

Centro Nacional de Investigación y Desarrollo Tecnológico

Subdirección Académica

Departamento de Ciencias Computacionales

TESIS DE MAESTRÍA EN CIENCIAS

**Almacenamiento y Uso de Patrones de Análisis Apegados a su
Semántica**

presentada por

Ing. Ramiro Mar López Ramírez

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

Director de tesis:
Dr. Moisés González García

Cuernavaca, Morelos, México. Junio de 2015.

"2015, Año del Generalísimo José María Morelos y Pavón"

Cuernavaca, Mor., 08/junio/2015
OFICIO No. DCC/055/2015

Asunto: Aceptación de documento de tesis

C. DR. GERARDO V. GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

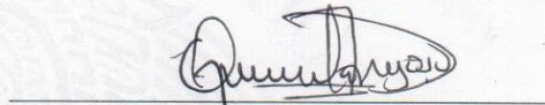
Por este conducto, los integrantes de Comité Tutorial del **C. Ramiro Mar López Ramírez**, con número de control M13CE030, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis profesional titulado "**Almacenamiento y uso de patrones de análisis apegados a su semántica**" y hemos encontrado que se han realizado todas las correcciones y observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

DIRECTOR DE TESIS

REVISOR 1



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



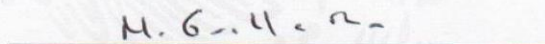
Dra. Olivia Graciela Fragosó Díaz
Doctora en Ciencias en Ciencias de la
Computación
7420199

REVISOR 2

REVISOR 3



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



M.C. Mario Guillén Rodríguez
Maestro en Ciencias con Especialidad en
Sistemas Computacionales
7573768

C.p. Lic. Guadalupe Garrido Rivera - Jefa del Departamento de Servicios Escolares.
Estudiante
Expediente

AMR/lmz

"2015, Año del Generalísimo José María Morelos y Pavón"

Cuernavaca, Mor., 9 de junio de 2015
OFICIO No. SAC/130/2015

Asunto: Autorización de impresión de tesis

ING. RAMIRO MAR LÓPEZ RAMÍREZ
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS
DE LA COMPUTACIÓN
PRESENTE

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado "**Almacenamiento y uso de patrones de análisis apegados a su semántica**", 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
"CONOCIMIENTO Y TECNOLOGÍA AL SERVICIO DE MÉXICO"



DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO



CENTRO NACIONAL DE
INVESTIGACIÓN Y
DESARROLLO
TECNOLÓGICO
SUBDIRECCIÓN
ACADÉMICA

C.p. Lic. Guadalupe Garrido Rivera.- Jefa del Departamento de Servicios Escolares.
Expediente

GVGR/mcr

*D*edicatoria

A Dios...

por ser mi amigo y compañero, por estar siempre presente cuando más te necesito, porque si te tengo a ti nada me falta. Por darme el don de la vida, por brindarme salud, por darme el temple y la inteligencia que se necesitan para afrontar los obstáculos que la vida conlleva, y por permitirme alcanzar este gran logro.

A mis padres...

Ramiro López de la Cruz y María Auxilio Ramírez Marín por regalarme su cariño y apoyo incondicional durante toda mi vida. Por brindarme el mejor de los ejemplos, por inculcarme grandes principios y valores, por impulsarme siempre a ser mejor, por apoyarme en todo momento, por estar siempre presentes en todos mis logros, porque les debo todo lo que soy. Estoy orgulloso de ser su hijo. Siempre estaré eternamente agradecido con Dios por regalarme a estos padres tan maravillosos.

A mis hermanos...

Jairo Rey y Luis Ronaldo por todas las sonrisas y alegrías que hemos compartido juntos. Por demostrarme siempre su cariño. Estoy orgulloso de ustedes tan pequeños y ya son exitosos, estoy seguro que un futuro, lo serán aún más.

A mi novia...

Paty López por ser muchas veces mi inspiración y mi motivación. Por darme su amor y paciencia, que a pesar de la distancia siempre estuvo conmigo, gracias por esos momentos inolvidables que hemos vivido juntos, gracias por tu apoyo y por animarme a seguir superándome día a día. Gracias por esas palabras de aliento que me diste, por entenderme y por seguir a mí lado a pesar de todo.

A mis abuelitas...

Belén Marín Q.E.P.D. y Juanita de la Cruz por regalarme una infancia llena de amor y alegría. Por sus atenciones y cuidados que me brindaron desde que era niño.

*A*gradecimientos

Al Consejo Nacional de Ciencia y Tecnología (CONACYT), por el gran apoyo económico recibido, ya que fue esencial para poder llevar a cabo mis estudios de maestría.

Al Centro Nacional de Investigación y Desarrollo Tecnológico CENIDET, al Departamento de Ciencias Computacionales y al área de Ingeniería de Software por darme la oportunidad de ser parte de sus filas y de sentirme orgulloso de pertenecer a este importante centro de investigación.

A mi director de tesis Dr. Moisés González García por su guía y confianza, por compartir su conocimiento y consejos, sin los cuales no hubiera sido posible la realización de esta investigación.

A mis revisores de tesis: Dr. René Santaolaya Salgado, Dra. Olivia Graciela Fragoso Díaz y M.C. Mario Guillén Rodríguez por sus valiosos consejos que me permitieron ser mejor y realizar un trabajo de calidad.

Al Dr. Guillermo Rodríguez, al M.C. Fernando Pech, al M.C. Osvaldo Daniel Fernández, y al Ing. José Domingo Juárez por su valioso tiempo y aportaciones a esta investigación.

A mis compañeros y amigos de generación (Vitervo, Marcos, Omar, Vicente, Balam, Miguel, Alma y Bernardo) por permitirme recorrer a su lado este largo camino, que con su apoyo, amistad y momentos que compartimos, fue una experiencia inolvidable.

A mis caseros Don Alfredo y Doña Georgina por abrirme las puertas de su casa durante estos dos años, por sus atenciones y generosidad, gracias.

RESUMEN

Al desarrollar un software, en la etapa de análisis, se han identificado situaciones semejantes, a partir de las que se producen soluciones parecidas a las sintetizadas en desarrollos previos. El análisis repetido de situaciones semejantes, en el dominio del problema, originó la necesidad de identificar patrones de análisis (PAs), los cuales representan las mejores prácticas para estructurar y describir problemas específicos.

La plantilla para descripción de PAs, propuesta por Ricardo Raminhos (Raminhos et al. 2006), se generó a partir de la experiencia, por analogías, y basándose en los patrones de diseño, y es una de las más utilizadas. Esta plantilla se refinó por nuestro grupo de investigación, en una investigación previa, incluyendo conceptos del modelo de estabilidad de software (Fayad & Altman 2001) y construyendo su ontología. Sin embargo los patrones de análisis son poco utilizados.

En esta investigación, consideramos que la escasa utilización de los PAs se debe a que sus descripciones no incluyen los conceptos del problema y sus relaciones (semántica). Para incrementar la utilización de los PAs, en esta tesis se propone almacenar los PAs como conocimiento, utilizando ontologías estructuradas en tres niveles de abstracción: 1) ontología de plantilla estable de PAs (una ontología única) y sus individuos (uno para cada PA), 2) ontologías de PAs (una ontología para cada PA) y los individuos de cada ontología (uno por cada caso de estudio), y 3) ontologías de casos de estudio (una para cada caso).

Para demostrar la factibilidad de este tipo de almacenamiento del conocimiento, se propone una metodología para construir la ontología de un PA. Utilizando dicha metodología y el editor de ontologías Protégé, se construyó la ontología del “PA para Reservación y Uso de Entidades Reutilizables”, tomando en consideración dos casos de estudio: 1) Reservación y uso de material bibliográfico en el “Centro de Información del CENIDET”, y 2) Reservación y renta de bienes raíces en la empresa “DreamHome”.

Demostrada la factibilidad del almacenamiento del conocimiento de PAs, mediante ontologías de los primeros dos niveles de abstracción, se especifica con la estructura del conocimiento de tres niveles de ontologías, un “Sistema para Almacenamiento y Utilización de Patrones de Análisis”. La conceptualización del sistema abarca: 1) la especificación de requerimientos, 2) la validación de dichos requerimientos mediante la técnica Delphi, la asignación de valores a los elementos de los diagramas realizados con base en el Lenguaje de Requerimientos Orientado a Metas (GRL, por sus siglas en inglés) y el análisis estadístico de los resultados obtenidos, y 3) la especificación del diseño de alto nivel, con trazabilidad hacia los requerimientos validados.

ABSTRACT

At the time of developing software, in the analysis phase, similar situations have been detected, producing similar solutions as the synthesized ones in previous developments. The repetitive analysis of similar situations, in the problem domain, generates the need of identify analysis patterns (APs), which represent the best practices for structuring and describing specific problems.

The template for APs description, proposed by Ricardo Raminhos (Raminhos et al. 2006), it was generated: from experience, by analogies, and based on design patterns, and is one of the most used. This template was refined by our research team, in a previous research, including concepts of the Software Stability Model (Fayad & Altman 2001) and building its ontology. However the analysis patterns are little used.

In this research, we consider that the limited use of APs it's because their descriptions don't include problem concepts and their relations (semantics). To increase de APs use in this thesis we proposed to store the APs as knowledge, using structured ontologies in three abstraction levels: 1) APs stable template ontology (an unique ontology) and its individuals, 2) APs ontologies (an ontology for every single AP) and the individuals for each ontology (one for each study case), and 3) case study ontologies (one for each case).

For demonstrate the feasibility of this kind of knowledge storage, it's proposed a methodology for build the ontology of an AP. Using that methodology and the ontology editor Protégé, it's built the "AP for reservation and use of reusable entities" ontology, taking in consideration two study cases: 1) Reservation and use of bibliographic material in the "CENIDET information center", and 2) Reservation and rent of real estates in the "DreamHome" enterprise.

Demonstrated the feasibility of the knowledge storage of APs, through ontologies of the first two abstraction levels, it's specified with the ontologies three levels knowledge structure, a "System for Storage and Use of Analysis Patterns". The conceptualization of the system covers: 1) the requirement specification, 2) the validation of those requirements through the Delphi technique, the values assignation to the GRL diagrams elements and the obtain results statistical analysis, and 3) the high level design specification, with traceability towards the validated requirements.

ÍNDICE

LISTA DE FIGURAS	IV
LISTA DE TABLAS.....	V
ABREVIATURAS Y CONCEPTOS	VI
CAPÍTULO 1 PROTOCOLO DE LA INVESTIGACIÓN	1
1.1 CONTEXTO DEL PROBLEMA	2
1.2 DESCRIPCIÓN DEL PROBLEMA.....	2
1.3 OBJETIVOS DE LA TESIS	3
1.3.1 <i>Objetivo general</i>	3
1.3.2 <i>Objetivos específicos</i>	3
1.4 JUSTIFICACIÓN	4
1.5 BENEFICIOS.....	4
1.6 ALCANCES Y LIMITACIONES	5
1.6.1 <i>Alcances</i>	5
1.6.2 <i>Limitaciones</i>	6
1.7 ORGANIZACIÓN DEL DOCUMENTO	6
CAPÍTULO 2 MARCO TEÓRICO	9
2.1 PATRONES DE ANÁLISIS.....	10
2.2 MODELO DE ESTABILIDAD DE SOFTWARE (SSM)	12
2.3 ONTOLOGÍAS.....	14
2.4 BASES DE DATOS BASADAS EN ONTOLOGÍAS (OBDB)	16
2.5 PROTÉGÉ	17
2.6 METHONTOLOGY	18
2.7 GRL	18
2.8 UCM	19
CAPÍTULO 3 ESTADO DEL ARTE.....	21
3.1 ANTECEDENTES.....	22
3.2 TRABAJOS RELACIONADOS.....	25
CAPÍTULO 4 ESTRUCTURA DE ALMACENAMIENTO DE PATRONES DE ANÁLISIS COMO CONOCIMIENTO. 31	31
4.1 ESTRUCTURA	32
4.2 ELEMENTOS DE LA ESTRUCTURA	33
4.3 AVANCES EN LA ESTRUCTURA Y CONCLUSIONES DEL CAPÍTULO	35
CAPÍTULO 5 METODOLOGÍA PARA LA CONSTRUCCIÓN DE ONTOLOGÍAS DE PATRONES DE ANÁLISIS.....	37
5.1 INTRODUCCIÓN.....	38
5.2 METODOLOGÍA PARA LA CONSTRUCCIÓN DE ONTOLOGÍAS DE PATRONES DE ANÁLISIS (MECOPA)	38
5.2.1 <i>Refinamiento del patrón de análisis</i>	39
5.2.2 <i>Refinamiento de los requerimientos</i>	40
5.2.3 <i>Determinación del ámbito de la ontología (Especificación)</i>	42

5.2.4	<i>Búsqueda de ontologías relacionadas (Integración)</i>	43
5.2.5	<i>Identificación de términos del dominio (Conceptualización)</i>	43
5.2.6	<i>Especificación de conceptos y relaciones (Formalización)</i>	44
5.2.7	<i>Definición de restricciones</i>	44
5.2.8	<i>Instanciación de la ontología</i>	45
CAPÍTULO 6	CONSTRUCCIÓN DE UNA ONTOLOGÍA DEL PATRÓN DE ANÁLISIS PARA RESERVACIÓN DE ENTIDADES	47
6.1	CONSTRUCCIÓN DE LA ONTOLOGÍA	48
6.1.1	<i>Refinamiento del patrón de análisis</i>	48
6.1.2	<i>Refinamiento de los requerimientos</i>	49
6.1.3	<i>Determinación del ámbito de la ontología (Especificación)</i>	61
6.1.4	<i>Búsqueda de ontologías relacionadas (Integración)</i>	63
6.1.5	<i>Identificación de términos del dominio (Conceptualización)</i>	65
6.1.6	<i>Especificación de conceptos y relaciones (Formalización)</i>	66
6.1.7	<i>Definición de restricciones</i>	71
6.1.8	<i>Instanciación de la ontología</i>	73
6.1.9	<i>Visualización de la ontología</i>	74
CAPÍTULO 7	ESPECIFICACIÓN DE REQUERIMIENTOS DEL SISTEMA AUPAS	75
7.1	ESPECIFICACIÓN DE REQUERIMIENTOS	76
7.2	DESCRIPCIÓN GENERAL	76
7.3	PERSPECTIVA DEL PRODUCTO	79
7.3.1	<i>Interfaces del sistema</i>	80
7.3.2	<i>Interfaces de usuario</i>	80
7.3.3	<i>Interfaces de Hardware</i>	80
7.3.4	<i>Interfaces de software</i>	80
7.3.5	<i>Interfaces de comunicación</i>	80
7.3.6	<i>Restricciones de memoria</i>	81
7.3.7	<i>Funcionamiento</i>	81
7.3.8	<i>Requerimientos de adaptación del sitio</i>	81
7.4	FUNCIONES DEL PRODUCTO	81
7.5	CARACTERÍSTICAS DE USUARIO.....	82
7.6	RESTRICCIONES.....	82
7.7	SUPOSICIONES Y DEPENDENCIAS.....	83
7.8	REQUERIMIENTOS FUNCIONALES (DIAGRAMAS).....	83
7.8.1	<i>Sistema de Almacenamiento y uso de Patrones de Análisis</i>	83
7.9	REQUERIMIENTOS FUNCIONALES (LISTA).....	98
7.10	REQUERIMIENTOS NO FUNCIONALES	101
7.11	RECURSOS (COMPONENTES)	101
7.12	ATRIBUTOS DEL SISTEMA DE SOFTWARE.....	102
7.12.1	<i>Confiability</i>	102
7.12.2	<i>Disponibilidad</i>	102
7.12.3	<i>Seguridad</i>	102

7.12.4	<i>Mantenibilidad</i>	102
7.12.5	<i>Portabilidad</i>	102
CAPÍTULO 8	DISEÑO DE ALTO NIVEL DEL SISTEMA AUPAS	105
8.1	DISEÑO	106
8.2	VISTAS DE DISEÑO	106
8.2.1	<i>Vista de contexto</i>	106
8.2.2	<i>Vista de interacción</i>	110
8.2.3	<i>Vista lógica</i>	111
8.2.4	<i>Vista de interfaz</i>	114
8.3	DISEÑO DE LA BASE DE DATOS.....	116
8.3.1	<i>Restricciones y/o recomendaciones de diseño</i>	116
8.3.2	<i>Estructura del repositorio de ontologías del sistema AUPAs</i>	117
CAPÍTULO 9	VALIDACIÓN DE LA INVESTIGACIÓN	119
9.1	VALIDACIÓN DE REQUERIMIENTOS	120
9.1.1	<i>Método Delphi y diseño de la encuesta</i>	121
9.1.2	<i>Participantes en el proceso Delphi</i>	124
9.1.3	<i>Recopilación de opiniones usando el método Delphi</i>	125
9.1.4	<i>Validación de fiabilidad de la encuesta</i>	126
9.1.5	<i>Validación de requerimientos mediante diagramas GRL</i>	132
CAPÍTULO 10	CONCLUSIONES	137
10.1	CONCLUSIONES	138
10.2	APORTACIONES	139
10.3	TRABAJOS FUTUROS	140
REFERENCIAS		141

LISTA DE FIGURAS

FIGURA 2.1 ARQUITECTURA DE MODELO DE ESTABILIDAD DE SOFTWARE	13
FIGURA 2.2 SIMBOLOGÍA DE GRL EN LA HERRAMIENTA JUCMNAV v5.5.0	18
FIGURA 2.3 SIMBOLOGÍA DE UCM EN LA HERRAMIENTA JUCMNAV v5.5.0.....	19
FIGURA 4.1 ESTRUCTURA DE ALMACENAMIENTO DE PATRONES DE ANÁLISIS COMO CONOCIMIENTO	32
FIGURA 5.1 CICLO DE VIDA DE MECOPA	39
FIGURA 5.2 VALORES DE PROPIEDADES PARA LA CLASE “MODELADO ESTRUCTURAL” CORRESPONDIENTE AL PUNTO 10.1 DE LA PLANTILLA PPAE	40
FIGURA 5.3 APLICANDO LA RESTRICCIÓN “SOME” EN PROTÉGÉ	45
FIGURA 5.4 INGRESANDO INDIVIDUOS EN PROTÉGÉ.....	46
FIGURA 6.1 INDIVIDUOS DE CLASES DE OPPAE EN PROTÉGÉ	49
FIGURA 6.2 DIAGRAMA GRL DEL REQUERIMIENTO 1. HACER UNA RESERVACIÓN.....	51
FIGURA 6.3 DIAGRAMA UCM DEL REQUERIMIENTO 1. HACER UNA RESERVACIÓN.....	52
FIGURA 6.4 DIAGRAMA GRL DEL REQUERIMIENTO 2. USAR UNA ENTIDAD RESERVADA.....	53
FIGURA 6.5 DIAGRAMA UCM DEL REQUERIMIENTO 2. USAR UNA ENTIDAD RESERVADA.....	54
FIGURA 6.6 DIAGRAMA GRL DEL REQUERIMIENTO 3. MODIFICAR UNA RESERVACIÓN	55
FIGURA 6.7 DIAGRAMA UCM DEL REQUERIMIENTO 3. MODIFICAR UNA RESERVACIÓN	56
FIGURA 6.8 DIAGRAMA GRL DEL REQUERIMIENTO 4. CANCELAR UNA RESERVACIÓN.....	57
FIGURA 6.9 DIAGRAMA UCM DEL REQUERIMIENTO 4. CANCELAR UNA RESERVACIÓN.....	58
FIGURA 6.10 DIAGRAMA GRL DEL REQUERIMIENTO 5. CREAR UN CONTRATO	59
FIGURA 6.11 DIAGRAMA UCM DEL REQUERIMIENTO 5. CREAR UN CONTRATO.....	60
FIGURA 6.12 REPRESENTACIÓN DE CLASES DE OPARE EN PROTÉGÉ	67
FIGURA 6.13 REPRESENTACIÓN DE RELACIONES DE OPARE EN PROTÉGÉ	69
FIGURA 6.14 REPRESENTACIÓN DE PROPIEDADES DE OPARE EN PROTÉGÉ	70
FIGURA 6.15 RESTRICCIONES DE CLASES DE OPARE EN PROTÉGÉ	72
FIGURA 6.16 INDIVIDUOS DE CLASES DE OPARE EN PROTÉGÉ.....	73
FIGURA 6.17 ONTOLOGÍA OPARE	74
FIGURA 7.1 SISTEMA AUPAS.....	78
FIGURA 7.2. GRL, UCM, Y OTRAS PIEZAS DE MODELADO (AMYOT 2003).....	79
FIGURA 7.3 DIAGRAMA GRL GENERAL DEL SISTEMA AUPAS.....	84
FIGURA 7.4 DIAGRAMA UCM GENERAL DEL SISTEMA AUPAS	86
FIGURA 7.5 DIAGRAMA GRL DE LA CREACIÓN DE UNA CUENTA DE USUARIO	87
FIGURA 7.6 DIAGRAMA UCM DE LA CREACIÓN DE UNA CUENTA DE USUARIO	88
FIGURA 7.7 DIAGRAMA GRL DEL INICIO DE SESIÓN.....	89
FIGURA 7.8 DIAGRAMA UCM DEL INICIO DE SESIÓN	89
FIGURA 7.9 DIAGRAMA GRL DE LA INSERCIÓN DE UNA NUEVA ONTOLOGÍA	90
FIGURA 7.10 DIAGRAMA UCM DE LA INSERCIÓN DE UNA NUEVA ONTOLOGÍA	91
FIGURA 7.11 DIAGRAMA GRL DE LA INSERCIÓN DE UN NUEVO INDIVIDUO	92
FIGURA 7.12 DIAGRAMA UCM DE LA INSERCIÓN DE UN NUEVO INDIVIDUO	93
FIGURA 7.13 DIAGRAMA GRL DE LA APROBACIÓN DE UNA APORTACIÓN AL CATÁLOGO DE PAS	94
FIGURA 7.14 DIAGRAMA UCM DE LA APROBACIÓN DE UNA APORTACIÓN AL CATÁLOGO DE PAS.....	95
FIGURA 7.15 DIAGRAMA GRL DE LA CONSULTA AL CATÁLOGO DE PAS	96
FIGURA 7.16 DIAGRAMA UCM DE LA CONSULTA AL CATÁLOGO DE PAS	97

FIGURA 8.1 DIAGRAMA DE CASOS DE USO GENERAL DEL SISTEMA AUPAS	109
FIGURA 8.2 DIAGRAMA DE SECUENCIA GENERAL DEL SISTEMA AUPAS.....	110
FIGURA 8.3 DIAGRAMA DE CLASES DEL SISTEMA AUPAS.....	113
FIGURA 8.4 DIAGRAMA DE COMPONENTES DEL SISTEMA AUPAS.....	115
FIGURA 8.5 ESTRUCTURA DEL REPOSITORIO DE ONTOLOGÍAS DEL SISTEMA AUPAS	117
FIGURA 9.1 PROCESO DE VALIDACIÓN DE REQUERIMIENTOS.....	121
FIGURA 9.2. EJEMPLO DE ASIGNACIÓN DE VALORES.	133
FIGURA 9.3 DIAGRAMA GRL GENERAL EVALUADO.	136

LISTA DE TABLAS

TABLA 2.1 CARACTERÍSTICAS DE LOS NIVELES DEL SSM.....	14
TABLA 3.1 COMPARATIVA DE ANTECEDENTES	24
TABLA 3.2 COMPARATIVA DE TRABAJOS RELACIONADOS	28
TABLA 5.1 EJEMPLO DE ACTORES, OBJETOS Y RESPONSABILIDADES	42
TABLA 6.1 ACTORES OBJETOS Y RESPONSABILIDADES DEL REQUERIMIENTO 1. HACER UNA RESERVACIÓN	51
TABLA 6.2 ACTORES OBJETOS Y RESPONSABILIDADES DEL REQUERIMIENTO 2. USAR UNA ENTIDAD RESERVADA	54
TABLA 6.3 ACTORES OBJETOS Y RESPONSABILIDADES DEL REQUERIMIENTO 3. MODIFICAR UNA RESERVACIÓN	56
TABLA 6.4 ACTORES OBJETOS Y RESPONSABILIDADES DEL REQUERIMIENTO 4. CANCELAR UNA RESERVACIÓN	58
TABLA 6.5 ACTORES OBJETOS Y RESPONSABILIDADES DEL REQUERIMIENTO 5. CREAR UN CONTRATO.....	60
TABLA 6.6 TABLA DE DEPENDENCIAS Y PRIORIDADES	61
TABLA 6.7 COMPARATIVA DE ONTOLOGÍAS.....	65
TABLA 6.8 RELACIONES ENTRE CLASES DE LA ONTOLOGÍA OPARE	68
TABLA 8.1 EJEMPLO DE LA TABLA “ONTOLOGÍAS PA” DE LA BD.....	116
TABLA 9.1. FIABILIDAD Y MÉTODOS UTILIZADOS.	127
TABLA 9.2. VALORES OBTENIDOS PARA CONSISTENCIA.....	128
TABLA 9.3. VALORES OBTENIDOS PARA ESTABILIDAD.	129
TABLA 9.4. VALORES OBTENIDOS PARA CONCORDANCIA DE LOS INTER-OBSERVADORES.	131
TABLA 9.5. RANGO DE COLORES OTORGADOS POR JUCMNAV.....	134
TABLA 9.6. REQUERIMIENTOS SUPERIORES A LA MEDIA.....	135

ABREVIATURAS Y CONCEPTOS

Aportación	Se refiere a las ontologías y/o individuos de ontología que son ingresados al sistema AUPAs por un usuario tipo autor.
AUPAs	Almacenamiento y Uso de Patrones de Análisis
BD	Base de Datos
BOs	Business objects (Objetos de negocio)
CC	Cuestionario de competencias
CENIDET	Centro Nacional de Investigación y Desarrollo Tecnológico
CyC	Del inglés en "cyc"lopedia es un proyecto de inteligencia artificial que intenta ensamblar una ontología comprensiva y una base de datos de conocimiento.
DAML	DARPA Agent Markup Language (Lenguaje de marcado para los agentes DARPA).
DARPA	Defense Advanced Research Projects Agency (Agencia de proyectos de investigación avanzados de defensa).
DMOZ	Nombrado así por su nombre de dominio original: directory.mozilla.org , es el directorio editado por personas más extenso y más completo de Internet
EBTs	Enduring Business themes (Temas de negocio perdurables)
eODL	Es el lenguaje de descripción de componentes de ITU. Es un lenguaje que permite la utilización de conceptos para una descripción de modelo independiente de la tecnología de los componentes durante su vida útil, desde diversas perspectivas.
FR	Por sus siglas en inglés Functional Requirements (Requerimientos Funcionales)
GRL	Goal-oriented Requirements Language (Lenguaje de requerimientos orientado a metas).

IEEE	Institute of Electrical and Electronics Engineers (Instituto de Ingenieros Eléctricos y Electrónicos).
Individuo	Es la instancia de ontología que se obtiene al poblar con datos cada una de las clases de dicha ontología.
IOs	Industrial objects (Objetos industriales)
IS	Ingeniería de Software
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector (Sector de estandarización de telecomunicaciones de la unión de telecomunicaciones internacional).
MECOPA	Metodología para la Construcción de Ontologías de Patrones de Análisis
MSC	Un diagrama de secuencia de mensajes o message sequence chart, por sus siglas en inglés, es un diagrama de interacción de la familia SDL estandarizada por la ITU-T.
NFR	Por sus siglas en inglés Non Functional Requirements (Requerimientos No Funcionales)
OBDB	Ontology Based Data Base (Base de datos basada en ontologías)
Ontolingua server	Servidor que permite acceder a una biblioteca de ontologías previamente definidas y cuyo lenguaje, también llamado Ontolingua.
OPARE	Ontología del Patrón de Análisis para Reservación de Entidades
OWL	Ontology Web Language (Lenguaje web de ontologías)
OWL-DL	OWL Description Logic (Es un tipo de OWL que incorpora funciones de lógica descriptiva).
PA	Patrón de Análisis
PPAE	Plantilla de Patrones de Análisis Estable
RosettaNet	Es un consorcio no lucrativo enfocado al establecimiento de la estandarización de procesos de negocio.

SDL	Es un lenguaje de especificación formal y visual normado por la ITU-T en el estándar Z.100, así como el acrónimo de Specification and Description Language (Lenguaje de Especificación y Descripción).
SDS	Software Design Specification (Especificación de Diseño de Software)
SMBD	Sistema Manejador de Base de Datos
SRS	Software Requeriments Specification (Especificación de Requerimientos de Software).
SSM	Software Stability Model (Modelo de estabilidad de software)
TTCN	Por sus siglas en inglés Tree and Tabular Combined Notation (Notación combinada de árbol y tabular) es un lenguaje de programación utilizado para pruebas de protocolos de comunicación y servicios web.
UCM	Use Case Maps (Mapas de casos de uso)
UML	Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language).
UNSPSC	United Nations Standard Products and Services Code (Código de productos y servicios estándar de las naciones unidas).
URN	User Requirements Notation (Notación de requerimientos de usuario)
XMI	Siglas en inglés de XML Metadata Interchange (XML de Intercambio de Metadatos) es una especificación para el Intercambio de Diagramas.
XML	Siglas en inglés de eXtensible Markup Language (“lenguaje de marcado extensible”)

Capítulo 1

*P*rotocolo de la *I*nvestigación

En el presente capítulo se expone el problema que dio origen al proyecto de investigación propuesto en esta tesis; se definen los objetivos, beneficios alcances y limitaciones de la investigación; y finalmente, en el último apartado se describe la forma en la que está estructurado este documento.

1.1 Contexto del problema

La fase de análisis en el desarrollo de software es una de las más importantes, pero a pesar de que ya existen metodologías y herramientas que dan soporte y buscan reducir tiempos de análisis, aún se presentan problemas. Algunos de estos problemas son: modelados incorrectos, requerimientos incompletos, aumento de tiempo de desarrollo por errores de análisis, entre otros.

Para reducir dichos problemas, en la fase de análisis se han identificado situaciones semejantes cada vez que se desarrolla una aplicación y se producen soluciones parecidas a las detectadas en proyectos previos. El análisis repetido de situaciones semejantes originó la identificación de lo que llamamos patrones de análisis (PAs), los cuales representan las mejores prácticas para problemas específicos.

Los patrones de análisis identificados, se han especificado mediante plantillas, que incluyen descripciones textuales, usando lenguaje natural, y visuales, mediante diagramas no estandarizados, que dificultan su manejo. Esta es una de las razones por las que no han sido incluidos en la fase del modelado del negocio. Sin embargo, en (Raminhos et al. 2006) se propuso una plantilla de patrones de análisis con el propósito de unificar los enfoques existentes hasta ese momento. A la fecha parece que ha dado resultado ya que está emergiendo como una de las plantillas más utilizadas. Sin embargo, esta plantilla no incluye la semántica del problema en la descripción de los patrones y esto lo consideramos una deficiencia.

1.2 Descripción del problema

Debido a que los trabajos realizados en materia de patrones de análisis no contemplan la descripción de los conceptos y sus relaciones (semántica), dichos patrones se utilizan poco en el ámbito del desarrollo de software (Hamza, 2005).

1.3 Objetivos de la tesis

1.3.1 *Objetivo general*

Facilitar la utilización de patrones de análisis como apoyo en la elaboración de software, mediante la incorporación de semántica a sus descripciones almacenadas en repositorios, empleando tres niveles de abstracción de ontologías para: la plantilla estable de patrones de análisis, los patrones de análisis mismos, y las instancias de patrones.

1.3.2 *Objetivos específicos*

Considerando el problema descrito, se ha propuesto la construcción de ontologías de patrones de análisis como medio para proveer la semántica a la descripción de patrones y así evitar problemas de ambigüedad e incrementar la viabilidad de utilización. Esta investigación tiene como uno de sus fines la especificación y diseño de un sistema para almacenamiento y utilización de patrones de análisis, aplicando la estructura de tres niveles de abstracción de ontologías: ontología de plantilla, ontología de patrón y ontología de instancia de patrón. El sistema permitirá exportar representaciones visuales de patrones de análisis, para utilizarlas en otras herramientas que soporten el desarrollo de software, en la etapa de análisis de sistemas y modelado del negocio.

Los objetivos específicos son:

- Construir un individuo de la ontología de la plantilla de patrones estables, propuesta por (Martínez Jiménez 2013).
- Generar una metodología para la construcción de ontologías de patrones de análisis, partiendo del individuo de la ontología de la plantilla de patrones estables.
- Construir una ontología para el patrón de análisis para reservación y uso de entidades reutilizables, así como su representación computacional.
- Construir un individuo de la ontología para el patrón de análisis para reservación y uso de entidades reutilizables.

- Proponer la estructura de un repositorio, donde se podrán almacenar, tanto los patrones de análisis como sus ontologías, en sus tres niveles de abstracción: plantilla, patrón e instancia.
- Sentar las bases, mediante la especificación de requerimientos, y diseño de alto nivel, de una herramienta de software para almacenaje y utilización de patrones de análisis. En la cual, un desarrollador de software, podrá acceder a las descripciones de patrones de análisis, y utilizar representaciones computacionales totales o parciales, como apoyo a su proceso de desarrollo.

1.4 Justificación

Al analizar los antecedentes y trabajos relacionados (descritos en el Capítulo 3), se observó que en la línea principal de producción de las empresas de software, se usan poco los patrones de análisis en el ámbito del desarrollo de aplicaciones. El escaso uso de los PAs tiene diversas causas, como son: falta de semántica en la descripción de los patrones, falta de estandarización, falta de resultados que garanticen que se obtendrá una ventaja comparativa al usarlos con respecto a no hacerlo.

Para utilizar patrones de análisis es conveniente la incorporación de repositorios que se actualicen continuamente (Bobkawska & Grabowski 2009; da Matta Vegi et al. 2011). Por esa razón esta investigación establecerá una infraestructura donde se puedan almacenar patrones e incorporar, mediante el uso de ontologías, aspectos de semántica, para utilizarlos de una manera práctica.

1.5 Beneficios

Los beneficios que se obtienen al realizar esta investigación son:

- Demostrar la factibilidad de describir el PA para reservación y uso de entidades reutilizables, mediante un individuo de la plantilla de patrones de análisis estables.

- Proveer al investigador del análisis del desarrollo de software, y potencial descriptor de PAs, una metodología para la construcción de ontologías de patrones de análisis.
- Demostrar la factibilidad de construcción de una ontología para el patrón de análisis para reservación y uso de entidades reutilizables, así como su representación computacional.
- Proponer un repositorio, donde se podrán almacenar, tanto los patrones de análisis como sus ontologías, en sus tres niveles de abstracción: plantilla, patrón e instancia.
- Sentar las bases, mediante la especificación de requerimientos, y diseño de alto nivel, de una herramienta para almacenaje y utilización de descripciones de patrones de análisis. En la cual, el desarrollador de software, podrá acceder a las descripciones de patrones de análisis, y utilizarlos como apoyo en su proceso de desarrollo.

1.6 Alcances y limitaciones

1.6.1 Alcances

- Se obtiene una metodología para construcción de ontologías de patrones de análisis.
- Se lleva a cabo la construcción de la ontología de un patrón de análisis y su representación computacional, pues la ontología de la plantilla estable ya existe y la ontología de la instancia esta fuera del alcance de esta investigación, es un trabajo a realizar en una investigación futura.
- Se evalúa el uso de una Base de Datos Basada en Ontologías o una herramienta de soporte a la construcción de ontologías (Ej. Protégé), para almacenar los patrones de análisis obtenidos.
- Se especifican los requerimientos y el diseño de alto nivel para un sistema de almacenaje y utilización de patrones de análisis, que pueda hacer uso de un repositorio de patrones de análisis.

1.6.2 Limitaciones

- Sólo se almacena una ontología en el repositorio para el nivel de abstracción de plantilla, pues sólo debe existir una plantilla para describir todos los PAs.
- Sólo se almacena una ontología en el repositorio para el nivel de abstracción de patrón de análisis, porque sólo se construirá la ontología de un patrón de análisis, que ejemplifique el uso de la metodología a proponer.
- Se utiliza el patrón de análisis para reservación y uso de entidades reutilizables expuesto en (Fernandez & Yuan 1999) para la ontología de patrón.
- Sólo se hace el prototipo de la interfaz de la herramienta para la utilización de los patrones de análisis (sus entradas y salidas) y se establecen sus requerimientos y diseño de alto nivel, para que se desarrolle en una investigación futura.

1.7 Organización del documento

En esta tesis se presenta el estudio del almacenamiento y uso de patrones de análisis apegados a su semántica. El trabajo está organizado en 8 secciones las cuales se listan a continuación.

- Capítulo 2. Marco teórico. En este capítulo se presentan los conceptos fundamentales que determinan el ámbito contextual en el cual se posiciona la investigación.
- Capítulo 3. Estado del arte. En este capítulo se presentan los resultados del proceso de búsqueda de información relevante con respecto a trabajos relacionados que sirven como punto de referencia y comparación con el trabajo propuesto.
- Capítulo 4 Estructura de Almacenamiento de Patrones de Análisis como Conocimiento. En este capítulo se propone una estructura de almacenamiento de los PAs como conocimiento, utilizando ontologías estructuradas en tres niveles de abstracción: 1) ontología de plantilla estable de PAs, 2) ontologías de PAs y 3) ontologías de casos de estudio de PAs.

- Capítulo 5. Metodología para la construcción de ontologías de patrones de análisis. En este capítulo se hace una adaptación de Methontology para la creación de ontologías de patrones de análisis.
- Capítulo 6. Construcción de una ontología del patrón de análisis para reservación de entidades. Capítulo en el cual, se presenta la documentación y los pasos que se llevaron a cabo para la construcción de la ontología del patrón de análisis para reservación y uso de entidades reutilizables.
- Capítulo 7. Especificación de Requerimientos del Sistema AUPAs. En este capítulo se presenta la documentación de los requerimientos del sistema de Almacenaje y Utilización de Patrones de Análisis.
- Capítulo 8. Diseño de Alto Nivel del Sistema AUPAs. En el presente capítulo se presenta la documentación del diseño de alto nivel del sistema de Almacenaje y Utilización de Patrones de Análisis.
- Capítulo 9. Validación de la investigación. En este capítulo se presenta una descripción general del proceso empleado para la validación de los requerimientos correspondientes al Sistema de almacenaje y utilización de patrones de análisis (AUPAs). Se explican los pasos que se siguieron para la validación y las conclusiones que se obtuvieron al realizarla.
- Capítulo 10. Conclusiones. Este capítulo proporciona los resultados obtenidos, las aportaciones realizadas y sugerencias para trabajos futuros en el ámbito de estudio.

Capítulo 2 *Marco Teórico*

El presente capítulo presenta los conceptos fundamentales que determinan el ámbito contextual en el cual se posiciona la investigación. Se detallan los conceptos de: patrones de análisis, modelo de estabilidad de software, ontologías; se presenta un editor de ontologías (protégé), además de una metodología para la construcción de ontologías (METHONTOLOGY), entre otros. Así como los lenguajes, GRL y UCM, para la especificación de los requerimientos utilizados en esta tesis.

2.1 Patrones de análisis

Existen muchas definiciones de patrones de análisis, una de éstas es: “son aquellos patrones cuyo propósito es construir un modelo de análisis, el cual se enfocará en los procesos de negocio en lugar de la implementación del software”. (Pantoquillo et al. 2003). Sin embargo, no nos podemos quedar con esta definición. Para entender a los patrones de análisis es necesario remontarnos un poco a los orígenes e ir viendo cómo se fue acuñando este término con el paso del tiempo.

Christopher Alexander en su libro publicado en 1977 introduce por primera vez el concepto de patrón, en el contexto de la arquitectura y lo define como: “cada patrón describe un problema que ocurre una y otra vez en nuestro ambiente, y luego describe el núcleo de la solución a ese problema, de tal manera que se pueda usar esa solución un millón de veces, sin hacerla dos veces de la misma manera” (Alexander et al. 1977). Posteriormente en 1994 Erick Gamma, escribe un libro acerca de patrones de diseño, basándose en el concepto de Alexander, pero aplicado esta vez al software, cambiando puertas y paredes, por términos de objetos e interfaces, pero manteniendo la esencia, de que un patrón, “es una solución a un problema recurrente, en un contexto y sistema de fuerzas en particular” (Gamma et al. 1994).

Es hasta 1997 cuando Martin Fowler introduce el concepto de patrón de análisis. Según Fowler: “un patrón es una idea que ha sido utilizada en un contexto práctico y que probablemente será utilizada en otros” (Fowler 1997). Además Fowler, en su libro (Fowler 1997), propone los primeros patrones de análisis (PAs). Estos son considerados avanzados, complicados y pobremente descritos (Bobkawska & Grabowski 2009).

Un año más tarde, Eduardo Fernández define al patrón de análisis como “el conjunto de clases y asociaciones que tienen alguna razón de ser en el contexto de una aplicación; esto es, un modelo conceptual de una parte de la aplicación” (Fernandez 1998) y propone tres patrones de análisis más, utilizando una forma más clara para describirlos. Ese mismo año, otros autores presentan más patrones de análisis, tal es el caso de Rosana Braga (Braga et al. 1998), describiéndolos de manera distinta a Fowler y a Fernández, y así con el paso de los años más autores fueron generando sus propios patrones de análisis, cada uno con un formato distinto de presentarlos.

Como un intento para unificar las plantillas de patrones de análisis, Ricardo Raminhos propuso una plantilla que combina las características comunes de las plantillas ya existentes y añade nuevas características que no habían sido consideradas (Raminhos et al. 2006).

La plantilla de Raminhos consta de 18 apartados, descritos a continuación:

1. Nombre: Identificador del patrón.
2. También conocido como: Otros nombres que hacen referencia a este patrón.
3. Historial: Registro cronológico de las versiones anteriores.
4. Ajustes estructurales: Definición de extensiones u omisiones de campos de la plantilla.
5. Problema: Breve descripción del problema que se resuelve a través del patrón.
6. Motivación: Descripción de las fuerzas involucradas y una situación representativa que motive el uso del patrón.
7. Contexto: Descripción detallada del contexto en el que el problema y la solución se repita y para el cual la solución es deseable.
8. Aplicabilidad: Descripción de las condiciones bajo las cuales el patrón puede ser aplicado.
9. Requerimientos: Listado de requerimientos funcionales, no funcionales, dependencias y contribuciones, identificación de conflictos y guía de soluciones, prioridades y participantes.
10. Modelado: Uso de modelos para representar procesos expresados en un nivel alto de abstracción.
11. Contexto resultante: Configuración del sistema después de aplicar el patrón.
12. Consecuencias: Ventajas y desventajas del uso del patrón.
13. Trampas anti-patrones: Las trampas más comunes que pueden originarse al aplicar el patrón.
14. Ejemplos: Uno o más ejemplos de aplicaciones que ilustren el uso del patrón.
15. Patrones relacionados: Lista de patrones similares que describan problemas o soluciones similares.
16. Patrones de diseño: Patrones de diseño o de arquitectura que pueden utilizarse para un mayor refinamiento.
17. Directrices de diseño: Consejos sobre la aplicabilidad del patrón, sin llegar a detalles específicos.
18. Usos conocidos: Ocurrencias conocidas de la aplicación del patrón en sistemas existentes. Se deben incluir al menos tres sistemas diferentes.

Los patrones de análisis se clasifican en cuatro tipos de acuerdo a su proceso de construcción o interpretación (Hamza & Fayad 2002):

- Los patrones construidos de acuerdo a la experiencia de su autor
- Los patrones construidos a partir de analogías.
- Los patrones construidos tomando como referencia a los patrones de diseño y su estructura.
- Los patrones construidos usando el enfoque de estabilidad de software.

La plantilla de Raminhos abarca los tres primeros tipos de patrones de análisis, sin embargo no contempla aquellos patrones construidos tomando como referencia el modelo de estabilidad de software (el cual se explica a continuación).

2.2 Modelo de Estabilidad de Software (SSM)

Introducido en (Fayad & Altman 2001), presenta tres niveles conceptuales clasificados de acuerdo a su naturaleza (Figura 2.1):

- Temas de negocios perdurables (EBTs): son las clases invariantes que presentan los conceptos básicos y perdurables del sistema. Por lo tanto, son extremadamente estables y forman el núcleo del SSM. Describen los principales objetivos del sistema.
- Objetos de negocios (BOs): son las clases que mapean las EBTs del sistema en objetos concretos. Los BOs son tangibles y externamente estables, pero son internamente adaptables.
- Objetos industriales (IOs): son las clases que mapean las BOs del sistema en objetos físicos.

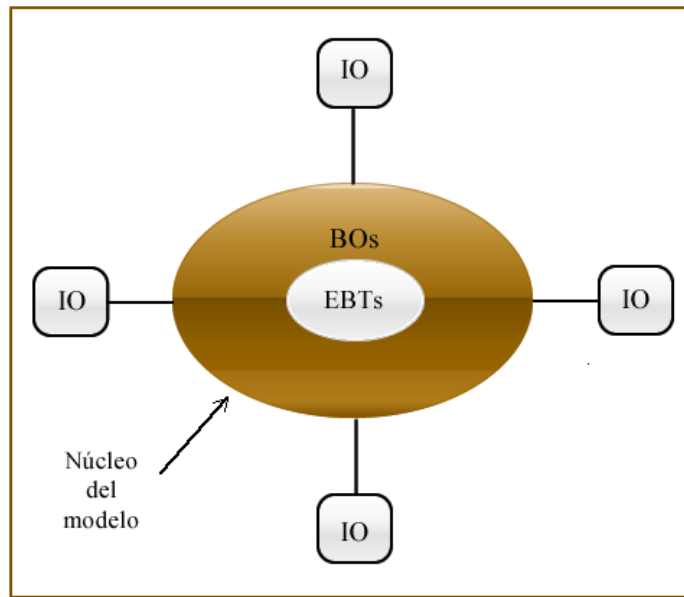


Figura 2.1 Arquitectura de Modelo de Estabilidad de Software

En la Tabla 2.1 se muestran las características (columnas) de los componentes de cada nivel (filas), del SSM. A continuación se describen los símbolos, usados en las columnas:

- Comportamiento (2da. columna). Define la manera de portarse del componente en el modelo:
 - Estable: El componente no cambia de conducta externa e internamente, todo el tiempo.
 - Estable-adaptable: El componente no cambia de conducta externamente y es adaptable (desempeña funciones distintas) internamente.
 - Inestable: El componente puede reemplazarse, agregarse o removerse del sistema, sin afectar el núcleo del sistema.

- Percepción (3ra. columna). Define la naturaleza del componente a través de los sentidos.
 - Intangible: El componente representan conceptos (no de sustancia física), del dominio.
 - Semi-tangible: El componente es tangible (se puede percibir por un sentido). Sin embargo aún no es un objeto concreto (específico).
 - Tangible: El componente es completamente un objeto concreto. Usualmente representa objetos reales del mundo.

- Especificidad (4ta. columna). Establece el grado de similitud del componente con el dominio.
 - General: El componente es común a todas las aplicaciones del dominio donde está identificado.
 - Particular: El componente puede ser completamente diferente de una aplicación a otra, incluso en el mismo dominio.

Tabla 2.1 Características de los niveles del SSM

Nivel de Estabilidad	Comportamiento	Percepción	Especificidad
EBTs	Estable	Intangible	General
BOs	Estable-adaptable	Semi-tangible	General
IOs	Inestable	Tangible	Particular

Tomando en cuenta este modelo surge la plantilla de patrones de análisis estable (Martínez Jiménez 2013). Plantilla de patrones de análisis, considerada como una modificación a la plantilla de (Raminhos et al. 2006), pero agregando el modelo de estabilidad expuesto en (Fayad & Altman 2001). Dicha modificación consiste básicamente en añadir aspectos dinámicos y estructurales. Para su implementación se optó por utilizar ontologías que permitieran clasificar los elementos de la plantilla en cada uno de los niveles del SSM y definir sus características, además de aprovechar todas las prestaciones que una ontología ofrece: formalización, consenso, compartición, semántica, etc. En la siguiente sección se explica más a detalle el concepto de ontología y sus características.

2.3 Ontologías

Según Studer (Studer et al. 1998), las dos definiciones que expresan de una 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, son explicados por el autor de la siguiente forma:

a) Compartida: refleja la noción de que una ontología captura conocimiento consensual, que no es objeto de un sólo individuo, sino aceptado por un grupo.

a) Conceptualización: se refiere a un modelo abstracto de algún fenómeno en el mundo proveniente de haber identificado los conceptos relevantes de dicho fenómeno.

b) Explícita: significa que el tipo de conceptos utilizados y las restricciones de su uso son explícitamente definidas.

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 de programación de ontologías. Uno de los más utilizados es el OWL, que se define a continuación.

Lenguaje de Ontologías Web (OWL): El Lenguaje de Ontologías Web (OWL) (L. McGuinness & van Harmelen 2004) está diseñado para ser usado en aplicaciones que necesitan procesar el contenido de la información en lugar de únicamente representar información para los humanos. OWL facilita un mejor mecanismo de interoperabilidad de contenido Web que los mecanismos admitidos por XML, RDF, y esquema RDF (RDF-S) proporcionando vocabulario adicional junto con una semántica formal. OWL tiene tres sub-lenguajes, con un nivel de expresividad creciente: OWL Lite, OWL DL, y OWL Full.

OWL proporciona más vocabulario que RDF para describir propiedades y clases tales como: relaciones entre clases (por ejemplo “clases disjuntas”), cardinalidad (por ejemplo “exactamente uno”) e igualdad. OWL también proporciona más tipos para las propiedades características (por ejemplo “simetría”), y clases enumeradas.

- OWL Lite está diseñado para aquellos usuarios que necesitan principalmente una clasificación jerárquica y restricciones simples.
- OWL DL está diseñado para aquellos usuarios que requieren la máxima expresividad conservando completitud computacional (se garantiza que todas las conclusiones sean computables), y decidibilidad (todos los cálculos acaban en un tiempo finito). OWL DL es denominado de esta forma debido a su correspondencia con la lógica de descripción (Description Logics, en inglés), un campo de investigación que estudia la lógica que compone la base formal de OWL.
- OWL Full está diseñado a usuarios que quieren máxima expresividad y libertad sintáctica de RDF sin garantías computacionales. Por ejemplo, en OWL Full una clase puede ser considerada simultáneamente como una

colección de clases individuales y como una clase individual propiamente dicha.

Para la implementación computacional de las ontologías se pueden utilizar dos técnicas de almacenamiento:

- Utilizar modelos de base de datos
- Utilizar editores de ontologías como Protégé

En las dos secciones siguientes se explican las prestaciones que ofrece cada técnica.

2.4 Bases de Datos Basadas en Ontologías (OBDB)

Una OBDB es un enfoque de almacenamiento, para guardar, en esquemas de base de datos, tanto ontologías como datos basados en ontologías, con el objetivo de obtener los beneficios y funcionalidades que ofrecen los Sistemas Manejadores de Base de Datos (SMBDs).

Las OBDBs se componen explícitamente de cinco elementos (Jean et al. 2007):

- Ontologías
- Esquema de datos
- Datos
- Las conexiones entre los datos y su esquema, y
- La conexión entre el esquema y la ontología

Al utilizar estos esquemas de base de datos para el almacenamiento de las ontologías, se obtienen los siguientes beneficios: rendimiento aceptable en consultas, manejo de grandes cantidades de datos, administración de transacciones, etc. (Dehainsala et al. 2007).

2.5 Protégé

Es un editor y/o creador de ontologías de código abierto, es un framework de base de conocimiento y, tal vez, es la herramienta de creación de ontologías más utilizado en el mercado. Protégé fue desarrollado por el Centro de Investigación de Informática Biomédica de la Escuela de Medicina de la Universidad de Stanford. Actualmente continua su desarrollo en colaboración con la Universidad de Manchester (Stanford Center For Biomedical Informatics Research 2015). Todos los archivos de software y su documentación están disponibles en el Sitio Web de Protégé.

Las ontologías de Protégé pueden ser exportadas en una gran variedad de formatos incluyendo RDF(S), OWL y XML Schema. Es un framework para el que otros proyectos sugieren utilizar plugins para su aplicación, está basado en Java, es extensible y usa Swing para crear su interfaz.

Usando Protégé, las ontologías pueden ser editadas y creadas usando lenguaje RDF/OWL (incluyendo OWL Full, DL y Light) a través de su entorno "plug-and-play" basado en Java. Este entorno proporciona una vista con pestañas de una ontología, permitiendo al usuario separar clases de subclases, clases de propiedades, clases de individuos, etc. y ver todas las características y las relaciones atribuidas a cada objeto.

A pesar de su facilidad de uso en comparación con muchos otros editores de ontologías comerciales y de código abierto, Protégé requiere un conocimiento fundamental de las ontologías y los tipos de objetos y relaciones definidas dentro de ellos. Sin este conocimiento un usuario principiante se pierde por completo dentro de la herramienta, sin embargo, hay numerosos sitios que ofrecen ontologías prediseñadas, plantillas y tutoriales guiados que permiten a los nuevos usuarios trabajar a través de los conceptos básicos y lograr una comprensión sólida en un plazo relativamente breve de tiempo.

La plataforma de Protégé admite dos formas principales de modelado de ontologías a través de los editores de Protégé-Frames, Protégé-OWL y Web Protégé.

Cabe mencionar que Protégé sólo cubre la fase de implementación de las ontologías, una etapa de todas las que componen el proceso de construcción de ontologías, aunque no existe una metodología estándar, la mayoría se basa en la metodología METHONTOLOGY, la cual se expone en la sección 2.6.

2.6 METHONTOLOGY

Es una metodología para construir ontologías, desde cero, que especifica qué y cómo llevar a cabo las actividades identificadas en el proceso de desarrollo de la ontología, qué clase de técnicas son las más apropiadas en cada actividad y qué produce cada una de ellas. Las actividades propuestas son: especificación, adquisición del conocimiento, conceptualización, integración, implementación, evaluación y documentación. Además, METHONTOLOGY, recomienda la reutilización de ontologías existentes. (Fernández et al. 1997).

En esta investigación consideramos que se puede hacer más eficiente el proceso de construcción de ontologías si la información de la que se va obtener la ontología se encuentra representada mediante diagramas o descripciones gráficas. Es por esta razón que se optó por buscar algún lenguaje gráfico para representar patrones de análisis, en específico, el apartado de requerimientos, debido a lo anterior se decidió utilizar los lenguajes GRL y UCM descritos a continuación.

2.7 GRL

GRL está dirigido a requerimientos no funcionales, y es un lenguaje que soporta modelado orientado a metas y razonamiento sobre requerimientos y atributos de calidad (ITU-T 2012). Además GRL provee representaciones visuales (ver Figura 2.2) para expresar varios tipos de conceptos que aparecen durante el proceso de requerimientos.

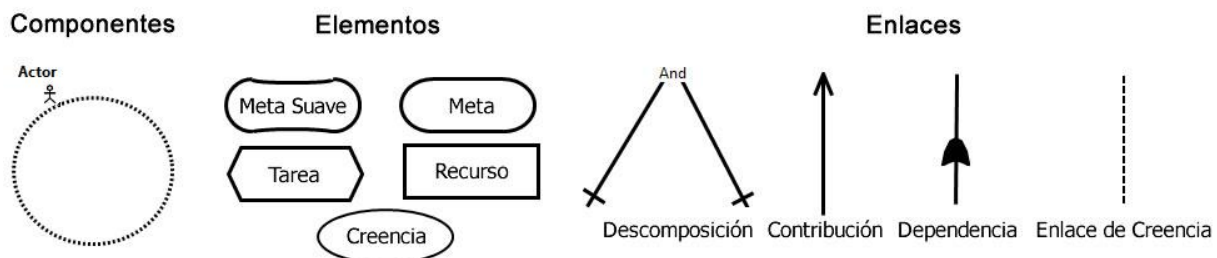


Figura 2.2 Simbología de GRL en la herramienta jUCMNav v5.5.0

2.8 UCM

En (ITU-T 2012) se menciona que UCM es un lenguaje de especificación que debido a su naturaleza visual, simple e intuitiva permite ser utilizado tanto por modeladores, así como los no especialistas, pero al mismo tiempo ofrece suficiente rigurosidad para los desarrolladores.

UCM describe visualmente relaciones causales entre responsabilidades sobrepuestas en estructuras organizacionales de componentes abstractos. Las responsabilidades representan procesos genéricos (acciones, actividades, operaciones, tareas, etc.) así como entidades que no son software (actores o hardware). Se dice que son causales debido a que las causas están vinculadas (precondiciones o eventos disparadores) al efecto (post condiciones y eventos resultantes) y que estas responsabilidades están ordenadas en secuencia, como alternativas o concurrentemente (Amyot & Mussbacher 2001). Las representaciones visuales que usa UCM se pueden ver en la Figura 2.3.

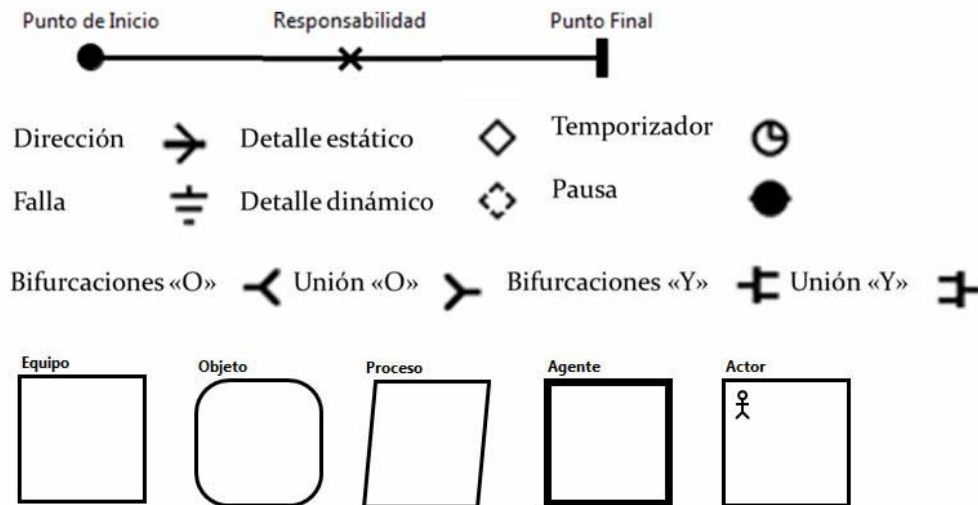


Figura 2.3 Simbología de UCM en la herramienta jUCMNav v5.5.0

Dentro de este capítulo se incluyeron conceptos que permiten sustentar la investigación reportada en la tesis. Se incluye información acerca de los patrones de análisis debido a que la investigación de esta tesis está enfocada principalmente al

estudio, almacenamiento y uso de los mismos. También se presentó información de las ontologías, las cuales nos dan la pauta para representar a los patrones de análisis como conocimiento.

Otro concepto importante incluido es el de estabilidad de software, el cual permite que los patrones de análisis sean más fáciles de reutilizar, además de incluir dos lenguajes de especificación de requerimientos los cuales se utilizan en el Capítulo 7.

En el capítulo siguiente se presentan algunos trabajos que utilizan patrones de análisis u ontologías, que sirven de comparativa con respecto a esta investigación.

Capítulo 3 Estado del Arte

En este capítulo se presentan los resultados del proceso de búsqueda de información relevante con respecto a antecedentes (investigaciones realizadas en CENIDET) y trabajos relacionados (realizados fuera de CENIDET) que sirven como punto de referencia y comparación con el trabajo propuesto

3.1 Antecedentes

De los trabajos que se han realizado en el departamento de Ciencias Computacionales en el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), en el área de patrones de análisis y de ontologías se pueden mencionar los siguientes:

Fernández Bonilla, Osvaldo Daniel, tesis de maestría: “Construcción de una Ontología Mediante Métodos Semiautomáticos”, CENIDET, Marzo 2010. En dicho trabajo se menciona que para la construcción de una ontología se necesitan tres elementos: conocimiento sobre el dominio de la ontología, una metodología para la construcción de ontologías y una herramienta de soporte para la construcción de las mismas como Protégé. Además proporciona una metodología que apoya al diseñador de ontologías en las tareas de lectura, organización de textos y búsquedas de relaciones semánticas a partir de un corpus de documentos de texto en el dominio del procesamiento de lenguaje natural. La tesis de Fernández además de aportar una metodología, realiza una aplicación de la misma construyendo una ontología, similitudes que tiene con nuestra investigación, sin embargo esta investigación se enfoca al dominio de los PAs y no sólo llegó hasta la construcción de una ontología sino a la especificación y diseño de alto nivel de un sistema de almacenamiento y uso de patrones de análisis, en el cual se almacenan las ontologías de los patrones de análisis así como sus individuos para fomentar su utilización.

Galindo García, Juan Antonio, tesis de maestría: “Generación de ontologías en OWL a partir de diagramas de clases de UML mediante MDA (Arquitectura Dirigida por Modelos)”, CENIDET, Enero 2011. Hace uso de la MDA y realiza transformaciones de metamodelos mediante el lenguaje de transformación ATL, aunque en este trabajo de investigación no se llega a definir reglas de transformación en un lenguaje de transformación, se construyen métodos que posteriormente podrían evolucionar a reglas de transformación. El trabajo de Galindo fue de interés ya que los patrones de análisis incluyen diagramas de clases en su descripción, y como el objetivo de nuestra investigación es la de generar una ontología de PA, fue interesante observar el cómo a partir de diagramas se puede facilitar la construcción de una ontología, cada clase del diagrama de clases se convierte en una clase de la ontología y cada atributo de clase se convierte en una propiedad (técnica que fue de gran ayuda para identificar algunas de las clases de la ontología de PA, realizada en nuestra investigación), aunque en la tesis de Galindo no se llega a generar la ontología como tal, (sólo se definen clases y propiedades, pero no relaciones semánticas entre clases), sienta las bases para definir ciertas reglas de transformación.

Valencia Vargas, Norma Elizabeth, tesis de maestría: “Creación de una Ontología para el Dominio de e-Gobierno en México”, CENIDET, Febrero 2011. Hace una adaptación de la metodología METHONTOLOGY para la construcción de ontologías adecuándola a sus necesidades y crea una ontología genérica del gobierno de México en sus tres niveles: Federal, Estatal y Municipal. Dicha tesis está muy relacionada con nuestra investigación, la única diferencia radica en el dominio, ya que para el presente trabajo se propone hacer una adaptación de METHONTOLOGY para la creación de ontologías de patrones de análisis además de construir una ontología para un patrón de análisis.

Martínez Jiménez, Rubí Celia, resultados de tesis de maestría: “Plantilla estable de patrones de análisis”, CENIDET, Noviembre 2013. El objetivo que se plantea es generar una plantilla de patrones de análisis que contenga elementos de estabilidad de software y la representación del conocimiento mediante ontologías. Cuyo producto resultado es una ontología para una plantilla de patrones de análisis, la cual sirve como base para la construcción de la ontología planteada en esta tesis, ya que a partir del individuo de la ontología de plantilla, se construye la ontología de ese individuo, es decir la ontología de PA.

De todos los trabajos arriba mencionados se obtuvieron ideas para el armado de esta investigación, ya que algunos de los problemas de los patrones de análisis son la escasez de semántica en su descripción, así como la falta de una representación formal de los mismos y varias tesis hablan de la importancia de utilizar ontologías para colaborar en la solución de esos problemas (trabajos 1,3 y 4 de la Tabla 3.1). Además de que detallan tres metodologías o formas diferentes para la construcción de ontologías: mediante métodos semiautomáticos, trabajo 1 de la Tabla 3.1 (Fernández Bonilla 2010), mediante diagramas de clases UML, trabajo 2 de la Tabla 3.1 (Galindo García 2011) y mediante una adaptación de METHONTOLOGY, trabajo 3 de la Tabla 3.1 (Valencia Vargas 2011), lo que también es de utilidad ya que en nuestra investigación se construye una ontología y además se establece una metodología de construcción, aunque la tesis que más influyó para inclinarse por esta investigación es la de Rubí Celia Martínez en la que se entregaron sus resultados y actualmente falta terminar el documento de tesis y presentar el examen de grado, pero que es el antecedente directo de esta tesis, ya que ella generó una ontología de plantilla de patrones de análisis, la cual se utilizará en la implementación del sistema propuesto en esta tesis.

En la Tabla 3.1 se presenta una comparativa de los antecedentes señalados en este apartado. A continuación se describen los símbolos y columnas de la tabla mencionada.

- Símbolos empleados:
 - El símbolo “X” en la intersección de una fila con una columna, significa que es falsa la característica.
 - El símbolo “✓” en la intersección de una fila con una columna, significa que es cierta la aseveración.
 - “Métodos Semiautomáticos” significa que en la tesis se utilizan dichos métodos para la construcción de ontologías.
 - “Mediante diagramas de clases de UML” significa que en la tesis se lleva a cabo una transformación de diagramas de clase de UML a ontologías.
 - “Adaptación de METHONTOLOGY” significa que en la tesis se lleva a cabo modificaciones a la metodología para la construcción de ontologías METHONTOLOGY, para adaptarla a un determinado dominio.

- Columnas:
 - Patrones de análisis (3ra columna): Un símbolo “✓”, en esta columna, Indica que en la tesis se habla acerca de los patrones de análisis.
 - Metodología para Ontologías (4ta columna): En esta columna, se incluye el nombre de la metodología para construcción de ontologías que se sigue en la tesis.
 - Producto resultado (5ta columna): Contiene una descripción de los productos resultantes de la tesis, los cuales pueden ser, metodologías, ontologías, sistemas, métodos, etc.

Tabla 3.1 Comparativa de antecedentes

No.	Trabajo	Patrones de Análisis	Metodología para construcción de Ontologías	Producto resultado
1	(Fernández Bonilla 2010)	X	Métodos Semiautomáticos	Ontología, Metodología
2	(Galindo García 2011)	X	Mediante diagramas de clases de UML	Métodos de transformación
3	(Valencia Vargas 2011)	X	Adaptación de METHONTOLOGY	Ontología
4	(Martínez Jiménez 2013)	✓	Adaptación de METHONTOLOGY	Ontología
5	Tesis	✓	Adaptación de METHONTOLOGY	Metodología, Ontología, Especificación y Diseño de Alto Nivel del sistema AUPAs

3.2 Trabajos relacionados

A continuación se muestra un análisis de los aspectos más relevantes que proponen otros autores con respecto a sus trabajos de investigación en temas de patrones de análisis, bases de datos, repositorios y ontologías.

En (Raminhos et al. 2006) se plantea la problemática referente a la necesidad de una plantilla genérica para la especificación de patrones de análisis, debido a que las existentes incluyen elementos diferentes. El objetivo central es la normalización de los enfoques existentes, mediante la creación de una plantilla que unifique las existentes y defina contenidos que llenen todas las brechas identificadas.

En (Dehainsala et al. 2007) se propone una técnica para la construcción de una base de datos basada en ontologías, la cual consiste en almacenar los datos de la ontología y sus instancias en la misma base de datos y se establece una regla a seguir que dice: por cada clase de la ontología se generará una tabla de la base de datos.

En (Bobkawska & Grabowski 2009) se presenta un estudio que tiene el objetivo de obtener los beneficios y problemas relacionados con el uso de patrones de análisis, el impacto de los patrones de análisis en el proceso de análisis y los aspectos relacionados con la eficiencia de los patrones de análisis.

Este estudio se llevó a cabo en un centro médico. Los patrones de análisis implementados fueron: “patrón de análisis para reservación y uso de entidades reutilizables” (Fernandez & Yuan 1999), “patrón de análisis para la orden y envío de un producto” (Fernandez et al. 2000), “patrón de análisis para inventarios” (Fernandez 2000) y los patrones para observaciones y mediciones (Fowler 1997).

Se muestra, que al usar los patrones de análisis se obtienen los siguientes resultados:

- Se redujo a 25 clases de un total de 49 con respecto a un análisis realizado sin patrones de análisis.
- Se efectuaron 9 modificaciones a las clases originales para mejorar su diseño. Dichas modificaciones se agrupan en los tipos siguientes:
 - Especialización.
 - Eliminación de clases originales debido a que eran redundantes.
 - Cambio de etiquetas de clase o de atributos.
 - Adición de nuevos atributos.

Como conclusión, en dicho trabajo se hace hincapié en la necesidad de un repositorio de patrones de análisis y la importancia de la actualización de los patrones de análisis existentes.

En (da Matta Vegi et al. 2011) se propone una arquitectura de patrones de análisis para reutilización (APRI), compuesta de servicios web y empleando una arquitectura orientada a servicios. La arquitectura posee un repositorio de patrones de análisis en formato XMI. El principal objetivo de este trabajo es catalogar y reutilizar los patrones de análisis. Emplea la plantilla de patrones de diseño planteada en (Raminhos et al. 2006), sin embargo no se hace mención del número de patrones empleados en el caso de estudio, ni se realizan transformaciones de ningún tipo.

En (Flores Vitelli 2011) se lleva a cabo la construcción de una ontología para el dominio de la microbiología utilizando la metodología METHONTOLOGY para la elaboración de ontologías. El reporte se divide en tres partes. La primera, consiste en una explicación breve de qué es una Ontología y sus características. La segunda, explica en detalle las actividades en el proceso de desarrollo de una Ontología y de manera particular en el caso de METHONTOLOGY. La última parte, se refiere a cómo se implementó dicha Ontología.

Al igual que los antecedentes, la información incluida en los trabajos relacionados detallados arriba, también se utilizó para ayudarnos en la conformación de nuestra propuesta de solución. Consideramos que la influencia de esta información es más directa, ya que hubo trabajos que utilizan repositorios de patrones (trabajos 3 y 4 de la Tabla 3.2) (da Matta Vegi et al. 2011) y (Bobkawska & Grabowski 2009) lo que motivó a pensar en uno. Sin embargo en esos trabajos se almacenan patrones de análisis sin contemplar su semántica, cosa que diferencia a nuestra investigación. También fueron de gran utilidad aquellos trabajos que se enfocan en el desarrollo y utilización de ontologías, trabajo 5 de la Tabla 3.2 (Flores Vitelli 2011), además de los trabajos enfocados al almacenamiento de ontologías en bases de datos, trabajo 2 de la Tabla 3.2 (Dehainsala et al. 2007), aunque en dicho trabajo se muestra cómo aplicar las bases de datos basadas en ontologías, el dominio que maneja es distinto al de esta tesis.

En la Tabla 3.2 se presenta una comparativa de los trabajos relacionados señalados en este apartado. A continuación se describen los símbolos y columnas de la tabla mencionada.

- Símbolos empleados:
 - El símbolo “X” en la intersección de una fila con una columna, significa que es falsa la característica.

- El símbolo “✓” en la intersección de una fila con una columna, significa que es válida la característica.
- “METHONTOLOGY” significa que el trabajo hace uso de esa metodología para la construcción de ontologías.
- Columnas:
 - Patrones de análisis (2da columna): Indica si en el trabajo se habla acerca de los patrones de análisis.
 - Ontologías (3ra columna): Indica si en el trabajo se habla acerca de ontologías.
 - Metodología para Ontologías (4ta columna): Indica si en el trabajo se sigue o se hace referencia a una metodología para construcción de ontologías.
 - Repositorios (5ta columna): Indica si en el trabajo se emplea algún tipo de repositorio.
 - Interfaz de captura y despliegue de PA (6ta columna): Indica si en el trabajo se emplea algún tipo de captura de patrones de análisis o de despliegue de los mismos.
 - Producto resultado (7ma columna): Contiene los productos resultantes del trabajo de investigación, los cuales pueden ser, plantillas, metodologías, ontologías, sistemas, arquitecturas, modelos etc.

Tabla 3.2 Comparativa de trabajos relacionados

No.	Trabajo	Patrones de Análisis	Ontologías	Metodología para Ontologías	Repositorios	Interfaz de captura y despliegue de PA	Producto resultado
1	(Raminhos et al. 2006)	✓	✗	✗	✗	✗	Plantilla
2	(Dehainsala et al. 2007)	✗	✓	✗	✓	✗	Arquitectura de OBDB
3	(Bobkawska & Grabowski 2009)	✓	✗	✗	✓	✗	Modelo conceptual de sistema de almacenaje de PA
4	(da Matta Vegi et al. 2011)	✓	✗	✗	✓	✓	Arquitectura conceptual de sistema de almacenaje de PA
5	(Flores Vitelli 2011)	✗	✓	METHONTOLOGY	✗	✗	Ontología construida mediante METHONTOLOGY
6	Tesis	✓	✓	METHONTOLOGY	✓	✓	Metodología, Ontología, Especificación y Diseño de Alto Nivel del sistema AUPAs

En este capítulo se realizó una comparativa de los principales antecedentes y trabajos relacionados previos a este trabajo, se explicaron las similitudes y las diferencias con la presente investigación, además de servir como punto de referencia para ver la importancia de esta aportación con respecto de las demás en la sociedad. De acuerdo a los trabajos presentados podemos afirmar que no existen repositorios de patrones de análisis que apliquen ontologías para su mejor descripción, lo que nos lleva a proponer la siguiente estructura de almacenamiento de patrones de análisis como conocimiento (ver Capítulo 4).

Capítulo 4

Estructura de Almacenamiento de Patrones de Análisis como Conocimiento

En este capítulo se propone una estructura de almacenamiento de los PAs como conocimiento, utilizando ontologías estructuradas en tres niveles de abstracción: 1) ontología de plantilla estable de PAs (una ontología única) y sus individuos (uno para cada PA), 2) ontologías de PAs (una ontología para cada PA) y los individuos de cada ontología (uno por cada caso de estudio) y 3) ontologías de casos de estudio (una para cada caso).

4.1 Estructura

Como resultado de los estudios en materia de patrones de análisis por parte de nuestro grupo de investigación surge la propuesta de una estructura para almacenar los patrones de análisis (ver Figura 4.1) de manera que promueva y facilite su utilización y reutilización, mediante la inclusión de conceptos de estabilidad de software y de semántica a los patrones.

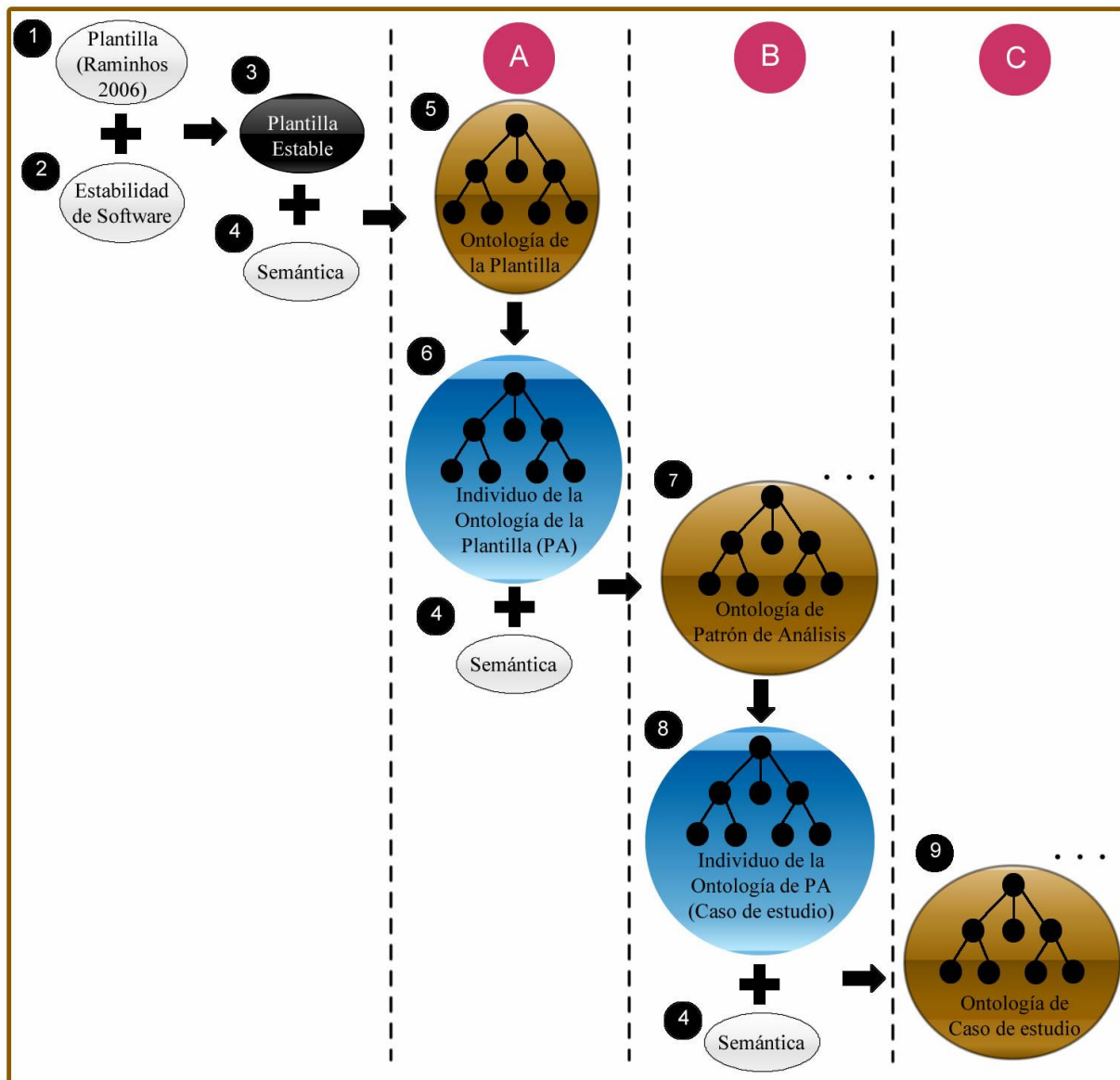


Figura 4.1 Estructura de Almacenamiento de Patrones de Análisis como Conocimiento

El trabajo realizado por Rubí Celia (Martínez Jiménez 2013) consistió en tomar la plantilla (correspondiente al punto marcado con el número 1 de la Figura 4.1) e incorporar los conceptos de estabilidad de software (Fayad & Altman 2001) (punto número 2), obteniendo como producto una plantilla estable de patrones de análisis (punto número 3), la cual, al aplicarle semántica (punto número 4) se obtuvo una ontología de plantilla (punto número 5).

La principal aportación de la estructura mostrada en la Figura 4.1, es la clasificación de las ontologías en tres niveles de abstracción: A) ontología de plantilla estable de PAs (una ontología única) y sus individuos (uno para cada PA), B) ontologías de PAs (una ontología para cada PA) y los individuos de cada ontología (uno por cada caso de estudio) y C) ontologías de casos de estudio (una para cada caso).

4.2 Elementos de la estructura

A continuación se presentan los elementos que conforman a la estructura de la Figura 4.1:

- 1 Plantilla: Se refiere a la plantilla para describir patrones de análisis propuesta por Ricardo Raminhos (Raminhos et al. 2006).
- 2 Estabilidad de software: Es un modelo propuesto por Fayad (Fayad & Altman 2001) que consta de tres elementos: Temas de negocios perdurables (EBTs): son las clases invariantes que presentan los conceptos básicos y perdurables del sistema. Objetos de negocios (BOs): son las clases que mapean las EBTs del sistema en objetos concretos. Objetos industriales (IOs): son las clases que mapean las BOs del sistema en objetos físicos. En la sección 2.2 se explican de una manera más detallada los conceptos de estabilidad.
- 3 Plantilla Estable: Es la plantilla de patrones de análisis desarrollada por Rubí Celia basada en los 18 puntos de la plantilla de Raminhos pero añadiendo una clasificación de acuerdo a los tres niveles de estabilidad mencionados en el punto anterior (Estabilidad de software).
- 4 Semántica: Consiste en determinar el significado de los conceptos, definir sus propiedades y las relaciones entre ellos. Así como las restricciones y/o posibles reglas.

- 5 Ontología de la Plantilla: Es una única ontología que representa el conocimiento de la Plantilla Estable, mediante clases, propiedades y relaciones entre clases.
- 6 Individuo de la ontología de plantilla: Es la instancia de ontología que se obtiene al poblar con datos cada una de las clases de la ontología de plantilla. Esos datos se obtienen de la descripción de un patrón de análisis específico. Un ejemplo de este tipo de individuo podría ser la descripción del patrón de análisis para reservación y uso de entidades reutilizables (ver sección 6.1.1 Refinamiento del patrón de análisis) o del patrón de análisis para inventarios.
- 7 Ontología de Patrón de Análisis: Para construir esta ontología se toma el individuo de la ontología anterior (punto 6) y se le añade semántica definiendo sus clases, propiedades y relaciones entre clases, así como las restricciones y/o posibles reglas. Habrá una ontología de este tipo para cada patrón de análisis que se desee representar como conocimiento. Ontologías de este tipo, son por ejemplo la ontología para el patrón de análisis para reservación y uso de entidades reutilizables (ver Capítulo 6 Construcción de una Ontología del Patrón de Análisis para Reservación de Entidades), o la ontología para el patrón de análisis para inventarios.
- 8 Individuo de la ontología de patrón de análisis: Un individuo de este tipo es aquella instancia que se obtiene al poblar con datos cada una de las clases de la ontología de un patrón de análisis. Esos datos se obtienen de la descripción de un caso de estudio de un patrón de análisis específico. Un ejemplo de este tipo de individuo podría ser la descripción del patrón de análisis para reservación y uso de entidades reutilizables con un caso de estudio en particular, como la renta de bienes raíces “Ramiro y Asociados”, otro individuo podría ser la reservación de habitaciones para el hotel “Rapin Inn”, o el individuo para el préstamo de libros en el centro de información del CENIDET (ver la sección 6.1.8 Instanciación de la ontología).
- 9 Ontología de Caso de estudio: Para construir esta ontología se toma el individuo de la ontología anterior (punto número 8) y se le añade semántica definiendo sus clases, propiedades y relaciones entre clases, así como las restricciones y/o posibles reglas. Habrá una ontología para cada individuo de un patrón de análisis. Por mencionar un ejemplo, para el patrón de análisis para reservación y uso de entidades reutilizables puede haber varios casos de

estudio, por ejemplo: 1) la reservación de libros en el centro de información del CENIDET, 2) la reservación y renta de bienes raíces en la empresa “DreamHome” etc. Para cada caso de estudio habrá una ontología que represente la semántica específica que lo diferencia de los otros casos del mismo patrón.

4.3 Avances en la estructura y conclusiones del capítulo

En la investigación realizada por Rubí Celia (Martínez Jiménez 2013) se logró implementar hasta el punto 5 de la estructura mostrada en la Figura 4.1.

En la presente investigación de tesis se logró completar el punto 6, 7 y 8, quedando pendiente únicamente el punto 9 para futuras investigaciones.

En capítulos posteriores se comprueba la factibilidad de implementar esta Estructura de Almacenamiento de Patrones de Análisis. También se detalla el proceso que se siguió para desarrollar los elementos 6, 7 y 8 de la misma. En el Capítulo 5, se propone una metodología para la construcción de ontologías de patrones de análisis, que se utiliza para documentar los resultados del Capítulo 6. Esta ontología corresponde al elemento 7 de la estructura de almacenamiento. En el Capítulo 6 se presenta la construcción de una ontología para el patrón de análisis para reservación y uso de entidades reutilizables.

Capítulo 5

Metodología para la Construcción de Ontologías de Patrones de Análisis

En el presente capítulo se hace una adaptación de METHONTOLOGY para la creación de ontologías de patrones de análisis. Se explican los dos procesos de refinamiento por los que deben pasar los patrones de análisis antes de ser considerados para la definición de su ontología, además de detallar las fases de especificación, integración, conceptualización y formalización de las ontologías.

5.1 Introducción

La ausencia de actividades estandarizadas, ciclos de vida, metodologías sistemáticas, técnicas y herramientas para la construcción de ontologías, que conforman un conjunto de criterios de diseño bien definidos, convierten al desarrollo de ontologías un arte en vez de una actividad de ingeniería (Cendón et al. 2005). Para que un arte se convierta en ingeniería debe existir una definición y estandarización de un ciclo de vida que vaya acorde a la definición de requerimientos para el mantenimiento del producto final, es ahí donde entra el presente trabajo, el cual busca incorporar una metodología o técnica que conduzca a los desarrolladores por el camino correcto.

5.2 Metodología para la construcción de ontologías de patrones de análisis (MECOPA)

Cuando se trata de construir una ontología, no existe una metodología, que se utilice de manera estándar, sino que cada diseñador opta por una o por otra de acuerdo a sus necesidades. Sin embargo una de las más utilizadas es METHONTOLOGY (Fernández et al. 1997), de la cual, para esta investigación, se tomaron las fases más relevantes: 1) especificación (fase donde se define el propósito, grado de formalidad, alcance y usuarios finales de la ontología), 2) integración (consiste en buscar ontologías relacionadas para su posible reutilización), 3) conceptualización (cuyo objetivo es construir un glosario de términos donde se incluyan conceptos, instancias, verbos y propiedades) y 4) formalización (consiste en definir los términos del punto anterior con un lenguaje computacional, ya sea de forma manual con la ayuda de un editor de ontologías). Además de incluir algunas otras (Corcho et al. 2003; Gruninger & Fox 1995) para crear una metodología especializada en ontologías de patrones de análisis, denominada MECOPA. Esta metodología consta de las fases descritas en la Figura 5.1.

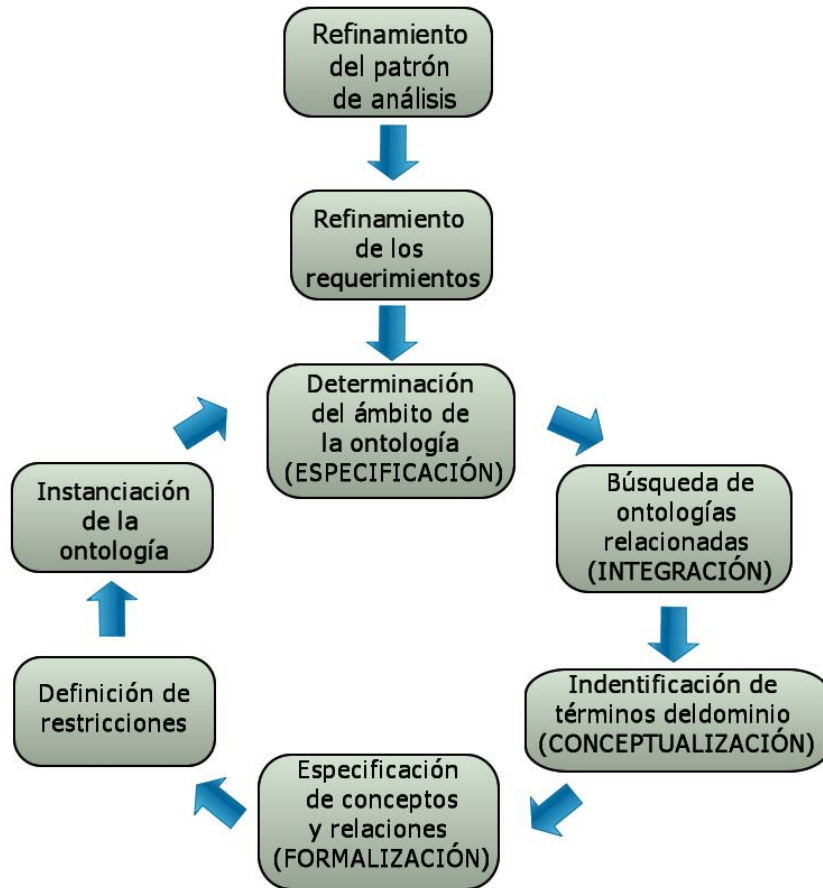


Figura 5.1 Ciclo de vida de MECOPA

A continuación se explica en que consiste cada una de las fases que componen al ciclo de vida de MECOPA:

5.2.1 Refinamiento del patrón de análisis

Consiste en adaptar la descripción del patrón de análisis a la estructura de la plantilla de patrones de análisis estable (PPAE). Para esto es necesario ingresar un nuevo individuo a la ontología OPPAE (Martínez Jiménez 2013).

La mayoría de las propiedades de cada una de las clases de la OPPAE son de tipo string. Es decir sólo se necesita ingresar texto, sin embargo hay sus excepciones como aquellos puntos de la plantilla donde se requiere el ingreso de diagramas como los son los puntos 9 (Requerimientos) y 10 (Modelado) de la PPAE. Para esto se sugiere que una vez creados los diagramas se exporten con una herramienta de

modelado como “Enterprise Architect” en formato XML, y se realice un compendio de todos los diagramas para el determinado PA, al que se esté desarrollando su ontología, en un directorio adjunto a la ubicación de la ontología OPPAE, y al llenar la propiedad donde vaya un diagrama sólo bastara con indicar la ruta donde se encuentra dicho diagrama (ver Figura 5.2).

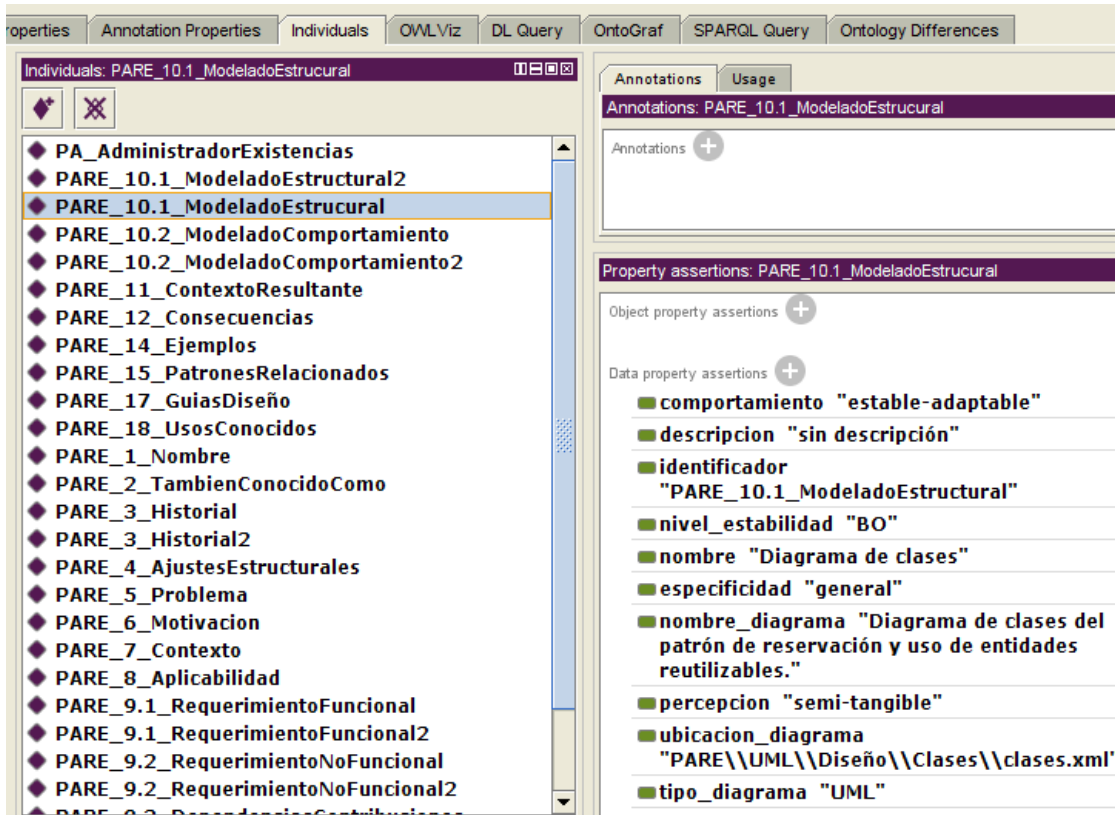


Figura 5.2 Valores de propiedades para la Clase “Modelado Estructural” correspondiente al punto 10.1 de la plantilla PPAE¹

5.2.2 Refinamiento de los requerimientos

Para el refinamiento de los requerimientos se debe seguir la recomendación del Sector de Estandarización de Telecomunicaciones de la Unión de Telecomunicaciones Internacional (ITU-T 2012), que permite realizar licitación, análisis, especificación, y validación de requerimientos, para lo cual se utilizan dos lenguajes de la Notación de Requerimientos de Usuario (URN, por sus siglas en inglés):

¹ PPAE: Hace referencia a la plantilla de patrones de análisis desarrollada por Rubí Celia (Martínez Jiménez 2013)

- GRL
- UCM

Es por esto, que esta fase consiste en elaborar dichos diagramas GRL y UCM correspondientes al apartado nueve de la plantilla de Raminhos (Raminhos et al. 2006). Y cuyo primer paso es la realización de un acomodo de requerimientos con sus descripciones, mediante una jerarquía, tomando en cuenta una secuencia en el tiempo en la que se deben hacer acciones más simples para cumplir con acciones más complejas y una vez cumplidas estas, cumplir con el requerimiento (ej. Se requiere consultar disponibilidad, antes de reservar entidad, etc).

Ejemplo:

1. Hacer una reservación

1.1 Solicitar una entidad: Acción efectuada por el cliente, indicándole al empleado, su interés por reservar una determinada entidad.

1.2 Reservar entidad: Proceso que efectúa un empleado indicándole al sistema que se desea realizar una reservación de una determinada entidad.

1.2.1 Consultar disponibilidad: Actividad que influye en el éxito de la reservación, indica que si una entidad está disponible se puede reservar, o de lo contrario, no disponible significa que ya ha sido reservada por otro cliente.

1.3 Registrar información: Actividad que realiza el empleado para registrar los detalles de una reservación, en un documento de reservación.

1.31 Documento reservación: Es un recurso informático donde se registran los detalles de la reservación.

Para posteriormente proceder a formular los diagramas GRL utilizando la simbología indicada en la Figura 2.2.

Como tercer paso se realiza un listado con los componentes que intervienen en el cumplimiento de cada requerimiento así como sus responsabilidades.

Ejemplo:

Tabla 5.1 Ejemplo de actores, objetos y responsabilidades

Actores	Responsabilidades
Cliente	Reservar, Registrar reservación
Empleado	Crear reservación, Retornar costo, Confirmar, Retornar error
Objetos	
Reservación	Verificar disponibilidad, Calcular costo, Retornar costo, Generar núm. De confirmación, Reservar entidad, Confirmación, Retornar error
Entidad	Disponible, No disponible

Y por último se opta por construir los diagramas UCM empleando la notación mostrada en la Figura 2.3.

Cabe mencionar que la mayoría de las veces los autores de los patrones de análisis no incluyen toda la descripción del patrón en los requerimientos, sino que lo complementan con otros puntos de la plantilla de Raminhos (Raminhos et al. 2006). Como el 10 donde se presenta un modelado del negocio, el 5 donde se expone el problema o el 6 donde se presenta la motivación.

Por lo anterior se recomienda considerar estos puntos además del 9 para la elaboración de los diagramas GRL y UCM.

5.2.3 Determinación del ámbito de la ontología (Especificación)

El objetivo de esta fase es producir un documento de especificación de la ontología ya sea informal, semi-formal o formal, escrito en lenguaje natural, utilizando un conjunto de representaciones intermedias o usando cuestionarios de competencias (CCs). (Fernández et al. 1997)

Dicho documento debe incluir el propósito, el dominio, los usuarios y el tipo de ontología. Contestando las siguientes preguntas:

- ¿Cuál es el dominio que cubre la ontología?
- ¿Para qué se usará la ontología?
- ¿Para qué tipo de cuestiones?
- ¿Quién usará o mantendrá la ontología?

La mayoría de las metodologías existentes sugieren la identificación de un cuestionario de competencias como la técnica para establecer los requerimientos de

la ontología (Gruninger & Fox 1995). El CC es un conjunto de preguntas en lenguaje natural, que debe responder la ontología que se está desarrollando.

Algunas estrategias para identificar los CCs son:

- Top-Down: El autor empieza con preguntas complejas, las cuales se van descomponiendo en preguntas más simples.
- Bottom-Up: El autor comienza con preguntas simples que se van estructurando para crear preguntas complejas.
- Middle-out: El autor empieza simplemente escribiendo preguntas importantes que se componen y descomponen después de manera abstracta o en preguntas más simples respectivamente.

El objetivo es extraer de la lista de preguntas y respuestas del CC un pre-glosario de términos. De los requerimientos en forma de preguntas, se extrae la terminología (nombres, adjetivos y verbos) que será formalmente representada en la ontología como conceptos, atributos, relaciones o instancias. Mientras que de las respuestas a las preguntas, se extrae la terminología que pudiera ser representada en la ontología como conceptos o instancias. (Suárez-Figueroa et al. 2009)

5.2.4 Búsqueda de ontologías relacionadas (Integración)

Con el objetivo de acelerar la construcción de la ontología, se debe considerar la reutilización de términos que ya han sido empleados en otras ontologías en lugar de empezar desde cero. (Fernández et al. 1997).

Para esto se debe realizar una investigación exhaustiva en repositorios, artículos, etc. de al menos 3 ontologías que se puedan reutilizar y que estén relacionadas al tema. Se pueden revisar ontologías públicas como: Ontolingua, DAML, UNSPSC, RosettaNet, DMOZ, CyC, etc.

5.2.5 Identificación de términos del dominio (Conceptualización)

En esta actividad, se debe estructurar el conocimiento del dominio en un modelo conceptual que describa el problema y su solución en términos del vocabulario del dominio identificado en la actividad de especificación de la ontología. (Fernández et al. 1997)

Se determinan los conceptos (clases) y propiedades (roles) de la ontología. Para esto es necesario escribir una lista de términos que se usarían para redactar oraciones sobre el tema del patrón de análisis.

- ¿Qué términos se preferiría usar?
- ¿Cuáles son las propiedades de estos términos?
- ¿Qué se puede decir sobre estos términos?

Para facilitar este punto se recomienda apoyarse de los diagramas GRL y UCM elaborados en el punto 5.2.2, ya que la mayoría de los términos del dominio se mencionan o son elementos de los diagramas. Además de ayudarse del cuestionario de competencias elaborado en el apartado 5.2.3 y de las ontologías relacionadas de la sección 5.2.4.

Otro aspecto muy importante a considerar es el Modelo de Estabilidad de Software (SSM) (Fayad & Altman 2001), tomando en cuenta atributos (propiedades) como comportamiento, percepción, especificidad, además del nivel de estabilidad de cada clase.

5.2.6 Especificación de conceptos y relaciones (Formalización)

Se lleva a cabo una especificación explícita de los conceptos y sus relaciones entre ellos, utilizando una jerarquía, además de especificar las propiedades de cada concepto.

Una vez especificados en borrador, se procede a pasarlos a un editor de ontologías gráfico como Protégé.

5.2.7 Definición de restricciones

En Protégé existen dos tipos de restricciones de cuantificación: las existenciales utilizando la palabra “some” (ver Figura 5.3), se usan para especificar que una clase se relaciona con al menos otra, y las universales empleando la palabra “only”, utilizadas para indicar que una clase se relaciona únicamente con otra.

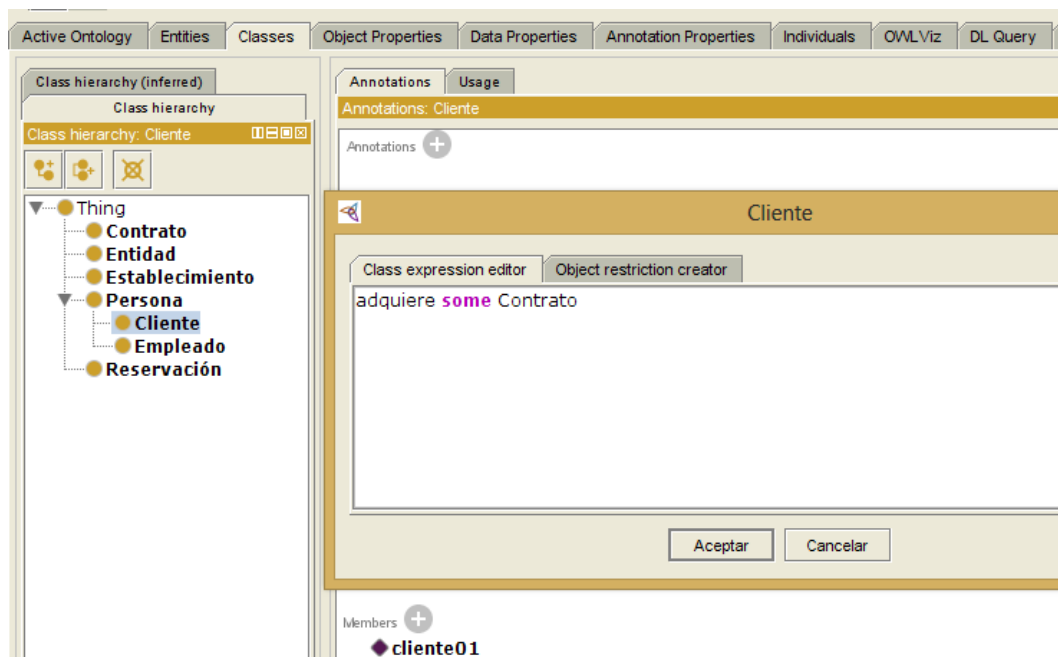


Figura 5.3 Aplicando la restricción “some” en Protégé

5.2.8 Instanciación de la ontología

En este paso se crean los datos de la ontología también llamados individuos, y consiste en instanciar cada una de las clases que componen a la ontología (ver Figura 5.4).

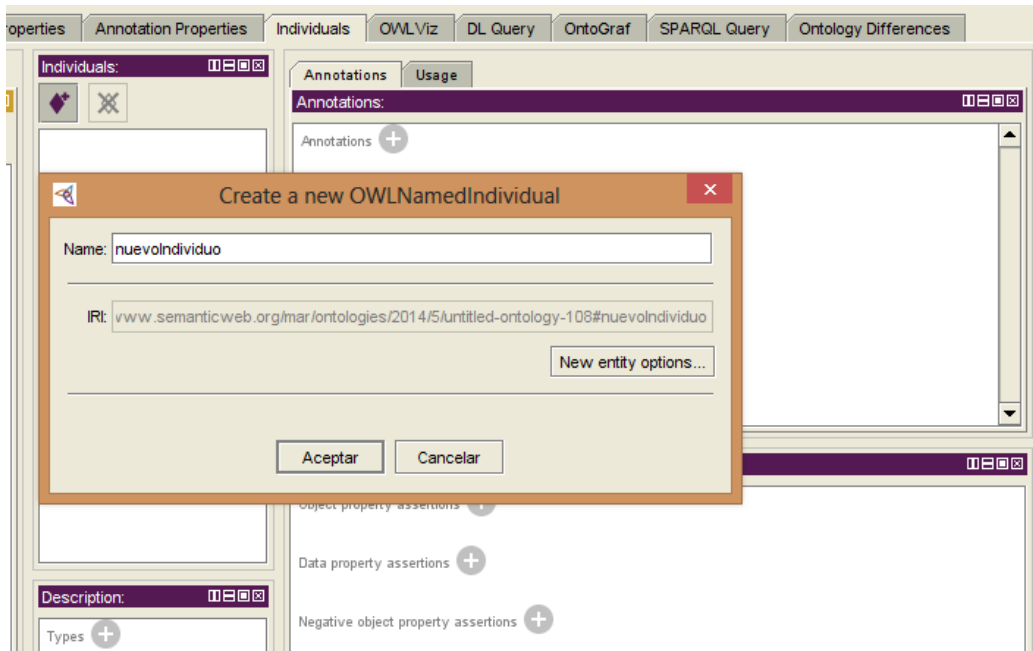


Figura 5.4 Ingresando Individuos en Protégé

En este capítulo pudimos apreciar algunas recomendaciones para llevar a cabo el proceso de construcción de ontologías, con detalles específicos para el dominio de los patrones de análisis. En el Capítulo 6 se muestra un ejemplo de una ontología generada siguiendo la metodología propuesta en este capítulo.

Capítulo 6

Construcción de una Ontología del Patrón de Análisis para Reservación de Entidades

En el presente capítulo se presenta la documentación y los pasos que se llevaron a cabo para la construcción de la ontología del patrón de análisis para reservación y uso de entidades reutilizables. Se detallan los dos procesos de refinamiento por los que pasó el patrón de análisis de reservación de entidades antes de ser considerado para la definición de su ontología, además se presentan los resultados de las fases de especificación, integración, conceptualización y formalización de la ontología.

6.1 Construcción de la Ontología

En este capítulo se encuentra detallado el proceso de creación de la ontología del patrón de análisis para reservación y uso de entidades reutilizables (Fernandez & Yuan 1999), a través del seguimiento de la metodología MECOPA propuesta en el capítulo anterior. En la Figura 5.1 se puede ver un esquema que representa el orden en el que se llevan a cabo las fases que constituyen el ciclo de vida de una ontología, según esta metodología.

Para facilitar la asimilación de cada una de las fases que componen a MECOPA se ha decidido emplear un caso de estudio a manera de ejemplo para explicar cada sección. Supongamos que tenemos un centro de investigación (CENIDET) que ofrece posgrados, y cuenta con una biblioteca, denominada centro de información, a la cual tienen acceso todos los alumnos y docentes que laboran en el CENIDET. El centro de información cuenta con un encargado, y una asistente, quienes atienden a los interesados en reservar algún libro. Para poder adquirir prestado un libro, es necesario que los alumnos tengan credencial del centro de información. Los libros sólo pueden ser prestados en caso de que haya dos o más ejemplares del mismo; en caso de que estén disponibles para préstamo, los alumnos llenan un formato de reservación donde viene la fecha de salida y fecha de entrega del libro, esta última es la fecha cuando el libro debe ser retornado a la biblioteca, después de haberlo utilizado.

A continuación se presenta la documentación de la ontología OPPAE correspondiente a cada fase de MECOPA:

6.1.1 Refinamiento del patrón de análisis

Enseguida se muestra el desarrollo de la primera fase de MECOPA:

En esta fase se describió el patrón de análisis para reservación y uso de entidades reutilizables, asignando valores a cada una de las propiedades de las clases de la ontología OPPAE.

En la Figura 6.1 se muestra la interfaz de la herramienta Protégé, donde se desarrolló la OPPAE. En la pestaña de “Individuos” se muestran los valores que fueron asignados a cada una de las propiedades de la clase Nombre (por mencionar

un ejemplo). Los valores de las propiedades de las demás clases se pueden observar seleccionando cada uno de los elementos que se encuentran en la ventana "Individuos" de la misma Figura 6.1.

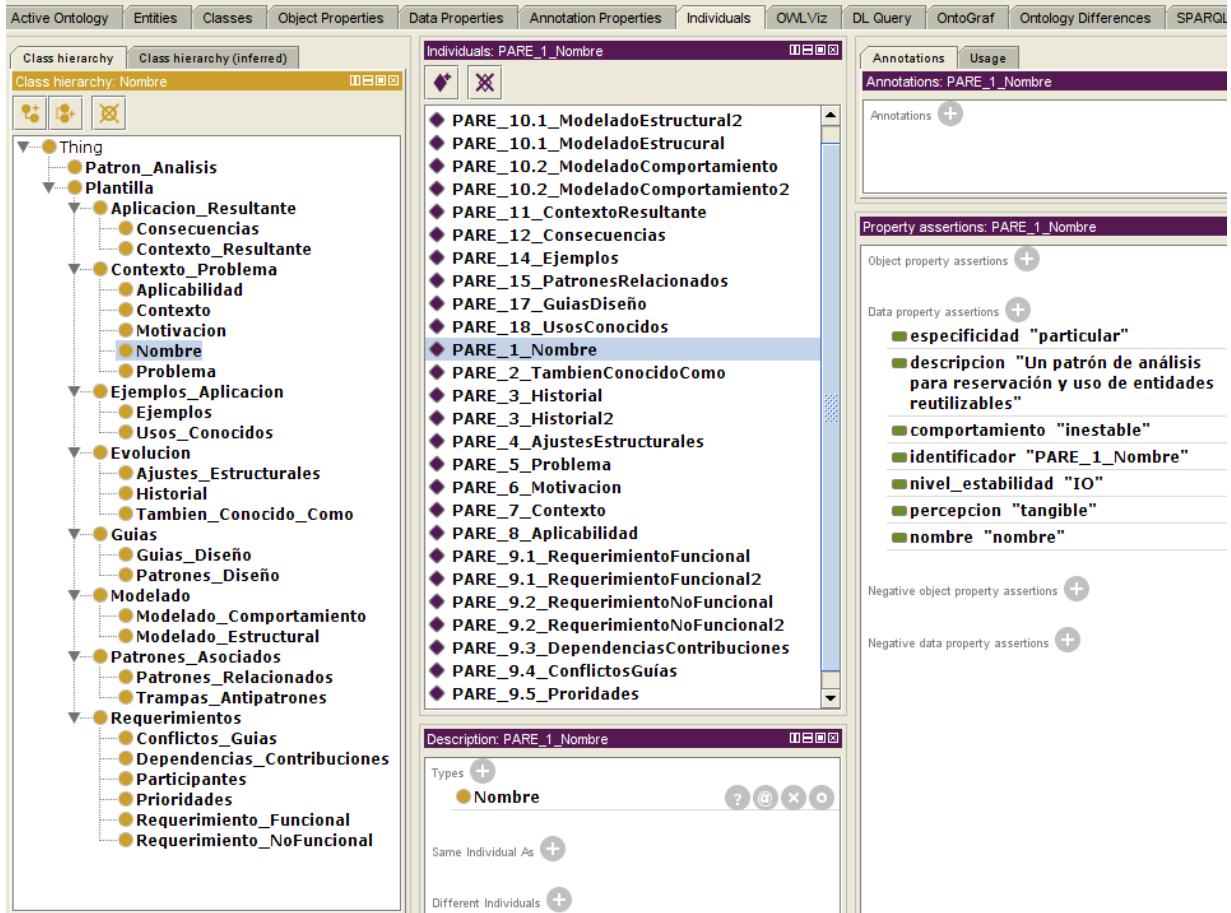


Figura 6.1 Individuos de clases de OPPAE en Protégé

6.1.2 Refinamiento de los requerimientos

A continuación se detalla la segunda fase de MECOPA:

Para llevar a cabo esta fase, nos dirigimos al apartado número 9 de la descripción del patrón de reservación y uso de entidades reutilizables, donde se encuentran los requerimientos de forma textual.

Primeramente se hizo una separación de las acciones más importantes que se deben llevar a cabo para que se cumpla el requerimiento (ejemplo: "Hacer una

reservación”), mediante una representación jerárquica (ver sección 6.1.2.1.1). Este proceso se hace para cada requerimiento.

Posteriormente, se procedió a la elaboración de los diagramas GRL de cada uno de los requerimientos, en la sección 6.1.2.1.2 se muestra un ejemplo.

Luego se hizo una lista de los posibles componentes (equipos, actores, objetos, etc.) que intervienen en el cumplimiento de cada requerimiento, así como de las posibles responsabilidades o tareas de cada componente (ver ejemplo en la sección 6.1.2.1.3).

Culminando con el diseño de los diagramas UCM de cada uno de los requerimientos (ejemplo sección 6.1.2.1.4), a continuación se muestran los resultados del refinamiento de cinco requerimientos:

6.1.2.1 Requerimiento 1. Hacer una reservación

En la Figura 6.2 y Figura 6.3 se presentan los diagramas GRL y UCM respectivamente, que describen las metas y el comportamiento del requerimiento: hacer una reservación:

6.1.2.1.1 Representación jerárquica:

1. Hacer una reservación

1.1 Solicitar una entidad: Acción efectuada por el cliente, que indica al empleado, su interés por reservar una determinada entidad.

1.2 Reservar entidad: Proceso que efectúa un empleado, que indica al sistema que se desea realizar una reservación de una determinada entidad.

1.2.1 Consultar disponibilidad: Actividad efectuada por el empleado, que influye en el éxito de la reservación. El resultado de llevar a cabo esta actividad se puede interpretar de la siguiente manera: si una entidad está disponible se puede reservar, de lo contrario, no disponible significa que ya ha sido reservada por otro cliente.

1.3 Registrar información: Actividad que realiza el empleado para registrar los detalles de una reservación, en un documento de reservación.

1.31 Documento reservación: Es un recurso informático donde se registran los detalles de la reservación.

6.1.2.1.2 Diagrama GRL²:

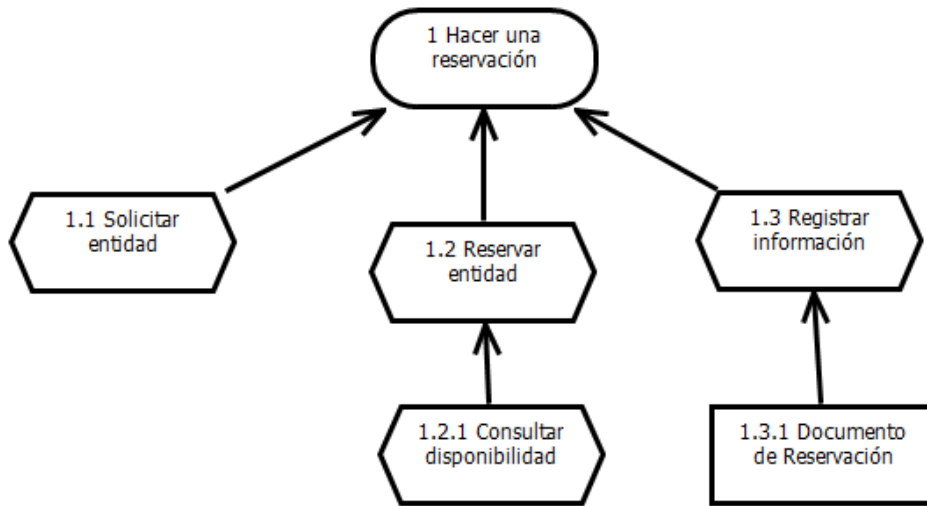


Figura 6.2 Diagrama GRL del requerimiento 1. Hacer una reservación

6.1.2.1.3 Tabla de responsabilidades:

Tabla 6.1 Actores objetos y responsabilidades del requerimiento 1. Hacer una reservación

Actores	Responsabilidades
Cliente	Reservar, Registrar reservación
Empleado	Crear reservación, Retornar costo, Confirmar, Retornar error
Objetos	
Reservación	Verificar disponibilidad, Calcular costo, Retornar costo, Generar núm. De confirmación, Reservar entidad, Confirmación, Retornar error
Entidad	Disponible, No disponible

² La nomenclatura de los diagramas GRL se encuentra en la sección 2.7 correspondiente al Capítulo 2.

6.1.2.1.4 Diagrama UCM³:

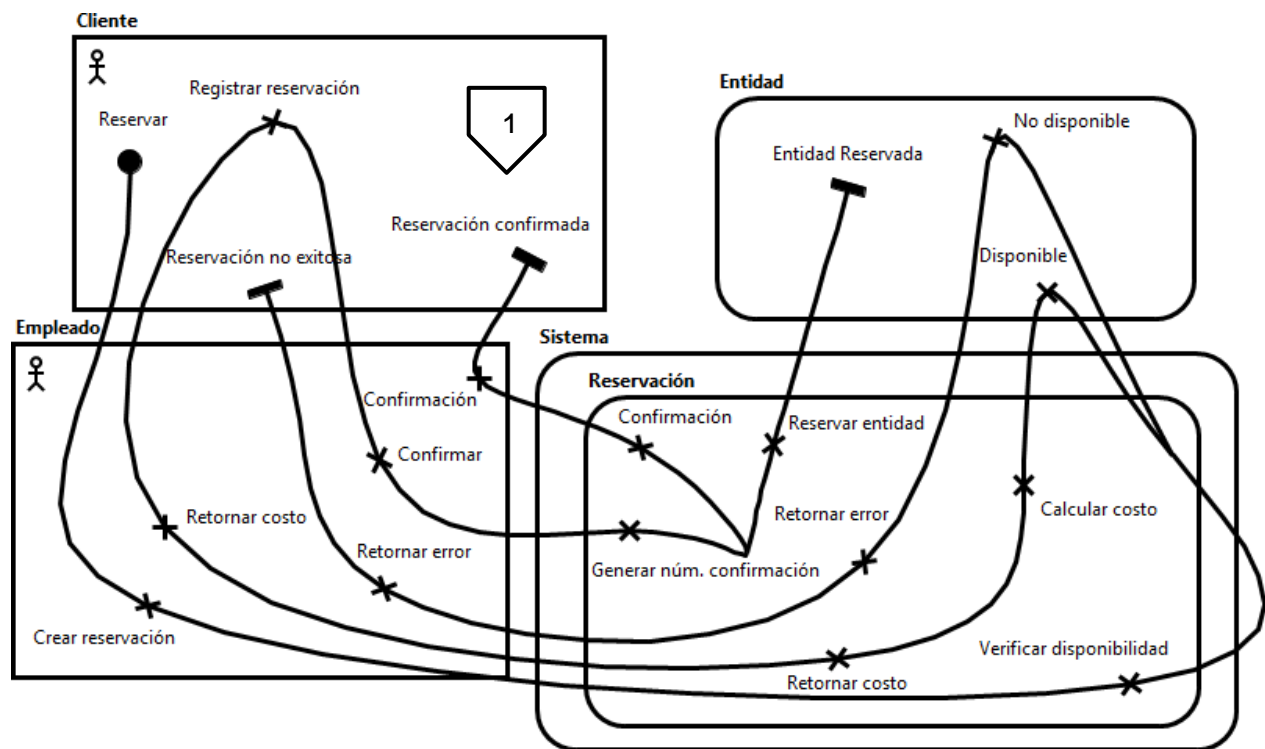


Figura 6.3 Diagrama UCM del requerimiento 1. Hacer una reservación

6.1.2.2 Requerimiento 2. Usar una entidad reservada

Los siguientes diagramas GRL y UCM describen las metas y el comportamiento del requerimiento usar una entidad reservada.

³ La nomenclatura de los diagramas UCM se encuentra en la sección 2.8 correspondiente al Capítulo 2.

6.1.2.2.1 Representación jerárquica:

2. Usar una entidad reservada

2.1 Usar entidad: Se refiere al hecho de que un cliente utiliza una entidad durante un tiempo determinado.

2.1.1 Crear registro de uso: Actividad que realiza un empleado previa a entregar la entidad al cliente.

2.2 Regresar disponibilidad: Acción que se lleva a cabo después de que la entidad ha sido usada y regresada, y consiste en volver al estado disponible, la entidad en cuestión.

2.2.1 Retornar entidad: Acción por parte del cliente en la que entrega la entidad al empleado, una vez finalizado el tiempo de reservación.

6.1.2.2.2 Diagrama GRL:

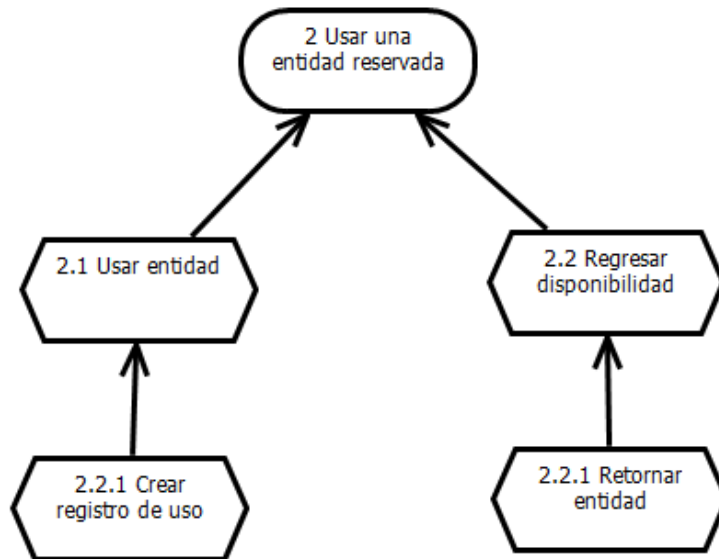


Figura 6.4 Diagrama GRL del requerimiento 2. Usar una entidad reservada

6.1.2.2.3 Tabla de responsabilidades:

Tabla 6.2 Actores objetos y responsabilidades del requerimiento 2. Usar una entidad reservada

Actores	Responsabilidades
Cliente	Usar entidad, Retornar entidad
Empleado	Crea, Entregar entidad, Terminar uso
Objetos	
Registro Uso	Quitar disponibilidad, Retornar disponibilidad
Entidad	Disponible , No disponible

6.1.2.2.4 Diagrama UCM:

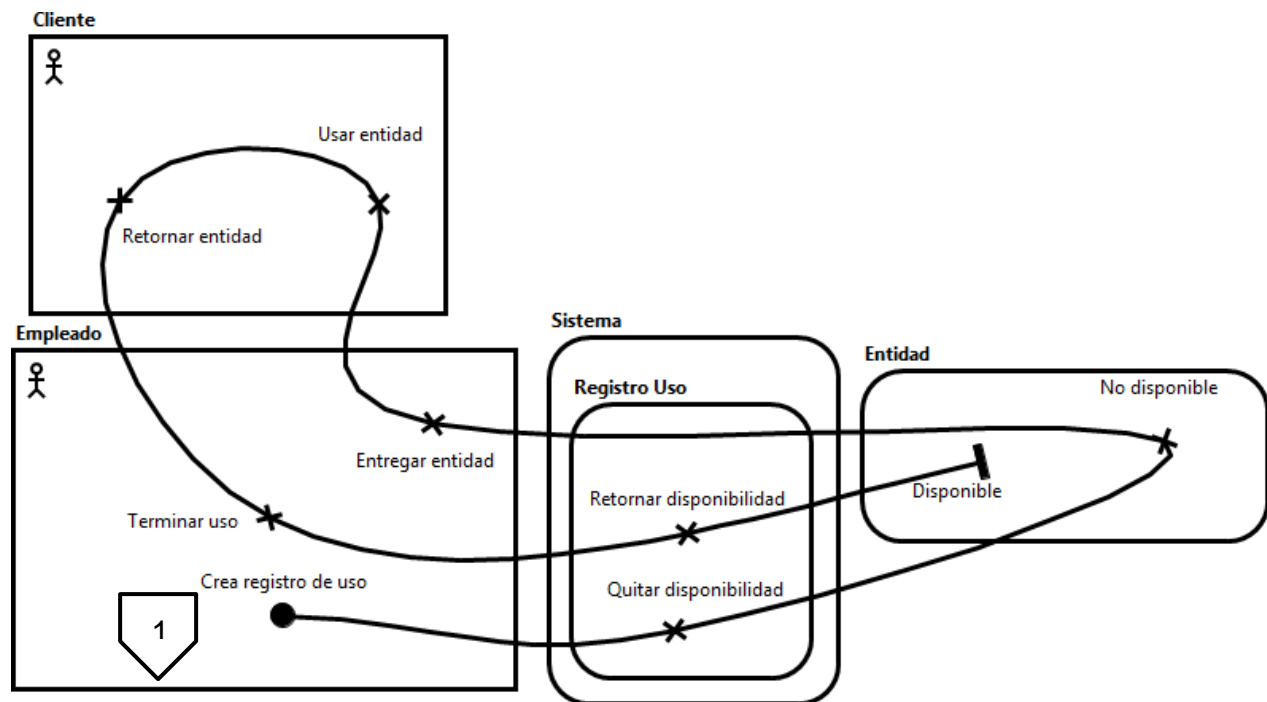


Figura 6.5 Diagrama UCM del requerimiento 2. Usar una entidad reservada

6.1.2.3 Requerimiento 3. Modificar una reservación

Los siguientes diagramas GRL y UCM describen las metas y el comportamiento del requerimiento modificar una reservación.

6.1.2.3.1 Representación jerárquica:

3. Modificar una reservación

3.1 Hacer cambios a reservación existente: Se hacen los cambios deseados a una reservación previa, y se crea una nueva.

3.1.1 Determinar disponibilidad: Actividad que influye en el éxito de la nueva reservación. Indica si la entidad de interés está disponible en el tiempo deseado.

3.1.2 Cancelar reservación previa: La reservación previa se cancela, para crear una nueva que incluya los cambios deseados.

6.1.2.3.2 Diagrama GRL:

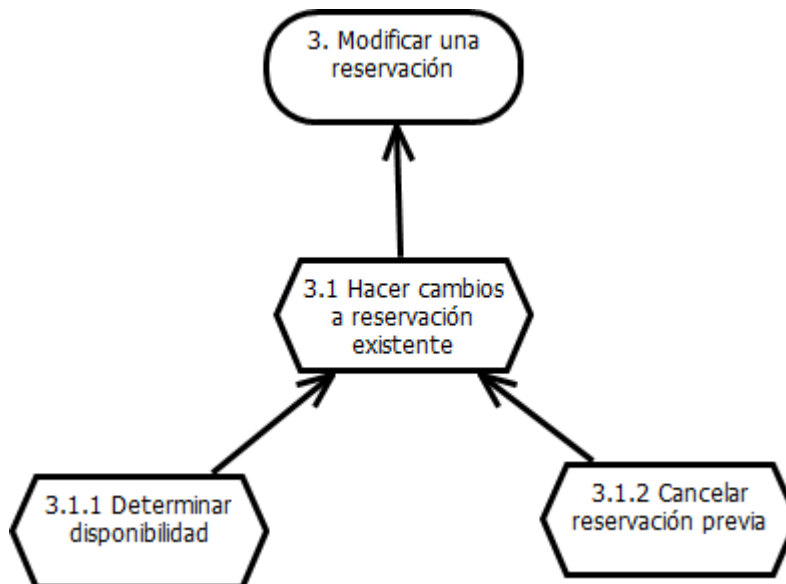


Figura 6.6 Diagrama GRL del requerimiento 3. Modificar una reservación

6.1.2.3.3 Tabla de responsabilidades:

Tabla 6.3 Actores objetos y responsabilidades del requerimiento 3. Modificar una reservación

Actores	Responsabilidades
Cliente	Solicitar modificación, Registrar cambios, Reservación confirmada
Empleado	Modificar, Retornar cambios, Confirmar, Confirmación, Retornar error
Objetos	
Reservación	Retornar cambios, Hacer cambios, Verificar disponibilidad, Cancelar reservación previa, Reservar entidad, Generar núm. De confirmación, Confirmación, Retornar error
Entidad	Disponible, No disponible , Entidad reservada

6.1.2.3.4 Diagrama UCM:

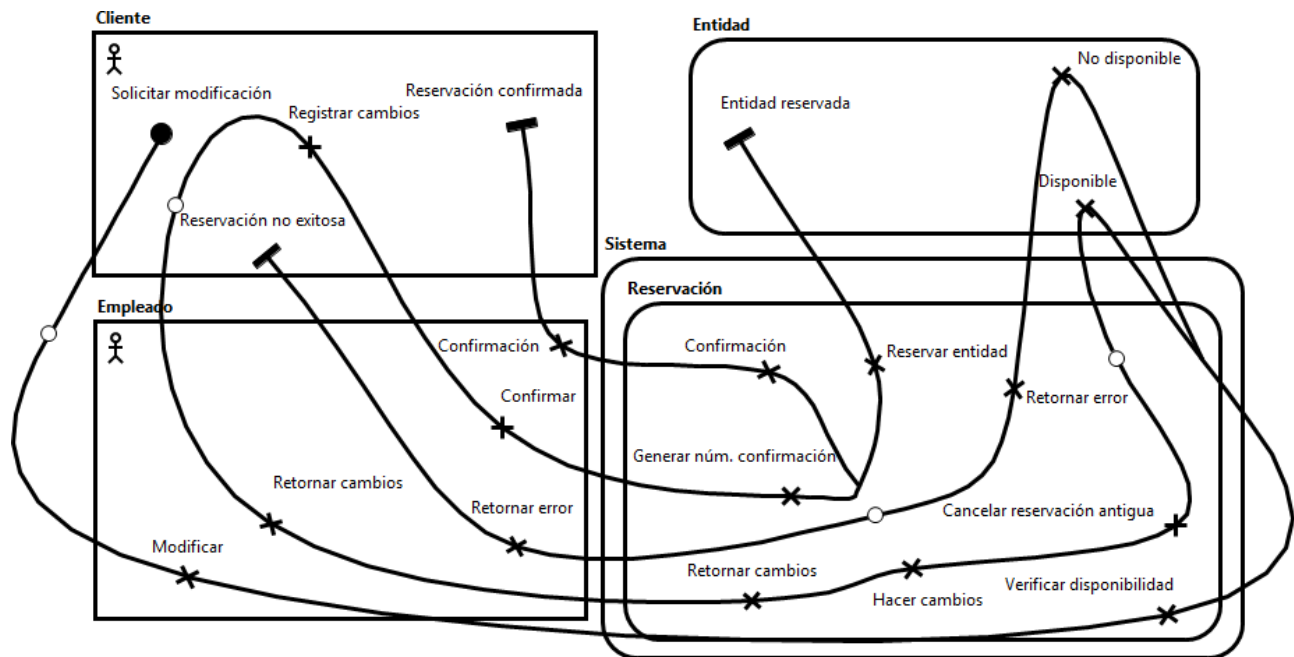


Figura 6.7 Diagrama UCM del requerimiento 3. Modificar una reservación

6.1.2.4 Requerimiento 4. Cancelar una reservación

Los siguientes diagramas GRL y UCM describen las metas y el comportamiento del requerimiento cancelar una reservación.

6.1.2.4.1 Representación jerárquica:

4. Cancelar una reservación

4.1 Cancelar reservación existente: Proceso de cancelar una reservación previamente solicitada por un cliente.

4.1.1 Definir consecuencias: Actividad que consiste en detectar las consecuencias negativas o positivas para el negocio.

4.1.1.1 Aplicar políticas: Medidas que se aplican según sea el caso del negocio para determinar los efectos de la cancelación, como encuestas a los clientes, etc.

6.1.2.4.2 Diagrama GRL:

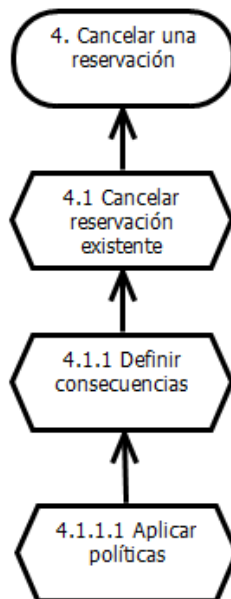


Figura 6.8 Diagrama GRL del requerimiento 4. Cancelar una reservación

6.1.2.4.3 Tabla de responsabilidades:

Tabla 6.4 Actores objetos y responsabilidades del requerimiento 4. Cancelar una reservación

Actores	Responsabilidades
Cliente	Solicitar cancelación, Responder
Empleado	Aplicar políticas, Definir consecuencias, Cancelar
Objetos	
Reservación	Retornar disponibilidad
Entidad	Disponible

6.1.2.4.4 Diagrama UCM:

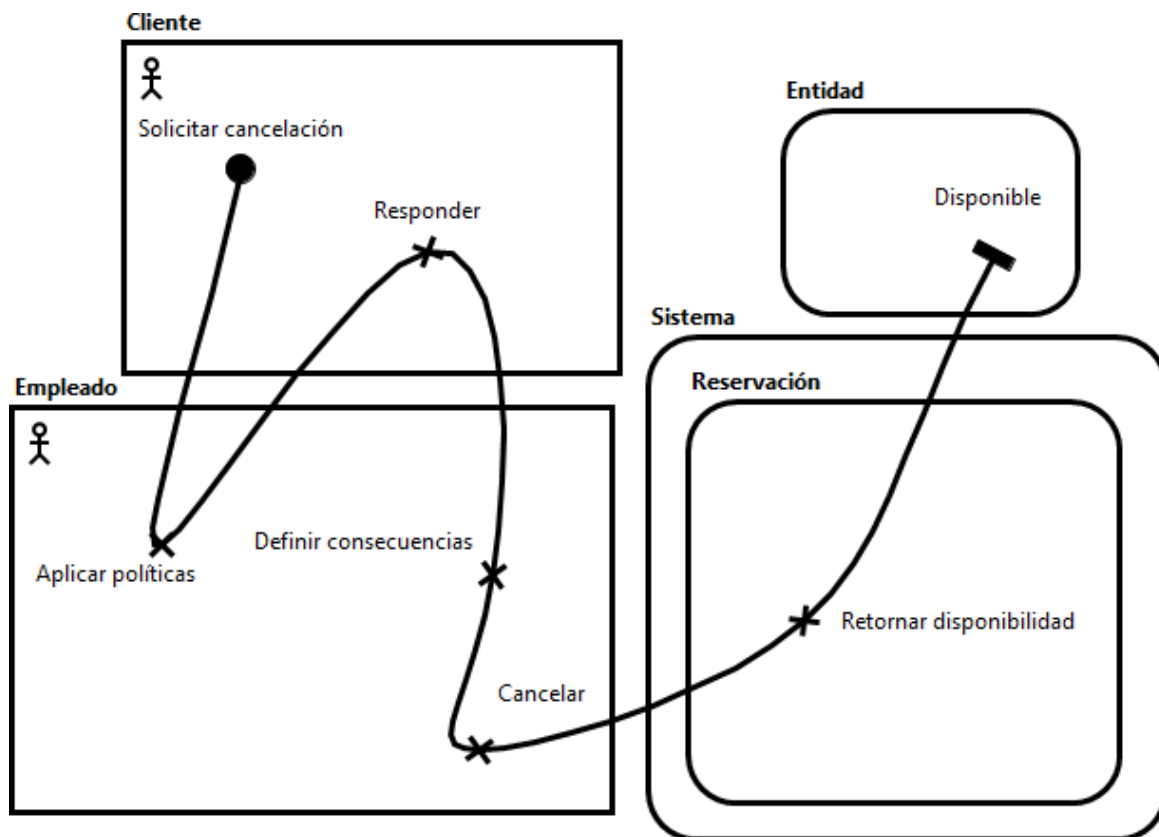


Figura 6.9 Diagrama UCM del requerimiento 4. Cancelar una reservación

6.1.2.5 Requerimiento 5. Crear un contrato

Los siguientes diagramas GRL y UCM describen las metas y el comportamiento del requerimiento crear un contrato.

6.1.2.5.1 Representación jerárquica:

5. Crear un contrato

5.1 Reservar una entidad: Actividad que corresponde al requerimiento 1.

5.1.1 Crear documento reservación: Esta actividad se lleva a cabo después de crearse una reservación, y consiste en crear un documento que respalde la transacción realizada por ambas partes.

6.1.2.5.2 Diagrama GRL:



Figura 6.10 Diagrama GRL del requerimiento 5. Crear un contrato

6.1.2.5.3 Tabla de responsabilidades:

Tabla 6.5 Actores objetos y responsabilidades del requerimiento 5. Crear un contrato

Actores	Responsabilidades
Cliente	Contrato recibido
Empleado	Imprimir contrato, Entregar contrato
Objetos	
Registro Uso	Generar contrato, contrato

6.1.2.5.4 Diagrama UCM:

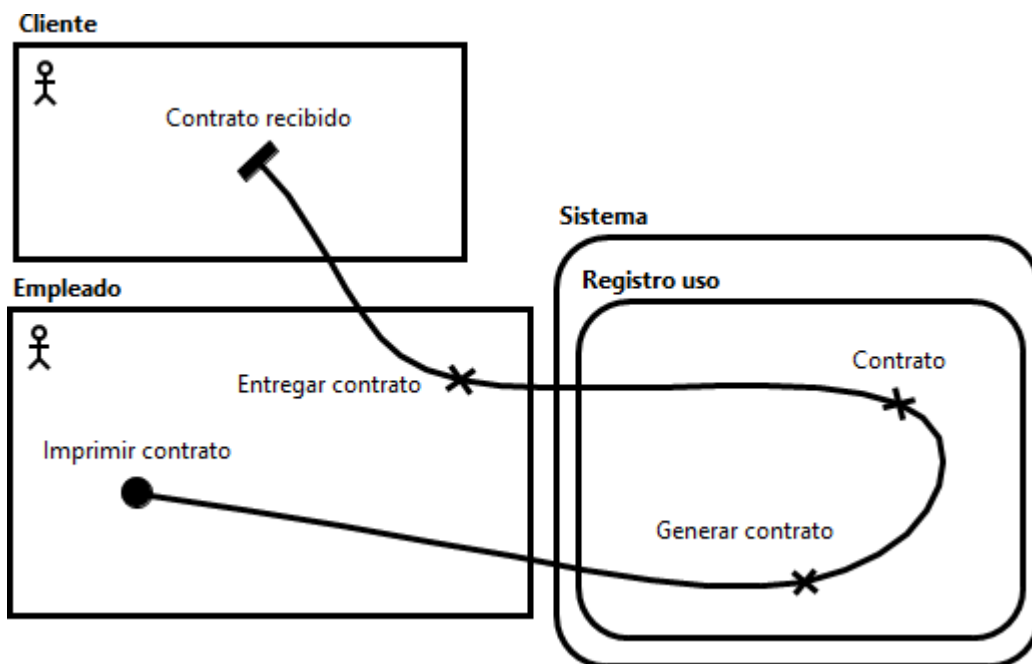


Figura 6.11 Diagrama UCM del requerimiento 5. Crear un contrato

Tanto para el diseño de los diagramas GRL como para los diagramas UCM se utilizó el ambiente de desarrollo Eclipse con el plugin jUCMNav v5.5.0, el cual se encuentra disponible en:

<http://jucmnav.softwareengineering.ca/ucm/bin/view/ProjetSEG/JUCMNavRelease550>

Los pasos de instalación se pueden consultar en:

<http://jucmnav.softwareengineering.ca/ucm/bin/view/ProjetSEG/DownloadingAndInstallation>

6.1.2.6 Requerimientos funcionales

La Tabla 6.6 muestra las dependencias y prioridades de los cinco requerimientos funcionales identificados del patrón de análisis para reservación y uso de entidades reutilizables, la columna número 1 muestra una lista de los requerimientos, la columna número 2 muestra si el requerimiento en cuestión, depende de otro(s) requerimiento(s) para su realización, y por último, la columna número 3 indica si la prioridad para el requerimiento en cuestión, es alta, media o baja.

Tabla 6.6 Tabla de dependencias y prioridades

Requerimiento (R)	Dependencias	Prioridad
1. Hacer una reservación		Alta
2. Usar una entidad reservada	R1, R5	Baja
3. Modificar una reservación	R1, R4	Baja
4. Cancelar una reservación	R1	Media
5. Crear un contrato	R1	Media

6.1.3 Determinación del ámbito de la ontología (Especificación)

Este apartado corresponde a la tercera fase de MECOPA (sección 5.2.3). Aquí se definen los usuarios, el dominio, el propósito y el tipo de ontología, así como el cuestionario de competencias:

6.1.3.1 Usuarios

Los usuarios finales previstos para la OPARE son:

- Responsables de análisis del dominio de reservación y uso de entidades reutilizables.
- Desarrolladores de aplicaciones para patrones de análisis.
- Investigadores y expertos del dominio, para la retroalimentación y mejora continua.

6.1.3.2 Dominio

La OPARE se desarrolló en el dominio de la Reservación y uso de entidades reutilizables. Cuando se habla de una entidad reutilizable nos referimos a aquellas que no son adquiridas por el cliente, él solamente tiene el derecho para usarlas por un cierto periodo de tiempo. Ejemplos de entidades reutilizables podrían ser habitaciones de hotel, asientos para un concierto y/o evento deportivo, películas, o en el caso ejemplo que se planteó al inicio de este capítulo, “libros”, etc.

6.1.3.3 Propósito

El propósito fundamental de la OPARE es representar, organizar, formalizar y estandarizar el conocimiento relacionado con el dominio del patrón de análisis para reservación y uso de entidades reutilizables.

6.1.3.4 Tipo de ontología

La OPARE es una ontología de dominio.

6.1.3.5 Cuestionarios de competencias

Para la elaboración del cuestionario de competencias se tomaron las consideraciones expuestas al final de la sección 5.2.3 y se optó por elegir el método “middle-out”⁴.

A continuación se muestran los requerimientos que deberá satisfacer la OPARE, la cual será capaz de contestar el siguiente cuestionario:

1. ¿Quiénes pueden ser clientes? **R:** Aquéllas personas que estén interesados en reservar una o más entidades.
2. ¿Qué actores están involucrados en una reservación? **R:** Cliente y empleado.

⁴ Middle-out se refiere a una técnica para la elaboración de cuestionarios de competencias. Se explica a más detalle en la sección 5.2.3 del Capítulo 5.

3. ¿Qué datos se necesitan para hacer una reservación? **R:** fecha de reservación, fecha de entrega, no. de confirmación y tipo de pago.
4. ¿Dónde se registra (consulta) la disponibilidad de una entidad? **R:** En el sistema.
5. ¿Qué datos se necesitan para llevar a cabo un registro de uso? **R:** Que la entidad a reservar esté disponible.
6. ¿Cuál es la diferencia entre registro de uso y contrato? **R:** El registro de uso es un documento informático y el contrato es un documento físico.
7. ¿Cuál es la diferencia entre reservación y registro de uso? **R:** En que el registro de uso se crea en el momento que el cliente solicita una entidad, antes de que la reservación se haya realizado
8. ¿Qué se requiere para cancelar una reservación? **R:** Que acuda el cliente que solicitó la reservación.
9. ¿Cuál es la diferencia entre reservar una habitación de un hotel y un boleto para un concierto? **R:** En que al reservar una habitación, ésta es asignada al cliente en el momento de uso, mientras que un asiento se asigna en el momento de la reservación.
10. ¿Qué representa un boleto de avión? **a) un contrato**, b) una reservación o c) un registro de uso

6.1.4 Búsqueda de ontologías relacionadas (Integración)

A continuación se muestra un análisis de las ontologías que proponen otros autores en temas relacionados a la reservación o alquiler de algún tipo de entidad.

Ontología de itinerario de viajes (Travel itinerary ontology)

Esta ontología hace una representación de conocimiento de todo lo relacionado al proceso de reservar un viaje, todo lo que conlleva, incluye clases como: Vuelo, reservación de hotel, itinerario, alquiler de auto, etc. La podemos consultar en:

<http://www.daml.org/ontologies/178>

Ontología de reservación de viajes (Trip Reservation Ontology)

Esta ontología funge como componente de una red de ontologías denominada: Ontologías de modelado de servicios web (WSMO: Web Service Modeling Ontologies) (Feier et al. 2005), dichas ontologías se distinguen por proveer de semántica a toda la información utilizada por cada uno de los actores implicados en el uso de servicios web. La ontología de reservación de viajes es una ontología que se enfoca en el servicio web de reservación de boletos por Internet y se encarga de definir la terminología necesaria para la descripción de la información relacionada a viajes y reservaciones. Incluye clases como: Viaje, Ticket, Reservación, etc. La especificación de WSMO la podemos consultar en:

<http://www.w3.org/Submission/WSMO-primer/#S41>

Ontología del patrón de alquiler de recursos

Esta ontología viene como ejemplo de construcción de ontologías paso a paso en (Hamza 2005), utilizando una metodología dirigida por modelos, partiendo del patrón de alquiler de recursos, hasta llegar a la representación ontológica. Contempla los aspectos de estabilidad de software (EBTs, BOs e IOs) e incluye clases como: Recurso, Cliente, Recibo, Reservación de recurso, Tipo de pago, Instancia de Recurso, Tipo de recurso, entre otras.

En la Tabla 6.7 se hace una comparativa de las tres ontologías enfocadas a la reservación de entidades. En la columna 3 de la tabla se muestran los elementos (o clases) comunes con la ontología OPARE. Tanto la ontología 1 como la ontología 2 de la Tabla 6.7 se dedican a la reservación de viajes, sin embargo la número 1 tiene un mayor alcance, ya que también incorpora reservación de autos y de hoteles. La ontología 3 es más general, no se enfoca en un tipo de reservación específico, sino que habla de una manera global. Esta última es la que más se asemeja a OPARE. En la columna 4 se indica si la ontología en cuestión maneja los conceptos del Modelo de Estabilidad de Software (SSM, por sus siglas en inglés). Finalmente en la columna 5 se indica el tipo de entidad que contempla cada ontología.

Tabla 6.7 Comparativa de Ontologías

No.	Ontología	Clases comunes a OPARE	SSM	Entidad
1	Ontología de itinerario de viajes	Reservación	No	Asiento de avión, Auto, Habitación de hotel
2	Ontología de reservación de viajes	Ticket, Reservación	No	Asiento de avión
3	Ontología del patrón de alquiler de recursos	Recurso, Cliente, Recibo, Reservación de recurso	Si	Recurso general

6.1.5 Identificación de términos del dominio (Conceptualización)

6.1.5.1 Clases

Una clase es un término que se utiliza para definir una categoría que incluye un conjunto de objetos (individuos) con características particulares, o incluso una categoría que englobe a más categorías. De acuerdo al ejemplo utilizado en este capítulo, una clase podría ser Cliente. Cliente es una categoría que puede englobar más categorías como alumnos y profesores, de las cuales, a su vez puede haber alumnos de computación, alumnos de mecánica, y de igual forma profesores.

Las clases identificadas se presentan a continuación:

1. Cliente
2. Empleado
3. Entidad
4. Reservación
5. Establecimiento
6. Contrato
7. Propietario

6.1.5.2 Propiedades

Una propiedad básicamente es una característica que posee un individuo y que lo hace distinguirse de los demás. Aunque en un sentido estricto es una relación entre un individuo de una clase y un tipo de dato del esquema XML. Por ejemplo el alumno de CENIDET Vitervo (Individuo de tipo Cliente) se relaciona mediante la propiedad “dirección” al tipo de dato “xsd:string” con valor “Apatzingán #13, col. Lázaro Cárdenas”.

Las propiedades identificadas para cada clase de la sección anterior son las siguientes:

1. Cliente: Nombre, dirección, teléfono, comportamiento, especificidad, nivel_estabilidad, percepción
2. Empleado: Nombre, idEmpleado, comportamiento, especificidad, nivel_estabilidad, percepción
3. Entidad: Nombre, idEntidad, Disponibilidad, Precio, comportamiento, especificidad, nivel_estabilidad, percepción
4. Reservación: FechaInicio, fechaFin, noConfirmacion, tipoPago, comportamiento, especificidad, nivel_estabilidad, percepción
5. Establecimiento: Nombre, giro, comportamiento, especificidad, nivel_estabilidad, percepción
6. Contrato: Tipo, folio, comportamiento, especificidad, nivel_estabilidad, percepción
7. Propietario: Nombre, RFC, comportamiento, especificidad, nivel_estabilidad, percepción

6.1.6 Especificación de conceptos y relaciones (Formalización)

6.1.6.1 Conceptos

En el dominio de las ontologías, un concepto es sinónimo de clase. En este apartado se lleva a cabo la especificación de las clases encontradas en la sección 6.1.5.1:

- Contrato
- Entidad
- Establecimiento
- Persona
 - a. Cliente
 - b. Empleado
- Propietario
- Reservación

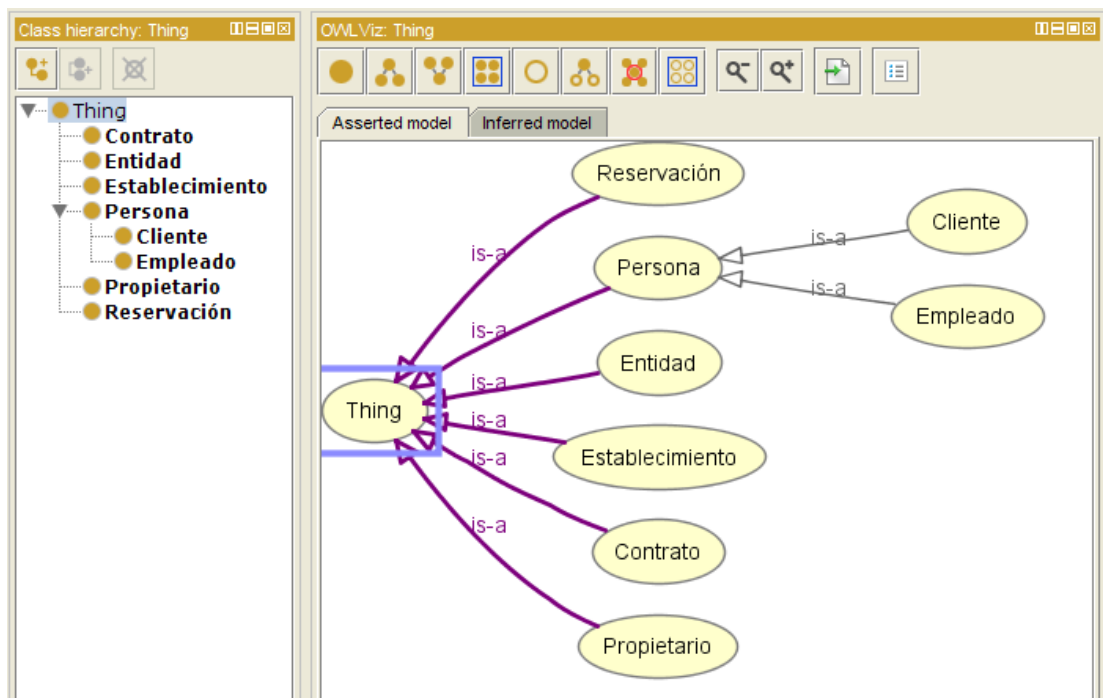


Figura 6.12 Representación de clases de OPARE en Protégé

6.1.6.1.1 Código OWL

A continuación se muestra el código OWL generado a partir del ingreso de las clases: “Cliente” y “Persona” en la herramienta Protégé:

```
<Declaration>  
  <Class IRI="#Cliente"/>  
</Declaration>  
  .  
  .
```

```

    .
    <Declaration>
      <Class IRI="#Persona"/>
    </Declaration>
    .
    .
    .
    <SubClassOf>
      <Class IRI="#Cliente"/>
      <Class IRI="#Persona"/>
    </SubClassOf>

```

6.1.6.2 Relaciones

Una relación representa la interacción y enlace entre dos individuos, regularmente se expresa como una acción que lleva a cabo un individuo del dominio (sujeto de la relación) y que para su realización se requiere de otro individuo del rango (objeto de la relación).

En el caso de ejemplo de la reservación de libros en CENIDET una relación podría ser: el alumno Omar (Individuo de la clase Cliente) “reserva” el libro “Autómatas” (Individuo de la clase “Entidad”).

Tabla 6.8 Relaciones entre clases de la ontología OPARE

Dominio	Relación	Rango	Tipo
Cliente	adquiere	Contrato	Inversa funcional
Empleado	crea	Reservación	N/A
Contrato	esAdquiridoPor	Cliente	Funcional
Reservación	esCreadaPor	Empleado	N/A
Propietario	esDueñoDe	Entidad	Inversa funcional
Contrato	esExpedidoPor	Empleado	Funcional
Entidad	esReservadaPor	Cliente	N/A
Entidad	esVerificadaPor	Empleado	N/A
Empleado	expide	Contrato	Inversa funcional
Empleado	laboraEn	Establecimiento	Funcional
Entidad	perteneceA	Propietario	Funcional
Cliente	reserva	Entidad	N/A
Empleado	verificaDisponibilidadDe	Entidad	N/A
Contrato	tieneDatosDe	Empleado Cliente Propietario Reservación Entidad Establecimiento	N/A

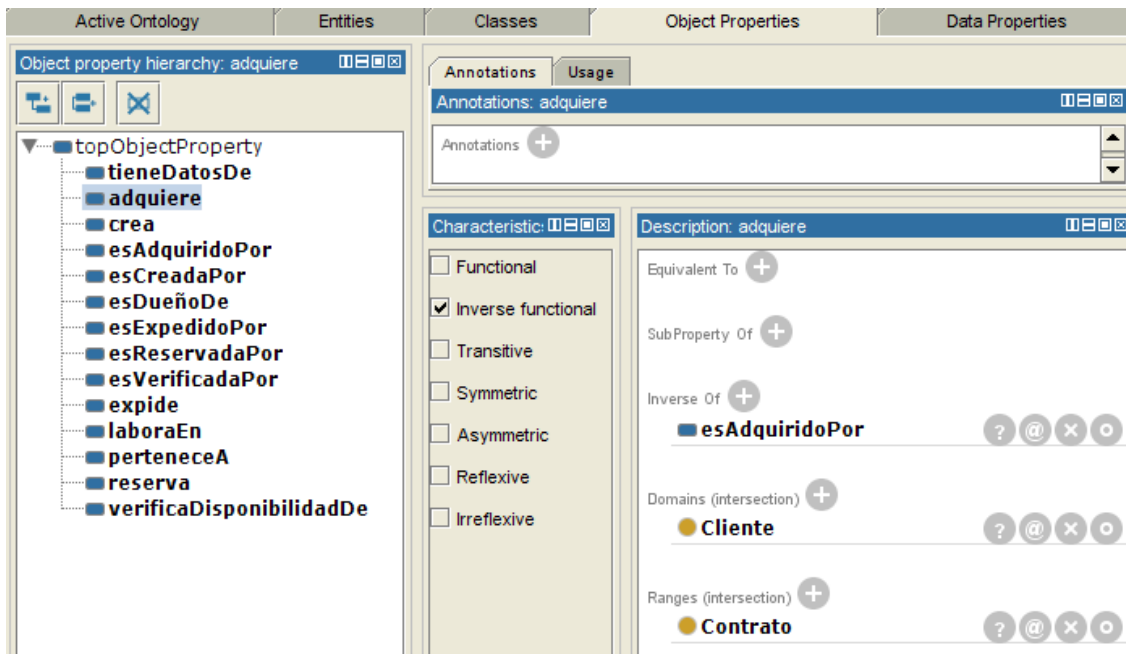


Figura 6.13 Representación de relaciones de OPARE en Protégé

6.1.6.2.1 Código OWL

A continuación se muestra el código OWL generado a partir del ingreso de la relación: “adquiere”, así como de su respectivo dominio (Cliente) y rango (Contrato), en la herramienta Protégé:

```

<Declaration>
  <ObjectProperty IRI="#adquiere"/>
</Declaration>
.
.
.
<SubClassOf>
  <Class IRI="#Cliente"/>
  <ObjectSomeValuesFrom>
    <ObjectProperty IRI="#adquiere"/>
    <Class IRI="#Contrato"/>
  </ObjectSomeValuesFrom>
</SubClassOf>

```

6.1.6.3 Propiedades

En este punto se acomodan las propiedades identificadas en 6.1.5.2, mediante una jerarquía, tal cual serán definidas en la especificación con la herramienta Protégé.

- Contrato: tipo, folio, noReservación, comportamiento, especificidad, nivel_estabilidad, percepción
- Entidad: nombre, idEntidad, disponibilidad, Precio, comportamiento, especificidad, nivel_estabilidad, percepción
- Establecimiento: nombre, giro, comportamiento, especificidad, nivel_estabilidad, percepción
- Persona: nombre, comportamiento, especificidad, nivel_estabilidad, percepción
 - a. Cliente: dirección, teléfono
 - b. Empleado: idEmpleado
- Propietario: nombre, RFC, comportamiento, especificidad, nivel_estabilidad, percepción
- Reservación: fechaInicio, fechaFin, noConfirmación, tipoPago, comportamiento, especificidad, nivel_estabilidad, percepción

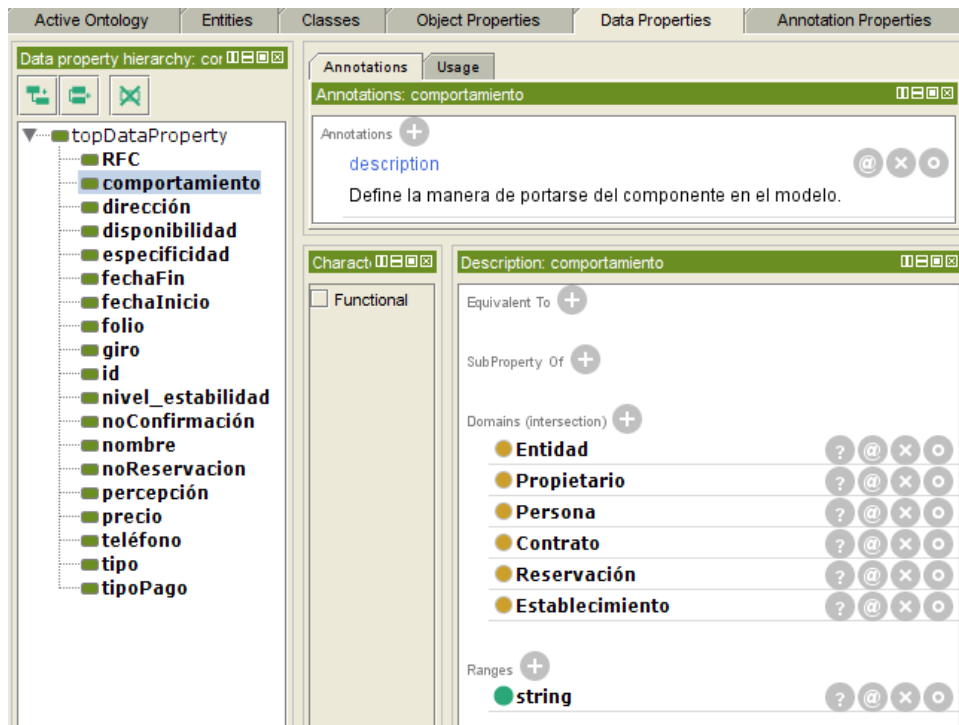


Figura 6.14 Representación de propiedades de OPARE en Protégé

6.1.6.3.1 Código OWL

A continuación se muestra el código OWL generado a partir del ingreso de la propiedad: “RFC”, así como de su respectivo dominio (Propietario) y rango (xsd:string), en la herramienta Protégé:

```
<Declaration>
  <DataProperty IRI="#RFC"/>
</Declaration>
.
.
.
<DataPropertyDomain>
  <DataProperty IRI="#RFC"/>
  <Class IRI="#Propietario"/>
</DataPropertyDomain>
.
.
.
<DataPropertyRange>
  <DataProperty IRI="#RFC"/>
  <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>
```

Cabe señalar, que tanto para las clases, relaciones y propiedades, sólo se muestran fragmentos de código a manera de ejemplo, el código completo de la ontología OPARE se encuentra en el archivo “ontologíaOPARE.owl”, adjunto a este documento de tesis.

6.1.7 Definición de restricciones

Las restricciones son aquellos requisitos que se deben cumplir para que un individuo pueda ser considerado individuo de su clase. De acuerdo al caso ejemplo del centro de información de CENIDET una restricción podría ser: “que para ser cliente o usuario activo de la biblioteca, se necesita reservar al menos un libro (entidad):

En esta fase se definieron dos tipos de restricciones:

6.1.7.1 Restricciones existenciales

Cliente adquiere *some* Contrato

Cliente reserva *some* Entidad

Propietario esDueñoDe *some* Entidad

6.1.7.2 Restricciones universales

Contrato esAdquiridoPor *only* Cliente

Contrato esExpedidoPor *only* Empleado

Empleado laboraEn *only* Establecimiento

Entidad perteneceA *only* Propietario

En la Figura 6.15 se muestran las restricciones de la clase Cliente:

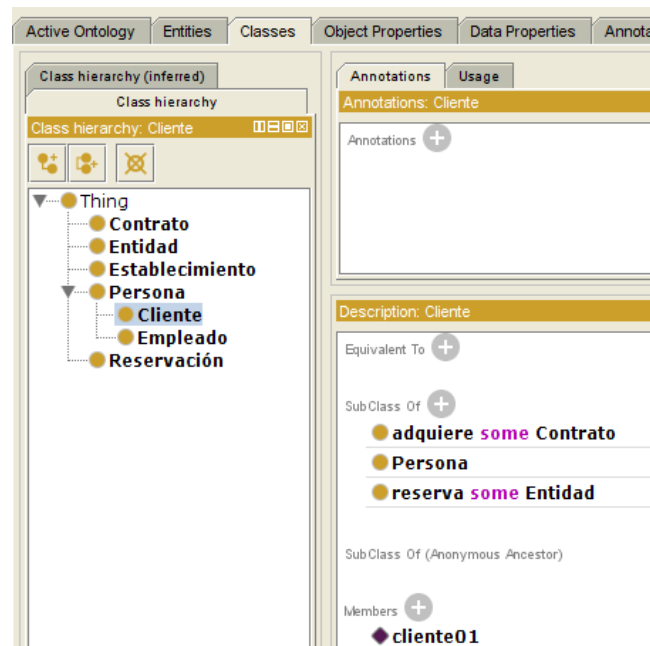


Figura 6.15 Restricciones de clases de OPARE en Protégé

6.1.8 Instanciación de la ontología

Una instancia o individuo de la ontología es el hecho de “ponerle nombre” a cada una de las clases, es decir si se habla de un libro, especificar exactamente de qué libro se trata, qué título tiene, qué tipo de entidad es, qué identificador tiene, etc.

En este paso se crearon los individuos de la ontología. Y consistió en poblar la ontología, asignando valores a cada una de las propiedades de las clases (ver Figura 6.16).

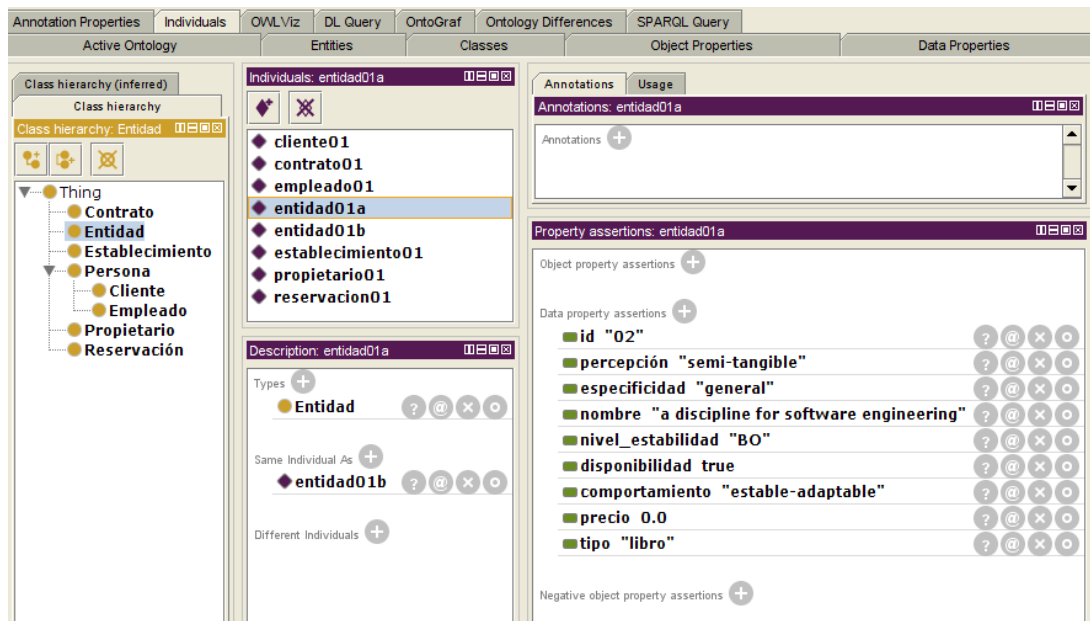


Figura 6.16 Individuos de clases de OPARE en Protégé

6.1.9 Visualización de la ontología

La ontología completa se visualiza en la Figura 6.17 que se muestra a continuación:

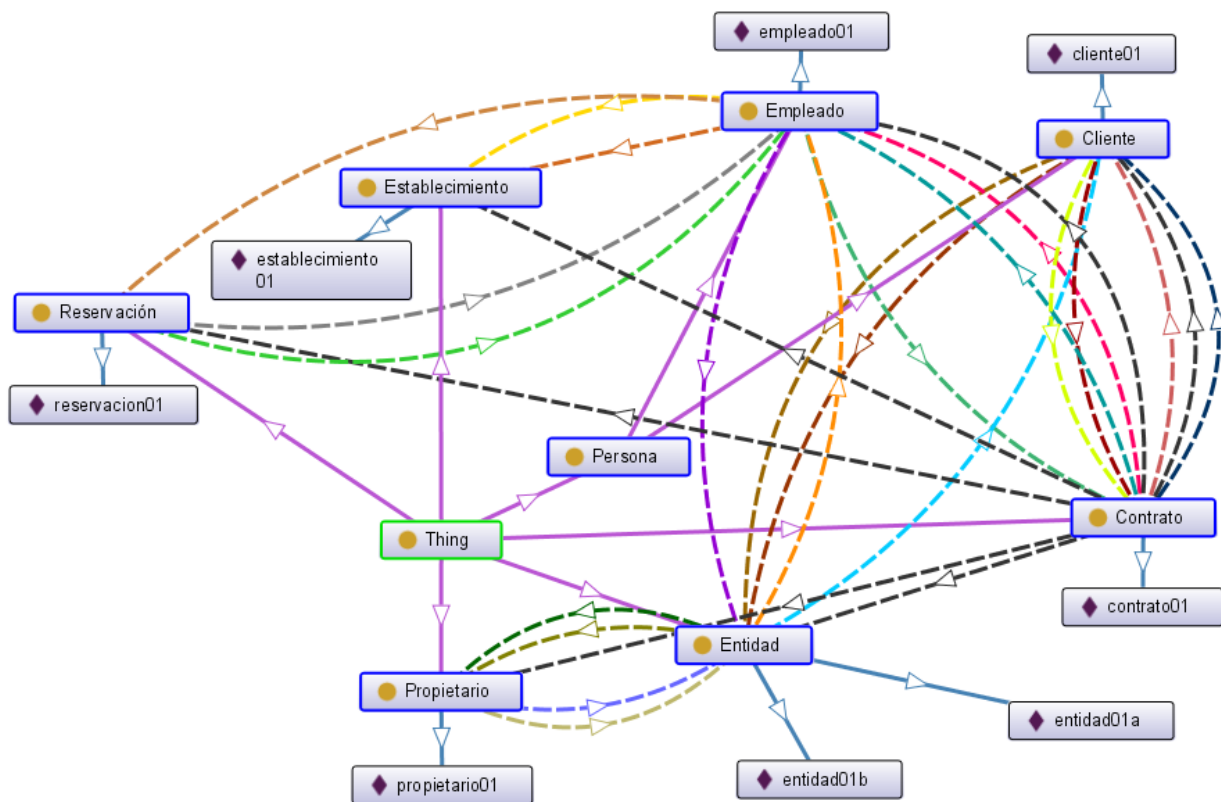


Figura 6.17 Ontología OPARE

En este capítulo se presentó toda la documentación necesaria y suficiente del proceso de creación de la ontología OPARE, como lo marca la metodología MECOPA. Esta documentación sirvió para reforzar el entendimiento de dicha metodología, fungiendo como un ejemplo directo de su utilización. Hasta este punto hemos cumplido con los puntos 5, 6, 7 y 8 de la estructura presentada en el Capítulo 4, por lo que podemos empezar a formular los requerimientos de un sistema computacional para el almacenamiento de dicha estructura, los cuales se especifican en el siguiente capítulo (Capítulo 7).

Capítulo 7

Especificación de

Requerimientos del Sistema

AUPAs

En el presente capítulo se presenta la documentación de los requerimientos del sistema de Almacenaje y Utilización de Patrones de Análisis. Este capítulo es un resumen del documento “SRS Sistema AUPAs 1.1.docx”. Adjunto a este documento de tesis, el cual se realizó tomando como referencia el estándar IEEE Std830. De este documento se detallan algunos puntos en este capítulo, como: descripción general, perspectiva, funciones y restricciones del producto, así como una especificación visual de los requerimientos, utilizando diagramas GRL y UCM además de una especiación textual en forma de lista.

7.1 Especificación de requerimientos

En este capítulo se describe la especificación de requerimientos para un sistema de almacenaje y utilización de patrones de análisis (AUPAs) reportada en el documento SRS Sistema AUPAs 1.1.docx adjunto a esta tesis.

El propósito de esta Especificación de Requerimientos es reunir en un documento los requerimientos funcionales y no funcionales del sistema de Almacenaje y Utilización de Patrones de Análisis (AUPAs). En el documento se describen los servicios que ha de ofrecer el sistema y las restricciones asociadas a su funcionamiento. Esto con la finalidad de ayudar en la selección de componentes que permitan crear la infraestructura, que dé soporte a un Repositorio de Ontologías de Patrones de Análisis, así como al sistema de software al que estará ligado dicho repositorio.

Esta especificación de requerimientos está dirigida al diseñador y/o desarrollador del sistema AUPAs. En la especificación, se describen las necesidades de los usuarios que deseen realizar algún desarrollo de software y estén interesados en aprovechar las ventajas que ofrecen los patrones de análisis.

7.2 Descripción general

El sistema de almacenamiento y uso de patrones de análisis asistirá al desarrollador de software, pues facilita la reutilización de PAs, ya que proporciona el soporte necesario para facilitar la etapa de análisis de un nuevo proyecto. Esta asistencia permitirá: reducir el tiempo de desarrollo⁵, así como minimizar el esfuerzo que se requiere elaborar el análisis si se partiera desde cero, además de facilitar la transformación de un modelo de análisis hacia un modelo de diseño, así como la obtención de un análisis más completo, etc.

El usuario podrá plantear al sistema AUPAs, un problema específico y consultar de un catálogo de PAs algún patrón de análisis genérico o abstracto, al cual se pueda ajustar la situación específica. (Ejemplo: reservación de entidades), y el sistema mostrará aquellos patrones relacionados al problema en cuestión, con la posibilidad de mostrar el patrón (individuo de la OPPAE), la ontología del patrón, sus individuos (casos de estudio particulares como el alquiler de bienes raíces en la empresa “Dream home”) y las posibles ontologías de instancias de patrones (ontologías de

⁵ Andreas Geyer (Geyer-schulz & M 2001) realizó un estudio comparativo de la etapa de análisis utilizando patrones de análisis versus su no utilización, donde presenta evidencia suficiente para soportar esta afirmación.

casos de estudio particulares).

Toda esta interacción entre el sistema y el usuario, será gráfica y amigable orientando al usuario en todo momento acerca de qué es lo que puede visualizar o consultar. El sistema deberá dar respuesta a las peticiones de una manera oportuna y fluida.

Todos los diagramas que presente el sistema podrán importarse y exportarse en formato XML desde y hacia otras herramientas de modelado, si es que así lo desea el usuario.

Para usuarios avanzados el sistema también ofrecerá la capacidad de ingresar sus propias ontologías. Una vez almacenada la ontología, en un repositorio temporal, se podrá realizar un procedimiento de evaluación y aprobación para incluirlas en el repositorio final y puedan formar parte del catálogo de PAs.

En la Figura 7.1 se muestra, a grandes rasgos, como se conformaría el sistema por tres elementos principales: 1) una interfaz de usuario (GUI) a la cual tengan acceso los usuarios, que podrán crear cuatro tipos de cuentas: lector, autor, evaluador y administrador, los cuales podrán hacer consultas, ingresar ontologías, evaluar ontologías, aprobar algunos tipos de cuentas, respectivamente; 2) el traductor OWL que será el encargado de transformar el código OWL, de las ontologías que se ingresen, en un modelo de base de datos relacional y de esta forma poder almacenar tanto la ontología como sus individuos; y 3) el repositorio de ontologías, donde se encuentra el catálogo de PAs y todas sus ontologías que ya han sido evaluadas y aprobadas.

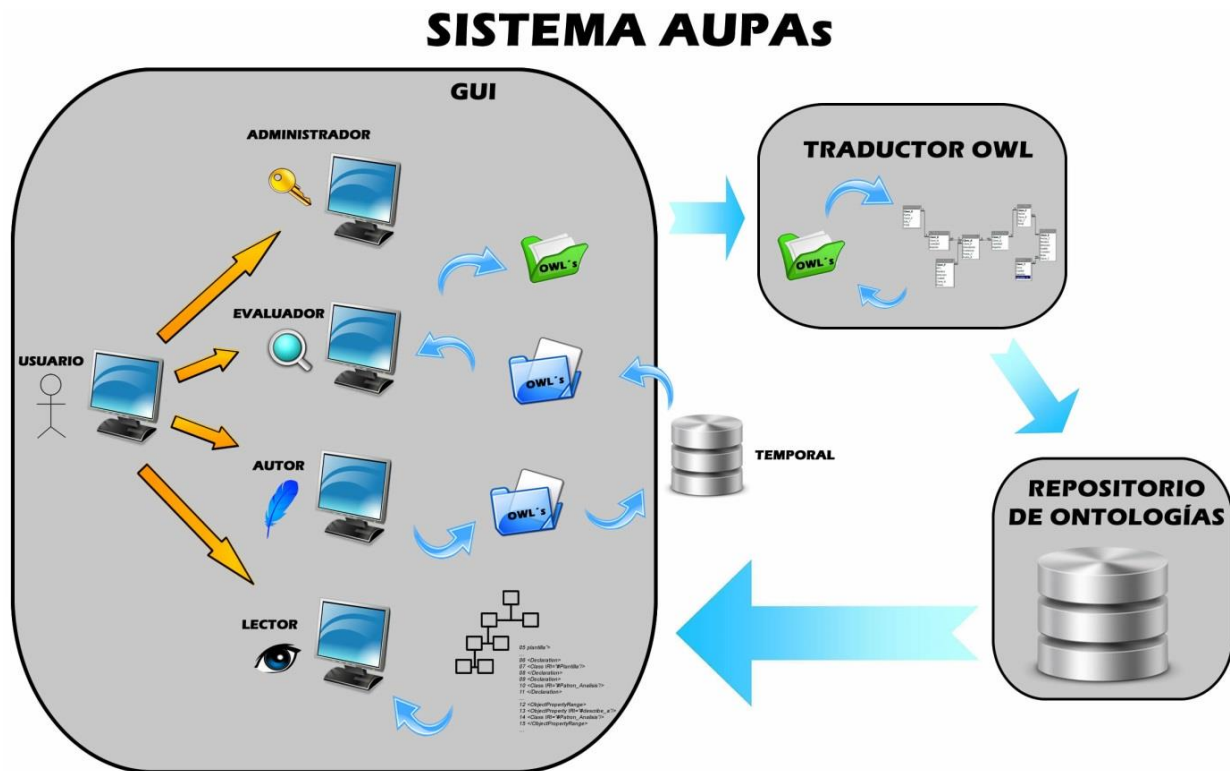


Figura 7.1 Sistema AUPAs

Para la especificación de requerimientos se utilizan dos lenguajes de URN (ver secciones 2.7 y 2.8), basados en la recomendación de la (ITU-T 2012). La notación URN incluye conceptos para la especificación de objetivos, requisitos no funcionales, fundamentos, indicadores, comportamientos, escenarios y estructuración. Esto mediante los dos lenguajes de URN:

- GRL
- UCM

Según [Amyot, 2003], los componentes UCM podrían también ser el punto de enlace entre los escenarios de URN y los conceptos del Lenguaje de Modelado Unificado (Diagramas UML, como los diagramas de casos de uso, de secuencia, etc.). Las relaciones de alto nivel que existen entre GRL, UCM, otros lenguajes ITU-T (como SDL⁶, eODL⁷ o MSC⁸), y UML se resumen en la Figura 7.2. URN representa la pieza

⁶ Lenguaje de Especificación y Descripción (SDL, por sus siglas en inglés).

⁷ Lenguaje de Descripción de Componentes.

⁸ Diagramas de Secuencia de Mensajes (MSC, por sus siglas en inglés)

faltante del rompecabezas de modelado que conecta los requisitos informales y los casos de uso (color amarillo) con los modelos estructurales (color azul) y de comportamiento (color azul turquesa).

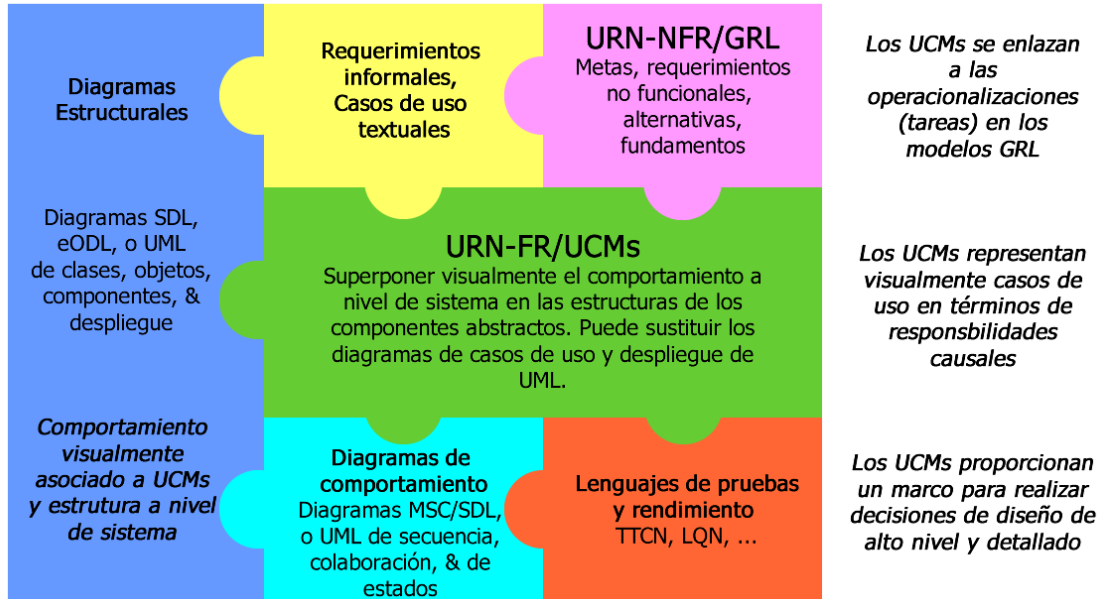


Figura 7.2. GRL, UCM, y otras piezas de modelado (Amyot 2003).

La realización del documento de especificación de requerimientos está basada en los lineamientos del estándar IEEE Std830 (IEEE 1998) que especifica el contenido, su estructura y sugerencias acerca de calidad de la Especificación de Requerimientos de Software (SRS).

Las secciones siguientes (7.3 a 7.12) son las elegidas del estándar IEEE Std830, para incluirlas en nuestro documento de especificación de requerimientos.

7.3 Perspectiva del producto

El sistema AUPAs será independiente, es decir, no dependerá de otro más grande.

El sistema AUPAs deberá almacenar patrones de análisis y sus ontologías, que sean útiles al proceso de análisis del desarrollo de software de acuerdo a los requerimientos presentados en la sección 7.9.

7.3.1 Interfaces del sistema.

- El sistema AUPAs no interactuará con otros sistemas.

7.3.2 Interfaces de usuario

- El sistema AUPAs será en forma visual, por lo que el usuario interactuará con el sistema mediante el ingreso de datos por teclado, así como en la selección de las funciones (consultas, aportaciones, evaluaciones, etc.) detalladas en la sección 7.4, visualizándolas en un monitor.
- No deberán contener una cantidad excesiva de gráficos/videos que no tengan utilidad al sistema, con la finalidad de agilizar el proceso de navegación en el sistema.
- Para el color de fondo, letras, tamaño de letras y tipos de letras, se deberá tomar en cuenta la comodidad visual del usuario (colores y tamaños adecuados).

7.3.3 Interfaces de Hardware

- El sistema AUPAs ocupará una computadora que interactúe con éste. La cual deberá ser capaz de soportar la cantidad de PAs almacenados (ver sección 7.6), además de cumplir con los requisitos mostrados en la sección 7.3.8

7.3.4 Interfaces de software

- El sistema deberá interactuar con bases de datos, las cuáles contendrán a los patrones de análisis (descripciones textuales, diagramas XML) y a sus ontologías (en formato OWL).

7.3.5 Interfaces de comunicación

- Para acceder al sistema se requerirá tener conexión a Internet. Mediante una conexión Cliente-Servidor.

7.3.6 *Restricciones de memoria.*

Se deberá considerar la cantidad necesaria de memoria para la ejecución del sistema, así como del acceso al repositorio de patrones de análisis, esto dentro de la computadora en la que estará montado el sistema. Este punto se desglosa a más detalle en la sección 7.6.

7.3.7 *Funcionamiento.*

La mayoría de los procesos son controlados por el usuario, pero existen procesos que realizará el sistema, siempre considerando las decisiones que el usuario tome.

- Un nuevo usuario creará su cuenta mediante un procedimiento de registro, donde se le asignará un nombre de usuario y contraseña.
- Los usuarios deben identificarse (autenticarse) mediante la pantalla principal del sistema.
- El sistema validará al usuario y le mostrará las opciones disponibles.
- El usuario una vez autenticado, podrá hacer uso de las ontologías de patrones de análisis disponibles por medio del sistema.
- Si no hay actividad en una cuenta del sistema por más de 15 minutos la cuenta inactiva será cerrada.

7.3.8 *Requerimientos de adaptación del sitio.*

En el caso del sistema, se requiere de una computadora configurada como servidor, que permita alojar los programas necesarios para que el sistema funcione. Además de contar con un motor de base de datos, conexión a Internet y un navegador.

En el caso de las máquinas cliente, también deberán contar con navegador de Internet así como con conexión de red y una IP accesible para establecer un enlace con la máquina servidor y mediante este enlace poder utilizar los recursos del mismo. De igual forma las computadoras cliente deberán contar con espacio de memoria necesario para alojar una especie de repositorio local.

7.4 Funciones del producto.

En las secciones 7.9 y 7.10 de este documento se encuentran las tareas o metas del producto que fueron identificadas como necesarias para la elaboración de un sistema de almacenamiento y uso de patrones de análisis.

7.5 Características de usuario.

Para cualquier usuario que tenga conocimientos de patrones de análisis, ontologías o de desarrollo de software. Debido a que el sistema cuenta con 4 diferentes tipos de cuenta (lector, autor, evaluador y administrador), los usuarios lo podrán utilizar independientemente del grado de experiencia, ya sean estudiantes, desarrolladores o expertos.

7.6 Restricciones

El sistema deberá seguir el formato de plantilla especificado por la OPPAE (Martínez Jiménez 2013) para el uso y descripción de los patrones de análisis contenidos en el repositorio.

Al inicio, la computadora en donde se encontrará montado el sistema, deberá contar con la memoria y capacidad de procesamiento necesaria para soportar 2 ontologías y 2 individuos, sin embargo deberá ser diseñado con cierta flexibilidad que permita almacenar por lo menos:

- 1 ontología de plantilla de PAs,
- de 40 a 50 patrones de análisis, que corresponden a los individuos de la ontología del punto anterior,
- de 40 a 50 ontologías de PAs,
- de 3 a 5 individuos (instancias de PAs) por cada una de las ontologías del punto anterior, así como
- de 3 a 5 ontologías de instancias de PAs.

Tomando en cuenta que los diagramas y otros gráficos que contengan tanto las ontologías como los patrones de análisis se almacenarán en formato XMI, la memoria promedio que ocupa un PA puede ir desde 80 KB a 100 KB.

Asimismo, la memoria que ocupa una ontología promedio calculada de acuerdo a su código puede variar dependiendo del tipo de lenguaje con el que esté desarrollada, sin embargo en promedio ocupa de 20 KB a 40 KB.

El sistema deberá ser capaz de soportar consultas concurrentes.

El sistema deberá ser diseñado con una arquitectura tipo cliente-servidor.

7.7 Suposiciones y dependencias

De cierta forma se puede asumir que la máquina servidor en la que estará alojado el sistema, en caso de que se sigan los requisitos anteriormente descritos, trabajará de una forma adecuada, sin embargo las máquinas cliente dependen completamente de los usuarios, es por eso que es importante detallar factores que puedan afectar a los requerimientos establecidos en este documento dentro de estos factores están:

- **Software:** Debido a que el sistema será accedido de forma remota desde un navegador Web, no importará de qué sistema operativo se acceda, aunque de preferencia deberá ser uno relativamente actual, así como una versión de navegador de Internet no obsoleto.
- **Comunicación:** La velocidad con la que se desenvuelva el sistema dependerá de la velocidad de la conexión a Internet que tenga cada usuario.
- **Recursos:** También existe otra dependencia de los recursos con los que cuente el usuario del sistema, como lo puede ser capacidad de procesamiento, memorias etc., lo que puede también afectar en la velocidad del despliegue, consultas, etc.
- **Herramientas:** Para sacar el máximo provecho al sistema los usuarios deberán contar con herramientas de modelado a donde poder importar los diagramas exportados del sistema.

Especificación visual de requerimientos

7.8 Requerimientos funcionales (diagramas)

7.8.1 Sistema de Almacenamiento y uso de Patrones de Análisis

En los siguientes diagramas GRL y UCM se describen las metas del Sistema de Almacenamiento y uso de Patrones de Análisis, así como el comportamiento de éstas. Cada uno de los componentes que integran a los diagramas GRL tienen un identificador, el cual indica el orden de precedencia de las tareas o metas, ejemplo el número 1 podrá tener hijos 1.1, 1.2, etc., los cuales deberán cumplirse antes, para que se lleve a cabo la meta padre, de igual forma el 1.1 podrá tener más hijos 1.1.1, 1.1.2, etc., siempre siguiendo el orden de que se cumplan los elementos hijos, para que se puedan efectuar los padres. En las secciones 7.9, 7.10 y 7.11, se muestra una lista en forma jerárquica de todas las metas, tareas y recursos, con explicación de cada una.

se procede a Iniciar sesión, para después ingresar al área de aportaciones, luego se puede elegir de entre 2 opciones ya sea ingresar nueva ontología o nuevo individuo una vez realizado esto, pasa por un proceso de aprobación, y finalmente, si es que cumple con los requisitos de contenido consensual, formal y compartido, se procede a realizar la aportación en el catálogo de PAs.

7.8.1.2 Diagrama UCM general del sistema AUPAs

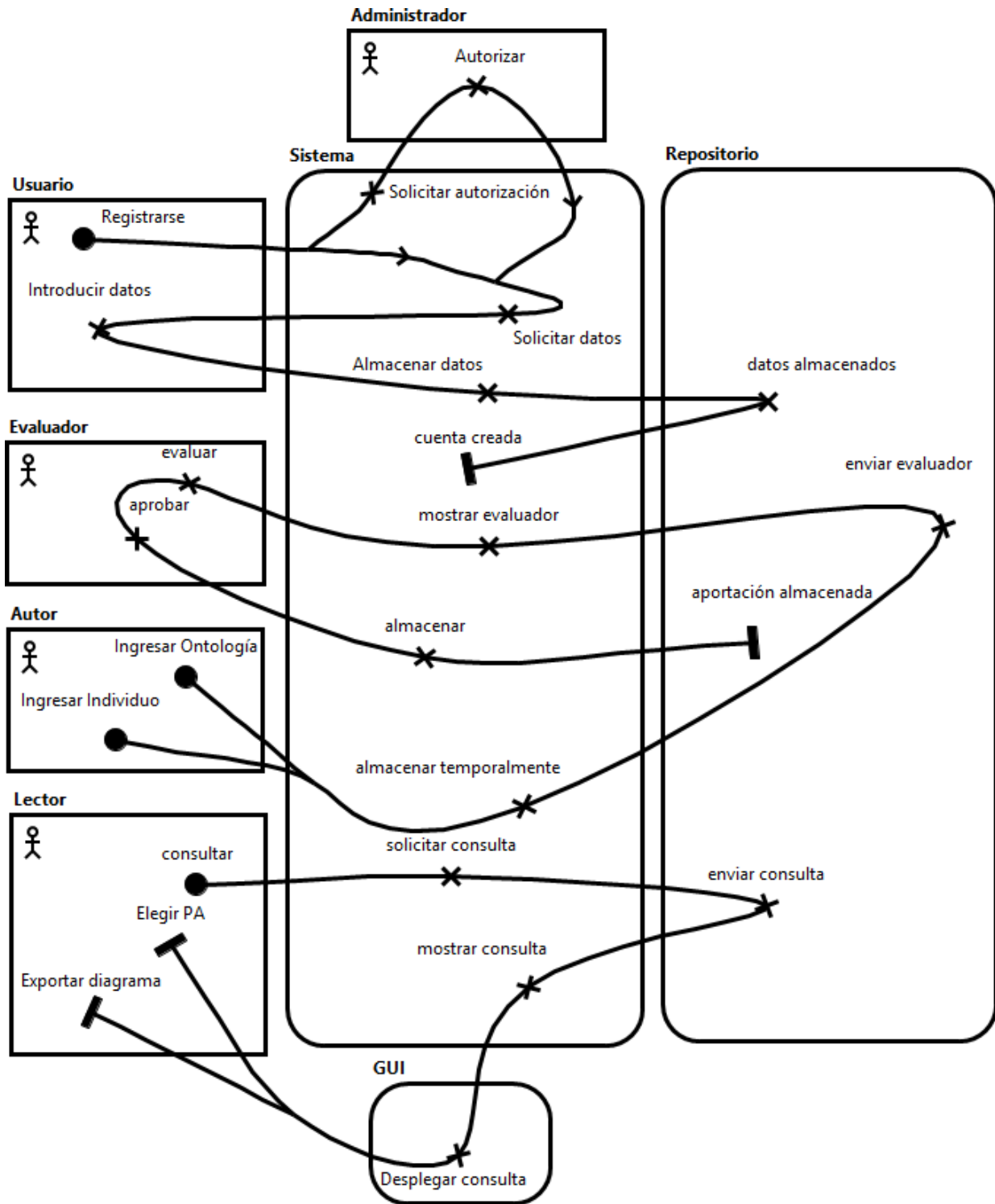


Figura 7.4 Diagrama UCM general del sistema AUPAs

7.8.1.3 Meta: Creación de una cuenta de usuario

En los siguientes diagramas GRL y UCM se describe a mayor detalle la meta 1.1.2.1.1.1.1 mostrada en el diagrama GRL general del sistema AUPAs.

7.8.1.3.1 Diagrama GRL de la creación de una cuenta de usuario

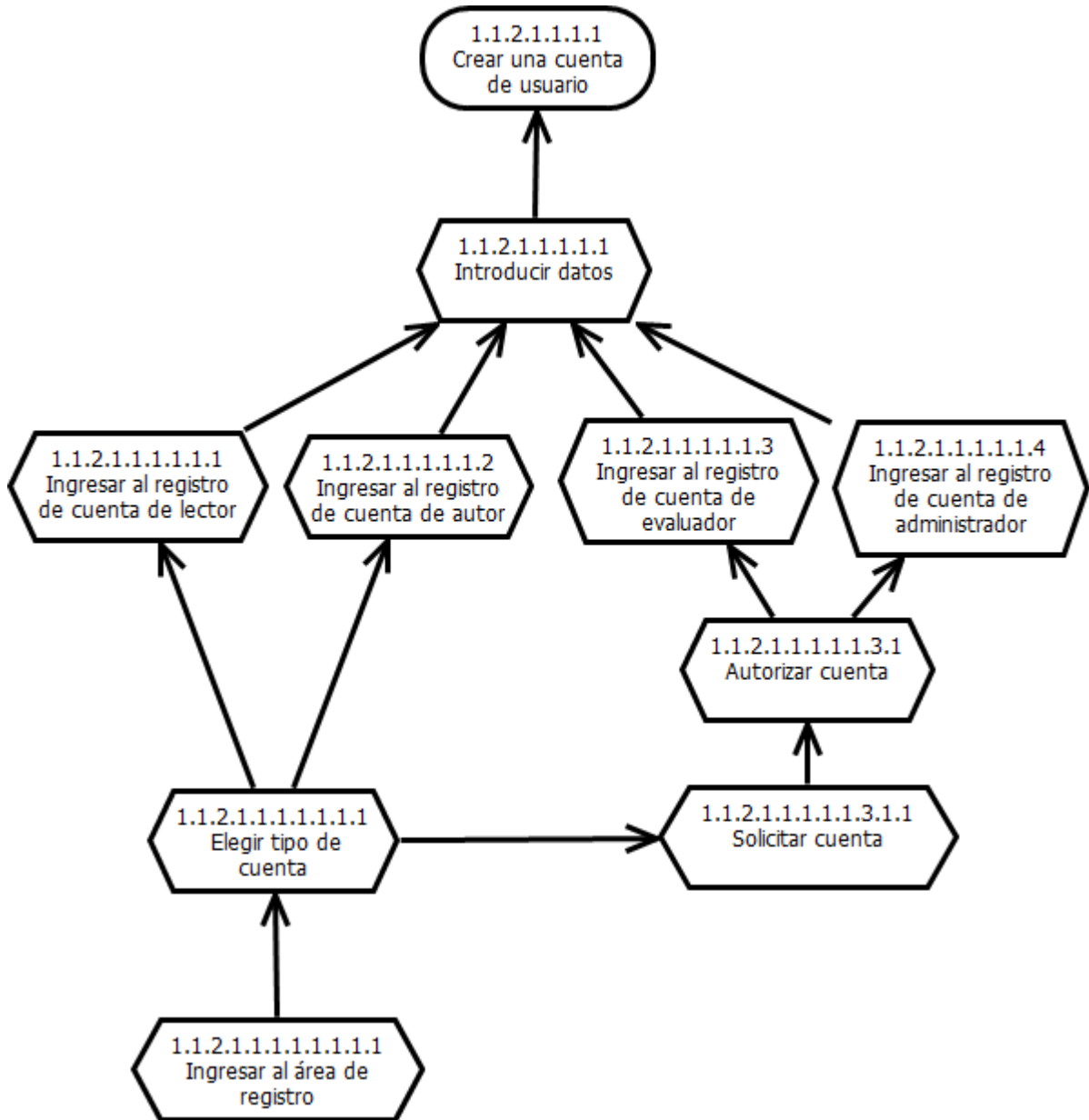


Figura 7.5 Diagrama GRL de la creación de una cuenta de usuario

7.8.1.3.2 Diagrama UCM de la creación de una cuenta de usuario

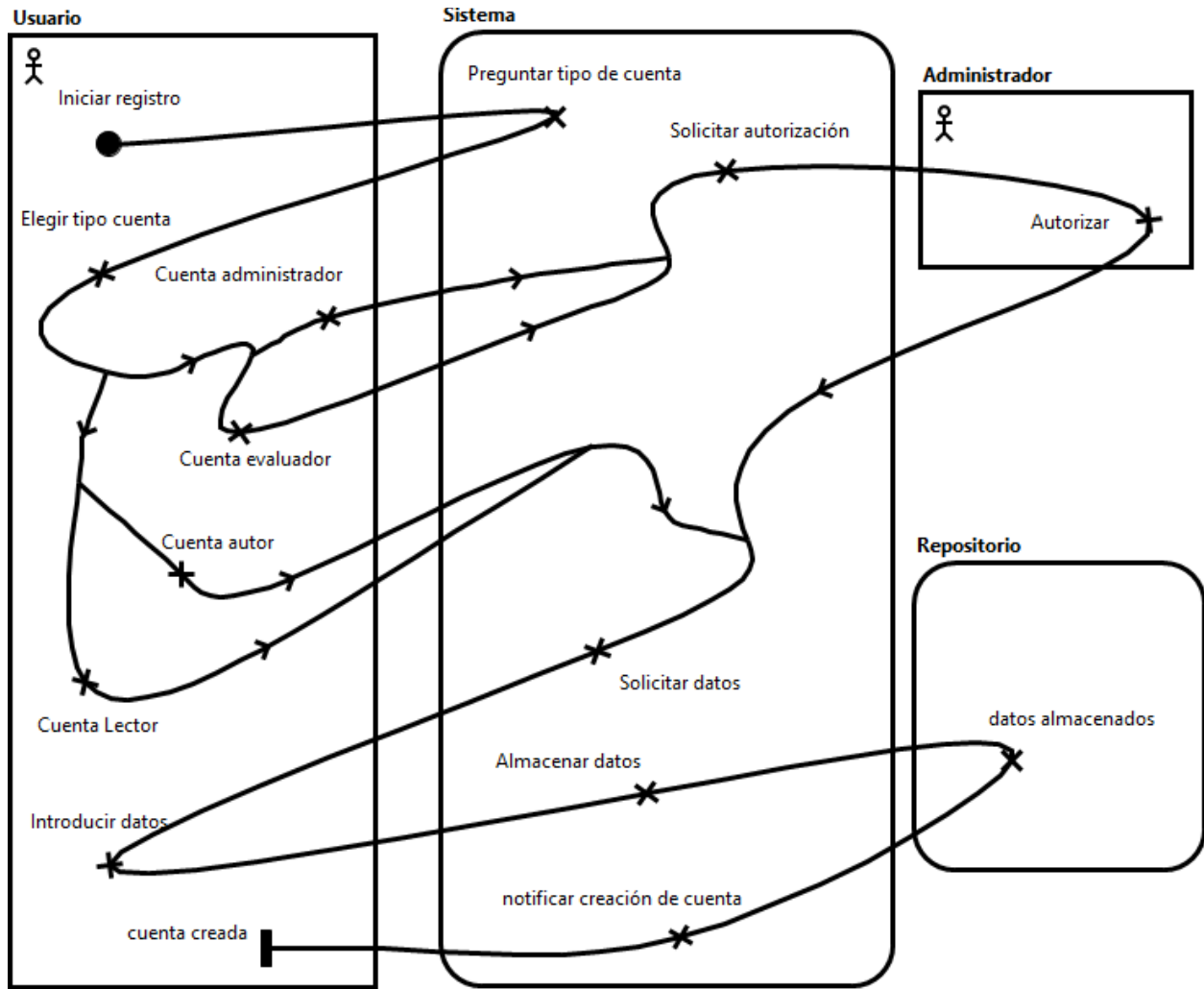


Figura 7.6 Diagrama UCM de la creación de una cuenta de usuario

7.8.1.4 Meta: Inicio de sesión

En los siguientes diagramas GRL y UCM se describe a mayor detalle la meta 1.1.2.1.1.1 mostrada en el diagrama GRL general del sistema AUPAs.

7.8.1.4.1 Diagrama GRL del inicio de sesión

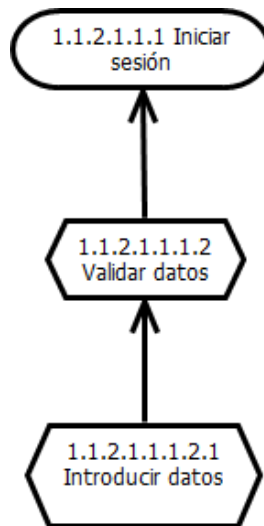


Figura 7.7 Diagrama GRL del inicio de sesión

7.8.1.4.2 Diagrama UCM del inicio de sesión

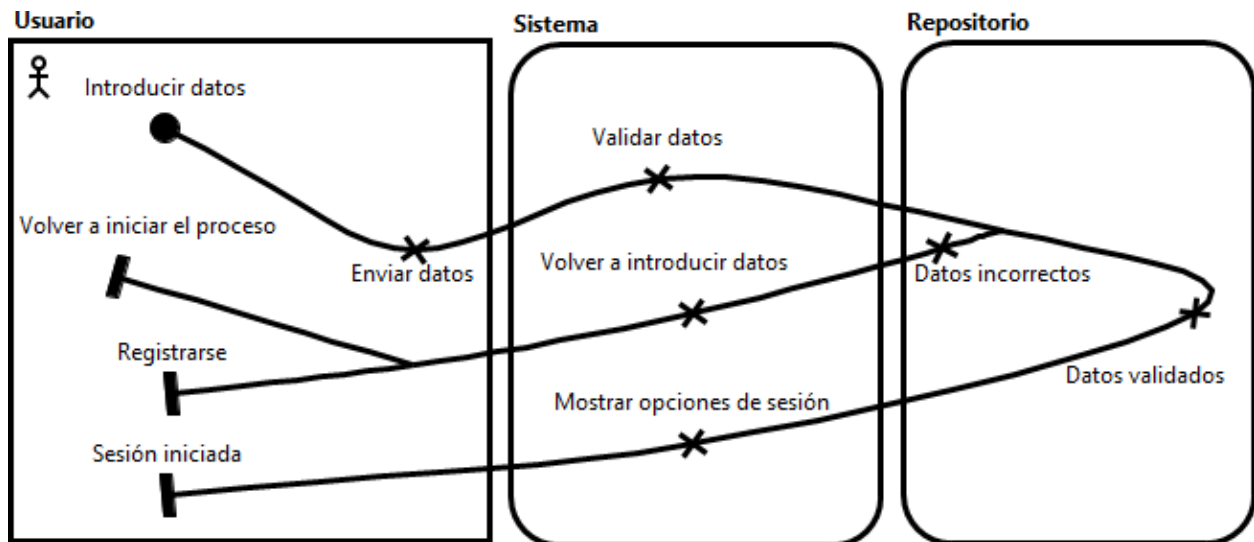


Figura 7.8 Diagrama UCM del inicio de sesión

7.8.1.5 Meta: Inserción de una nueva ontología

En los siguientes diagramas GRL y UCM se describe a mayor detalle la meta 1.1.3.1.1.1 mostrada en el diagrama GRL general del sistema AUPAs.

7.8.1.5.1 Diagrama GRL de la inserción de una nueva ontología

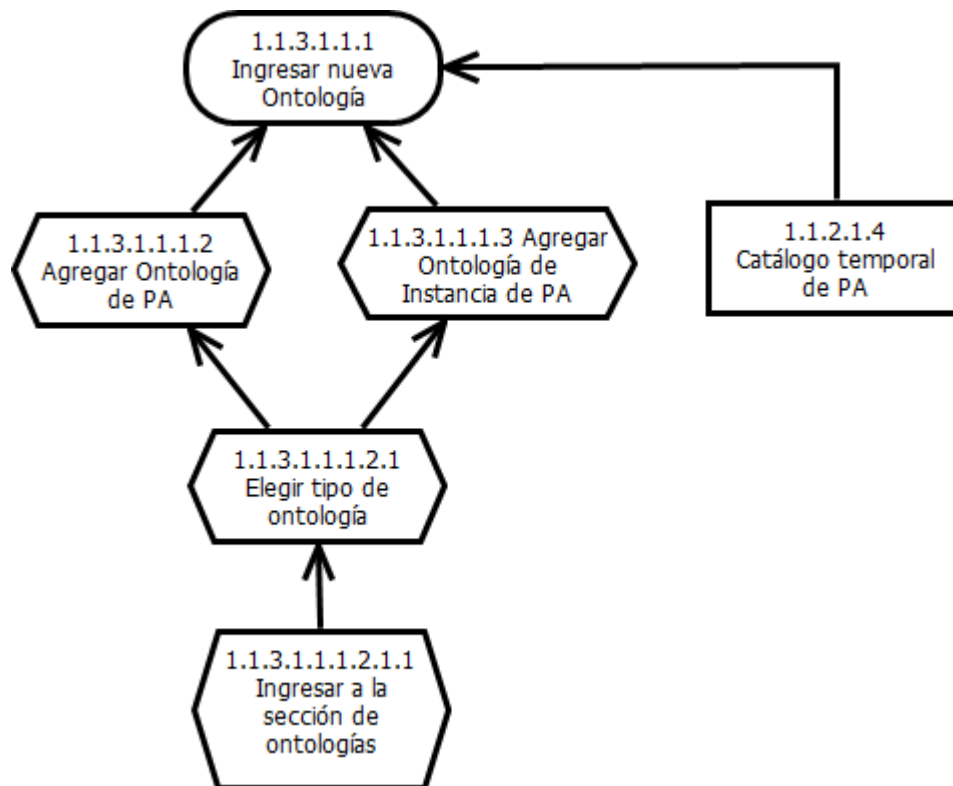


Figura 7.9 Diagrama GRL de la inserción de una nueva ontología

7.8.1.5.2 Diagrama UCM de la inserción de una nueva ontología

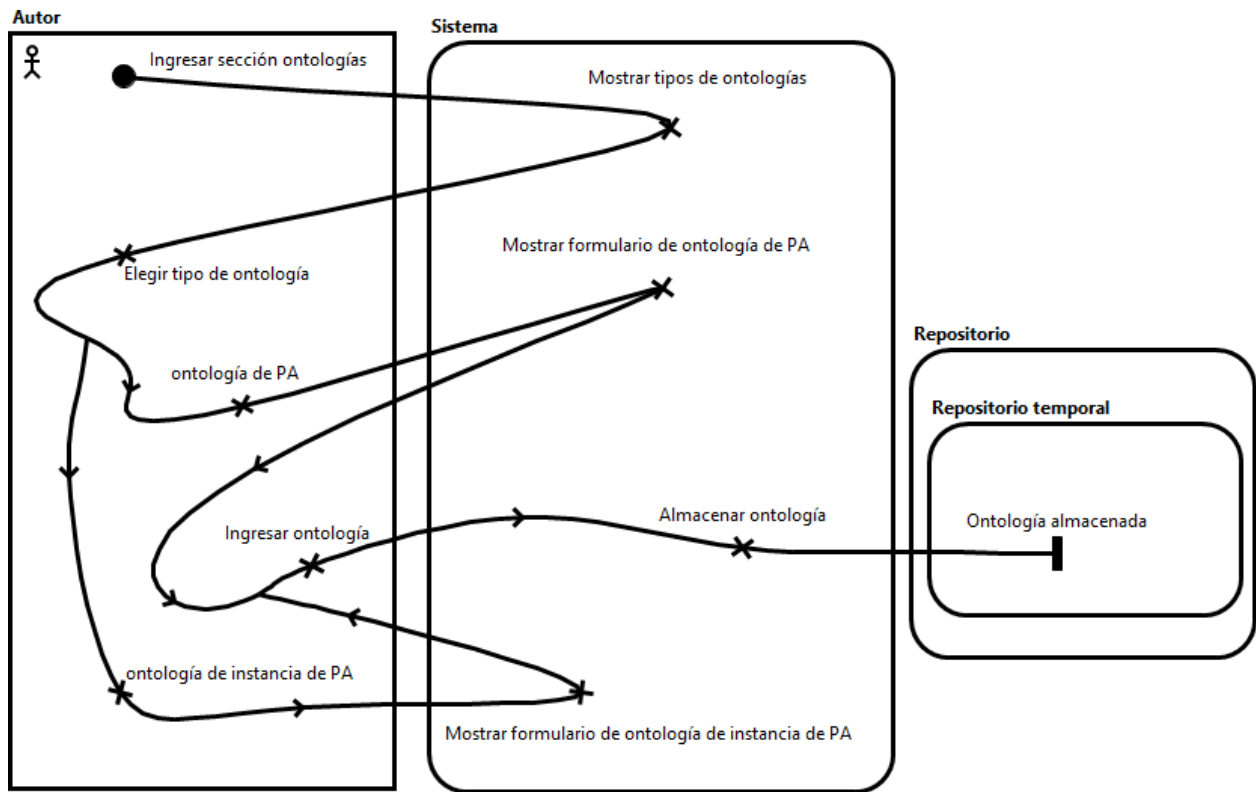


Figura 7.10 Diagrama UCM de la inserción de una nueva ontología

7.8.1.6 Meta: Inserción de un nuevo individuo

En los siguientes diagramas GRL y UCM se describe a mayor detalle la meta 1.1.3.1.1.2 mostrada en el diagrama GRL general del sistema AUPAs.

7.8.1.6.1 Diagrama GRL de la inserción de un nuevo individuo

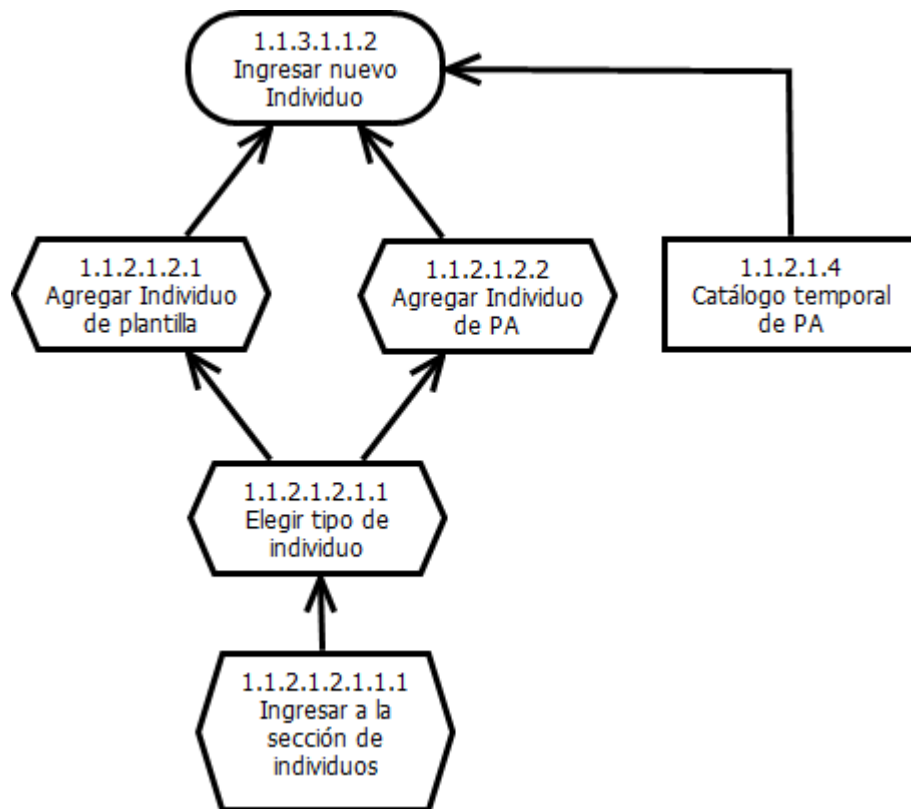


Figura 7.11 Diagrama GRL de la inserción de un nuevo individuo

7.8.1.6.2 Diagrama UCM de la inserción de un nuevo individuo

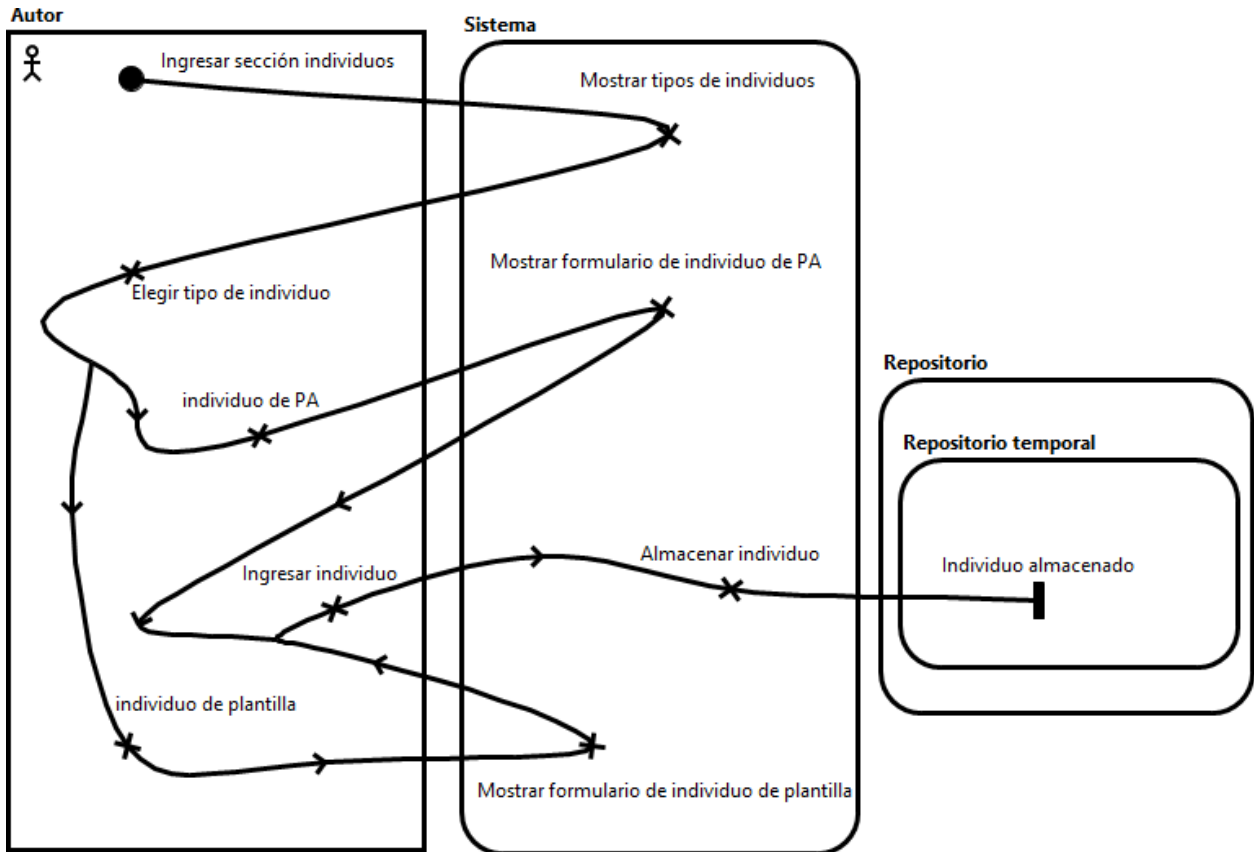


Figura 7.12 Diagrama UCM de la inserción de un nuevo individuo

7.8.1.7.2 Diagrama UCM de la aprobación de una aportación al catálogo de PAs

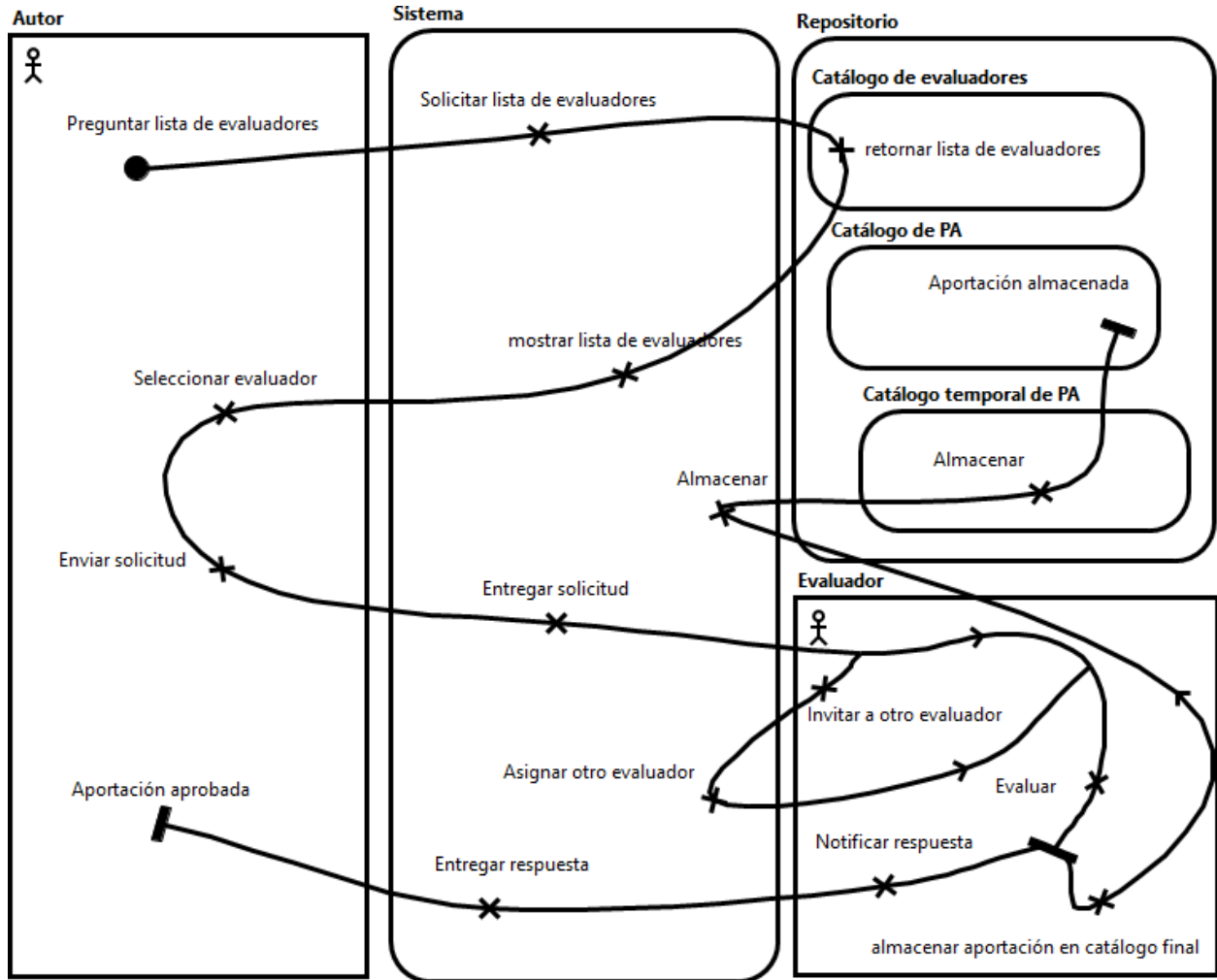


Figura 7.14 Diagrama UCM de la aprobación de una aportación al catálogo de PAs

7.8.1.8 Meta: Consulta al catálogo de PAs

En los siguientes diagramas GRL y UCM se describe a mayor detalle la meta 1.1.2.1 mostrada en el diagrama GRL general del sistema AUPAs.

7.8.1.8.1 Diagrama GRL de la consulta al catálogo de PAs

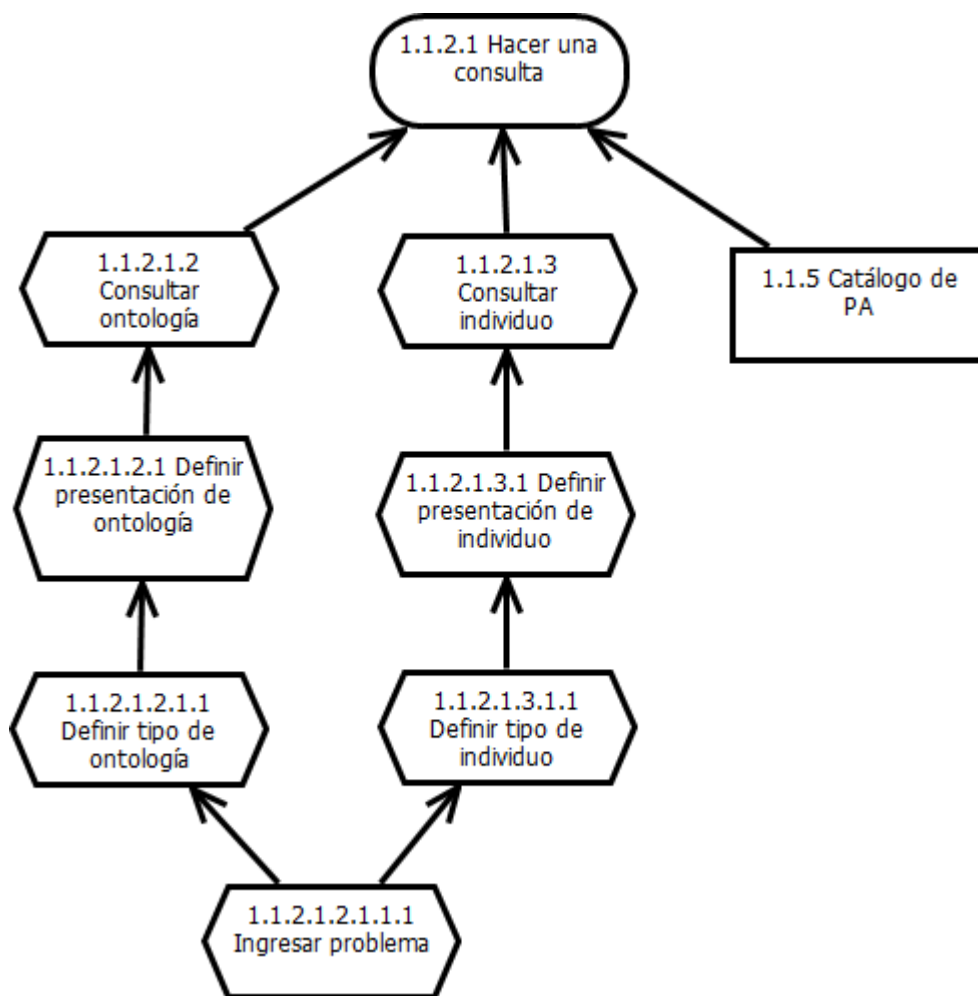


Figura 7.15 Diagrama GRL de la consulta al catálogo de PAs

7.8.1.8.2 Diagrama UCM de la consulta al catálogo de PAs

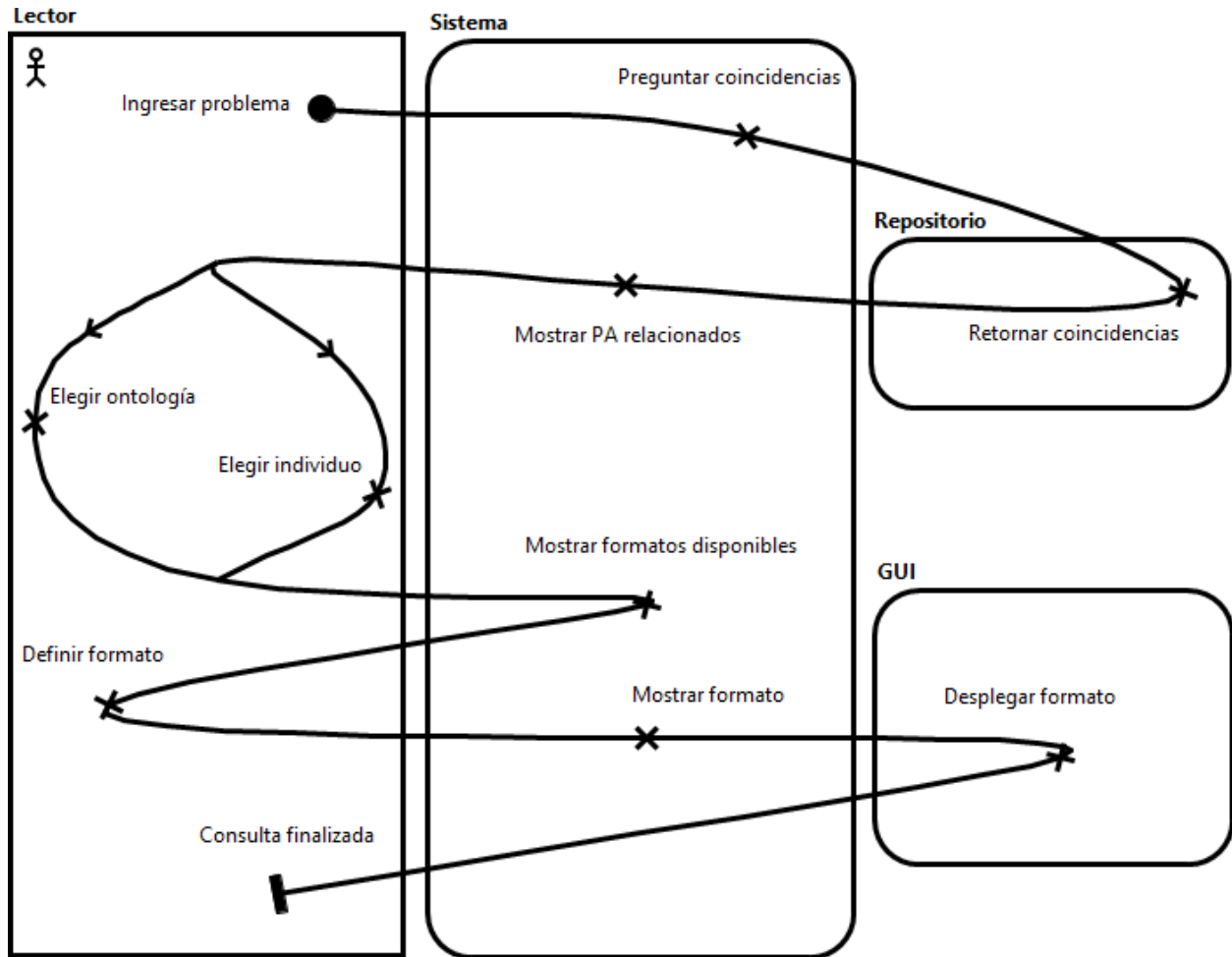


Figura 7.16 Diagrama UCM de la consulta al catálogo de PAs

Especificación textual de requerimientos

Se obtuvo un total de 52 elementos de los diagramas GRL, los cuales se pueden dividir en:

- 41 Requerimientos funcionales,
- 7 Requerimientos no funcionales y
- 4 Recursos.

7.9 Requerimientos funcionales (lista)

A continuación se enlistan las metas y tareas mostradas en los diagramas GRL de la sección anterior (7.8):

1.1.1.1 Exportar diagrama: Poder exportar diagramas en formato XMI a otras herramientas de modelado.

1.1.2.1 Hacer una Consulta: El usuario tipo Lector podrá realizar diversas consultas al repositorio, ya sea la consulta de alguna ontología o individuo de ontología, así como sus respectivos diagramas y códigos.

1.1.2.1.1 Ingresar al área de consultas

1.1.2.1.1.1 Iniciar sesión

1.1.2.1.1.1.1 Crear una cuenta de usuario: Al crear una nueva cuenta (cuenta tipo lector), el usuario dispondrá de un conjunto de permisos asignados de forma predefinida y podrá disponer en un futuro, si se desea, de más permisos, los cuáles se irán otorgando de manera flexible dependiendo del tipo de cuenta, con el cumplimiento de ciertos requisitos.

1.1.2.1.1.1.1.1 Introducir datos: Se ingresarán los datos de la cuenta del nuevo usuario.

1.1.2.1.1.1.1.1.1 Ingresar al registro de cuenta de lector

1.1.2.1.1.1.1.1.1.1 Elegir tipo de cuenta: El usuario tendrá la posibilidad de elegir entre cuatro tipos de cuenta (Administrador, Evaluador, Autor o Lector).

1.1.2.1.1.1.1.1.1.1 Ingresar al área de registro

1.1.2.1.1.1.1.1.2 Ingresar al registro de cuenta de autor

1.1.2.1.1.1.1.1.3 Ingresar al registro de cuenta de evaluador

1.1.2.1.1.1.1.3.1 Autorizar cuenta: El administrador será el único tipo de usuario que podrá autorizar más cuentas de administrador, así como de evaluador.

1.1.2.1.1.1.1.3.1.1 Solicitar cuenta: El usuario deberá hacer una solicitud para la creación de cuentas tipo Evaluador o Administrador.

1.1.2.1.1.1.1.4 Ingresar al registro de cuenta de administrador

1.1.2.1.1.2 Validar datos: Se validará si los datos ingresados son correctos, además de identificar qué tipo de cuenta es la que intenta iniciar sesión.

1.1.2.1.1.2.1 Introducir datos: Se ingresan los datos de usuario y contraseña.

1.1.2.1.2 Consultar ontología

1.1.2.1.2.1 Definir presentación de ontología: Se podrá elegir si se visualizará la ontología de una forma gráfica (diagramas) o simplemente ver el código fuente (owl).

1.1.2.1.2.1.1 Definir tipo de ontología: Se podrá elegir entre consultar la ontología de la OPPAE, consultar ontologías de PAs o consultar ontologías de Instancias de PAs.

1.1.2.1.2.1.1.1 Ingresar problema: Se deberá introducir un tema o problema de interés. Y el sistema desplegará los resultados disponibles relacionados a ese tema que se encuentren en el catálogo de PAs.

1.1.2.1.3 Consultar individuo

1.1.2.1.3.1 Definir presentación de individuo: Se podrá elegir si se visualizará el individuo en forma de tabla o simplemente ver el código fuente (owl).

1.1.2.1.3.1.1 Definir tipo de individuo: Se podrá elegir de entre dos tipos de individuos: de OPPAE o de ontología de PA.

1.1.3 Hacer una aportación al catálogo de PAs: El usuario podrá ingresar al sistema una ontología o un individuo según sea el caso, para esto será necesario que se tengan permisos de “Autor”.

1.1.3.1.1 Aprobar aportación: Un usuario podrá evaluar las aportaciones que se realicen al repositorio y decidir si se aprueban o se rechazan, el usuario encargado deberá contar con los privilegios que otorgan la cuenta tipo “Evaluador”.

1.1.3.1.1.1 Ingresar nueva Ontología: Se ingresarán las ontologías en formato OWL.

1.1.3.1.1.1.1 Ingresar al área de aportaciones

1.1.3.1.1.1.2 Agregar Ontología de PA:

1.1.3.1.1.1.2.1 Elegir tipo de ontología: Se podrá elegir si se desea ingresar una ontología de PA o una ontología de instancia de PA.

1.1.3.1.1.1.2.1.1 Ingresar a la sección de ontologías

1.1.3.1.1.1.3 Agregar Ontología de instancia de PA

1.1.3.1.1.2 Ingresar nuevo Individuo: Se ingresarán los individuos mediante un formulario desplegado por el sistema.

1.1.2.1.2.1 Agregar Individuo de plantilla:

1.1.2.1.2.1.1 Elegir tipo de individuo: Se podrá elegir si se desea ingresar un individuo: de OPPAE o de ontología de PA.

1.1.2.1.2.1.1.1 Ingresar a la sección de individuos

1.1.2.1.2.2 Agregar Individuo de PA

1.1.3.1.1.3 Notificar respuesta: El evaluador dará su veredicto de acuerdo a la aprobación e informara al autor si fue aceptada o rechazada.

1.1.3.1.1.3.1 Evaluar: El evaluador realizará el proceso de evaluación de la aportación.

1.1.3.1.1.3.1.1 Enviar solicitud: El autor deberá ponerse en contacto con el evaluador para ver la posibilidad que su aportación sea evaluada.

1.1.3.1.1.3.1.1.1 Seleccionar evaluador: Se deberá seleccionar al evaluador que tenga menos tareas asignadas para acelerar el proceso de evaluación.

1.1.3.1.1.3.1.2 Solicitar otro evaluador: El evaluador solicitado podrá solicitar otro evaluador ya sea para en conjunto hacer la evaluación o para asignarle dicho trabajo.

7.10 Requerimientos no funcionales

A continuación se enlistan las metas suaves mostradas en los diagramas GRL de la sección 7.8:

1. Permitir dar soporte al proceso de desarrollo de software

1.1 Dar soporte a la etapa de análisis

1.1.1 Dar soporte a la fase de representación del problema

1.1.2 Dar soporte para elegir el patrón de análisis adecuado

1.1.3.1 Lograr contenido consensual, formal y compartido

1.1.4 Proveer administración del sistema

El sistema deberá dar respuesta a las peticiones de una manera oportuna y fluida.

7.11 Recursos (Componentes)

A continuación se enlistan los recursos mostrados en los diagramas GRL de la sección 7.8:

Repositorio (OBDB)

1.1.2.1.4 Catálogo temporal de PAs: Catálogo donde se encuentran almacenadas las ontologías de PAs y de instancias de PAs, así como los individuos de plantilla de PAs, y los individuos de PAs; de reciente creación, que aún no han sido aprobados para incorporarse al catálogo de PAs final.

1.1.3.1.1.3.1.1.1 Catálogo de evaluadores:

Catálogo que contiene almacenados los datos de todos los evaluadores, así como su disponibilidad.

1.1.5 Catálogo de PAs: Catálogo donde se encuentran almacenadas las ontologías de plantilla, de PAs y de instancias de PAs, así como sus respectivos individuos.

7.12 Atributos del Sistema de software

7.12.1 Confiabilidad

Las Ontologías disponibles en el catálogo del repositorio pasarán por un proceso de validación, antes de ser aceptadas e incluidas en el catálogo. Por lo que toda información que presente el sistema será consensuada, formal y compartida.

7.12.2 Disponibilidad

El Sistema estará disponible en todo momento debido a que estará almacenado en una computadora servidor y mientras haya conexión a Internet el Sistema podrá ser accedido.

7.12.3 Seguridad

El ambiente tendrá diferentes perfiles de usuario: Administrador, Evaluador, Autor y Lector, lo cual permitirá un esquema de seguridad y la aplicación de restricciones en el acceso a las opciones que ofrece el sistema.

7.12.4 Mantenibilidad

Debido a que el sistema se presenta como un trabajo de investigación a nivel maestría tendrá diversos desarrolladores que irán enriqueciéndolo y manteniéndolo funcional, en investigaciones posteriores a esta.

7.12.5 Portabilidad

El sistema inicialmente estará alojado en una computadora servidor con sistema operativo Linux, sin embargo será capaz de emigrar a otras plataformas sin necesidad de hacer modificaciones drásticas o costosas. Al contrario estará en la posibilidad de adaptar alguna otra de una manera práctica.

Dentro de este capítulo se presentaron los requerimientos funcionales y no funcionales, además de los diagramas GRL y UCM, los cuales darán la pauta para el diseño de alto nivel del sistema de Almacenaje y Utilización de Patrones de Análisis, el cual se realizó en el siguiente capítulo (Capítulo 8).

Capítulo 8

Diseño de Alto Nivel del Sistema AUPAs

En el presente capítulo se presenta la documentación del diseño de alto nivel del sistema de Almacenaje y Utilización de Patrones de Análisis. Se manejan diferentes vistas de diseño para representar al sistema como: vista de contexto, vista de interacción, vista lógica, vista de interfaz, además de mostrar el diseño de la base de datos donde estarán almacenados los PAs y sus diferentes tipos ontologías.

8.1 Diseño

La etapa de diseño, de alto nivel, realizada en esta investigación, tuvo como finalidad describir la arquitectura de los componentes de software que formarán parte de la infraestructura para la creación del sistema AUPAs.

Para su realización se elaboró un Documento de Especificación de Diseño de Software (SDS) donde se presenta la estructura del sistema de almacenaje y utilización de patrones de análisis (AUPAs), desde distintas perspectivas. Este documento se desarrolló utilizando el estándar de la IEEE 1016 (Engineering et al. 2009), que especifica el contenido, su estructura y sugerencias acerca de calidad de la Especificación Diseño de Software (SDSS).

8.2 Vistas de diseño

8.2.1 *Vista de contexto*

8.2.1.1 Elementos de diseño

8.2.1.1.1 Actores

En este punto se detallan las funciones o permisos con las que cuenta cada tipo de usuario.

Usuario.- Se le denomina usuario al actor que accede al sistema AUPAs y que aún no ha sido registrado en el sistema, o que no cuenta con uno de los 4 tipos de cuenta, por lo que su estatus puede cambiar a cualquiera de los 4 tipos de usuario siguientes: Lector, Autor, Evaluador, o Administrador, después de pasar por el proceso de registro correspondiente.

Administrador.- Es el tipo de usuario que posee todos los privilegios para acceder a todas las funciones del sistema y que además tiene la capacidad de otorgar o quitar permisos a los demás usuarios, es quién se encarga de gestionar todas las cuentas,

es el único tipo de usuario que puede autorizar a otros usuarios obtener las cuentas tipo administrador y evaluador.

Evaluador.- Son aquellos usuarios que se encargan de autorizar el ingreso de aportaciones al sistema, fungen como un filtro, que a través de él pasan todas las ontologías o individuos de ontología antes de aceptarse para formar parte del catálogo de PAs final. De este tipo de usuario depende la decisión de aprobar o rechazar una aportación.

Autor.- Dícese de los usuarios que se encargan de aportar, alguna ontología o individuo de una ontología ya existente, para el catálogo de PAs.

Lector.- Se refiere a todo desarrollador de software cuyo único objetivo sea consultar el catálogo de PAs del sistema, para usar los servicios que apoyan en la elección del PA adecuado, con el objetivo de utilizarlo para describir algún problema de interés.

8.2.1.1.2 Casos de uso

En este apartado se detallan los casos de uso mostrados en la Figura 8.1.

Hacer una aportación: una “aportación” se refiere a las ontologías y/o individuos de ontología que son ingresados al sistema AUPAs por un usuario tipo autor. Ahora bien, el caso de uso “hacer una aportación” es realizado una vez que una aportación ha sido aprobada, y consiste en un proceso de conversión de código OWL-DL a un esquema de BD, por medio del cual la aportación se almacena en el repositorio de Ontologías.

Aprobar aportación: Consiste en revisar una aportación mediante un proceso de evaluación, en el cual se decide si dicha aportación cumple con los requisitos para formar parte del catálogo de PAs.

Ingresar individuo: Se ingresarán los individuos mediante un formulario desplegado por el sistema. Se podrá elegir si se desea ingresar un individuo: de OPPAE o de ontología de PA.

Ingresar ontología: Se ingresarán las ontologías en formato OWL. Se podrá elegir si se desea ingresar una ontología de PA o una ontología de instancia de PA.

Autenticación: Se ingresan los datos de usuario y contraseña. Se validará si los datos ingresados son correctos, además de identificar qué tipo de cuenta es la que intenta iniciar sesión.

Hacer una consulta: El usuario introducirá un tema o problema de interés. Y el sistema desplegará los resultados disponibles relacionados a ese tema que se encuentren en el catálogo de PA, ya sean ontologías o individuos. Posteriormente el usuario elegirá qué es lo que desea consultar, así como la presentación o el formato de su consulta (ej. Gráfica, en lista, etc).

Exportar diagrama: Se refiere al hecho de poder exportar diagramas en formato XML a otras herramientas de modelado.

Crear una cuenta de usuario: El usuario tendrá la posibilidad de solicitar una cuenta, eligiendo entre cuatro tipos de cuenta (Administrador, Evaluador, Autor o Lector). Pero siempre iniciando por el primer nivel (cuenta tipo lector), el usuario dispondrá de un conjunto de permisos asignados de forma predefinida y podrá disponer en un futuro, si se desea, de más permisos, los cuáles se irán otorgando de manera flexible dependiendo del tipo de cuenta, con el cumplimiento de ciertos requisitos. De igual forma si un usuario sube de nivel de cuenta ganará permisos pero sin perder los que ya tenía, ejemplo: si un usuario que era “lector” cambia a “autor” seguirá contando tanto con los privilegios de autor como de lector y así sucesivamente, si un autor cambia a evaluador.

8.2.1.1.3 Consideraciones

1. Para llevar a cabo el caso de uso “Aprobar aportación”, es necesario realizar alguno de los 2 casos de uso: “Ingresar individuo” o “Ingresar ontología”, previamente.

2. Para emplear el caso de uso “Ingresar Individuo” será necesario haber ingresado previamente la ontología a la que pertenece dicho individuo, es decir haber cumplido con el caso de uso “Ingresar Ontología”, una vez cumplido este último, se podrán ingresar tantos Individuos se requieran, de acuerdo a lo anterior podemos decir que el caso de uso “Ingresar Individuo” sólo dependerá la primera vez del caso de uso “Ingresar Ontología”, para las demás veces será independiente.

3. Para utilizar el caso de uso “Autenticación”, sólo dependerá la primera ocasión del caso de uso “Crear una cuenta de usuario”, para las demás ocasiones será independiente.

Los elementos de las secciones 8.2.1.1.1 y 8.2.1.1.2 se pueden apreciar de una mejor manera mediante la Figura 8.1.

8.2.1.2 Diagrama de casos de uso general del sistema AUPAs

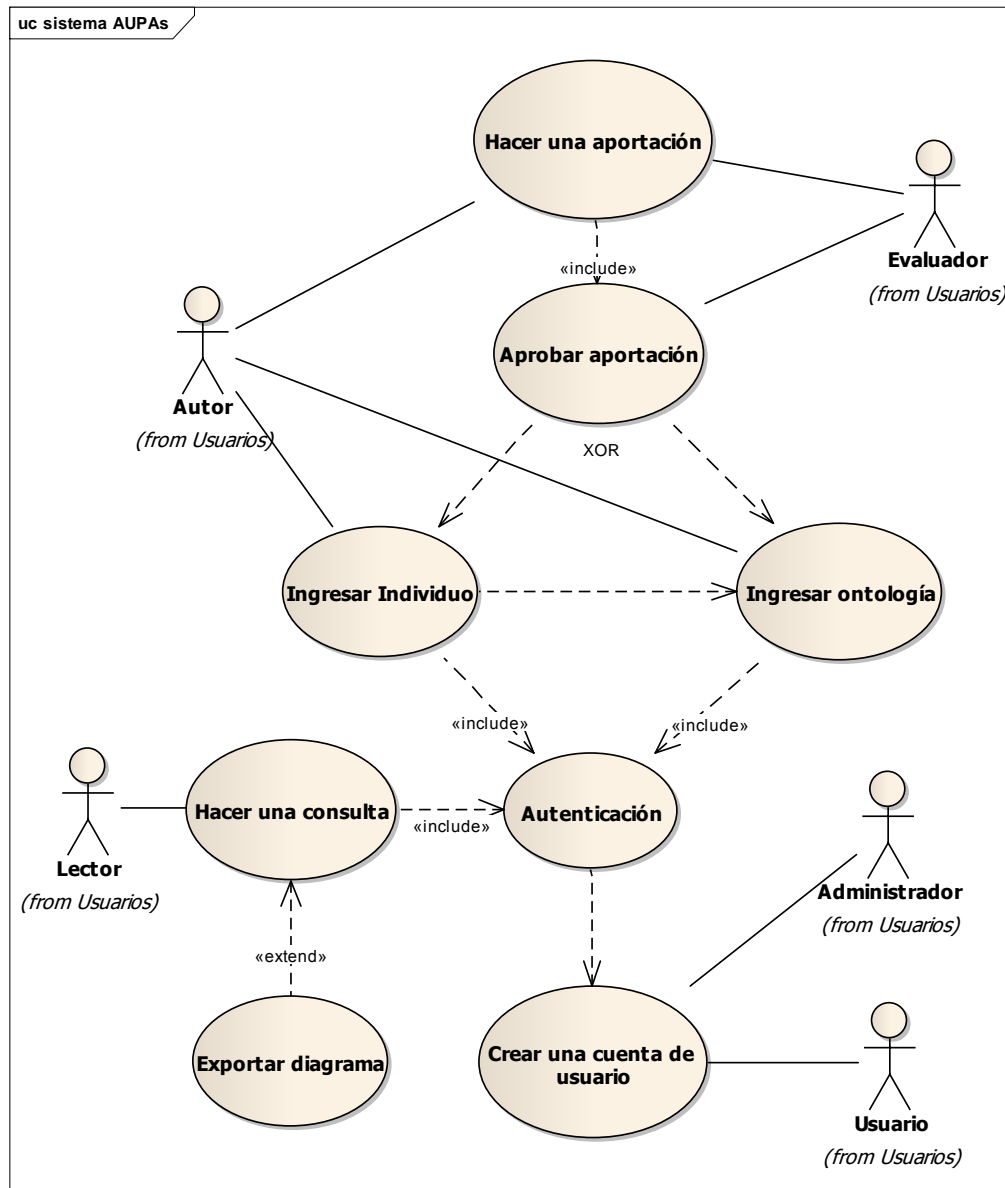


Figura 8.1 Diagrama de casos de uso general del sistema AUPAs

8.2.2 Vista de interacción

8.2.2.1 Diagrama de secuencia general del sistema AUPAs

En el siguiente diagrama (Figura 8.2) se muestran las acciones que debe llevar a cabo, de manera cronológica, cada tipo de usuario para generar un flujo general exitoso en el sistema.

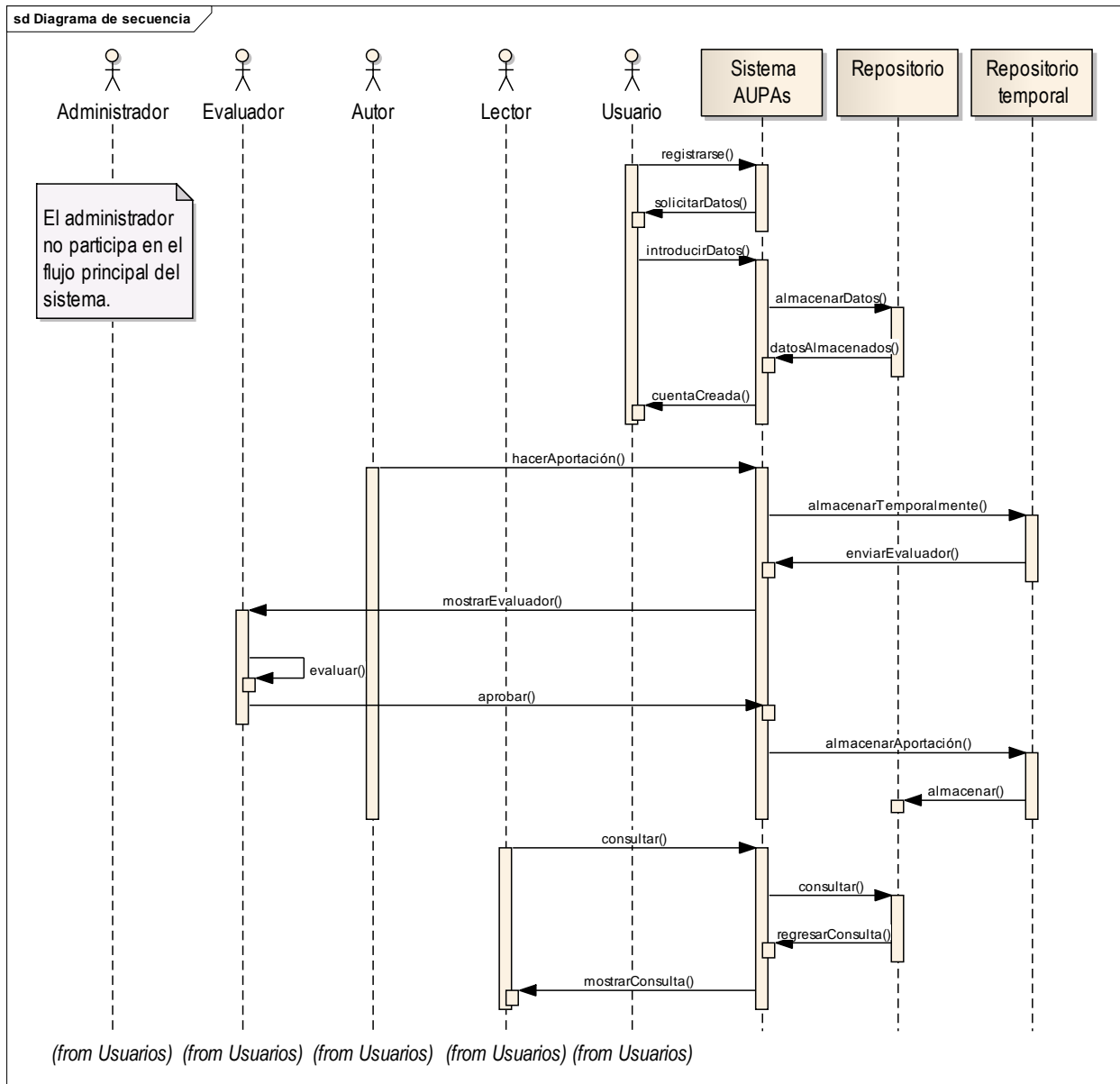


Figura 8.2 Diagrama de secuencia general del sistema AUPAs

8.2.3 Vista lógica

8.2.3.1 Elementos de diseño

8.2.3.1.1 Clases

En el caso de sistema AUPAs se identificaron las clases⁹ mostradas en la Figura 8.3 y se detallan a continuación:

Sistema AUPAs: Clase principal donde se llevan a cabo todas las funciones del sistema mediante las llamadas a otras clases, creación de objetos, etc.

Usuario: Clase abstracta que puede ser implementada en los 4 siguientes subtipos de clases:

Administrador: Clase que debe describir las propiedades y el comportamiento propios del tipo de usuario Administrador descrito en el punto 8.2.1.1.1.

Evaluador: Clase que debe describir las propiedades y el comportamiento propios del tipo de usuario Evaluador descrito en el punto 8.2.1.1.1.

Autor: Clase que debe describir las propiedades y el comportamiento propios del tipo de usuario Autor descrito en el punto 8.2.1.1.1.

Lector: Clase que debe describir las propiedades y el comportamiento propios del tipo de usuario Lector descrito en el punto 8.2.1.1.1.

Aportación: Clase abstracta que puede ser implementada en los 2 siguientes tipos de clases:

Ontología: Clase que debe describir las propiedades y el comportamiento de todo lo referente a las ontologías almacenadas, como nombre de ontología, tipo de ontología, clases que la componen, propiedades, relaciones, etc.

Individuo: Clase que debe describir las propiedades y el comportamiento de todo lo referente a los individuos de ontologías almacenados, como nombre del individuo, ontología a la que pertenece, contenido (valores de las propiedades de las clases de esa ontología), etc.

⁹ Una clase es un conjunto de objetos con las mismas características. Clases típicas son Persona, Factura, Compañía, Proveedor, Orden, Producto, y Meta. (Eriksson & Penker 2000)

Catálogo: Clase abstracta que puede ser implementada en los 3 siguientes tipos de clases:

CatálogoPA: Clase que debe describir las propiedades y el comportamiento del catálogo donde se encuentran almacenadas las ontologías de plantilla, de PA y de instancias de PA, así como sus respectivos individuos.

CatálogoPATemporal: Clase que debe describir las propiedades y el comportamiento del catálogo donde se encuentran almacenadas las ontologías de PA y de instancias de PA, así como los individuos de plantilla de PA, y los individuos de PA; de reciente creación, que aún no han sido aprobados para incorporarse al catálogo de PA final.

CatálogoEvaluadores: Clase que debe describir las propiedades y el comportamiento del catálogo que contiene almacenados los datos de todos los evaluadores, así como su disponibilidad.

8.2.3.2 Diagrama de clases del sistema AUPAs

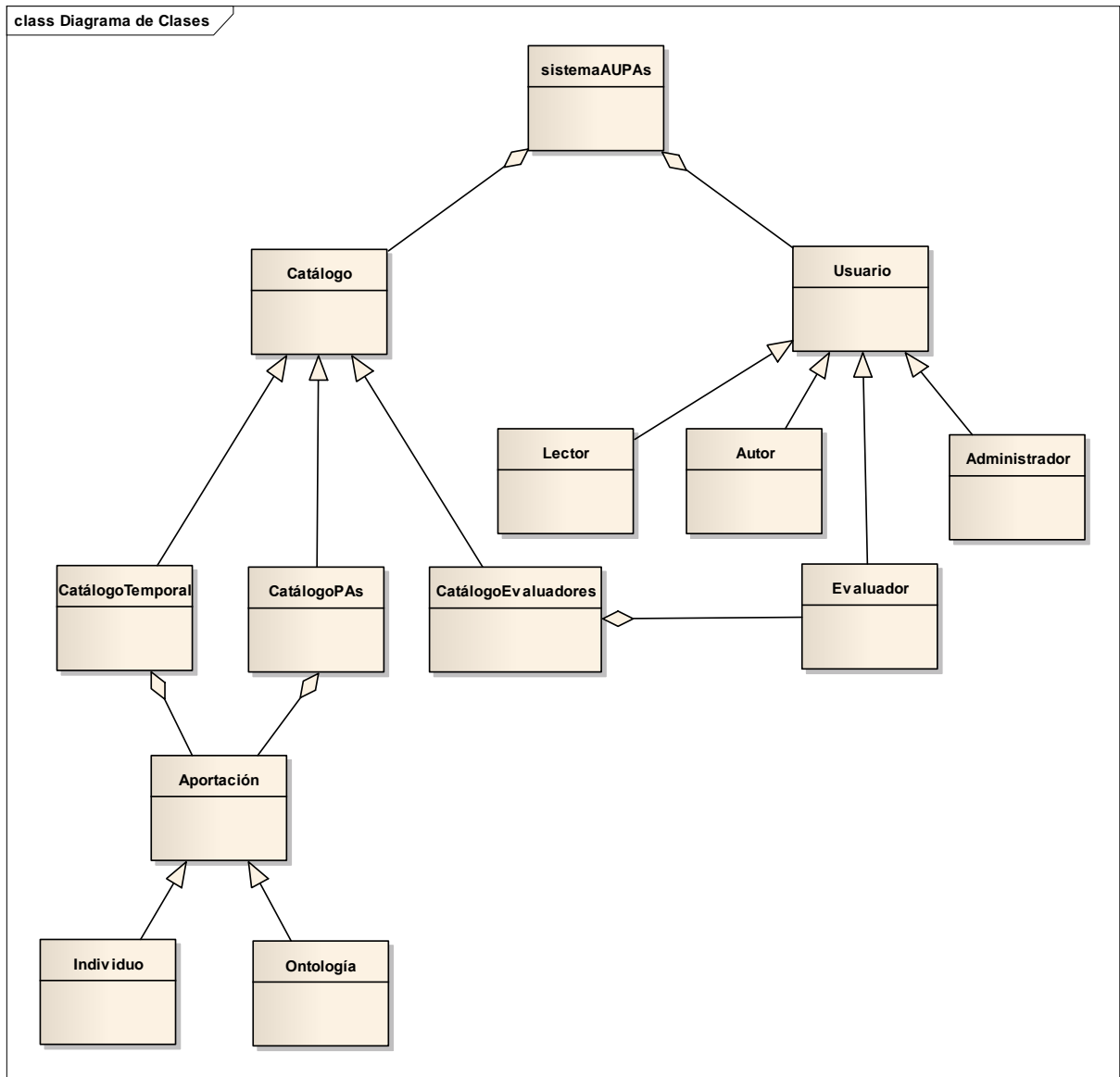


Figura 8.3 Diagrama de clases del sistema AUPAs

8.2.4 Vista de interfaz

8.2.4.1 Elementos de diseño

8.2.4.1.1 Componentes

Los componentes son trozos autónomos de código que se pueden volver a usar desplegándolos de manera independiente. (Los componentes no tienen que ser grandes, pero en general, son mucho más que una sola clase o un par de clases vagamente relacionadas). (Kimmel 2008)

Los componentes mostrados en la Figura 8.4 se detallan a continuación:

Aplicación sistema AUPAs: Aplicación que deberá instalarse en la máquina Servidor.

Autenticación: El propósito de este componente es revisar los datos de los usuarios al momento de ingresar al sistema tanto los errores en la escritura de los parámetros como en la existencia de los usuarios, además de definir los permisos de acceso a las actividades del sistema para los usuarios registrados.

Repositorio: Este componente se detalla más abajo en el punto 8.3.

Página de Internet: Interfaz que le permite al usuario tener acceso al sistema.

Traductor OWL-DL a SQL: Componente que se encarga de la conversión de archivos OWL a modelos o esquemas de base de datos.

Graficador de OWL-DL a XMI: Componente que se encarga de la conversión de código OWL a diagramas que puedan ser visualizados por el usuario.

Consultas: Aquí se encuentran todos los elementos encargados para hacer conexión con la base de datos y mediante peticiones del usuario generar consultas

Aportaciones: Se refiere al conjunto de elementos que están involucrados en la recepción de aportaciones por parte de los usuarios y el almacenamiento de estas en el repositorio.

Registro: Componente encargado de recibir los datos personales de los usuarios, con la posibilidad de crear hasta cuatro tipos de cuenta (Administrador, Evaluador, Autor o Lector).

Cuentas: Es un elemento primordial ya que es la única forma en que los usuarios pueden tener acceso al sistema.

Administración del sistema: Dentro de los propósitos de este componente se encuentran: la administración de los datos de usuarios que estarán utilizando el sistema mediante operaciones realizadas sobre la BD del sistema, además de dedicarse a administrar las aportaciones al catálogo de PAs, entre otros.

8.2.4.2 Diagrama de componentes del sistema AUPAs

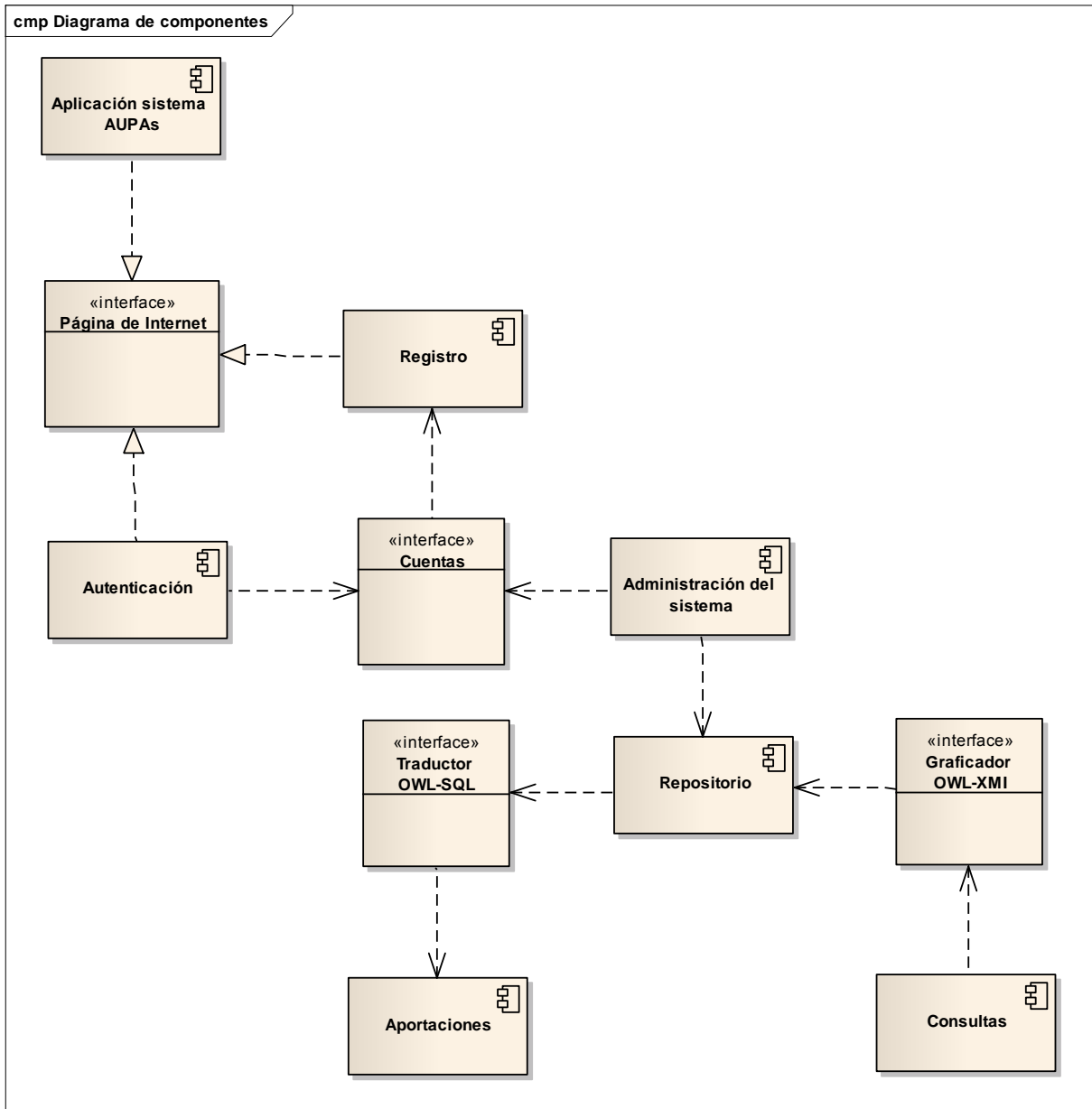


Figura 8.4 Diagrama de componentes del sistema AUPAs

8.3 Diseño de la base de datos

8.3.1 Restricciones y/o recomendaciones de diseño

- 1 Habrá un BD que contenga tres tablas: “Ontología Plantilla”, “Ontologías PA”, y “Ontologías Instancias de PA”, las cuales tendrán apuntadores a los archivos OWL de todas las ontologías que formen parte del catálogo (Un ejemplo de la tabla “Ontologías PA” se muestra en la Tabla 8.1).

Tabla 8.1 Ejemplo de la tabla “Ontologías PA” de la BD

ID	NOMBRE	OWL	OBDB
1	RESERVACION DE ENTIDADES	...\RESERVACION.OWL	...\RESERVACION.SQL
2	TALLER DE REPARACION	...\REPARACION.OWL	...\REPARACION.SQL
3	INVENTARIOS	...\INVENTARIOS.OWL	...\INVENTARIOS.SQL

- 2 Las tablas de la BD del punto anterior de igual forma contarán con los apuntadores a la OBDB de cada OWL.
- 3 El repositorio contará con un directorio donde se encuentren almacenadas las ontologías en formato OWL.
- 4 Habrá una OBDB por cada Ontología, ya sea de plantilla, patrón o instancia de patrón.
- 5 Los individuos de cada ontología estarán almacenados en la OBDB de su ontología.
- 6 El repositorio también contará con un directorio donde se encuentren almacenados los diagramas de los PAs en formato XMI.
- 7 El Catálogo o Repositorio temporal que aparece en las vistas de diseño de interacción y lógica (puntos 8.2.2 y 8.2.3) puede implementarse tanto dentro como fuera del Repositorio final, si es por fuera debe tener la misma estructura de éste, sólo que de forma independiente (aislada), y si se desea

Dentro de este capítulo se presentaron los diagramas UML correspondientes al diseño de alto nivel, los cuales sentarán las bases para un desarrollo posterior del diseño detallado e implementación del sistema de Almacenaje y Utilización de Patrones de Análisis. El diseño de alto nivel se realizó partiendo de los requerimientos presentados en el Capítulo 7, cuya validación se encuentra en el siguiente capítulo (Capítulo 9).

Capítulo 9

Validación de la Investigación

En este capítulo se presenta una descripción general del proceso empleado para la validación de los requerimientos correspondientes al Sistema de almacenaje y utilización de patrones de análisis (AUPAs). Se explican los pasos que se siguieron para la validación, (como el diseño de la encuesta, la selección de participantes, el análisis de las encuestas, etc., con base en el método Delphi), además de las conclusiones que se obtuvieron al realizarla.

9.1 Validación de requerimientos

Para la validación de los requerimientos fue necesario implementar una forma de evaluar diferente ya que normalmente tanto los requerimientos como el diseño de software son evaluados mediante prototipos y la aprobación de un cliente potencial. En el caso de esta investigación, se tienen varios clientes potenciales (estudiantes, desarrolladores o expertos, con conocimientos de patrones de análisis, ontologías o de desarrollo de software), y no se cuenta con un prototipo para probar la validez de los requisitos. Debido a esto, se usó una forma de validación de los requerimientos diferente, mediante el uso de encuestas y la herramienta jUCMNav¹⁰.

Se realizó el proceso de validación de los requerimientos, mediante una metodología “Delphi” mostrado en la Figura 9.1 mediante el uso de encuestas aplicadas a varios participantes con experiencia en ingeniería de software, bases de datos u ontologías (Linstone & Turoff 2002). Se encuestó también a quien realizó el análisis y la obtención de requerimientos y se validó la encuesta (ver sección 9.1.4 Validación de fiabilidad de la encuesta) mediante métodos estadísticos, para después validar los requerimientos, mediante los resultados obtenidos del tratamiento de las respuestas de los participantes. Los resultados estadísticos así obtenidos, se utilizaron en la determinación de valores a incluir en los diagramas GRL, mediante la herramienta jUCMNav. La inclusión de esos valores permitió visualizar la satisfacción y contribución de la jerarquía de elementos (metas, metas blandas, tareas y recursos) representados mediante GRL, para validar los requerimientos identificados.

En el proceso mostrado en la Figura 9.1 se parte de los 41 requerimientos funcionales, 7 requerimientos no funcionales y los 4 recursos descritos en el documento SRS del sistema AUPAs.

Considerando el conjunto de requerimientos mencionado, se establecieron los conocimientos y características de los participantes en el proceso Delphi, que se siguió para obtener opiniones acerca de la utilidad de dichos requerimientos. Se recopilaron las opiniones mediante encuestas. Una vez que se determinó la validez de la encuesta, se obtuvieron los porcentajes de evaluación para cada requerimiento, para asignarlos como valores numéricos a los elementos de los diagramas GRL, de acuerdo a las opiniones de los participantes. Estos valores se utilizaron para evaluar el nivel de satisfacción y contribución de sus elementos intencionales.

¹⁰ jUCMNAV: es un plug-in de Eclipse y editor gráfico para modelado, análisis y transformaciones con la notación de requerimientos de usuario (URN), el cual incluye UCM y GRL.

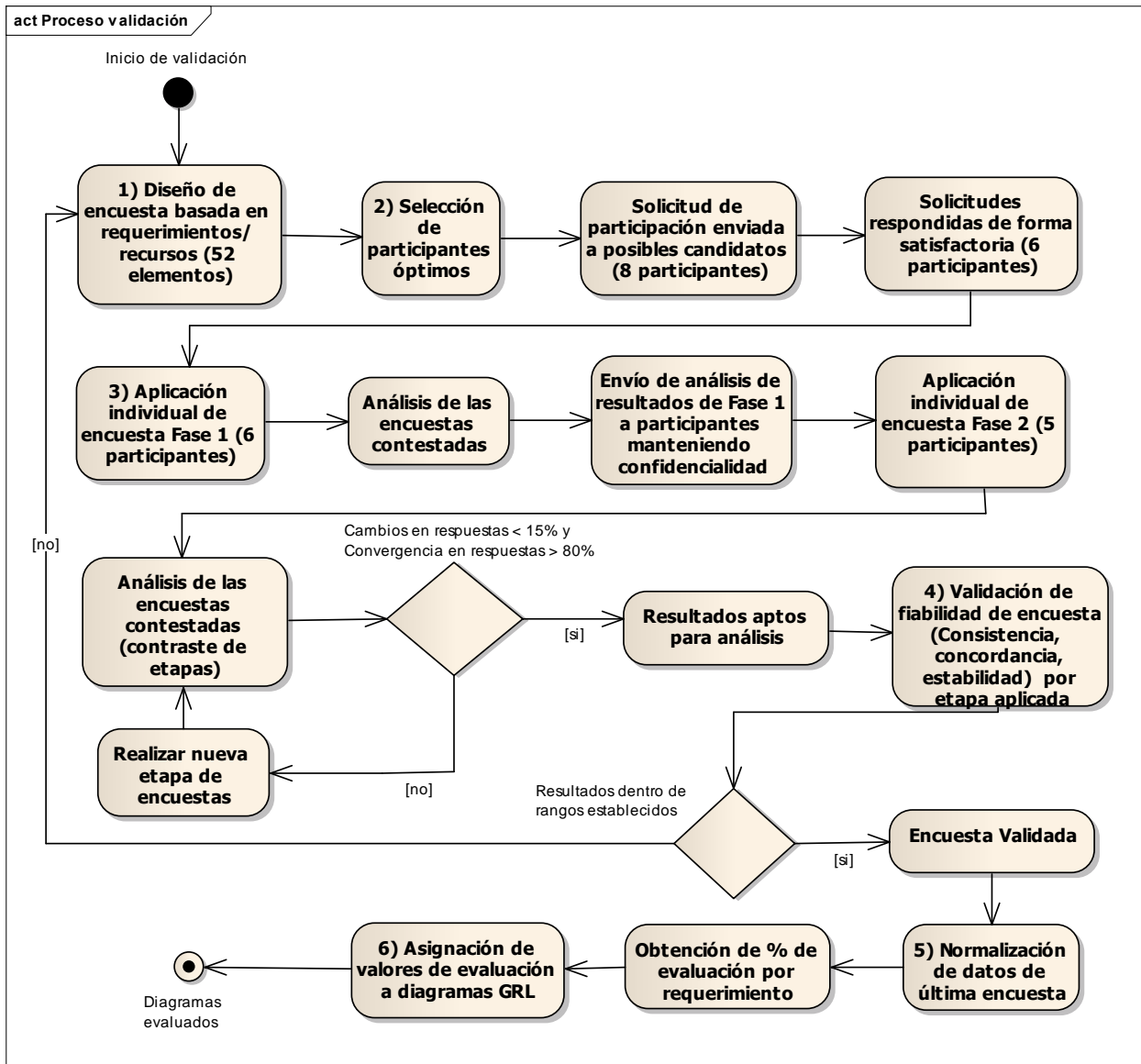


Figura 9.1 Proceso de validación de requerimientos.

9.1.1 Método Delphi y diseño de la encuesta

Para la recolección de la información se optó por seguir el método Delphi, el cual propone la evaluación mediante un cuestionario o encuesta, aplicada a expertos en el tema a evaluar, manteniendo siempre la confidencialidad de los participantes. Estas encuestas se aplican en varias fases, hasta cumplir con los dos criterios de paro: consenso y estabilidad. Lo que significa: 1) para el consenso, que los participantes deben lograr una convergencia en sus respuestas individuales con

respecto a las del grupo en más del 80% (Pozo et al. 2007), o lo que es lo mismo una desviación estándar en las opiniones de menos del 20%; y 2) para la estabilidad, que los cambios observados en las respuestas de una fase con respecto a otra, sean menores al 15% (Linstone & Turoff 2002).

Según Pozo (Pozo et al. 2007), este criterio de paro, se toma como base, para asegurar convergencia y no variabilidad de las estimaciones (respecto a la variabilidad: qué tan influenciados son los participantes con respecto a las respuestas grupales). Para el caso de la presente investigación, los criterios de paro se cumplieron en la segunda fase de aplicación de la encuesta, alcanzando en promedio un cambio en las respuestas del 4.4% y una convergencia del 84.2%.

A cada participante se le explicó el motivo de la aplicación de dicha encuesta, así como el método de evaluación y el propósito de las escalas, que se explican más adelante. Se pidió a los participantes que evaluaran el contenido de la encuesta utilizando como base sus conocimientos y experiencias, razonando cuidadosamente sus respuestas.

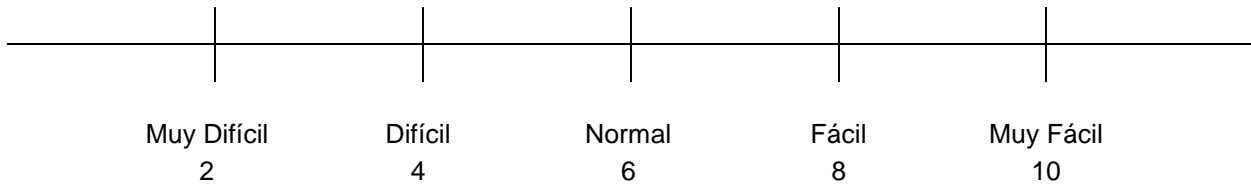
Se cuidó que ningún participante conociera las respuestas proporcionadas por otros participantes. En la segunda aplicación (fase) de la encuesta (y sucesivas aplicaciones, si hubieran sido necesarias), se proporcionó a cada participante, el análisis de las respuestas de la fase anterior, identificando su respuesta, y se presentaron las respuestas de los demás sin especificar el nombre de los participantes. De esta manera se pretendió que los participantes contrastaran sus respuestas contra las respuestas de los demás participantes. Así en cada fase, se pudo dar una nueva respuesta, considerando la opinión grupal o bien mantener sus opiniones. Esta estrategia ocasiona que las respuestas converjan.

Para el diseño de la encuesta (tarea marcada con el número 1 en la Figura 9.1) se utilizaron las descripciones de las tareas terminales (que no se desglosan en sub-tareas), asociadas a metas (requerimientos funcionales), las descripciones de las metas suaves (requerimientos no funcionales) y las descripciones de los recursos, representados en los diagramas GRL del documento SRS. Los elementos se ordenan agrupándolos en concordancia con cada uno de los requerimientos funcionales del sistema y agregando espacios para que los participantes pudieran evaluarlos y plasmar sus observaciones.

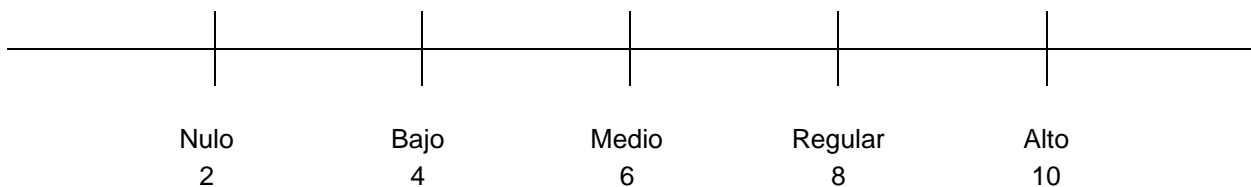
En el caso de los elementos asociados a requerimientos (funcionales y/o no funcionales), se evalúa la factibilidad de las tareas asociadas y el beneficio obtenido de su realización. En el caso de los elementos asociados a recursos, se evalúan la

importancia de tenerlos disponibles y el beneficio de utilizarlos. Para las evaluaciones de los elementos, se utilizan las dos escalas siguientes:

Nivel de factibilidad para su implementación:



Nivel de beneficio para el usuario:



Aunque se tienen 41 requerimientos funcionales, 7 no funcionales y 4 recursos, sólo se incluyeron en la encuesta 40 elementos relacionados con 35 tareas terminales (que no se desglosan en sub-tareas), 2 requerimientos no funcionales y 3 recursos. En el caso de tareas terminales, sus valores se promedian, para obtener el valor de los nodos padres, en niveles más altos de la jerarquía (Ej. la tarea 1.1.3.1.1.1 Ingresar nueva Ontología, que tiene tres sub-tareas: 1.1.3.1.1.1.1 Ingresar al área de aportaciones, 1.1.3.1.1.1.2 Agregar Ontología de PA y 1.1.3.1.1.1.3 Agregar Ontología de Instancia de PA).

Así, si un participante evalúa estas sub-tareas para factibilidad, con los valores “Difícil” (valor 4), “Normal” (valor 6) y “Fácil” (valor 8), se le asigna el promedio de estos valores: “Normal” (valor 6), a la tarea 1.1.3.1.1.1 Aunque en el ejemplo se manejan los valores simbólicos como: “Difícil” y “Fácil”, el tratamiento de la encuesta ocupa los valores numéricos).

9.1.2 *Participantes en el proceso Delphi*

Se seleccionaron 8 participantes de la población (profesores y estudiantes) de CENIDET (Centro Nacional de Investigación y Desarrollo Tecnológico), con conocimientos y experiencia en ingeniería de software, bases de datos u ontologías (tarea marcada con el número 2 en la Figura 9.1), y se les envió la solicitud de participación para la evaluación de requerimientos de forma individual. A esta solicitud de participación respondieron satisfactoriamente 6 participantes, que se consideraron adecuados, debido al nivel de conocimientos que tienen de ingeniería de software, bases de datos u ontologías.

El número reducido de participantes se debe a los siguientes factores:

- El tiempo de respuesta de las encuestas es mayor a lo esperado, debido a la cantidad de elementos a evaluar. Algunos de los posibles participantes no contaban con el tiempo necesario.
- El tema de las encuestas está enfocado en su mayoría a dos tópicos PAs y Ontologías, por lo tanto los participantes deben tener conocimientos de, por lo menos alguna de las dos áreas. Para el caso de los PAs, el número de candidatos a participar con estos conocimientos es reducido.
- Sólo se consultaron participantes dentro de CENIDET, pues es mayor la disponibilidad de los participantes dentro de la institución.

Los 6 participantes en este proceso tienen conocimientos y experiencia en ingeniería de software, bases de datos u ontologías. Sus respuestas se contrastaron con las respuestas del analista, que se trata como la respuesta esperada, para detectar una desviación relativa respecto a la respuesta esperada, que representa el error del grupo. El número de participantes que respondieron la encuesta es reducido, pero esta situación se espera para un proceso Delphi. Brockhoff (Brockhoff 2002) deliberadamente se concentra en estudiar el desempeño de grupos integrados por 4 a 11 participantes, debido a que es común que sólo se pueda encontrar un número reducido de expertos.

Brockhoff estableció que 5 participantes es el tamaño mínimo para los grupos, pues consideró determinar dos participantes cuyas respuestas cayeran fuera del valor de los cuartiles 0.25 y 0.75, y todavía determinar una correlación lineal. Aunque observó

que con 4 participantes los resultados eran aceptables. Su estudio menciona que en un grupo pequeño puede existir un nivel significativo de “ruido” por lo que es básico utilizar la mediana de las respuestas individuales, aunque no sea sensible a dispersiones grandes. Si se utiliza la mediana en la determinación de la desviación relativa, se tendría el error mediana del grupo (MGE, por sus siglas en inglés Median Group Error).

9.1.3 Recopilación de opiniones usando el método Delphi

La aplicación (tarea marcada con el número 3 en la Figura 9.1) se realizó en fases, de acuerdo al método Delphi, utilizando como criterios de paro para las rondas de aplicación de encuesta: 1) un valor menor o igual al 15% de cambios en las respuestas dadas por los participantes y 2) una convergencia en las respuestas de por lo menos 80%. Estos valores de 15% y 80% indican que las respuestas dadas por los participantes alcanzaron una estabilidad y un grado de consenso adecuados, respectivamente.

Fase 1.

Se le pidió a cada usuario que evaluara los requerimientos/recursos con base en su experiencia, pudiendo además agregar observaciones a las encuestas. Por cuestiones de disponibilidad de expertos y de recursos, la encuesta se aplicó a 6 participantes. Se mantuvo la confidencialidad de las respuestas de los demás participantes. Las encuestas se aplicaron por separado para evitar que los participantes influenciaran las respuestas que dieron los demás. Se trató de evitar al máximo el efecto líder, que implica que los participantes evalúen con respecto a lo que dice otro de ellos. Se promueve que se basen en su experiencia o conocimientos.

Fase 2.

A los 6 participantes se les aplicó nuevamente la misma encuesta, dándoles además los valores de las respuestas de la 1ª fase de todos los participantes, identificando su respuesta (indicando el número de participante que le corresponde a cada encuestado, al inicio de la encuesta, así ellos se podrían identificar en las gráficas), sin mencionar a quien pertenecían las demás respuestas. Con el objetivo de que decidieran si cambiaban sus respuestas o si mantenían su opinión. Desgraciadamente un participante no pudo continuar en el proceso y se descartó por falta de disponibilidad. Por lo que, esta fase se evaluó sólo con 5 participantes.

De esta manera se pudo determinar si el porcentaje de cambios era el indicado para detener la aplicación de encuestas o bien si era necesaria la aplicación de encuestas en una nueva fase de evaluación. Se consideró que en esta fase se consiguieron resultados satisfactorios para el análisis, debido a que los cambios en las respuestas fueron menores al 15% (4.4%), y la convergencia en las mismas fue mayor al 80% (84.2%). Se consideró que estos porcentajes indican un nivel aceptable de estabilidad y convergencia en las opiniones, y que era conveniente iniciar el análisis de los resultados del proceso.

Las alternativas consideradas para la toma de decisiones, dependiendo de los criterios de paro fueron: 1) si se cumplían, se consideraban los resultados de la última muestra como aceptables, 2) si no se cumplían, se tendría que volver a realizar la aplicación de la encuesta (otra fase) y buscando la convergencia efectuar reuniones grupales, para contrastar opiniones y lograr que se cumplan las condiciones necesarias para detener las encuestas.

Encuestas de contraste

En el método Delphi se requiere tener las respuestas esperadas de todos los elementos de la encuesta. En esta investigación no se cuenta con valores esperados de las evaluaciones de los requerimientos, por lo que se decidió considerar como valores esperados los asignados por el analista (el autor de esta tesis), al responder el cuestionario utilizado, en dos fases como los demás participantes. Esta decisión se tomó considerando la experiencia obtenida por el analista, mediante la obtención y especificación de requerimientos. Las respuestas del analista, en las dos fases realizadas, se promediaron y se utilizaron para determinar la desviación relativa respecto a las respuestas esperadas.

9.1.4 Validación de fiabilidad de la encuesta

Para validar la fiabilidad de un instrumento de medición (tarea marcada con el número 4 en la Figura 9.1), se utiliza la consistencia de un resultado, la estabilidad temporal y la concordancia de los inter-observadores (Cohen 1996). Esto es, la encuesta será consistente después de pruebas repetidas, utilizando los mismos o diferentes valores, para reactivos iguales o similares. Cada una de las técnicas desarrolladas para probar la fiabilidad proporciona una medida para un aspecto específico.

Para este proceso se verificó la fiabilidad de la encuesta mediante métodos estadísticos. Para esto se dividió el proceso en grupos, la primera fase de aplicación de la encuesta fue el grupo uno, la segunda fase de aplicación fue el grupo dos y finalmente, las encuestas llenadas por el analista fueron el grupo tres. Este último, se utilizó como respuestas esperadas, para determinar la desviación estándar.

Para cada grupo se realizó un análisis para ver los resultados obtenidos, además se realizó un análisis general tomando en consideración los resultados de todos los grupos. En el caso de este trabajo se evalúa la fiabilidad de los tres factores mostrados en la columna uno, de la

Tabla 9.1.

Tabla 9.1. Fiabilidad y métodos utilizados.

Fiabilidad evaluada mediante:	Con el Método:
La consistencia	Coeficiente alfa de Cronbach
La estabilidad temporal	Coeficiente de correlación
La concordancia de los inter-observadores	Índice Kappa

Consistencia de la escala utilizada

Para cada uno de los grupos se realizó el cálculo del coeficiente alfa de Cronbach, con la finalidad de medir la fiabilidad de la escala de medida utilizada para la encuesta.

Para realizar el cálculo del coeficiente alfa de Cronbach, se tienen dos métodos uno utiliza las varianzas de cada uno de los elementos y otro método usa el coeficiente de correlación. Para nuestro caso se utilizaron ambos métodos para comprobar los resultados.

Como primer paso se toman los resultados obtenidos por cada elemento (requerimiento) de cada grupo y se obtiene su varianza, después se obtiene la sumatoria de las varianzas obtenidas (suma de varianzas), una vez que se tiene este resultado debemos obtener la varianza de la sumatoria de todos los valores del grupo (varianza de la sumatoria), utilizando también el número de elementos que

tiene la encuesta (k). Se utiliza la siguiente formula y obtenemos el coeficiente Alfa de Cronbach, mediante varianzas.

$$\text{Alfa de Cronbach} = \frac{k}{k-1} * \left(1 - \frac{\text{suma de varianzas}}{\text{varianza de la sumatoria}} \right)$$

Para el cálculo del coeficiente utilizando la correlación, se tiene que obtener primero la matriz de correlación, en nuestro caso se utilizó la matriz de coeficiente de correlación de Pearson. Primero se obtiene el promedio de los resultados obtenidos en la matriz (p), luego obtenemos el número de elementos que tiene la encuesta (k). Se utiliza la siguiente fórmula para obtener el coeficiente alfa de Cronbach mediante el coeficiente de correlación.

$$\text{Alfa de Cronbach} = \frac{k p}{1 + p (k-1)}$$

Mediante el cálculo del coeficiente Alfa de Cronbach, utilizando los dos métodos mencionados, se obtuvieron los resultados que se muestran en la Tabla 9.2.

Tabla 9.2. Valores obtenidos para consistencia.

Grupo	Valor de consistencia utilizando:	
	Varianza de los ítems	Correlación
1	0.98134408	0.98275733
2	0.98400689	0.98531209
3	0.98722519	0.98937934
General	0.982206347	0.983364109

Si se truncan los valores hasta los centésimos, tenemos valores de 0.98 para los métodos de varianza y correlación, lo que indica que existe una buena consistencia interna.

En todos los casos el valor obtenido para el coeficiente es cercano a 1 y está dentro de los límites de 0.7 a 1.0, que se especifican como aceptables, lo que indica que la escala de medida utilizada es confiable.

La mayoría de los resultados tienen variaciones dentro de los valores de milésimas, estas variaciones se deben directamente a las respuestas dadas por los participantes. Se observa, por ejemplo como entre los grupos 1 y 2 hay un aumento

de 2 milésimas aproximadamente, causado por la mayor concordancia entre las respuestas de los participantes, en la 2ª fase de aplicación de la encuesta.

Con respecto al grupo 3 que corresponde a las encuestas del analista, se reporta el promedio de los valores obtenidos, donde se observa un ligero aumento en la consistencia con respecto a los demás grupos, debido a que estas encuestas se llenaron por una misma persona en 2 fases, los valores fueron muy similares.

Estabilidad temporal de los valores asociados a los requerimientos

Para evaluar la estabilidad temporal se utilizó el coeficiente de correlación de Pearson. Se evaluó cada grupo, por separado, los cálculos se realizaron mediante el programa de hoja de cálculo Excel. Para calcular la correlación, podían usarse funciones o computarla de forma directa, utilizando Excel. Se decidió obtenerla mediante ambos métodos, para comprobar los resultados.

Primero se tomaron los resultados obtenidos de cada grupo y se compararon los resultados del primer elemento con los demás resultados mediante la siguiente función =COEF.DE.CORREL(rango1,rango2), donde los valores de los rangos estaban dados por los resultados de las respuestas a comparar (ej. 0,2,4,6). Luego se tomaron los resultados del segundo elemento y se compararon con los demás (ej. 0,2,4,6), de esta manera se repitió la comparación con todos los elementos, hasta obtener una matriz de resultados. Este sería un proceso tedioso, si se hace de forma manual, ya que se requiere comparar de 2 en 2 los resultados.

Una vez realizado el procedimiento de la matriz de correlación, se obtuvo el promedio general por cada grupo. Los resultados se presentan en la Tabla 9.3.

Tabla 9.3. Valores obtenidos para estabilidad.

Grupo	Estabilidad	Interpretación
1	0.58761042	Existe una correlación positiva
2	0.62645877	Existe una correlación positiva
3	0.69960078	Existe una correlación positiva
General	0.596412088	Existe una correlación positiva

En todos los grupos el resultado está dentro de los rangos aceptables de 0 a 1, notando un aumento en el tiempo, de la estabilidad de las encuestas de un primer momento en el grupo 1 y un segundo momento en el grupo 2. Este aumento indica

que en la segunda fase los participantes tenían respuestas más parecidas. En el caso del grupo 3, la correlación es aún mayor debido a la similitud en las respuestas por parte del analista.

Considerando el valor de la escala para este procedimiento, se tiene como resultado que existe una correlación positiva entre los elementos de la encuesta. Esto nos indica que existe una dependencia directa entre las respuestas dadas por los participantes para cada elemento (requerimientos), con una tendencia a aumentar la correlación, cuando las respuestas de los participantes son similares, o a disminuirla, cuando las respuestas cambian de un participante a otro.

Concordancia de los inter-observadores

Se utiliza el índice Kappa para evaluar a cada grupo por separado y se analizan los resultados obtenidos de las encuestas realizadas en la última fase (la segunda fase en este caso), este método permite únicamente comparaciones de 2 en 2, por lo que en el caso de 5 participantes (Ps) se realizaron 10 comparaciones (ej. P1-P2, P1-P3, P1-P4, P1-P5, P2-P3, P2-P4, P2-P5, P3-P4, P3-P5 y P4-P5) para cada característica evaluada mediante las encuestas (para dos aspectos: Beneficio y Factibilidad).

El procedimiento consiste en comparar y medir el número de concordancias y discrepancias en las respuestas de cada participante con respecto a los demás, para esto se hace una matriz en la que se registran los resultados obtenidos del análisis de concordancias y discordancias, después se obtienen el total de elementos que coinciden totalmente, sumándolos, y después se dividen entre el total de elementos de la encuesta. El resultado obtenido es P_o (Porcentaje observado de acuerdo entre participantes). Luego se obtienen la matriz de concordancias debidas al azar, la que se obtiene calculando los elementos de su diagonal, mediante la multiplicación de los valores de las orillas de la matriz de concordancias (suma de concordancias en el renglón multiplicada por la suma de concordancias en la columna), y dividiendo este resultado entre el total de elementos que concuerdan. El valor de P_e (Porcentaje esperado de acuerdo entre participantes), se genera con la suma de los valores de la matriz de concordancias debidos al azar, que coinciden exactamente para los dos participantes evaluados, dividida entre el total de elementos. Una vez que se tienen estos valores se evalúa la fórmula para obtener el índice kappa.

$$\text{kappa} = \frac{P_o - P_e}{1 - P_e}$$

Los valores obtenidos para kappa se colocan nuevamente dentro de una matriz para realizar el cálculo de promedios obtenidos para los dos aspectos: Beneficio y Factibilidad. Este procedimiento se repitió para cada uno de los grupos.

La concordancia se midió con base en los aspectos: factibilidad y beneficio, para cada grupo, obteniendo los resultados presentados en la Tabla 9.4.

Tabla 9.4. Valores obtenidos para concordancia de los inter-observadores.

Grupo	Valor de Concordancia			
	Factibilidad	Beneficio	Promedio	Análisis
1	0.059464	0.012010	0.035737	Leve
2	0.039008	0.105602	0.072305	Leve
3	0.595142	0.696203	0.645672	Considerable
General	0.231205	0.271272	0.251238	Aceptable

Se puede observar entre los grupos 1 (primer momento en el tiempo) y 2 (segundo momento en el tiempo) una disminución en la concordancia para factibilidad, y un aumento para beneficio en el grupo 2 (factibilidad=0.039008, beneficio=0.105602) respecto al grupo 1 (factibilidad=0.059464, beneficio=0.012010). En la segunda etapa del proceso Delphi los participantes coincidieron en ser más optimistas, ya que, para los elementos, disminuyeron su percepción de factibilidad y aumentaron su percepción de beneficio. En el caso del grupo 3 la concordancia de las respuestas es mayor debido a la similitud de las encuestas por parte del analista.

Los grupos 1 y 2 se encuentran dentro del rango de 0.01 a 0.20, que de acuerdo a la escala del índice kappa, equivale a una concordancia leve, es decir los participantes difieren en sus respuestas y opiniones. El grupo 3 se encuentra dentro del rango de 0.61 a 0.80 que de acuerdo a la escala del índice kappa equivale a una concordancia considerable.

Considerando en general, a los tres grupos como uno solo, la mayoría se encuentra dentro del rango aceptable para la escala del índice kappa, dentro del intervalo de 0.21 a 0.40. Esto significa que los participantes coinciden en su medición de forma aceptable, situación que puede deberse a la cantidad reducida de personas que participaron en la realización de las encuestas y a la variedad en las opiniones dadas por cada participante.

9.1.5 Validación de requerimientos mediante diagramas GRL

Una vez realizada la validación de la fiabilidad de la encuesta aplicada, se colocaron valores dentro de cada uno de los elementos de los diagramas, tomando como base los datos, obtenidos de la 2ª fase de las encuestas aplicadas a los 5 participantes. Los elementos (metas, recursos y tareas) que forman los diagramas, en el jUCMNav se utilizaron para listar los requerimientos, en forma genérica. Es decir sin identificar cuáles requerimientos son funcionales, no funcionales o recursos. En la explicación que se presenta a continuación se mencionan sólo los elementos de diagramas GRL, sin hacer referencia a los requerimientos asociados.

Los valores a colocar en los diagramas se normalizaron antes de incluirlos, para resaltar las evaluaciones con valores superiores al valor medio de la escala (5), y minimizar las evaluaciones inferiores al valor medio. Así se obtiene una respuesta con valores binarios, otorgando a cada respuesta valores de 0 o 1. El valor es 1 si la respuesta es igual o mayor a 5 (considerado como “Normal”), en caso de que la respuesta sea menor a 5 su valor es 0. De esta forma se pretende evitar falsos positivos o falsos negativos.

Una vez normalizados los datos, se realiza el conteo de 1s para cada elemento (meta/recurso/tarea), luego se realiza el cálculo del porcentaje para cada elemento. El porcentaje se calcula mediante el número de respuestas con valor 1, dividido entre el número total de respuestas para el elemento. Este porcentaje se utiliza para otorgar la evaluación numérica (o peso) a cada elemento dentro de los diagramas generados en GRL.

La herramienta jUCMNav permite colocar dos valores de evaluación: Evaluación numérica y Contribución numérica. En la Evaluación numérica se colocan los valores obtenidos del análisis de las encuestas (colocado en la parte superior del símbolo del elemento, ver Figura 9.2).

Los valores obtenidos en las encuestas se asignan como evaluación numérica de los elementos que no tienen hijos, que están hasta el final de las ramas de los diagramas. Por lo tanto, los valores de todos los elementos padre se calculan a partir de estos. Por ejemplo en la Figura 9.2, en el caso de los elementos del “Diagrama de la fase de inserción de una nueva ontología” (meta 1.1.3.1.1.1 de la Figura 7.3 Diagrama GRL general del sistema AUPAs), la meta “1.1.3.1.1.1 Ingresar nueva Ontología” tiene 3 sub-elementos (en este caso son tareas), para los cuales se

calculan los valores de evaluación numérica a asignar, usando los valores obtenidos del análisis de las encuestas (en porcentajes), ingresando el valor (peso) a la parte superior de cada elemento del diagrama GRL.

En este ejemplo, el primer sub-elemento (1.1.3.1.1.1.1), fue evaluado de forma positiva (arriba del valor medio de la escala) por el 90% de los participantes, el segundo (1.1.3.1.1.1.2), fue evaluado positivamente por el 70% de los participantes y por último el tercero (1.1.3.1.1.1.3) también evaluado positivamente por el 70%. Así, el valor para la evaluación numérica, a considerar para la tarea superior (1.1.3.1.1.1) se calcula de la siguiente manera: $= (90+70+70) / 3$ lo que nos da un total de 76.7. Esto significa que del total de las sub-tareas, se tiene una evaluación numérica del 77% de los participantes, después de redondear.

A la tarea superior (1.1.3.1.1.1) se le asigna el valor 77 a la evaluación numérica. Así la tarea superior (1.1.3.1.1.1), cuando se utilice en el cálculo, junto con sus hermanos aportará un valor de evaluación numérica de 77, que se sumará con las evaluaciones numéricas de sus hermanos para determinar el valor de la evaluación numérica de su elemento padre. De esta manera se obtienen los valores para cada elemento padre en los diagramas GRL.

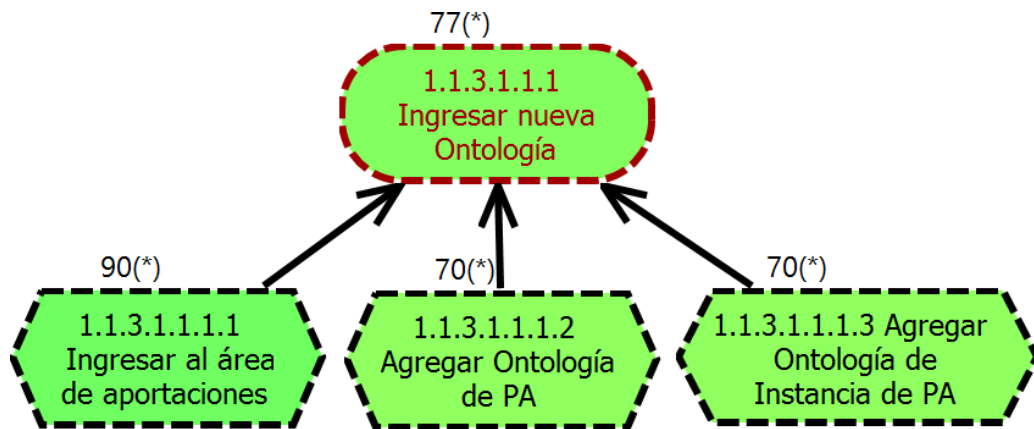


Figura 9.2. Ejemplo de asignación de valores.

Una vez establecido esto, se indica al jUCMNav que evalúe la estrategia de evaluación numérica establecida mediante los resultados de las encuestas. El resultado que presenta, esta opción de la herramienta, es colorear los diagramas GRL. Coloca rangos de colores de acuerdo a los valores que están dentro de la

Evaluación numérica (valores fuera de las figuras en la parte superior), estos colores van desde el rojo hasta el verde, dependiendo del valor numérico que tenga asignado. El rango de colores que maneja jUCMNav se muestra en la Tabla 9.5.

Tabla 9.5. Rango de colores otorgados por jUCMNav.

Evaluación numérica	Rango de colores	Evaluación numérica	Rango de colores
-100	-100(*) Meta1	25	25(*) Meta6
-75	-75(*) Meta2	50	50(*) Meta7
-50	-50(*) Meta3	75	75(*) Meta8
-25	-25(*) Meta4	100	100(*) Meta9
0	0(*) Meta5		

Dentro de este capítulo se describió el procedimiento para analizar la fiabilidad de la encuesta, que aplica métodos estadísticos, y se puede realizar utilizando fórmulas y funciones de Excel. Con base en los resultados obtenidos mediante las pruebas de consistencia, estabilidad temporal y concordancia, se determinó que la fiabilidad de la encuesta aplicada es aceptable.

En la Figura 9.3 se presenta el diagrama GRL evaluado, de la validación de los requerimientos. Observando el diagrama se determinó que el 100% de los

elementos de los diagramas se consideran prioritarios para integrarse en el diseño del sistema en una primera etapa.

En la Tabla 9.6 se observa que la totalidad de los requerimientos están por encima del valor 5 que equivale a una factibilidad y beneficio normal, lo que indica que estos requerimientos/tareas/recursos pueden integrarse en una primera etapa de diseño del sistema.

Tabla 9.6. Requerimientos superiores a la media.

Requerimientos	Requerimientos mayores superiores a escala considerada como normal (5)			
	Grupo 1	Grupo 2	Grupo 3	General
Aceptados	40	40	40	40
Rechazados	0	0	0	0



Figura 9.3 Diagrama GRL General evaluado.

A lo largo de este capítulo se demostró que es factible la especificación de requerimientos presentada en el Capítulo 7, mediante la utilización del método Delphi cuyos resultados fueron exitosos, con esto queda sustentada la investigación, y lo único que nos resta es revisar las conclusiones generales así como los posibles trabajos futuros, mostrados en el siguiente capítulo (Capítulo 10).

Capítulo 10 *C*onclusiones

En este capítulo se presentan las conclusiones, las aportaciones obtenidas mediante esta investigación y se indican los trabajos futuros.

10.1 Conclusiones

De acuerdo a los resultados presentados podemos concluir que se cumplió con el alcance y el objetivo general establecido en la tesis ya que se alcanzaron todos y cada uno de los objetivos específicos (ver Capítulo 1 sección 1.3.2 Objetivos específicos):

- Se construyó un individuo de la ontología OPPAE con el patrón de análisis de reservación y uso de entidades reutilizables, el cual se muestra en la sección 6.1.1.
- Se generó una metodología para la construcción de ontologías de patrones de análisis (MECOPA), la cual se explica en el Capítulo 5.
- Se construyó una ontología con base en dicha metodología para el patrón de análisis de reservación y uso de entidades reutilizables (OPARE), cuya documentación se encuentra en el Capítulo 6.
- Se propuso la estructura de un repositorio (ver sección 8.3), para almacenar los tres niveles de abstracción de ontologías mencionados en el objetivo general (ver Capítulo 1 sección 1.3.1 Objetivo general).
- Asimismo se realizó la especificación de requerimientos y el diseño de alto nivel del sistema AUPAs (Capítulo 7 y Capítulo 8, respectivamente).

Después de haber sometido la especificación de requerimientos por un proceso de validación basado en encuestas a expertos (método Delphi), se determinó la factibilidad para desarrollar el sistema de almacenamiento y uso de patrones de análisis.

Con la incorporación de semántica al patrón de análisis para reservación y uso de entidades reutilizables mediante ontologías, se lograron entre otras cosas: la formalización del patrón mediante la utilización del lenguaje web de ontologías OWL (ver sección 6.1.6), la facilitación de su almacenamiento, la adecuación de la descripción del patrón para adaptarse a diferentes dominios, así como la eliminación de ambigüedades.

Con la incorporación de diagramas GRL y UCM a los requerimientos se logra su representación visual, esto es una ventaja ya que por lo regular los requerimientos se expresan de forma textual, dificultando su manejo, sin embargo con estos diagramas se obtiene un mejor entendimiento de los requerimientos y la posibilidad de mapearlos más fácilmente para los modelados posteriores como el diseño. Además de que los diagramas GRL contemplan la representación de requerimientos tanto funcionales como no funcionales, lo que traduce en una especificación de requerimientos completa.

Es complicado plasmar en un proyecto de dos años, todas las ideas y trabajar en el desarrollo de las mismas, en lo que respecta a la solución de la poca utilización de los patrones de análisis. Sin embargo, este trabajo intenta sentar las bases para difundir y facilitar la utilización de patrones de análisis. Queda mucho camino por recorrer pero esta investigación representa una gran base para que surjan más ideas o se desprendan más proyectos con la finalidad de innovar el desarrollo de software y aprovechar al máximo las ventajas que nos dan los patrones de análisis.

10.2 Aportaciones

Las principales aportaciones de este trabajo de tesis son:

1. La propuesta de una metodología para la construcción de ontologías de patrones de análisis (MECOPA).
2. El refinamiento de los requerimientos del patrón de análisis para reservación y uso de entidades reutilizables (Fernandez & Yuan 1999) mediante la representación de los mismos con diagramas GRL y UCM.
3. La construcción de una ontología para el patrón de análisis para reservación y uso de entidades reutilizables (Fernandez & Yuan 1999).
4. La especificación de requerimientos y el diseño de alto nivel del sistema AUPAs.
5. La documentación del proceso de validación de los requerimientos de software.
6. La identificación de una estructura de base de datos para el almacenamiento de ontologías y patrones de análisis.

10.3 Trabajos futuros

Esta tesis es la base para el diseño detallado y la implementación del sistema AUPAs.

Adicionalmente, se identificaron otras áreas para continuar la investigación, algunas de estas son:

1. Incluir más patrones de análisis en el repositorio.
2. Llevar a cabo la construcción de más ontologías de otros patrones de análisis, para incluirlas en el sistema.
3. Llevar a cabo el desarrollo de ontologías de instancia de patrones de análisis, para completar los tres niveles de ontologías que contemplará el sistema.
4. Realizar pruebas al sistema.

Referencias

- (Alexander et al. 1977) Alexander, C., Ishikawa, S. & Silverstein, M., 1977. A Pattern Language. *Ch. Alexander*, pp.1–1218.
- (Amyot 2003) Amyot, D., 2003. Introduction to the User Requirements Notation: learning by example. *Computer Networks*, 42(3), pp.285–301.
- (Amyot & Mussbacher 2001) Amyot, D. & Mussbacher, G., 2001. Bridging the Requirements/Design Gap in Dynamic Systems with Use Case Maps (UCMs). In *Proceedings of the 23rd International Conference on Software Engineering. Toronto, Ontario, Canada: IEEE Computer Society.*
- (Bobkawska & Grabowski 2009) Bobkawska, A. & Grabowski, J., 2009. The Role of Analysis Patterns in Systems Analysis. In *EuroPLoP 2009: 14th Annual European Conference on Pattern Languages of Programming.*
- (Borst 1997) Borst, W.N., 1997. *Construction of engineering ontologies for knowledge sharing and reuse,*
- (Braga et al. 1998) Braga, R.T.V., Germano, F.S.R. & Masiero, P.C., 1998. A Confederation of Patterns for Resource Management. *Pattern Languages of Programs 4*, 2002, pp.1–13.
- (Brockhoff 2002) Brockhoff, K., 2002. The Performance of Forecasting Groups in Computer Dialogue and Face-to-face Discussion. In *The Delphi method: Techniques and applications.* pp. 285–311.
- (Cendón et al. 2005) Cendón, J. et al., 2005. Sistema informático para la gestión del conocimiento y su representación. El lenguaje +gbit.
- (Cohen 1996) Cohen, R.S., 1996. Adverse symptomatology and suicide associated with the use of methylenedioxymethamphetamine (MDMA, ecstasy). In *Biol Psychiatry.* pp. 819–820.

- (Corcho et al. 2003) Corcho, O., Fernández-López, M. & Gómez-Pérez, A., 2003. Methodologies, tools and languages for building ontologies. Where is their meeting point? *Data and Knowledge Engineering*, 46, pp.41–64.
- (Dehainsala et al. 2007) Dehainsala, H., Pierra, G. & Bellatreche, L., 2007. OntoDB: An Ontology-Based Database for Data Intensive Applications. In *Proc. of Database Systems for Advanced Applications (DASFAA'2007)*.
- (Engineering et al. 2009) Engineering, S., Committee, S. & Computer, I., 2009. *IEEE Std 1016-2009 (Revision of IEEE Std 1016-1998), IEEE Standard for Information Technology—Systems Design—Software Design Descriptions*,
- (Eriksson & Penker 2000) Eriksson, H.-E. & Penker, M., 2000. *Business Modeling with UML*, John Wiley & Sons, Inc.
- (Fayad & Altman 2001) Fayad, M.E. & Altman, A., 2001. An Introduction to Software Stability. *Communications of the ACM*, 44(9), pp.95–98.
- (Feier et al. 2005) Feier, C. et al., 2005. Towards Intelligent Web Services: The Web Service Modeling Ontology (WSMO).
- (Fernández Bonilla 2010) Fernández Bonilla, O.D., 2010. *Construcción de una Ontología Mediante Métodos Semiautomáticos*. Centro Nacional de Investigación y Desarrollo Tecnológico.
- (Fernandez 1998) Fernandez, E.B., 1998. Building systems using analysis patterns. *Proceedings of the third international workshop on Software architecture - ISAW '98*, pp.37–40.
- (Fernandez 2000) Fernandez, E.B., 2000. Stock Manager: An analysis pattern for inventories. In *Proc. of the Pattern Languages of Programs Conference (PLoP'00)*.
- (Fernandez & Yuan 1999) Fernandez, E.B. & Yuan, X., 1999. An analysis pattern for reservation and use of reusable entities. In *Proc. of the Pattern Languages of Programs Conference (PLoP'99)*.

- (Fernandez et al. 2000) Fernandez, E.B., Yuan, X. & Brey, S., 2000. Analysis pattern for the order and shipment of a product. In *Proc. of the Pattern Languages of Programs Conference (PLoP'00)*.
- (Fernández et al. 1997) Fernández, M., Gómez-Pérez, A. & Juristo, N., 1997. METHONTOLOGY: From Ontological Art Towards Ontological Engineering. In *Spring Symposium Series*. pp. 33–40.
- (Flores Vitelli 2011) Flores Vitelli, I., 2011. *Aplicación de METHONTOLOGY para la Construcción de una Ontología en el Domino de la Microbiología. Caso de Estudio: Identificación de Bacilos Gram Negativos no Fermentadores de la Glucosa (BGNNF)*, Caracas.
- (Fowler 1997) Fowler, M., 1997. *Analysis Patterns: Reusable Object Models* Primera Ed., Addison-Wesley Professional.
- (Galindo García 2011) Galindo García, J.A., 2011. *Generación de ontologías en OWL a partir de diagramas de clases de UML mediante MDA (Arquitectura Dirigida por Modelos)*. Centro Nacional de Investigación y Desarrollo Tecnológico.
- (Gamma et al. 1994) Gamma, E. et al., 1994. Design Patterns Elements of Reusable Object-Oriented Software.
- (Geyer-schulz & M 2001) Geyer-schulz, A. & M, I.E., 2001. Software Engineering with Analysis Patterns. , (November), pp.1–19.
- (Gruber 1993) Gruber, T.R., 1993. A Translation Approach to Portable Ontology Specifications by A Translation Approach to Portable Ontology Specifications. , (April).
- (Gruninger & Fox 1995) Gruninger, M. & Fox, M.S., 1995. Methodology for the Design and Evaluation of Ontologies. *Industrial Engineering*, 95, pp.1–10.
- (Hamza & Fayad 2002) Hamza, H. & Fayad, M.E., 2002. Model-based Software Reuse Using Stable Analysis Patterns. In *ECOOP02 - Workshop on Model-based Software Reuse*.

- (Hamza 2005) Hamza, H.S., 2005. Developing business object models with patterns and ontologies. *Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications - OOPSLA '05*, p.106.
- (IEEE 1998) IEEE, 1998. *IEEE Recommended Practice for Software Requirements Specifications*,
- (ITU-T 2012) ITU-T, 2012. *User Requirements Notation (URN) - Language definition*,
- (Jean et al. 2007) Jean, S. et al., 2007. OntoDB: It is Time to Embed your Domain Ontology in your Database. In *Proc. of Database Systems for Advanced Applications (DASFAA'2007)*.
- (Kimmel 2008) Kimmel, P., 2008. *Manual de UML*, McGraw-Hill.
- (L. McGuinness & van Harmelen 2004) L. McGuinness, D. & van Harmelen, F., 2004. OWL Web Ontology Language Overview. *W3C Recommendation*. Available at: <http://www.w3.org/TR/2004/REC-owl-features-20040210/>.
- (Linstone & Turoff 2002) Linstone, H.A. & Turoff, M., 2002. *The Delphi Method - Techniques and applications*,
- (Martínez Jiménez 2013) Martínez Jiménez, R.C., 2013. *Resultados de la tesis de maestría: "Plantilla estable de patrones de análisis."* Centro Nacional de Investigación y Desarrollo Tecnológico.
- (da Matta Vegi et al. 2011) Da Matta Vegi, L.F., Alves Peixoto, D. & Santos Soares, L., 2011. An Infrastructure Oriented for Cataloging Services and Reuse of Analysis Patterns.
- (Pantoquilha et al. 2003) Pantoquilha, M., Raminhos, R. & Araujo, J., 2003. Analysis Patterns Specifications: Filling the Gaps.

- (Pozo et al. 2007) Pozo, M.T., Gutiérrez, J. & Rodríguez, C., 2007. El uso del método Delphi en la definición de los criterios para una formación de calidad en animación sociocultural y tiempo libre. *Revista de investigación educativa*, 25, pp.351–66.
- (Raminhos et al. 2006) Raminhos, R. et al., 2006. A Systematic Analysis Patterns Specification. *8th International Conference on Enterprise Information Systems (ICEIS)*.
- (Stanford Center For Biomedical Informatics Research 2015) Stanford Center For Biomedical Informatics Research, 2015. Stanford Medicine. Available at: <http://bmir.stanford.edu/>.
- (Studer et al. 1998) Studer, R., Benjamins, V. & Fensel, D., 1998. Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25, pp.161–197.
- (Suárez-Figueroa et al. 2009) Suárez-Figueroa, M.C., Gómez-Pérez, A. & Villazón-Terrazas, B., 2009. How to write and use the ontology requirements specification document. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5871 LNCS(PART 2), pp.966–982.
- (Valencia Vargas 2011) Valencia Vargas, N.E., 2011. *Creación de una Ontología para el Dominio de e-Gobierno en México*. Centro Nacional de Investigación y Desarrollo Tecnológico.