>
<i>DateTimeInterval(Req8-DTI8)</i>	\Rightarrow <i>ClassAssertion(time:DateTimeInterval domain-PAIyER:Req8-DTI8)</i>
<i>Instant(Req8-DTI8-Sins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req8-DTI8-Sins)</i>
<i>Instant(Req8-DTI8-Eins)</i>	\Rightarrow <i>ClassAssertion(time:Instant domain-PAIyER:Req8-DTI8-Eins)</i>
-- Aserciones de propiedades de objeto --	
<i>obtieneUbicacionDe(Valentino,R eq8-Ev8)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:obtieneUbicacionDe domain-PAIyER:Valentino domain-PAIyER:Req8-Ev8)</i>
<i>obtieneUbicacionDe(Req8-Ev8,Libro2)</i>	\Rightarrow <i>ObjectPropertyAssertion(domain-PAIyER:obtieneUbicacionDe domain-PAIyER:Req8-Ev8 domain-PAIyER:Libro2)</i>
<i>during(Req8-Ev8,Req8-DTI8)</i>	\Rightarrow <i>ObjectPropertyAssertion(timeEvents:during domain-PAIyER:Req8-Ev8 domain-PAIyER:Req8-DTI8)</i>
<i>hasBeginning(Req8-DTI8,Req8-DTI8-Sins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasBeginning domain-PAIyER:Req8-DTI8 domain-PAIyER:Req8-DTI8-Sins)</i>
<i>hasEnd(Req8-DTI8,Req8-DTI8-Eins)</i>	\Rightarrow <i>ObjectPropertyAssertion(time:hasEnd domain-PAIyER:Req8-DTI8 domain-PAIyER:Req8-DTI8-Eins)</i>
}	
Caso de fracaso 2 (A₉, A₁₀, A₁₁, A₁₂):	
<i>A₉: [X (Empleado \sqsubseteq \existsubica.Entidad)]: El empleado Valentino ubica el Entidad Libro3.</i>	
...	
<i>A₁₀: [XX (Empleado \sqsubseteq \existsverificaExistenciaDe.Entidad)]: El empleado Valentino verifica la existencia de la entidad Libro3.</i>	
...	
<i>A₁₁: [\negXXX (Empleado \sqsubseteq \existssolicitaUbicacionDe.Entidad)]: El empleado Valentino no solicita al inventario la ubicacion del Entidad Libro3.</i>	
...	
<i>A₁₂: [\negXXXX (Empleado \sqsubseteq \existsobtieneUbicacionDe.Entidad)]: El empleado Valentino no obtiene la ubicación del Entidad Libro3.</i>	
}	
Caso de éxito 2 (A₁₃, A₁₄, A₁₅, A₁₆):	
<i>A₁₃: [X (Empleado \sqsubseteq \existsubica.Entidad)]: El empleado Valentino ubica el Entidad Libro4.</i>	
...	
<i>A₁₄: [XX (Empleado \sqsubseteq \existsverificaExistenciaDe.Entidad)]: El empleado Valentino verifica la existencia de la entidad Libro4.</i>	
...	
<i>A₁₅: [XXX (Empleado \sqsubseteq \existssolicitaUbicacionDe.Entidad)]: El empleado Valentino solicita al inventario la ubicacion del Entidad Libro4.</i>	
...	
<i>A₁₆: [XXXX (Empleado \sqsubseteq \existsobtieneUbicacionDe.Entidad)]: El empleado Valentino obtiene la ubicación del Entidad Libro4.</i>	
}	
Caso de éxito o fracaso n	
<i>A_n</i>	
...	

Tabla D.36 Reglas SWRL para la SoIs de casos de éxito y fracaso de acuerdo al RF Realizar seguimiento de las ubicaciones de las entidades. (cont.)

ABox	Regla SWRL
A₃	<i>Event(?e, during(?e, ?i), overlaps(?i, Req8-DTI3), solicitaUbicacionDe(?e, Libro1), solicitaUbicacionDe(Valentino, ?e) -> Nothing(Valentino)</i>
A₄	<i>Event(?e, during(?e, ?i), overlaps(?i, Req8-DTI4), obtieneUbicacionDe(?e, Libro1), obtieneUbicacionDe(Valentino, ?e) -> Nothing(Valentino)</i>
A₁₁	<i>Event(?e, during(?e, ?i), overlaps(?i, Req8-DTI11), solicitaUbicacionDe(?e, Libro3), solicitaUbicacionDe(Valentino, ?e) -> Nothing(Valentino)</i>
A₁₂	<i>Event(?e, during(?e, ?i), overlaps(?i, Req8-DTI12), obtieneUbicacionDe(?e, Libro3), obtieneUbicacionDe(Valentino, ?e) -> Nothing(Valentino)</i>

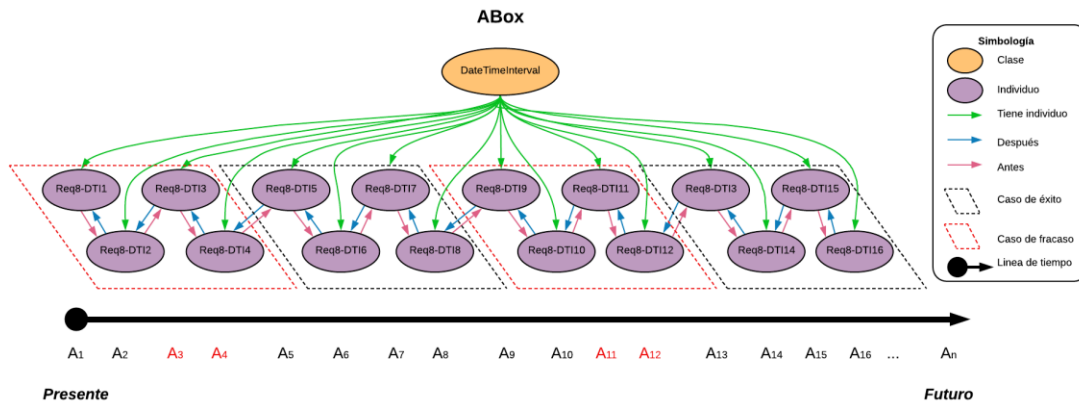


Figura D.32 SoIs en la ABox de casos de éxito y fracaso de acuerdo al RF8. Realizar seguimiento de las ubicaciones de las entidades.

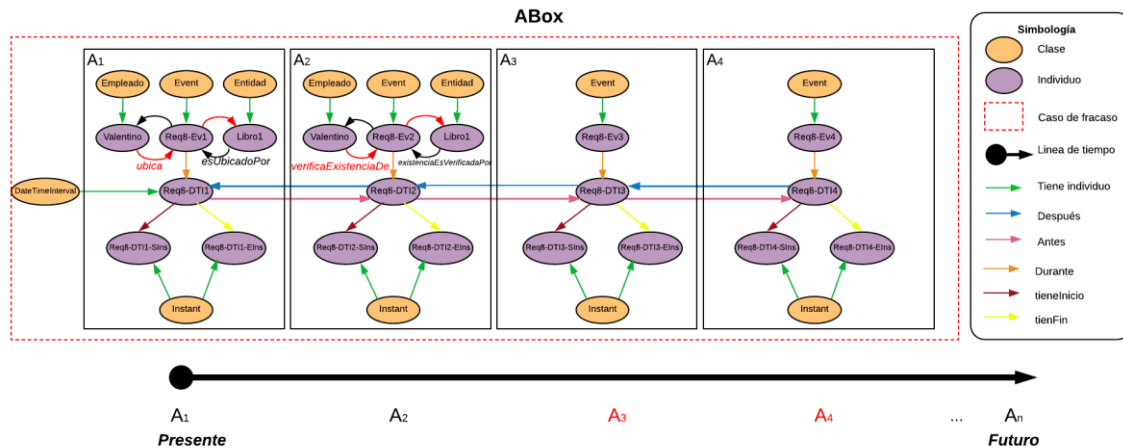


Figura D.33 Caso de fracaso 1 en la ABox de casos de éxito y fracaso de acuerdo al RF8. Realizar seguimiento de las ubicaciones de las entidades.

Anexo E. Documentación TSCC

En este anexo, se describe el conjunto de documentos utilizados (propuestos en (Piedrola Domínguez N., 2019)) en esta investigación para la realizar la Verificación de Consistencia Semántica Temporal (TSCC) en ontologías OTPAEs.

La documentación utilizada para llevar a cabo una verificación TSCC, está basada en el Estándar para la Documentación de Pruebas de Sistema y Software IEEE (IEEE, 2008). Consiste en seis documentos, los cuales se describen a continuación:

1. Plan

Documento que describe los detalles del plan de verificación. Se integra de los siguientes apartados:

1. Introducción
 - 1.1 Identificador del Documento
 - 1.2 Alcance
 - 1.3 Referencias
2. Detalles del Plan de verificación
 - 2.1 Especificaciones de comportamiento a verificar
 - 2.2 Matriz de Trazabilidad de verificación
 - 2.3 Criterio de Aceptación de la verificación
 - 2.4 Criterio para Suspensión y Reanudación de la verificación
 - 2.5 Entregables
3. Administración de Verificación
 - 3.1 Actividades Planificadas
 - 3.2 Infraestructura
4. Referencias

2. Diseño

Documento que describe los detalles del diseño de la verificación:

1. Introducción
 - 1.1 Identificador de documento
 - 1.2 Alcance
 - 1.3 Referencias
2. Detalles del Diseño de la verificación
 - 2.1 Características a verificar
 - 2.2 Estrategia de verificación
 - 2.3 Criterio de aceptación del caso de verificación
 - 2.4 Entregables de verificación
3. Referencias

3. Casos

Documento que describe los objetivos, requisitos y resultados de cada caso. Se integra de los siguientes apartados:

1. Introducción
 - 1.1 Alcance

- 1.2 Referencias
- 2. Caso de verificación C-01
 - 2.1 Identificador del caso de verificación
 - 2.2 Objetivo
 - 2.3 Requisitos
 - 2.4 Resultado
 - 2.4.1 Protégé
 - 2.4.2 Consola de Protégé
 - ...
 - ...
 - ...
- n. Caso de verificación C-n
 - n.1 Identificador del caso de verificación
 - n.2 Objetivo
 - n.3 Requisitos
 - n.4 Resultado
 - n.4.1 Protégé
 - n.4.2 Consola de Protégé
- x. Referencias

4. Procedimiento

Documento que describe los pasos a ejecutar para examinar el conjunto de verificaciones especificadas en el documento de Casos:

- 1. Introducción
 - 1.1 Alcance
 - 1.2 Referencias
- 2. Procedimiento de verificación PrV-01
 - 2.1 Requerimientos especiales
 - 2.2 Descripción del conjunto de pasos a seguir para ejecutar el caso de verificación CV-01
 - ...
 - ...
 - ...
- n. Procedimiento de verificación PrV-n
 - 2.1 Requerimientos especiales
 - 2.2 Descripción del conjunto de pasos a seguir para ejecutar el caso de verificación CV-n
- x. Referencias

5. Registro

Documento que provee un registro cronológico acerca de detalles relevantes en la ejecución de la verificación. Se integra de los siguientes apartados:

1. Introducción

1.1 Alcance

1.2 Referencias

2. Registro de verificación RegV-01

2.1 Descripción

2.2 Actividades y eventos de la verificación

2.3 Inconsistencias

2.4 Ejemplo de secuencia de interpretaciones temporales utilizada para la verificación

...

...

...

n. Registro de verificación RegV-n

n.1 Descripción

n.2 Actividades y eventos de la verificación

n.3 Inconsistencias

n.4 Ejemplo de secuencia de interpretaciones temporales utilizada para la verificación

x. Referencias

6. Reporte

Documento que resume las actividades de verificación y sus resultados, además contiene evaluaciones de las verificaciones correspondientes. Se integra de los siguientes apartados:

1. Introducción

1.1 Alcance

1.2 Referencias

2. Detalles

2.1 Resultados generales

2.2 Resultados detallados

2.3 Conclusiones y recomendaciones

3. Referencias

Un ejemplo que demuestra el uso de esta documentación, es la verificación TSCC en el patrón de análisis PAI contenida en la carpeta *Entregables/Entregable 9/TSCC*, almacenado en el disco que acompaña a esta tesis (para mayor referencia consultar el *Anexo G. Contenido del disco*).

Anexo F. Documentación SC

En este anexo, se presenta el conjunto de documentos utilizados (propuestos en (Piedrola Domínguez N., 2019)) en esta investigación para llevar a cabo la verificación SC de especificaciones de comportamiento en trayectorias de estado, derivadas de la reducción de secuencias de interpretaciones en las ontologías OTPAEs.

La documentación utilizada para llevar a cabo una verificación SC, está basada en el Estándar para la Documentación de Pruebas de Sistema y Software IEEE (IEEE, 2008). Consiste en seis documentos, los cuales se describen a continuación:

1. Plan

Documento que describe los detalles del plan de verificación. Se integra de los siguientes apartados:

1. Introducción
 - 1.1 Identificador del Documento
 - 1.2 Alcance
 - 1.3 Referencias
2. Detalles del Plan de verificación
 - 2.1 Especificaciones de comportamiento a verificar
 - 2.2 Matriz de Trazabilidad de verificación
 - 2.3 Criterio de Aceptación de la verificación
 - 2.4 Criterio para Suspensión y Reanudación de la verificación
 - 2.5 Entregables
3. Administración de Verificación
 - 3.1 Actividades Planificadas
 - 3.2 Infraestructura
4. Referencias

2. Diseño

Documento que describe los detalles del diseño de la verificación:

1. Introducción
 - 1.1 Identificador de documento
 - 1.2 Alcance
 - 1.3 Referencias
2. Detalles del Diseño de la verificación
 - 2.1 Características a verificar
 - 2.2 Estrategia de verificación
 - 2.3 Criterio de aceptación del caso de verificación
 - 2.4 Entregables de verificación
3. Referencias

3. Casos

Documento que describe los objetivos, requisitos y resultados de cada caso. Se integra de los siguientes apartados:

1. Introducción
 - 1.1 Alcance

- 1.2 Referencias
- 2. Caso de verificación C-01
 - 2.1 Identificador del caso de verificación
 - 2.2 Objetivo
 - 2.3 Requisitos
 - 2.4 Resultado
 - 2.4.1 nuXmv
 - ...
 - ...
 - ...
- n. Caso de verificación C-n
 - n.1 Identificador del caso de verificación
 - n.2 Objetivo
 - n.3 Requisitos
 - n.4 Resultado
 - n.4.1 nuXmv
- x. Referencias

4. Procedimiento

Documento que describe los pasos a ejecutar para examinar el conjunto de verificaciones especificadas en el documento de Casos:

- 1. Introducción
 - 1.1 Alcance
 - 1.2 Referencias
- 2. Procedimiento de verificación PrV-01
 - 2.1 Requerimientos especiales
 - 2.2 Descripción del conjunto de pasos a seguir para ejecutar el caso de verificación CV-01
 - ...
 - ...
 - ...
- n. Procedimiento de verificación PrV-n
 - 2.1 Requerimientos especiales
 - 2.2 Descripción del conjunto de pasos a seguir para ejecutar el caso de verificación CV-n
- x. Referencias

5. Registro

Documento que provee un registro cronológico acerca de detalles relevantes en la ejecución de la verificación. Se integra de los siguientes apartados:

- 1. Introducción
 - 1.1 Alcance

- 1.2 Referencias
- 2. Registro de verificación RegV-01
 - 2.1 Descripción
 - 2.2 Actividades y eventos de la verificación
 - 2.3 Anomalías
 - 2.4 Ejemplo de archivo .smv para la verificación
 - ...
 - ...
 - ...
- n. Registro de verificación RegV-n
 - n.1 Descripción
 - n.2 Actividades y eventos de la verificación
 - n.3 Anomalías
 - n.4 Ejemplo de archivo .smv para la verificación
- x. Referencias

6. Reporte

Documento que resume las actividades de verificación y sus resultados, además contiene evaluaciones de las verificaciones correspondientes. Se integra de los siguientes apartados:

- 1. Introducción
 - 1.1 Alcance
 - 1.2 Referencias
- 2. Detalles
 - 2.1 Resultados generales
 - 2.2 Resultados detallados
 - 2.3 Conclusiones y recomendaciones
- 3. Referencias

Un ejemplo que demuestra el uso de esta documentación, es la verificación SC en el patrón de análisis PAI contenida en la carpeta *Entregables/Entregable 9/SC*, almacenado en el disco que acompaña a esta tesis (para mayor referencia consultar el *Anexo G. Contenido del disco*).

Anexo G. Contenido del disco

En este anexo, se describe la estructura de las carpetas y archivos contenidos en el disco que acompaña este trabajo de tesis.

1. Directorios

A continuación, se presenta la lista de directorios contenidos en el disco que acompaña esta tesis.

- Entregables
 - Entregable 1
 - Entregable 2
 - Entregable 3 y 4
 - Entregable 5
 - Entregable 6
 - Entregable 7
 - Entregable 8
 - Razonador Pellet
 - Protégé 4.3
 - DRT
 - Chronos
 - Ontologías
 - OPAE PAI
 - OTPAE PAI
 - Entregable 9
 - TSCC
 - SC
 - Entregable 10
 - Entregable 11
 - Ontologías
 - OPAE PAIyER
 - OTPAE PAIyER
 - Entregable 12
 - TSCC
 - SC
 - Entregable 13
 - Ontologías
 - OPAE PAI
 - OTPAE PAI
 - OPAE PAIyER
 - OTPAE PAIyER

2. Documentación generada

Para lograr el objetivo general (presentado en la sección *1.3.1. Objetivo general*), se realizaron las actividades presentadas a continuación.

1. Se realizó un estudio y análisis de los trabajos relacionados al tema de investigación.
2. Se realizó un estudio y análisis sobre el uso de la Lógica Descriptiva y Temporal.
3. Se realizó un análisis de la estructura de la OTPAE del PAER con la finalidad de conocer los elementos que componen a la ontología temporal.
4. Se realizó un análisis de la estructura de la OTPAE del PAC con la finalidad de conocer los elementos que componen a la ontología temporal.
5. Se realizó un estudio y análisis de técnicas utilizados para la combinación de lógicas descriptiva y temporal.
6. Se describió la combinación de la OTPAE del PAC con la OTPAE del PAER.
7. Se describió de forma preliminar el método utilizado para realizar la actividad 6. Posteriormente, se refinó este método para obtener finalmente el método COMOTPA.
8. Con la finalidad de obtener una nueva OTPAE de un nuevo patrón de análisis, se creó y documentó la OTPAE del Patrón de Análisis para Inventarios (PAI). Además, se realizó la verificación TSCC y SC de la OTPAE del PAI.
9. Se realizó un caso de estudio combinando la OTPAE del PAI con la OTPAE del PAER. Además, se realizó la verificación de la OTPAE compuesta.

A continuación, en la Tabla G.1, se muestra la lista de actividades mencionadas anteriormente clasificadas de acuerdo al objetivo específico (ver sección *1.3.2. Objetivos específicos*). Se incluyen los entregables generados (documentación), así como su ubicación en el directorio *Entregables*.

Tabla G.1 Documentación generada (cont.)

Objetivo específico	Actividad	Documentación
1	1	Entregable 1/Informe sobre trabajos relacionados.pdf
	2	Entregable 2/Informe sobre el uso de la Lógica Descriptiva y Lógica Temporal.pdf
	3 y 4	Entregable 3 y 4/Estructura de la OTPAE del PAER y de la OTPAE del PAC.pdf.
	5	Entregable 5/ Revisión y estudio de técnicas y/o métodos para la combinación de lógicas descriptiva y temporal.pdf
2	6	Entregable 6/Especificación de la primera ontología compuesta.pdf
	7	Entregable 7/Metodología para la combinación de OTPAEs (Preliminar).pdf.
	8	Entregable 8/Reporte técnico - Creación de la OTPAE del PAI.pdf

Tabla G.1 Documentación generada (cont.)

Objetivo específico	Actividad	Documentación
		<p>Verificación de Consistencia Semántica Temporal PAI (TSCC) Entregable 9/TSCC/1. Plan de TSCC PAI-PV.pdf Entregable 9/TSCC/2. Diseño de TSCC PAI -DV.pdf Entregable 9/TSCC/3. Casos de TSCC PAI -CV.pdf Entregable 9/TSCC/4. Procedimiento de TSCC PAI -PrV.pdf Entregable 9/TSCC/5. Registro de TSCC PAI -RegV.pdf Entregable 9/TSCC/6. Reporte de TSCC PAI -RepV.pdf</p> <p>Verificación de Satisfacibilidad PAI (SC) Entregable 9/SC/1. Plan de SC PAI -PV.pdf Entregable 9/SC/2. Diseño de SC PAI -DV.pdf Entregable 9/SC/3. Casos de SC PAI -CV.pdf Entregable 9/SC/4. Procedimiento de SC PAI-PrV.pdf Entregable 9/SC/5. Registro de SC PAI -RegV.pdf Entregable 9/SC/6. Reporte de SC PAI -RepV.pdf</p>
	7	Entregable 10/ Reporte técnico - Metodología para la Combinación de Ontologías Temporales de Patrones de Análisis (COMOTPA).pdf
3	9	<p>Entregable 11/Caso de estudio – Combinación de la OTPAE del PAER con la OTPAE del PAI.pdf</p> <p>Verificación de Consistencia Semántica Temporal PAIyER (TSCC) Entregable 12/TSCC/1. Plan de TSCC PAIyER-PV.pdf Entregable 12/TSCC/2. Diseño de TSCC PAIyER-DV.pdf Entregable 12/TSCC/3. Casos de TSCC PAIyER-CV.pdf Entregable 12/TSCC/4. Procedimiento de TSCC PAIyER-PrV.pdf Entregable 12/TSCC/5. Registro de TSCC PAIyER-RegV.pdf Entregable 12/TSCC/6. Reporte de TSCC PAIyER-RepV.pdf</p> <p>Verificación de Satisfacibilidad PAIyER (SC) Entregable 12/SC/1. Plan de SC PAIyER-PV.pdf Entregable 12/SC/2. Diseño de SC PAIyER-DV.pdf Entregable 12/SC/3. Casos de SC PAIyER-CV.pdf Entregable 12/SC/4. Procedimiento de SC PAIyER-PrV.pdf Entregable 12/SC/5. Registro de SC PAIyER-RegV.pdf Entregable 12/SC/6. Reporte de SC PAIyER-RepV.pdf</p>

En el directorio *Entregables/Entregable 13* se encuentra la documentación de los resultados obtenidos en esta investigación.

Finalmente, en el directorio *Ontologías*, se encuentran las versiones finales de las ontologías estáticas y temporales de los patrones de análisis PAI y PAIyER (resultado de la combinación de los patrones de análisis PAI y PAER), respectivamente. Por cada especificación de comportamiento dinámico en los requerimientos funcionales de los patrones de análisis, se generaron dos SoIs (basadas en casos de éxito (E) y casos de éxito y fracaso/alternativo (EF)). En la Tabla G.2 y en la Tabla G.3, se menciona el tipo de conocimiento almacenado por las ontologías, el anexo correspondiente y la sección en donde puede encontrarse el detalle de las SoIs.

Tabla G.2 Versiones finales de las ontologías del patrón de análisis PAI (cont.)

Ontología	Conocimiento	Anexo	Sección
OPAE PAI/opaePAI.owl	Estático	Anexo A. Ontología OPAE del patrón de análisis PAI	
OTPAE PAI/Req1-otpaepAI-E.owl	Temporal	Anexo B. Ontología OTPAE del patrón de análisis PAI	1.2.1.1. Casos de éxito
OTPAE PAI/Req1-otpaepAI-EF.owl			1.2.1.2. Casos de éxito y fracaso
OTPAE PAI/Req2-otpaepAI-E.owl			1.2.2.1. Casos de éxito
OTPAE PAI/Req2-otpaepAI-EF.owl			1.2.2.2. Casos de éxito y fracaso
OTPAE PAI/Req3-otpaepAI-E.owl			1.2.3.1. Casos de éxito
OTPAE PAI/Req3-otpaepAI-EF.owl			1.2.3.2. Casos de éxito y fracaso

Tabla G.3 Versiones finales de las ontologías del patrón de análisis PAIyER (cont.)

Ontología	Conocimiento	Anexo	Sección
OPAE PAIyER/opaePAIyER.owl	Estático	Anexo C. Ontología OPAE del patrón de análisis PAIyER	
OTPAE PAIyER/Req1-otpaepAIyER-E.owl	Temporal	Anexo D. Ontología OTPAE del patrón de análisis PAIyER	1.2.1.1. Casos de éxito
OTPAE PAIyER/Req1-otpaepAIyER-EF.owl			1.2.1.2. Casos de éxito y fracaso
OTPAE PAIyER/Req2-otpaepAIyER-E.owl			1.2.2.1. Casos de éxito
OTPAE PAIyER/Req2-otpaepAIyER-EF.owl			1.2.2.2. Casos de éxito y fracaso
OTPAE PAIyER/Req3-otpaepAIyER-E.owl			1.2.3.1. Casos de éxito
OTPAE PAIyER/Req3-otpaepAIyER-EF.owl			1.2.3.2. Casos de éxito y fracaso
OTPAE PAIyER/Req4-otpaepAIyER-E.owl			1.2.4.1. Casos de éxito
OTPAE PAIyER/Req4-otpaepAIyER-EF.owl			1.2.4.2. Casos de éxito y fracaso
OTPAE PAIyER/Req5-otpaepAIyER-E.owl			1.2.5.1. Casos de éxito
OTPAE PAIyER/Req5-otpaepAIyER-EF.owl			1.2.5.2. Casos de éxito y fracaso
OTPAE PAIyER/Req6-otpaepAIyER-E.owl			1.2.6.1. Casos de éxito
OTPAE PAIyER/Req6-otpaepAIyER-EF.owl			1.2.6.2. Casos de éxito y fracaso
OTPAE PAIyER/Req7-otpaepAIyER-E.owl			1.2.7.1. Casos de éxito

Tabla G.3 Versiones finales de las ontologías del patrón de análisis PAIyER (cont).

Ontología	Conocimiento	Anexo	Sección
OTPAE PAIyER/Req7- otpaePAIyER-EF.owl			1.2.7.2. Casos de éxito y fracaso
OTPAE PAIyER/Req8- otpaePAIyER-E.owl			1.2.8.1. Casos de éxito
OTPAE PAIyER/Req8- otpaePAIyER-EF.owl			1.2.8.2. Casos de éxito y fracaso