

Tecnológico Nacional de México

Centro Nacional de Investigación
y Desarrollo Tecnológico

Tesis de Maestría

Editor para el Lenguaje de Dominio Específico
Diagramas de Entidad

presentada por

Ing. Karen Loreli Zaragoza Jiménez

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

Director de tesis

Dr. Moisés González García

Cuernavaca, Morelos, México. Junio de 2019.



SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MEXICO

Centro Nacional de Investigación y Desarrollo Tecnológico

"2019, Año del Caudillo del Sur, Emiliano Zapata"

Cuernavaca, Morelos a 20 de junio del 2019
OFICIO No. DCC/066/2019

Asunto: **Aceptación de documento de tesis**

DR. GERARDO V. GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial de la Ing. Karen Loreli Zaragoza Jiménez, con número de control M15CE098, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis profesional titulado "Editor para el lenguaje de dominio específico Diagramas de Entidad" 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

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

REVISOR 1

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

REVISOR 2

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

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

NACS/lmz

cenidet[®]
Centro Nacional de Investigación
y Desarrollo Tecnológico

Interior Internado Palmira S/N, Col. Palmira, C. P. 62490, Cuernavaca, Morelos.
Tel. (01) 777 3 62 77 70, ext. 4106, e-mail: dir_cenidet@tecnm.mx
www.tecnm.mx | www.cenidet.edu.mx





SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MEXICO

Centro Nacional de Investigación y Desarrollo Tecnológico

"2019, Año del Caudillo del Sur, Emiliano Zapata"

Cuernavaca, Mor., 20 de junio de 2019
OFICIO No. SAC/230/2019

Asunto: Autorización de impresión de tesis

ING. KAREN LORELI ZARAGOZA JIMÉNEZ
CANDIDATA AL GRADO DE MAESTRA 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 "Editor para el Lenguaje de Dominio Especifico Diagramas de Entidad", ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

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

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

DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO



SEP TecNM
CENTRO NACIONAL
DE INVESTIGACIÓN
Y DESARROLLO
TECNOLÓGICO
SUBDIRECCIÓN
ACADÉMICA

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

GVGR/mcr

cenidet®
Centro Nacional de Investigación
y Desarrollo Tecnológico

Interior Internado Palmira S/N, Col. Palmira, C. P. 62490, Cuernavaca, Morelos.
Tel. (01) 777 3 62 77 70, ext. 4106, e-mail: dir_cenidet@tecnm.mx
www.tecnm.mx | www.cenidet.edu.mx



Dedicatoria

A Dios, por darme la vida y estar siempre conmigo, guiándome en mi camino.

A mi mamá Juana, a quien recuerdo con mucho cariño, y sé que estaría orgullosa.

A mi madre por ser el pilar más importante y por demostrarme siempre su cariño y apoyo incondicional.

A mi padre por todo su esfuerzo y cariño.

A mis hermanas por estar siempre presentes, acompañándome y por el apoyo moral, que me brindaron a lo largo de esta etapa.

A mí, por todo el esfuerzo y dedicación para lograr este trabajo.

Agradecimientos

A CONACYT por el apoyo económico que durante el transcurso de la maestría recibí y ayudó a culminar esta meta.

Al CENIDET por darme la oportunidad de realizar mis estudios de maestría. Por abrirme sus puertas para realizar este proyecto personal y profesional.

A mi director de tesis, Dr. Moisés González García, por su orientación, apoyo y enseñanzas para llevar a cabo este trabajo de tesis, sobretodo su paciencia ante mi inconsistencia. ¡Muchas gracias!

A mis revisores de tesis, la Dra. Olivia Graciela Fragoso Díaz y al Dr. René Santaolaya Salgado por sus observaciones y sugerencias para mejorar este trabajo de investigación.

A la Dra. Olivia Fragoso por todo su apoyo y confianza para culminar de este proyecto de investigación.

Al Dr. René Santaolaya por la orientación y ayuda que me ha brindado en la realización de este proyecto de investigación.

A mi madre por su cariño, comprensión y colaboración durante este camino que no ha sido sencillo. Sin ti, esta meta no estaría cumplida. Lo hicimos posible.

A mi padre por sus consejos y por creer en mí en todo momento.

A mis hermanas, Lisset y Karla, por motivarme a seguir adelante y no darme por vencida cuando más lo he necesitado.

A Karla, Deisy, Amado, Víctor y Pavel por ser parte significativa de mi vida y ser como mi segunda familia. Gracias por su apoyo, comprensión y amistad.

A mis mejores amigas Mariela, Karen y Lupita, que conocí en la maestría. Gracias por estar conmigo durante todo el proceso, por su apoyo y esfuerzo para hacer lo imposible, posible.

A mis amigos, José Carlos (el hermano) por las largas pláticas y su guía en diversos temas, Robertico por los gratos momentos compartidos, y Gibrán, por sus palabras de aliento.

A los compañeros que conocí durante mi estancia en el CENIDET: Antonio, Jessica, Honorio y Dianita.

Al maestro José Luis por su comprensión y apoyo para concluir este periodo.

A todas las personas que me apoyaron, cada una a su manera y han hecho que este trabajo se realice con éxito: Ady, Libni, Inés, Marco, el maestro Felipe, Jenny, Kristian, Edgardo, Pedrito, Eduardo.

Resumen

Un modelo de dominio del problema es el conocimiento organizado y estructurado acerca de un problema, representa el vocabulario y los conceptos claves considerados para sintetizar el dominio de la aplicación. Sin embargo, la falta de experiencia o desconocimiento en el dominio del problema, de los desarrolladores, ocasiona la intervención constante de los expertos del dominio, además de obtener un modelo incompleto.

En el presente trabajo de investigación se tuvo por objetivo evaluar distintas herramientas de modelado que permitan la generación y aplicación de un perfil UML para generar un editor, que ayude en la construcción de los diagramas de entidades a nivel usuario, combinado y aplicación.

Una vez seleccionada la herramienta Enterprise Architect, y debido a que en la revisión del estado del arte no se encontraron herramientas de edición que permitan la construcción de tales diagramas; se analizó la utilización de esta herramienta, en el proceso de desarrollo de un perfil UML y los pasos para generar un editor mediante un perfil UML.

Construido el editor que permita la construcción de los diagramas de entidades, éste ayuda a los desarrolladores a definir un modelo de dominio del problema más completo, sin la intervención constante de los expertos del dominio.

Para validar la utilidad del editor resultante se documentaron las pruebas de esta herramienta, editando tres diagramas de entidad (a nivel usuario, combinado y aplicación), para el caso de estudio del centro de información del CENIDET.

Mediante la edición de los diagramas de entidad del caso de estudio se identificaron y documentaron los requisitos para especificar la transformación de varios diagramas de entidad a nivel usuario hacia un diagrama de entidad combinado. Se realizó esto debido a que el modelado realizado está en el contexto del Desarrollo Dirigido por Modelos (DDM), donde se automatizan las transformaciones.

Abstract

A domain model of the problem is the organized and structured knowledge about a problem, it represents the vocabulary and the key concepts considered to synthesize the domain of the application. However, the lack of experience or lack of knowledge in the domain of the problem, of the developers, causes the constant intervention of domain experts, in addition to obtaining an incomplete model.

In the present research work, the objective was to evaluate different modeling tools that allow the generation and application of a UML profile to generate an editor, which helps in the construction of the diagrams of entities at user, combined and application level.

Once the Enterprise Architect tool was selected, and because in the revision of the state of the art no editing tools were found that allow the construction of such diagrams; the use of this tool was analyzed in the process of developing a UML profile and the steps to generate an editor using a UML profile.

Built the editor that allows the construction of the diagrams of entities, this helps developers to define a domain model of the most complete problem, without the constant intervention of domain experts.

To validate the utility of the resulting editor, the tests of this tool were documented, editing three entity diagrams (user, combined and application level), for the case study of the CENIDET information center.

By editing the entity diagrams of the case study, the requirements to specify the transformation of several entity-level diagrams at the user level to a combined entity diagram. This was done because the modeling performed is in the context of Model-Driven Development (DDM), where transformations are automated.

Contenido

Resumen.....	I
Abstract.....	II
Lista de figuras.....	VII
Lista de tablas.....	IX
Acrónimos.....	XI
CAPÍTULO 1. INTRODUCCIÓN.....	1
1.1. Descripción del problema.....	2
1.2. Objetivos.....	2
1.2.1. Objetivo general.....	2
1.2.2. Objetivo específicos.....	3
1.3. Alcances.....	3
1.4. Limitaciones.....	3
1.5. Metodología de solución.....	3
1.6. Organización del documento.....	5
CAPÍTULO 2. ESTADO DEL ARTE.....	7
2.1. Antecedentes.....	8
2.1.1 Método de Desarrollo Arquitectónico en Grupo.....	8
2.1.2. Caracterización de las fronteras de los modelos MDA mediante UML.....	9
2.1.3. Definición e implementación de un perfil de UML para la adquisición de requerimientos funcionales centrados en el usuario.....	10
2.2. Trabajos relacionados.....	11
2.2.1. An EMF-based Toolkit for Creation of Domain Specif Data Services.....	11
2.2.2. Domain modeling as a basis for building a meshing tool software product line.....	11
2.2.3. A Tool-Supported Methodology for Validation and Refinement of Early-Stage Domain Models.....	12
2.2.4. An UML profile for modeling RESTful services.....	12
2.2.5. Perfil UML para el desarrollo de aplicaciones WAP.....	13
2.2.6. Una Introducción a los Perfiles UML.....	14
2.2.7. Análisis de los trabajos relacionados.....	14
CAPÍTULO 3. MARCO TEÓRICO.....	17

3.1. Conceptos generales.....	18
3.1.1. Modelo.....	18
3.1.2. Dominio.....	18
3.1.3 Modelo de dominio.....	18
3.1.4. Dominio del problema.....	19
3.1.5. Arquitectura Dirigida por Modelos (MDA).....	19
3.2. Diagramas de entidades.....	20
3.2.1. Diagrama de entidades a nivel usuario.....	21
3.2.2. Diagrama de entidades combinado.....	22
3.2.3. Diagrama de entidades a nivel aplicación.....	22
3.2.4. Representación de los diagramas.....	23
3.3.z UML (<i>Unified Modeling Language</i>).....	26
3.3.1. Perfil UML.....	26
3.3.2. Diagrama de Clases.....	29
3.4. Enterprise Architect.....	31
3.4.1. Generación dirigida por Modelos.....	32
3.4.2. Estructura de un proyecto.....	33
CAPÍTULO 4. DIAGRAMAS DE ENTIDAD Y SU PERFIL UML.....	35
4.1. Definición de los diagramas de entidades para un editor.....	36
4.1.1. Identificación de elementos y restricciones.....	36
4.2. Diseño del perfil UML.....	42
4.2.1 Creación del metamodelo.....	42
4.2.2. Definición del perfil UML.....	43
4.2.3. Identificación de los elementos UML a extender.....	44
4.2.4. Definición de los valores etiquetados.....	45
4.2.5. Definición de las restricciones.....	47
4.2.6. Diseño del perfil UML.....	48
4.3 Representación de los diagramas de entidades a partir del perfil UML.....	49
4.3.1. Diagramas de entidades conforme al perfil UML.....	49
4.3.2. Adecuación del diagrama de entidades a nivel usuario.....	51
4.3.3. Adecuación del diagrama de entidades combinado a nivel usuario.....	52

4.3.4. Adecuación del diagrama de entidades a nivel aplicación	53
CAPÍTULO 5. CREACIÓN DEL PERFIL UML PARA EL DSL: DIAGRAMAS DE ENTIDAD Y GENERACIÓN DE HERRAMIENTA DE EDICIÓN	54
5.1. Análisis y evaluación de las herramientas de modelado	55
5.1.1. Papyrus	55
5.1.2. Enterprise Architect	58
5.1.3. Análisis de otras herramientas de modelado	61
5.1.4. Tabla comparativa	61
5.2. Desarrollo del perfil UML en <i>Enterprise Architect</i>	64
5.2.1. Generar proyecto	64
5.2.2. Desarrollo del perfil UML	69
5.3. Instanciar el editor en Enterprise Architect	91
5.3.1. Ejemplo del editor	92
5.3.2. Pasos para construir un diagrama de entidades en Enterprise Architect.	94
CAPÍTULO 6. ESPECIFICACIÓN DE REQUERIMIENTOS PARA LA CONSTRUCCIÓN DE DIAGRAMA DE ENTIDADES COMBINADO	97
6.1. Transformación de diagramas ENU a diagrama EC	98
6.1.1. Requerimiento para la creación de un diagrama ENU	99
6.1.2. Requerimientos para la transformación de Diagramas ENU a un Diagrama EC.....	101
6.1.3. Resultado.....	105
CAPÍTULO 7. PRUEBAS Y RESULTADOS	112
7.1. Pruebas	113
7.1.1. Plan de Pruebas.....	113
7.1.2. Diseño de Pruebas.....	114
7.1.3. Casos de Pruebas	115
7.1.4. Procedimiento de Pruebas	115
7.1.5. Registro de Pruebas	116
7.1.6. Reporte de Pruebas.....	119
7.2. Resultados.....	120
7.2.1. Creación de un diagrama de entidades.....	120
CAPÍTULO 8. CONCLUSIONES	130
8.1. Introducción	131

8.2. Aportaciones	131
8.3. Trabajos futuros	132
Referencias	133
ANEXO A. PLAN DE PRUEBAS	135
A.1. Plan de pruebas.....	136
A.1.1. Introducción	136
A.1.2. Detalles del plan de pruebas del perfil UML.....	137
A.1.3. Administración de Pruebas	138
A.2. Diseño de pruebas.....	139
A.2.1. Introducción	139
A.2.2. Detalles del diseño de la prueba.....	140
A.3. Casos de pruebas.....	144
A.3.1. Introducción	144
A.3.2. Casos de Prueba	144
A.4. Procedimiento de pruebas	149
A.4.1. Introducción	149
A.4.2. Procedimiento de Prueba TPr-01.....	149
A.4.3. Procedimiento de Prueba TPr-02.....	151
A.4.4. Procedimiento de Prueba TPr-03.....	152
A.5. Registro de pruebas	154
A.5.1. Introducción	154
A.5.2. Registro de prueba TL -01.....	154
A.5.3. Registro de prueba TL - 02	157
A.5.4. Registro de prueba TL - 03.....	160
A.6. Reporte de pruebas.....	163
A.6.1. Introducción	163
A.6.2. Detalles.....	163
ANEXO B. CASO DE ESTUDIO.....	165
B.1. Introducción.....	166
B.2. Diagrama de entidades combinado	166
B.2.1. Identificación de los elementos.....	166
B.2.2. Construcción	172
B.2.3. Resultado.....	176

B.3. Diagrama de entidades a nivel aplicación	176
B.3.1. Identificación de los elementos.....	176
B.3.2. Construcción	179
B.2.3. Resultado.....	183

Lista de figuras

Figura 1.1 Metodología de solución.....	4
Figura 3.1 Diagrama de entidades a nivel usuario (1)	23
Figura 3.2 Diagrama de entidades a nivel usuario (3).....	24
Figura 3.3 Diagrama de entidades combinado	24
Figura 3.4 Diagrama de entidades combinado con frontera	25
Figura 3.5 Diagrama de entidades a nivel aplicación	25
Figura 3.6 Diagrama de clases	30
Figura 3.7 Diagrama de clases con la relación asociación	31
Figura 3.8 Estructura de un proyecto	34
Figura 4.1 Diagrama de entidades a nivel usuario (1)	40
Figura 4.2 Diagrama de entidades a nivel usuario (3)	40
Figura 4.3 Diagrama de entidad a nivel combinado.....	41
Figura 4.4 Diagrama de entidades a nivel aplicación	41
Figura 4.5 Modelo del dominio	42
Figura 4.6 Perfil UML para los diagramas de entidades.....	48
Figura 4.7 Adecuación del diagrama de entidades del usuario 1	51
Figura 4.8 Adecuación del diagrama de entidad a nivel usuario (2)	52
Figura 4.9 Adecuación del diagrama de entidades combinado	52
Figura 4.10 Adecuación del diagrama de entidad a nivel aplicación	53
Figura 5.1 Perfil UML elaborado en Papyrus.....	56
Figura 5.2 Aplicación del perfil UML (Papyrus)	57
Figura 5.3 Perfil UML elaborado en Enterprise Architect.....	59
Figura 5.4 Archivo XML generado de EA	59
Figura 5.5 Aplicación del Perfil UML (Enterprise Architect).....	60
Figura 5.7 Seleccionar la opción New Project	65
Figura 5.8 Guardar proyecto	65
Figura 5.9 Model	66
Figura 5.10 Agregar vista (View).....	66
Figura 5.11 Generar vista (View)	67
Figura 5.12 Vista (View).....	67
Figura 5.13 Agregar diagrama.....	68
Figura 5.14 Seleccionar diagrama	68
Figura 5.15 Área de trabajo para generar el diagrama	69
Figura 5.16 Panel Toolbox.....	70
Figura 5.17 Seleccionar opción Profile.....	70

Figura 5.18 Elemento MDG Technology.....	71
Figura 5.19 Nombre del perfil	71
Figura 5.20 Diagramas para generar el perfil UML.....	72
Figura 5.21 Diagrama Profile	72
Figura 5.22 Seleccionar elemento Metaclass.....	73
Figura 5.23 Agregar meta-clases (Class, Attribute y AssociationClass)	73
Figura 5.24 Metaclases: Class, Attribute y AssociationClass.....	74
Figura 5.25 Agregar estereotipos	74
Figura 5.26 Definir estereotipos: Entidad, Mensaje y RegistroMensaje	75
Figura 5.27 Agregar relación extensión entre estereotipos y metaclases	75
Figura 5.28 Relación Extensión.....	76
Figura 5.29 Valores etiquetados.....	76
Figura 5.30 Seleccionar Attributes	77
Figura 5.31 Agregar valores etiquetados	77
Figura 5.32 Profile DE.....	78
Figura 5.33 Generar archivo XML para el diagrama Profile.....	78
Figura 5.34 Guardar archivo Profile	79
Figura 5.35 Archivo XML del diagrama Profile	79
Figura 5.36 Diagrama Toolbox.....	80
Figura 5.37 Ventana Add ToolBox Page	80
Figura 5.38 Agregar estereotipos (Entidad, Mensaje y RegistroMensaje).....	81
Figura 5.39 Elementos de la caja de herramientas	81
Figura 5.40 Diagrama Toolbox.....	82
Figura 5.41 Guardar archivo XML del diagrama Toolbox	82
Figura 5.42 Área de trabajo para el digrama Diagram Profile.....	83
Figura 5.43 Seleccionar meta-clase	83
Figura 5.44 Diagram Profile	84
Figura 5.45 Archivo XML de Diagram Profile	84
Figura 5.46 Generate MDG Technology	85
Figura 5.47 Winzard MDG Technology	85
Figura 5.48 Crear archivo MST.....	86
<i>Figura 5.49 Especificaciones de la tecnología a crear.....</i>	86
Figura 5.50 Seleccionar meta-modelos para el archivo MTS	87
Figura 5.51 Seleccionar dirección de los archivos XML.....	87
Figura 5.52 Elegir archivo Profile.....	88
Figura 5.53 Elegir archivo Diagram Profile	88
Figura 5.54 Elegir archivo Toolbox.....	89
Figura 5.55 Resumen de los archivos seleccionados.....	89
Figura 5.56 Mensaje del archivo MTS generado exitosamente	90
Figura 5.57 Archivo MTS	90
Figura 5.58 Instanciar archivo MTS.....	91
Figura 5.59 Elegir archivo.....	91
Figura 5.60 Propiedades del editor.....	92
Figura 5.61 Generar un diagrama de entidades.....	93

Figura 5.62 Área de trabajo para construir un diagrama de entidades	93
Figura 6.1 Proceso para la construcción de un diagrama de entidades a nivel usuario	100
Figura 6.2 Proceso para la generación de un diagrama de entidades combinado	105
Figura 6.3 Diagrama de entidades a nivel Auxiliares de Biblioteca (AB)	106
Figura 6.4 Diagrama de entidades a nivel Jefa de organización bibliográfica (JOB).....	106
Figura 6.5 Diagrama de entidades combinado	111
Figura 7.1 Diagrama de entidades a nivel “Jefa de organización bibliográfica” a nivel CIM.....	121
Figura 7.2 Agregar Entidades	124
Figura 7.3 Definición de una entidad principal	125
Figura 7.4 Definición de una entidad secundaria	125
Figura 7.5 Nombre de la relación	126
<i>Figura 7.6 Relaciones entre entidades</i>	127
Figura 7.7 Agregar mensaje	127
Figura 7.8 Propiedades del mensaje	128
Figura 7.9 Diagrama de entidades a nivel usuario	128
Figura A.1 Diagrama de entidades a nivel usuario.....	146
Figura A.2 Diagrama de entidades combinado a nivel usuario	147
Figura A.3 Diagrama de entidades a nivel aplicación	148

Lista de tablas

Tabla 2.1 Comparativa de los trabajos relacionados.....	15
Tabla 3.1 Elementos de un diagrama de clase	29
Tabla 4.1 Elementos del diagrama de entidades (Warnier/Orr)	37
Tabla 4.2 Clasificación de las entidades.....	38
Tabla 4.3 Descripción de los estereotipos	43
Tabla 4.4 Asociación de los estereotipos y meta-clases.....	45
Tabla 4.5 Descripción de los elementos	46
Tabla 4.6 Elementos de la adecuación de los diagramas de entidades.....	50
Tabla 5.1 Elementos definidos.....	57
Tabla 5.2 Comparación de las herramientas de modelado	63
Tabla 6.1 Elementos utilizados en los diagramas de entidades.....	98
Tabla 6.2 Requerimientos para crear un diagrama ENU.....	99
Tabla 6.3 Requerimiento RF02	102
Tabla 6.4 Requerimiento RF02-01.....	102
Tabla 6.5 Requerimiento RF02-02	103
Tabla 6.6 Requerimiento RF02-03	104
Tabla 6.7 Entidades.....	107
Tabla 6.8 Relaciones del diagrama de entidades a nivel AU	107

Tabla 6.9 Relaciones del diagrama de entidades a nivel JOB.....	107
Tabla 6.10 Mensajes de la relación JOB a AB.....	108
Tabla 6.11 Mensajes de la relación JCI – AB	108
Tabla 6.12 Mensajes de la relación U – AB.....	108
Tabla 6.13 Mensajes de la relación U - JOB	109
Tabla 6.14 Mensajes de la relación de U – JOB	109
Tabla 6.15 Mensajes de la relación JCI – JOB	110
Tabla 7.1 Casos de Pruebas.....	114
Tabla 7.2 Pruebas a realizar	114
Tabla 7.3 Registro de la prueba TPr -01	117
Tabla 7.4 Datos empleados en la prueba	119
Tabla 7.5 Resultados obtenido de las pruebas.....	119
Tabla 7.6 Funcionalidades generales del editor	120
Tabla 7.7 Entidades identificadas.....	122
Tabla 7.8 Relación de Jefa de Organización Bibliográfica (JOB) y Auxiliares de Biblioteca (AB)	122
Tabla 7.9 Relación de Usuarios (U) y Auxiliares de Biblioteca (AB).....	122
Tabla 7.10 Relación de Jefa de Organización Bibliográfica (JCI) y Jefe del Centro de Información (JCI)	123
Tabla A.1 Nomenclatura para identificar los documentos generados.....	136
Tabla A.2 Matriz de trazabilidad de prueba.....	137
Tabla B.10 Relación RF y JCI	171
Tabla B.14 Entidades definidas (nivel aplicación)	177

Acrónimos

AGD	Desarrollo Arquitectónico en Grupo
MDA	Arquitectura dirigida por Modelos
MGD	Gereneración dirigida por Modelos
DSL	Lenguaje de Dominio Especifico
UML	Lenguaje Unificado de Modelado
OMG	Object Management Group
EA	Enterprise Architect
ENU	Entidad a nivel usuario
EC	Entidades combinado

CAPÍTULO 1. INTRODUCCIÓN

La elaboración de modelos de dominio del problema a veces resulta difícil por parte de los desarrolladores, debido a que no poseen un conocimiento extenso de este dominio, y tampoco manejan la terminología adecuada empleada por los expertos del dominio. Elaborar y analizar los diagramas de entidades propuesto por Wanier/Orr, que retratan a los usuarios involucrados y las entidades con las que intercambian productos de los procesos involucrados (entradas/salidas), es un medio para ayudar a definir un modelo de dominio del problema.

En esta investigación se presenta la definición e instanciación de un editor a través de un perfil UML, que permita la construcción de los diagramas de entidades.

En este capítulo se expone el problema a resolver y la propuesta general de solución. Asimismo, se incluyen los objetivos, alcances y limitaciones, establecidos para esta investigación.

1.1. Descripción del problema

Un modelo de dominio del problema es el conocimiento organizado y estructurado acerca de la operación en un dominio. Representa el vocabulario y los conceptos claves del dominio del problema. Considerando lo expresado antes, un dominio consta de dos partes importantes: 1) el dominio del problema, que define el entorno donde se trabajará, es decir, las ideas, el conocimiento y la información del problema que se intenta resolver, donde se incluyen las reglas, procesos y sistemas existentes que necesitan se sintetice una solución que mejore la operación; y 2) el dominio de solución, que define el entorno donde y los elementos incluidos en la solución. Así, en este entorno se provee una solución simple y estructurada basada en el conocimiento del problema específico.

La creación de los modelos de dominio requiere cierto conocimiento del dominio y el uso de la terminología apropiada de dicho dominio. Cuando un desarrollador desconoce o no tiene la experiencia suficiente en el dominio que se pretende soportar mediante una aplicación (por ejemplo, Sector Salud, Farmacéutico, Financiero, etc.), le resulta difícil realizar un modelo de dominio, y en ocasiones requiere la ayuda de los expertos del dominio para generarlo. Esta situación puede resultar en un modelo de dominio incompleto y/o software inestable. Aunque existen herramientas y metodologías que permiten la creación de modelos de dominio del problema, no existe una herramienta de edición que soporte la creación de diagramas de entidad Warnier/Orr incluidos en la metodología AGD (*Architecture and Group Development* (González, 2006)), que ayudan al modelado del dominio del problema. En este tipo de diagramas se identifican las entidades, relaciones, entradas y salidas que tendrá una aplicación.

1.2. Objetivos

1.2.1. Objetivo general

Desarrollar una herramienta de edición que permita la construcción de los diagramas de entidades utilizando un ambiente de modelado.

1.2.2. Objetivo específicos

Los objetivos específicos que se definen son:

- Creación de un perfil UML para los diagramas de entidades.
- Generación del editor para elaborar diagramas de entidades.
- Creación de diagramas de entidades para el caso de estudio.
- Especificación de requerimientos para automatizar la construcción de un diagrama de entidades combinado.

1.3. Alcances

A continuación, se presentan los alcances de esta investigación:

- Generar los diagramas de entidades: diagrama de entidades a nivel usuario, diagrama de entidades combinado y diagrama de entidades a nivel aplicación.
- Realizar un caso de estudio para validar el editor.
- Especificar los requerimientos para la transformación de varios diagramas a nivel usuario, hacia un diagrama de entidades combinado.

1.4. Limitaciones

Las limitaciones que se presentan en esta investigación:

- No se incluirá un análisis de la metodología AGD, ni la comparación con otras metodologías para el desarrollo de la aplicación.
- No se incluirá la definición de las transformaciones entre los diagramas propuestos.
- No se construirá un modelo de dominio del problema.

1.5. Metodología de solución

En la figura 1.1 se muestra el diagrama que representa la metodología de solución general para desarrollar esta investigación.

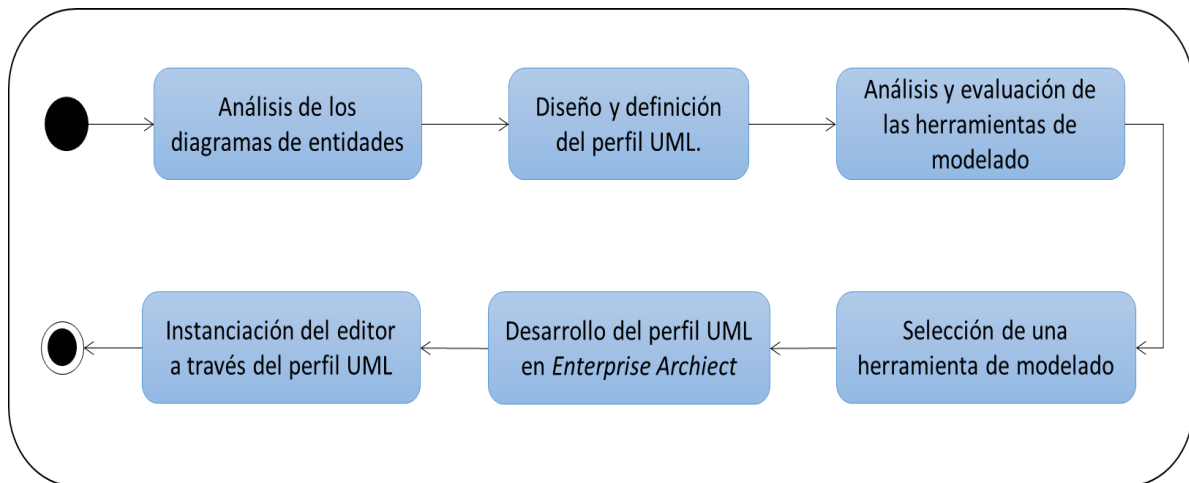


Figura 1.1 Metodología de solución

La investigación se dividió en 6 etapas:

1. **Análisis de los diagramas de entidades.**

En la primera etapa se analizó la construcción de los diagramas de entidades: diagrama de entidades a nivel usuario, diagrama de entidades combinado y diagramas de entidades a nivel aplicación para desarrollar el editor. Este análisis se hizo con el fin de identificar los elementos, restricciones y las características en general que poseen los diagramas de entidades, con base a lo estipulado en la metodología de *Warnier/Orr*.

2. **Diseño y definición del perfil UML.**

En esta etapa se decidió desarrollar el editor utilizando perfiles UML, para ello se requirió mapear los elementos identificados en la primera etapa siguiendo el procedimiento establecido por la OMG. Establecer las restricciones, meta-clases, y estereotipos que se emplearán en el perfil UML.

3. **Análisis y evaluación de las herramientas de modelado.**

En esta etapa se analizaron y evaluaron diversas herramientas de modelado, que permitirán el desarrollo e implementación de los perfiles UML, hasta la instanciación del editor.

4. **Selección de una herramienta de modelado.**

Derivado de la evaluación de las herramientas de modelado, se identificó la herramienta que se ajustó mejor al propósito de esta investigación,

considerando principalmente los siguientes criterios de evaluación: soporte para Perfiles UML, soporte XML para perfiles UML y existencia de documentación actualizada.

5. **Desarrollo del perfil UML en *Enterprise Architect*.**

Para el desarrollo del perfil UML se utilizaron los mecanismos de la herramienta de modelado *Enterprise Architect*, que facilita la elaboración de los perfiles UML a través del mecanismo MGD (*Model Driven Generation* (Ltd., 2019)).

6. **Instanciación del editor a través del perfil UML.**

En esta última etapa, se instanció el editor en el entorno *Enterprise Architect*, donde se definen los diagramas de entidades como un tipo de diagrama a generar. Se obtiene así una herramienta para editar un lenguaje de dominio específico (DSL, *Domain Specific Language*, (Gronback, 2009)): Diagramas de Entidad.

1.6. Organización del documento

Este documento se organiza de la siguiente manera:

Capítulo 2. Estado del arte. En este capítulo se exponen las investigaciones realizadas en CENIDET (antecedentes) y los trabajos relacionados (trabajos de investigación realizados fuera del CENIDET).

Capítulo 3. Marco Teórico. En este capítulo se definen los conceptos utilizados a lo largo de este documento, como: qué es un perfil UML, modelo, modelo del dominio del problema, entre otros

Capítulo 4. Diagramas de entidad y su perfil UML. En este capítulo se presenta el proceso realizado para diseñar el perfil UML, que contempla los elementos y la construcción de los diagramas de entidades.

Capítulo 5. Creación del perfil UML para el DSL: diagramas de entidad y generación de herramienta de edición. En este capítulo se muestra el desarrollo del perfil UML y la generación de la herramienta de edición.

Capítulo 6. Especificación de requerimientos para la construcción de diagrama de entidades combinado. En este capítulo se describe la especificación de requerimientos para construir los diagramas de entidades combinado.

Capítulo 7. Pruebas y Resultados. En este capítulo se exponen los resultados obtenidos de la investigación y los casos de pruebas con base al estándar de la IEE (*IEEE Computer Society, 2008*) para los documentos de prueba.

Capítulo 8. Conclusiones. En este capítulo se presentan las conclusiones obtenidas a partir de los resultados de la investigación, así como las aportaciones y trabajos futuros.

Anexo A. Plan de prueba. En este anexo se describen los documentos de las pruebas: plan de pruebas, casos de pruebas, diseño de pruebas, procedimientos de las pruebas y registro de las pruebas.

Anexo B. Caso de estudio. En este anexo se presentan la elaboración de los diagramas de entidades a nivel usuario y el diagrama de entidades combinado haciendo uso de la herramienta de edición.

CAPÍTULO 2. ESTADO DEL ARTE

En este capítulo se presentan los resultados del proceso de búsqueda de información relevante con respecto a las investigaciones realizadas en CENIDET y los trabajos relacionados que se utilizan como punto de referencia y comparación con el presente trabajo de investigación.

2.1. Antecedentes

En esta sección se describen de forma general las investigaciones realizadas en el Centro Nacional de Investigación y desarrollo Tecnológico, y una tesis de doctorado producto de la investigación doctoral del Doctor Moisés González García, miembro del *CENIDET*.

2.1.1 Método de Desarrollo Arquitectónico en Grupo

En este trabajo de tesis de doctorado realizado por (González, 2006), se define el método Desarrollo Arquitectónico en Grupo (*AGD, Architectural and Group Development*) con el propósito de asignarle mayor importancia al producto de software y su estructura, abarcando un proceso de grupo, con el que se produce cada uno de los productos del trabajo, requeridos para construir el resultado.

Esta metodología emplea un conjunto de productos de ingeniería que abarca los modelos de la Arquitectura Dirigida por Modelos (MDA): Modelos Independientes del Cómputo (*CIM, Computationally-Independent Model*), Modelos Independientes de la Plataforma (*PIM, Platform-Independent Model*) y Modelos de Plataforma Específica (*PSM, Platform-Specific Model*) e incluye dos modelos más: Modelo de Implementación (IM) y el Modelo de Operación (OM). Además, en MDA el conjunto de productos se organiza en una estructura gerárquica de cuatro niveles: PS1Set (el nivel más abstracto), PS2Set, PS3Set, y de el cuarto nivel (el nivel más detallado: modelos o productos específicos).

En el *CIM*, entre otros, se incluyen tres diagramas: diagramas de entidades a nivel usuario, diagramas de entidades combinado de usuarios y el diagrama de entidades de la aplicación, para mostrar una representación abstracta y estática, de la forma en que trabaja la organización. Estos diagramas son versiones modificadas de los diagramas de entidades propuesto por *Ken Orr*, que se proponen para facilitar la interacción con los expertos del dominio y los participantes en la operación, donde se encuentra el problema. En el diagrama de entidades a nivel usuario, se identifican los actores involucrados y se recopila la información de cada actor para entender su participación y la interacción entre ellos. Para obtener una idea completa, se integran los diagramas a nivel usuario en un diagrama de

entidades combinado, donde se contiene la información suficiente para identificar todas las interacciones entre las entidades relacionadas, que constituyen el dominio del problema. No obstante, para tener sólo las interacciones críticas (o las que se pretende abarcar en el siguiente esfuerzo de soportar a la organización), se construye el diagrama de entidades a nivel aplicación. Obteniendo este diagrama se genera la lista de objetivos y el diagrama de líneas de producción, el cual muestra las salidas/entradas y procesos principales.

En los diagramas de entidades, el nivel de abstracción es mayor y permiten representar la realización de las operaciones en el ambiente del problema (espacio del problema).

Los diagramas propuestos en este trabajo de investigación se tomaron como base para definir un Lenguaje de Dominio Específico, que se implementó para el desarrollo de una herramienta de edición, con el objetivo de obtener un modelo de dominio del problema.

2.1.2. Caracterización de las fronteras de los modelos MDA mediante UML

En este trabajo de tesis realizada por (López, 2007), se implementa la metodología *AGD (Desarrollo Arquitectónico en Grupo)*, producto de la investigación doctoral del Doctor Moisés González García para identificar la estructura de los modelos *MDA (Model Driven Architecture)* y sus características, dado que, cuando se desarrolla un software con el enfoque MDA, las transformaciones entre modelos se realizan de forma gradual y no se determina cuando un tipo de modelo se convierte en otro. Para solucionar el problema, la autora realiza un análisis para delimitar cada uno de los modelos, definiendo su estructura y contenido por medio de los elementos de los diagramas seleccionados (casos de uso, diagrama de clases, diagrama de secuencias y diagrama de actividad).

Además, la autora aplica la definición de los modelos y sus fronteras para solucionar el problema clásico (o problema modelo) de biblioteca, se realiza cada uno de los modelos (uno a partir de otro u otros) hasta obtener, mediante estas transformaciones, la implementación de un producto de

software ejecutable. Respetando las fronteras que existen entre los modelos (CIM, PSM, PIM, etc.), y demostrar que es factible usarlas como referencia.

En este trabajo de investigación se incluye un caso de estudio que se enfoca en resolver el problema clásico de biblioteca aplicando las transformaciones entre los modelos de MDA, e incluyendo la construcción de los diagramas de entidades (a nivel usuario, combinado y aplicación) referentes al problema propuesto. Se tomaron tres diagramas de entidad como base para validar la utilización de la herramienta de edición desarrollada en la presente investigación.

2.1.3. Definición e implementación de un perfil de UML para la adquisición de requerimientos funcionales centrados en el usuario

En este trabajo de investigación realizado por (Albores Villatoro, 2007), se plantea la definición e implementación de un perfil del UML que quede integrado a un sistema interactivo, en el cual, los propios clientes capturen sus requerimientos funcionales, generando diagramas de casos de uso y, se demuestre que es indispensable la intervención directa del cliente no sólo en la definición sino también en la especificación de sus requerimientos.

Además, describe cada uno de los pasos que deben seguirse para desarrollar e instanciar un perfil UML para extender los diagramas de casos de uso. Incluye también la especificación del perfil UML conforme lo establecido en el catálogo de la OMG (OMG, 2019).

Este trabajo de investigación, como se menciona anteriormente, proporciona un conjunto de pasos definidos y realizados para elaborar un perfil UML. Este proceso en general es un punto muy importante para el desarrollo del perfil UML, que se utilizó para la elaboración del editor propuesto en esta investigación.

2.2. Trabajos relacionados

En este apartado se analizan los aspectos más relevantes que proponen diversos autores respecto a sus resultados de investigación en temas relacionados a perfiles UML, modelos de dominio, editores y herramientas de modelado. En cada tema incluido, se describe la relación con la investigación que se reporta en este documento.

2.2.1. An EMF-based Toolkit for Creation of Domain Specific Data Services

Los autores (Andreas, Bozic, & Kondov, 2014) presentan una solución para la administración de datos dirigidos por modelos en aplicaciones de ingeniería y composición científica, para ello proponen un framework genérico basado en EMF, que consiste de un conjunto de *plugins* de Eclipse que permite a los expertos en el dominio desarrollar modelos de datos y de estos generar automáticamente un servicio de acceso a datos autónomo. Los autores definen un metamodelo basado en Ecore, el cual se puede utilizar fácilmente para crear modelos de datos de dominio específico.

El autor propone un framework para crear modelos de datos, pero en un dominio específico, permitiendo a los expertos en el dominio desarrollar dichos modelos.

Este trabajo se tomó como base en el análisis para la identificación de las herramientas de modelado y la importancia de desarrollar un editor que ayude a construir un modelo de dominio del problema.

2.2.2. Domain modeling as a basis for building a meshing tool software product line

Los autores (Rossel, Bastarrica, Hitschfeld-Kahler, Díaz, & Medina, 2014) proponen una metodología para la construcción de modelo de dominio, dado que las herramientas de mallas son muy complejas y desarrolladas por usuarios finales (expertos del dominio) sin la aplicación de prácticas establecidas de ingeniería software. Esto lo solucionan mediante un proceso de análisis de dominio, mostrando como sus características se pueden especificar por medio de un modelo basado en el léxico, las características,

objetivos y escenarios. Además, el enfoque que proponen, implica un proceso con actividades claras, roles y condición de terminación, y una formalización del modelo especificado en un lenguaje formal.

Este trabajo se propone una metodología para un modelo de dominio, sin embargo, solo se enfoca en un área específica, sin darle importancia a la construcción general de un modelo completo. Por tanto, se toma como apoyo para el análisis de la complejidad en el desarrollo de modelos del dominio.

2.2.3. A Tool-Supported Methodology for Validation and Refinement of Early-Stage Domain Models

Los autores (Autili, Bertolino, Angelis, Ruscio, & Sandro, 2016), proponen una metodología y el desarrollo de una herramienta para permitir un mejor entendimiento entre el modelo generado por los expertos en modelado y los requisitos que tienen en mente los expertos del dominio. El *plugin* está basado en las herramientas de *Eclipse Modeling Framework* y utiliza variedad de modelos (diagramas UML, clases *Ecore*), con el objetivo de contribuir a reducir la brecha de conocimientos entre los Expertos del Dominio y Expertos del Modelado. Los resultados que obtuvieron fueron evoluciones positivas sobre la eficiencia del *plugin* en la revelación de errores de modelado.

Este trabajo de investigación se tomó como referencia para la evaluación de las herramientas de modelado que permitiera la construcción de un editor. Además, la relación que tiene con la presente investigación es que la creación de modelos de dominio del problema.

2.2.4. An UML profile for modeling RESTful services

Los autores (Ormeño, Lund, Aballay, & Aciar, 2012) presentan un perfil UML para modelar los servidores RESTful, porque a pesar que las herramientas Spring y Spring Roo, se facilitan el desarrollo y el mantenimiento de las aplicaciones, que utilizan estos servicios como controladores, solo se centran en los conceptos de la arquitectura REST, dejando de lado el uso y los elementos tecnológicos de dichos servicios. Conforme a lo anterior, los autores se basan en los controladores y artefactos

por la herramienta Spring Roo para realizar los modelos que se utilizaran en el perfil.

Los modelos que utilizan representan los diversos usos de los servicios, y esta la posibilidad de aplicar la transformación de un modelo a una aplicación Spring que funcione. Utilizan Enterprise Architect para realizar el perfil UML.

Los modelos presentados en este trabajo se emplean para un área específica, resaltando la importancia de tener herramientas que permitan la construcción de modelos de dominios y ayuden a los desarrolladores a entender un dominio en particular (dominio de aplicación). La elaboración del perfil UML en la herramienta Enterprise Architect, sirve como apoyo para el análisis y evaluación de las herramientas de modelado.

2.2.5. Perfil UML para el desarrollo de aplicaciones WAP

En este artículo, los autores (Ricardo & Nibaldo, 2005) presentan un perfil UML para el desarrollo de aplicaciones WAP (*Wireless Applications Protocol*). El objetivo principal del perfil propuesto, es extender UML para entregar elementos específicos (clases, estereotipos, valores etiquetados y restricciones) que permitan a los desarrolladores modelar aplicaciones WAP. Definen una serie de pasos proceso para el desarrollo del perfil, que abarca la identificación de las meta-clases hasta la aplicación del perfil en un caso de estudio.

Mencionan que es importante que el desarrollador debe contar con procedimientos, restricciones, elementos y notaciones específicas al dominio que pretende modelar. De esta forma se crean diagramas de mayor definición y expresividad al dominio en cuestión, obteniendo como resultado una disminución en el tiempo utilizado en avanzar desde las etapas de modelado a las etapas de implementación.

De este trabajo de investigación se tomó como punto de referencia el proceso de 5 etapas que utilizan para obtener un perfil UML, y la realización del caso de estudio a través del perfil UML propuesto.

2.2.6. Una Introducción a los Perfiles UML

Los autores (Fuentes & Antonio, 2004) proponen un proceso para definir un perfil UML, dado que es un mecanismo para la definición de lenguajes específicos de dominio cuya semántica sea una extensión de la de UML. También mencionan la importancia de los perfiles dentro del desarrollo de Aplicaciones Dirigido por Modelos (MDA) y la definición de modelos. La versión de UML que ellos consultan es 1.5.

Este trabajo de investigación incluye los conceptos básicos que se requieren para comprender el proceso de construcción de un perfil UML. Asimismo, resalta la importancia de construir un perfil y la finalidad de definir un lenguaje de dominio específicos. Estos puntos son importantes para la realización de este trabajo de investigación.

2.2.7. Análisis de los trabajos relacionados

En la tabla 2.1, se muestra una comparativa de los trabajos relacionados presentados en este apartado. A continuación, se describen las columnas de la tabla mencionada:

Columnas:

- Objetivo. Finalidad para la cual se realiza el trabajo.
- Herramienta empleada. Medio que se utilizó para desarrollar el producto.
- Metodología o enfoque. Define que la perspectiva se empleó para definir el producto.
- Producto. Contiene el producto resultante del trabajo de investigación, que pueden ser: procedimientos, metodologías, editores, perfiles, entre otros.

Tabla 2.1 Comparativa de los trabajos relacionados

Título	Año	Objetivo	Herramienta empleada	Metodología /Enfoque	Producto
An EMF-based Toolkit for Creation of Domain Specific Data Services	2014	Servicio web de acceso a datos de un dominio específico	Eclipse Modelling Framework	MDE	Framework/ Editor
Domain modeling as a basis for building a meshing tool software product line	2014	Metodología para la construcción de modelos de dominio para herramientas de mallas	Meshing Tool Generator.	No se especifica	Metodología
A Tool-Supported Methodology for Validation and Refinement of Early-Stage Domain Models	2016	Generar una variedad de preguntas LN a partir de Diagramas UML, clases ECORE	Eclipse Modeling Framework	MDE	Editor
An UML profile for modeling RESTful services	2012	Generar un perfil UML para modelar los servicios REST.	Enterprise Architect/Spring Roo	No se especifica	Perfil UML
Perfil UML para el desarrollo de aplicaciones WAP	2005	Generar un Perfil UML para entregar elementos específicos que permitan a los desarrolladores modelar aplicaciones WAP.	No se especifica	No se especifica	Procedimiento
Una Introducción a los Perfiles UML	2004	Analizar los mecanismos de extensión que se utilizan para definir un Perfil UML.	No se especifica	No se especifica	Procedimiento
Trabajo de investigación	2016	Generar un editor para los diagramas de entidades.	Enterprise Architect	AGD	Perfil UML /Editor

En la tabla 2.1 se describe lo que se ha realizado hasta ahora respecto al desarrollo de herramientas de edición y metodologías que permitan la construcción de un modelo del dominio del problema. Un framework realizado en Eclipse para generar modelos en un dominio específico, que ayude a los expertos en la construcción de modelos de datos (Andreas, Bozic, & Kondov, 2014). La propuesta de una metodología que ayude al análisis de un dominio en particular, aplicando las especificaciones de un modelo específico (Rossel, Bastarrica, Hitschfeld-Kahler, Díaz, & Medina, 2014). La importancia de reducir la brecha de conocimiento entre los expertos del dominio y los desarrolladores mediante una metodología y herramienta que permita la construcción de un modelo de dominio (Autili, Bertolino, Angelis, Ruscio, & Sandro, 2016). El desarrollo de perfiles UML en Enterprise Architect para la construcción de modelo en particular (Ormeño, Lund, Aballay, & Aciar, 2012). La definición de un proceso que permita diseñar un perfil UML y ayude a los desarrolladores a modelar un Lenguaje de Dominio Específico (Ricardo & Nibaldo, 2005). La importancia de utilizar los perfiles para extender UML y generar editores que empleen lenguajes de dominio específicos (Fuentes & Antonio, 2004). Tomando en cuenta estos trabajos de investigación, en esta tesis se propone desarrollar un editor que soporte el lenguaje de dominio específico *Diagramas de entidad*, haciendo uso de los perfiles UML. Con el objetivo de proveer a los desarrolladores con una herramienta de edición que facilite la elaboración de modelos del dominio del problema.

En este capítulo se exponen los antecedentes y una comparativa de los trabajos relacionados con el presente trabajo de investigación, así como la importancia de la aportación que se realiza con respecto a los demás. De acuerdo a esta comparativa, podemos afirmar que no existe una herramienta de edición que permita la construcción de diagramas de entidad para la definición de modelos del dominio del problema.

En el siguiente capítulo se definen los conceptos fundamentales para el desarrollo de esta investigación y que son mencionados en los antecedentes y trabajos relacionados.

CAPÍTULO 3. MARCO TEÓRICO

En este capítulo se describen los conceptos fundamentales que determinan el ámbito contextual en el cual se posiciona la investigación. Se detallan los conceptos de: diagramas de entidades, perfil UML y aquellos que se manejan en la herramienta de modelado *Enterprise Architect*.

3.1. Conceptos generales

En este apartado se presentan conceptos básicos para una mejor comprensión del presente documento.

3.1.1. Modelo

El autor (Kimmel, 2008) define un modelo como una colección abstracta que puede representar diferentes aspectos de un sistema.

De acuerdo a (Object Management Group, 2015), en el contexto de MDA, un modelo debe incluir el conjunto de información acerca de un sistema que está dentro de su dominio, las reglas de integridad aplicables a ese sistema, y el significado de los términos utilizados.

3.1.2. Dominio

En (Czarnecki & Eisenecker., 2000), un dominio es una área de conocimiento que incluye:

- un conjunto de conceptos y terminología que comprenden los expertos del área.
- el conocimiento de cómo construir sistemas de software (o partes del sistema de software) en esa área.

3.1.3 Modelo de dominio

En (Larman, 2003), un modelo de dominio es una representación visual de las clases conceptuales y objetos del mundo real en un dominio de interés. También se le denomina modelo conceptual, modelo de objetos del dominio y modelos de objetos de análisis.

En (Taylor, Medvidovic, & Dashofy, 2008), los autores definen un modelo de dominio como un conjunto de artefactos que capturan información sobre un dominio. En un modelo de dominio se define un vocabulario para el dominio, se describen las entidades y datos en el dominio, se define cómo las entidades y los datos se combinarán para proporcionar características. Además, proporciona la base para la estandarización de las descripciones de los problemas a resolver en el dominio.

3.1.4. Dominio del problema

Un dominio del problema en (Boone, 2014) es un término de ingeniería en referencia a toda la información que define el problema y las limitaciones de la solución (las limitaciones son parte del problema). Incluye las metas que el propietario del problema desea alcanzar, el contexto en que el problema existe, y todas las reglas que definen las funciones esenciales u otros aspectos de cualquier producto de la solución.

3.1.5. Arquitectura Dirigida por Modelos (MDA)

Arquitectura dirigida por Modelos (MDA, *Model Driven Architecture*) es una propuesta de la OMG (Object Management Group, 2015) para el desarrollo de software. La idea principal de MDA es especificar sistemas con modelos independientes de plataforma, separando la especificación independiente la plataforma, de la especificación para implementar en una plataforma específica. El modelo se convierte en el elemento más valioso, puesto que a partir de él y mediante una serie de transformaciones se puede obtener el código de la aplicación.

Esta arquitectura considera para los modelos, tres niveles de abstracción:

- **Modelo Independiente de Cómputo (CIM, *Computation Independent Model*):** Modelo que tiene relación con el negocio, donde se presenta una vista del sistema. CIM es llamado comúnmente modelo de dominio o de negocio, porque utiliza un lenguaje que es familiar para los expertos del dominio. Juega un papel muy importante en la reducción de la brecha que existe entre los expertos del dominio y los tecnólogos de la información, responsables de la implementación del sistema.
- **Modelo Independiente de Plataforma (PIM, *Platform Independent Model*):** Modelo de un subsistema que no contiene información específica de la plataforma o tecnología específica que se utiliza, posee un alto nivel de abstracción, por lo cual no es necesario repetir el proceso de modelado cada vez que se introduzca una tecnología al sistema y representa la estructura, funcionalidad y restricciones del sistema, independiente de la plataforma tecnológica sobre la que se va a implementar.

- **Modelo de Plataforma Específica (PSM, *Platform Specific Model*).** Modelo de un subsistema que incluye información sobre la tecnología específica que se utiliza; este modelo completa al PIM especificando cómo toma forma el sistema al implementarse en una plataforma determinada. Los conceptos abstractos del PIM se detallan y describen de forma acorde a los recursos de la plataforma elegida y es la representación de un sistema, con detalles específicos de la plataforma en la que será implantado.

3.2. Diagramas de entidades

Los diagramas de entidades fueron propuestos en la metodología *Warnier/Orr*, y ayudan a definir los requerimientos de una aplicación, de manera clara, completa y coherente (Orr, 1981).

Los diagramas propuestos son:

- Diagramas de entidades a nivel usuario.
- Diagramas de entidades combinado.
- Diagrama de entidades a nivel aplicación.

Todos estos diagramas permiten mostrar una representación abstracta y estática, de la forma en que trabaja actualmente la organización, es decir, un modelo conceptual. Además, permiten la intervención de los expertos en el dominio de la aplicación cuando se inicia con elementos proporcionados para la representación estática del Diagrama de Entidades a Nivel Aplicación (último diagrama, que representa las entidades, entrada y salidas que abarca la aplicación, que será una solución que mejore la situación problemática).

Dentro los diagramas de entidades se manejan los siguientes conceptos generales:

- *Entidad*. Son todos los usuarios involucrados en el dominio del problema. Estas entidades se dividen en: internas (pertenecientes a la organización) y externas (que no pertenecen a la organización).
- *Evento*. Representa la información que existe entre una entidad interna y una entidad externa.

- *Transacciones.* Se representan por flechas para indicar el conjunto de todas las interacciones entre entidades de la organización (internas) y entidades del ambiente (externas).

Los diagramas mencionados en las secciones 3.2.1 Diagramas de entidades a nivel usuario, 3.2.2 Diagramas de entidades combinado y 3.2.3 Diagrama de entidades a nivel aplicación y diagrama de línea de producción; se consideran en esta investigación como un lenguaje de dominio específico (DSL, *Domain Specific Language*).

3.2.1. Diagrama de entidades a nivel usuario

Este diagrama es un medio para recopilar información inicial a través de investigaciones y entrevistas. Estos diagramas proporcionan un mecanismo para capturar una vista general del dominio del problema, que se soportará mediante la aplicación a desarrollar.

Se deben identificar todas las entidades principales que interactúan en el contexto de la aplicación, dado que por cada usuario involucrado en el dominio del problema/aplicación se genera un diagrama, de acuerdo a las siguientes reglas (Orr, 1981):

1. Escribir el nombre de la organización, usuario y aplicación en la parte superior del diagrama.
2. Colocar el nombre del usuario (participante) principal en una elipse, colocada en el centro del diagrama.
3. Organizar las entidades que están relacionadas con el usuario principal.
4. Dibujar las flechas (interacciones) entre las entidades.
5. Revisar el diagrama y volver a dibujar si es necesario.

Cuando se terminan de dibujar todos los Diagramas de Entidad a Nivel Usuario, de todos los actores involucrados, se tienen las representaciones separadas de las tareas que abarca la aplicación. Si se requiere tener una idea completa, se integran todos los diagramas en el Diagrama de Entidades Combinado.

3.2.2. Diagrama de entidades combinado

Como su nombre lo indica este diagrama representa la vista general de todas las entidades involucradas en el sistema.

Para crear el diagrama combinado se tienen las siguientes reglas:

1. Identificar las entidades y relaciones en cada diagrama de entidades a nivel usuario.
2. Unir las entidades que sean iguales o similares, encontradas en los diagramas a nivel usuario, para evitar redundancias.
3. Descartar aquellas entidades que no son necesarias.

En el diagrama de entidades combinado se tiene mucha información, incluye todas las interacciones entre las entidades relacionadas con el sistema, como se mencionó anteriormente. Para tener sólo las interacciones críticas se construye el Diagrama de Entidades a Nivel Aplicación, que se logra dibujando una frontera alrededor de las entidades internas. Se representa la aplicación cuando se consolidan todas las entidades internas en una sola elipse.

3.2.3. Diagrama de entidades a nivel aplicación

El diagrama de entidades a nivel aplicación muestra las principales entidades que se incluirán en la aplicación.

El proceso para desarrollar el diagrama es el siguiente (Orr, 1981):

1. Dibujar una frontera alrededor de las entidades internas para representar la aplicación en el diagrama de entidades combinado.
2. Dibujar el diagrama a nivel aplicación, consolidando todas las entidades internas en una sola elipse que representa a la empresa (aplicación, dentro de la organización). Identificando las transacciones que interactúan con las entidades externas.

A partir este diagrama, es posible identificar los objetivos principales del sistema (aplicación). Para definir estos objetivos, se parte de la entidad principal del diagrama de entidad a nivel aplicación, donde cada transacción representa un objetivo. Si la flecha que simboliza la transacción sale de la

entidad principal, entonces el objetivo se declara como “Enviar nombre de la transacción”. En caso que la flecha vaya en dirección contraria, se declara “recibir nombre de la transacción”.

Después de definir los objetivos y de elabora el diagrama de entidades a nivel aplicación, se construyen los diagramas de línea de producción. Estos diagramas están fuera del alcance de esta investigación.

3.2.4. Representación de los diagramas

En esta apartado se muestra una representación de los diagramas descritos anteriormente.

El diagrama de entidades a nivel usuario es un medio para recopilar información inicial a través de investigaciones y entrevistas. En las figuras 3.1 y 3.2, se muestran dos diagramas de entidades a nivel usuario, las cuales se emplean para generar el segundo diagrama, el diagrama de entidades combinado.

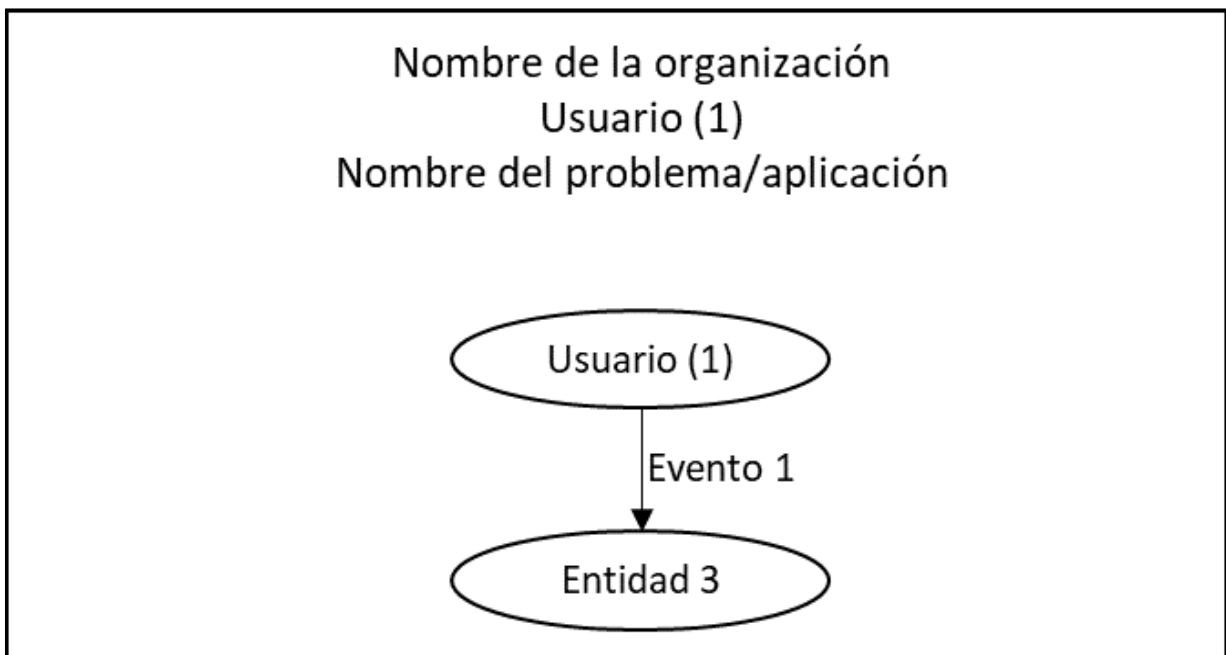


Figura 3.1 Diagrama de entidades a nivel usuario (1)

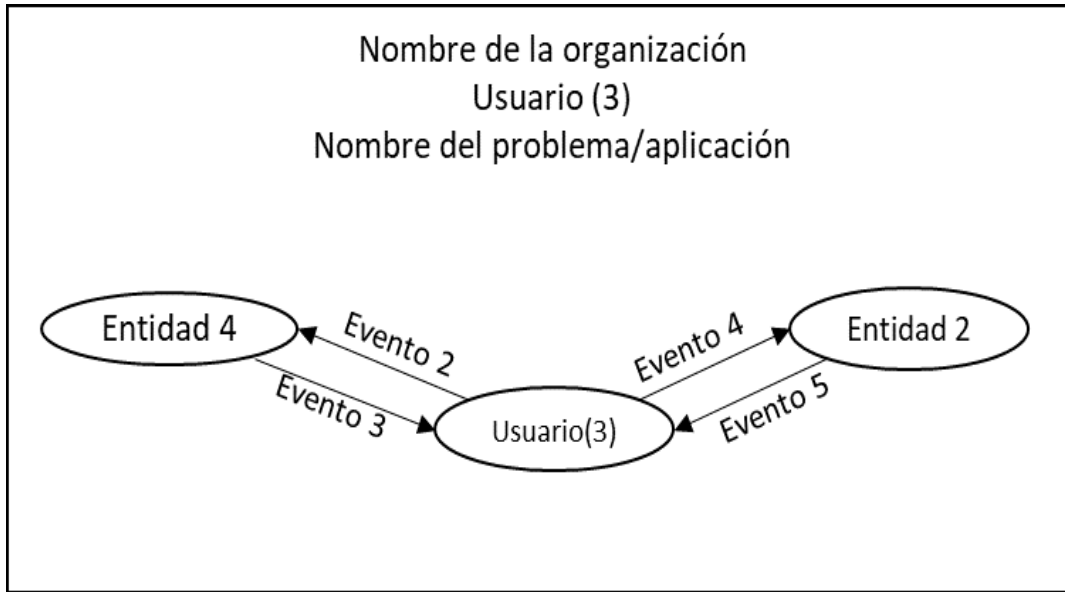


Figura 3.2 Diagrama de entidades a nivel usuario (3)

Al construir el diagrama de entidades combinado a nivel usuario se pueden descubrir inconsistencias en la información (eventos). Por ejemplo, una transacción solo puede mostrarse en un diagrama, y no verse reflejada en otros diagramas.

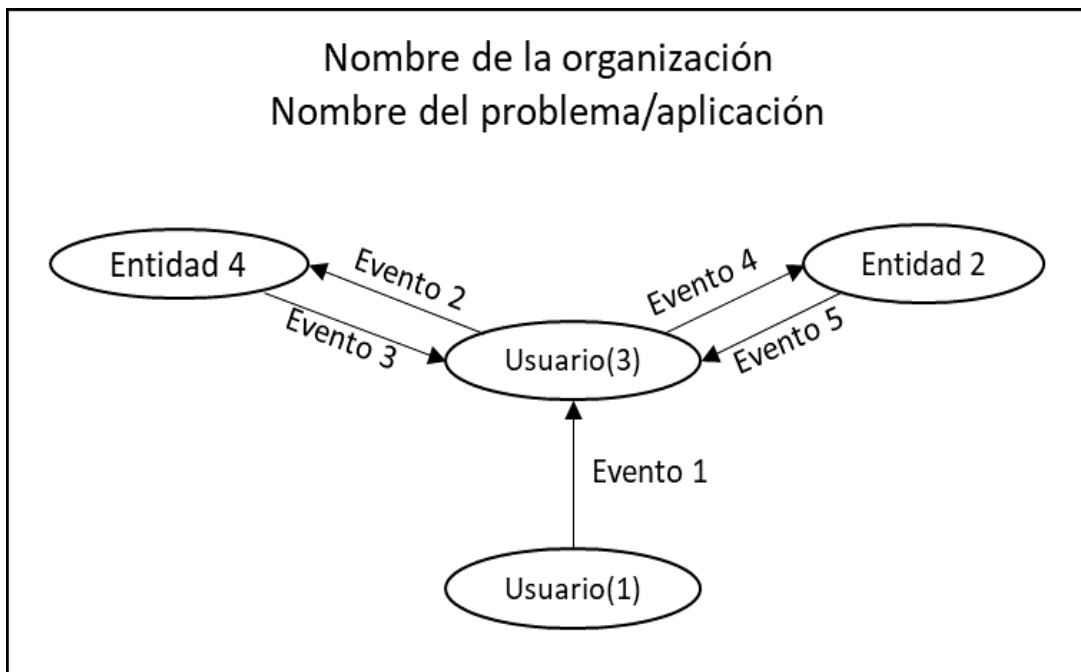


Figura 3.3 Diagrama de entidades combinado

El diagrama de entidades combinado contiene una gran cantidad de información, dado que se representan todas las interacciones dentro del sistema. Tomando como referencia el diagrama anterior (ver figura 3.3), se obtiene el diagrama de entidades a nivel aplicación, donde se delimitará mediante una línea puntada las entidades que formarán parte del sistema, como se muestra en la figura 3.4.

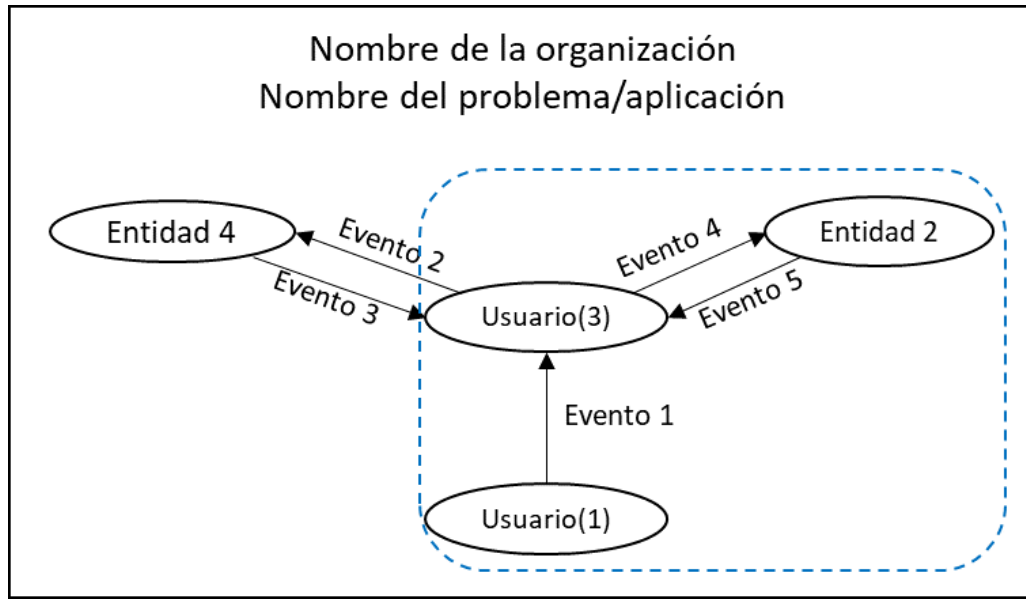


Figura 3.4 Diagrama de entidades combinado con frontera

Una vez seleccionadas las entidades internas, se definió una nueva entidad, que representa la aplicación o problema. En este ejemplo, la nueva entidad es *Aplicación* (ver figura 3.5).

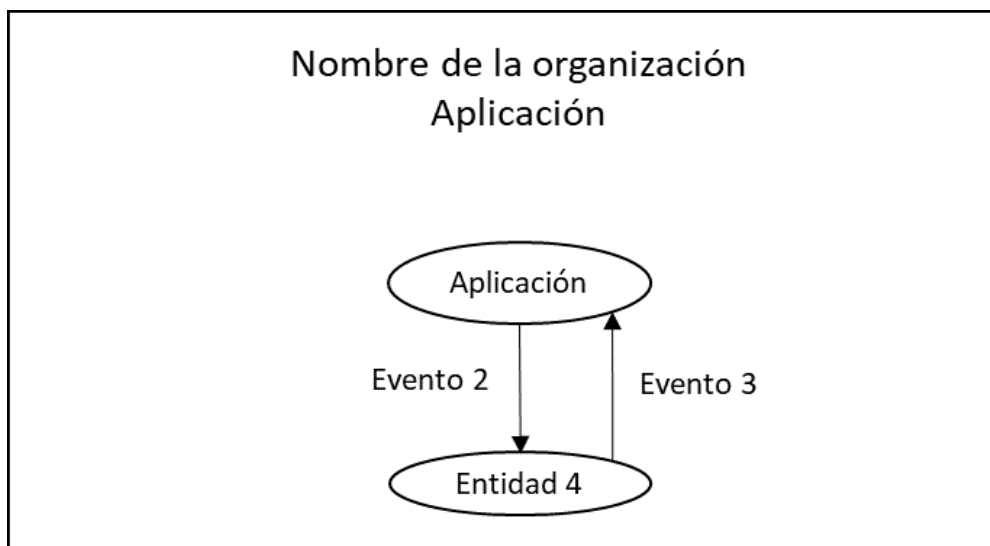


Figura 3.5 Diagrama de entidades a nivel aplicación

3.3. UML (*Unified Modeling Language*)

El Lenguaje de Unificado de Modelado (por sus siglas en inglés, *UML*), es un lenguaje estándar de modelado para describir la estructura de una aplicación (Booch, Rumbaugh, & Jacobson, El Lenguaje Unificado de Modelado, 2000). Puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema de software, desde un sistema de manejo de información hasta aplicaciones web distribuidas e incluso sistemas en tiempo real.

Como lenguaje, provee un vocabulario y reglas para combinar las palabras de ese vocabulario con el propósito de comunicar. Como lenguaje de modelado su vocabulario y reglas hacen foco en la representación conceptual y física de un sistema.

Dado que UML es un lenguaje de propósito general, este proporciona una gran flexibilidad y expresividad para modelar sistemas. Sin embargo, el hecho de que UML sea una notación de propósito muy general obliga a que muchas veces sea deseable poder contar con algún lenguaje más específico para modelar y representar los conceptos de ciertos dominios particulares, a este tipo de lenguaje se le conoce como Lenguaje de Dominio Específico (DSL).

El OMG propone dos posibilidades para definir un DSL: o se define un nuevo lenguaje (alternativo al UML, mediante un metamodelo para el dominio específico), o se extiende el propio UML, especializando algunos de sus conceptos y restringiendo otros, pero, respetando la semántica original de los elementos (mediante un perfil UML) (Fuentes & Antonio, 2004).

3.3.1. Perfil UML

Un perfil UML constituye un mecanismo que proporciona el propio UML para extender su sintaxis y su semántica para expresar los conceptos específicos de un determinado dominio de la aplicación (Fuentes & Antonio, 2004).

Para definir un perfil UML se utilizan tres mecanismos:

- **Estereotipos.** Están definidos por un nombre y por una serie de elementos del metamodelo de UML, a los que puede asociarse. Los estereotipos representan las nuevas características que se agregan al metamodelo UML para extender este lenguaje.
- **Valores Etiquetados.** Son meta-atributos que se asocian a una metaclase del metamodelo UML, extendido por un perfil. Todo valor etiquetado ha de contar con un nombre y un tipo, y se asocia a un determinado estereotipo.
- **Restricciones.** Las restricciones se asocian a los estereotipos e imponen condiciones sobre los elementos del metamodelo UML que se estereotiparon. Las restricciones se transcriben en lenguaje natural o en Lenguaje de Restricción de Objetos (OCL, *Object Constraint Language*), definido por la OMG como el estándar para transcribir restricciones y consultas.

3.3.1.1. Importancia de los perfiles UML

En UML 2.5 se señalan varias razones por las cuales un diseñador puede querer extender y adaptar un metamodelo existente, sea del propio UML u otro perfil en particular (OMG Unified Modeling Language, 2015):

- Disponer de una terminología y vocabulario propio de un dominio de aplicación o de una plataforma de implementación concreta.
- Definir una sintaxis para construcciones que no cuentan con una notación propia (como sucede con las acciones).
- Definir una nueva notación para símbolos ya existentes, más acorde con el dominio de la aplicación (por ejemplo, una figura con un ordenador, en lugar del símbolo para representar un nodo que ofrece UML, para representar ordenadores en una red).
- Agregar cierta semántica no determinada de forma precisa en el meta modelo (por ejemplo, la incorporación de prioridades en la recepción de señales, en una máquina de estados de UML).
- Agregar tipos que no existe en el metamodelo (por ejemplo, relojes, tiempo continuo, etc.).
- Agregar restricciones a las existentes en el metamodelo, restringiendo su forma de utilización (por ejemplo, impidiendo que ciertas acciones

se ejecuten en paralelo, dentro de una transición, o forzando la existencia de ciertas asociaciones entre las clases de un modelo).

- Agregar información que puede ser útil a la hora de transformar el modelo a otros modelos, o a código.

3.3.1.2. Elaboración de un perfil UML

Los autores (Fuentes & Antonio, 2004) proponen los siguientes pasos para definir un perfil UML:

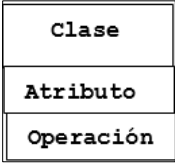

- **Crear el metamodelo.** Obtener la definición del metamodelo de la plataforma o dominio de la aplicación a modelar con un perfil. Si no existe dicho metamodelo, entonces definir uno utilizando los mecanismos del propio UML (clases, relaciones de herencia, asociaciones, etc.). Incluyendo las definiciones de las entidades necesarias, las relaciones que existen entre ellas y las restricciones que puedan existir en el uso de estas.
- **Definir el perfil UML.** Se debe crear un estereotipo por cada elemento del metamodelo que deseamos incluir en el perfil. Los estereotipos deben tener el mismo nombre que los elementos del metamodelo, para así mantener una unión entre ambas etapas. Cualquier elemento que exista en el metamodelo podrá representarse más adelante en un estereotipo.
- **Identificar los elementos a extender.** Consiste en identificar los elementos de UML que se van a extender con cada estereotipo. Ejemplo de tales elementos son las clases, asociaciones, atributos, paquetes, etc.
- **Definir los valores etiquetados.** Por cada atributo presente en el metamodelo se debe agregar el valor etiquetado correspondiente en el perfil UML. Los valores etiquetados son atributos adicionales que se asocian al perfil.
- **Definir las restricciones.** El último paso es definir las restricciones que forman parte del perfil. Las restricciones se definirán en lenguaje OCL, lenguaje definido por OMG como el estándar para transcribir restricciones y consultas.


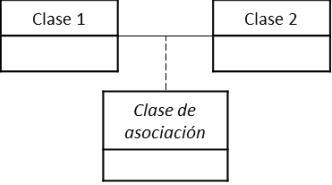
3.3.2. Diagrama de Clases

Un diagrama de clases es un diagrama que muestra un conjunto de interfaces, colaboraciones y sus relaciones. Se utilizan para modelar la vista de diseño estática de un sistema. Esta vista soporta principalmente los requerimientos funcionales de un sistema, los servicios que el sistema debe proporcionar a sus usuarios finales.

Los elementos de un diagrama de clases se muestran en la tabla 3.1.

Tabla 3.1 Elementos de un diagrama de clase

Elemento	Definición	Notación
<i>Clase</i>	Es una descripción de un objeto que tiene los mismos atributos, operaciones, relaciones y semántica.	
<i>Atributo</i>	Es una propiedad de una clase identificada con un nombre, que describe un rango de valores que pueden tomar las instancias de la propiedad.	<i>atributo</i>
<i>Operación</i>	Es una implementación de un servicio que puede requerirse a cualquier objeto representado por la clase, para que muestre un comportamiento.	<i>operación()</i>
<i>Interfaces</i>	Es una colección de operaciones que especifican un servicio de un clase o componente.	
<i>Relación de dependencia</i>	Es una relación de uso que declara un cambio en la especificación de un elemento que puede afectar a otro elemento que utiliza.	----->
<i>Relación de generalización</i>	Es una relación entre un elemento general (llamado superclase o padre) y un caso más específico de ese elemento.	—————>

<i>Relación de asociación</i>	Es una relación estructural que especifica que los objetos de un elemento están conectados con los objetos de otro.	
<i>Relación de clase de asociación</i>	Es una relación híbrida con las características de una asociación y una clase.	

3.3.2.1. Presentación de un diagrama de clases

En la figura 5 se muestra un diagrama de clases de la aplicación de la taquilla del teatro. Este diagrama contiene parte del dominio del modelo “venta de entradas”, que es un ejemplo del libro El Lenguaje Unificado de Modelado, Manual de Referencia de los autores (Rumbaugh, Jacobson, & Booch, 1999).

Los clientes pueden tener muchas reservas, pero cada reserva la hace un cliente (ver figura 3.6).

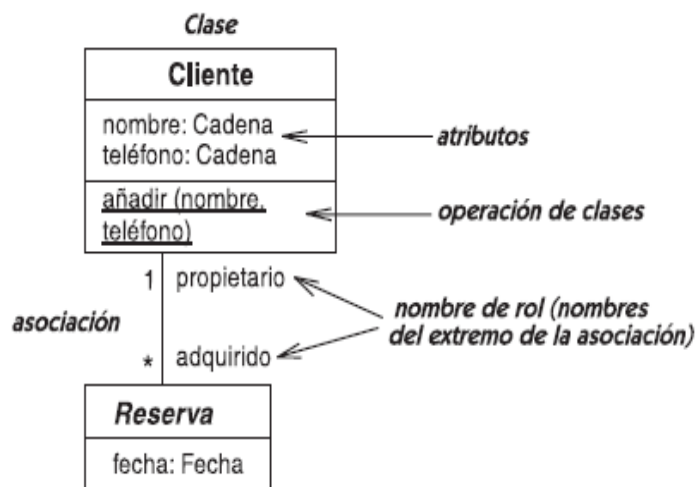


Figura 3.6 Diagrama de clases

Dentro de UML, existe una relación llamada *Clase de asociación*, que puede verse como una asociación que también tiene propiedades de clase, o como una clase que también tiene propiedades de asociación. Este elemento de

modelado posee las propiedades tanto de una relación de asociación como del elemento clase (OMG Unified Modeling Language, 2015). Aunque su notación consiste tanto de un símbolo de asociación, como de un símbolo de clase, realmente es un único elemento del modelo.

Este elemento se puede añadir a un diagrama de clases, en caso que la relación de asociación tenga atributos como se muestra en la figura 3.7. En este ejemplo la relación de asociación “Acción” tiene el atributo “cantidad”.



Figura 3.7 Diagrama de clases con la relación asociación

3.4. Enterprise Architect

Enterprise Architect (EA) es una herramienta de diseño y modelado visual basada en el UML de la OMG. La plataforma soporta: el diseño y construcción de sistemas de software; modelado de procesos de negocio; y modelado de dominios basados en la industria.

EA tiene un mecanismo de perfil UML genérico para cargar y trabajar con diferentes perfiles. Los perfiles UML se presenta en archivos XML, con un formato específico. Estos archivos XML se pueden importar en la herramienta. Una vez importados, puede arrastrar y soltar elementos de perfil en el diagrama actual. EA adjuntará el estereotipo, los valores y los valores predeterminados, notas etiquetadas e incluso un meta-archivo si se especifica uno, para el nuevo elemento.

3.4.1. Generación dirigida por Modelos

La *Tecnología de Generación dirigida por Modelos (MDG, Model Driven Generation Technology)*, es un mecanismo para proporcionar acceso a los recursos de una tecnología disponible o una tecnología que se crea.

Con *Enterprise Architect*, se pueden desarrollar modelos basados en las especificaciones de UML, como son los valores etiquetados (*Tagged Values*), estereotipos (*Stereotypes*), perfiles (*Profiles*) y patrones de diseño.

Las tecnologías principales son:

- La tecnología UML 2 básica, en esta se implementa el modelado y comportamiento de UML 2.5.
- La tecnología *Core Extensions*, que aplica perfiles y estereotipos para proporcionar un modelado extendido, para algún dominio específico, por ejemplo: requisitos, interfaz del usuario y modelado de datos.

Una vez presentados los conceptos anteriores, ahora se describe cómo se desarrolla una tecnología MDG. Para construir un perfil UML se generan tres tipos de diagramas:

- Perfil (*Profile*). En este diagrama se definen los estereotipos, las meta-class (elementos UML a extender) y los valores etiquetados (propiedades del estereotipo).
- Caja de herramientas (*ToolBox*). En este diagrama se define una caja de herramientas, donde cada estereotipo es un elemento en la caja y un atributo define una propiedad del elemento.
- Perfil del diagrama (*Diagram profile*). Este diagrama se utiliza para vincular el diagrama de la caja de herramientas y el perfil UML.

Además, se crea un archivo MTS (*MDG, Technology Selection*) en formato XML, el cual almacena los elementos que se definen en los tres diagramas anteriores. Este archivo se utiliza para instanciar el editor para crear diagramas ajustados al perfil UML construido.

3.4.2. Estructura de un proyecto

EA cuenta con una serie de elementos que conforman un proyecto. Estos elementos están clasificados en una arquitectura propia.

1. *Modelos (models)*. Son el nivel superior de un proyecto en *Enterprise Architect*. Un proyecto puede contener uno o más modelos.
2. *Puntos de vista (Views)*. Los puntos de vista proporcionan una ventana para explorar las relaciones entre los elementos en el modelo (*model*).
3. *Paquetes (Packages)*. Son los contenedores para los elementos y los diagramas. Los contenidos pueden visualizarse de diferentes maneras dependiendo de las necesidades.
4. *Diagramas (Diagrams)*. Los diagramas se utilizan para definir relaciones específicas de un número de elementos seleccionados.
5. *Elementos (Elements)*. Son los componentes básicos del modelo, para crear y construir los sistemas.
6. *Conectores (Connectors)*. Estos componentes definen las relaciones entre los elementos.
7. *Valores etiquetados (Tagged Values)*. Amplían las posibilidades para especificar las características del elemento o conector estereotipado.
8. *Plantillas (templates) y asistentes (Wizards)*. Las plantillas y asistentes son modelos predefinidos para ayudar en la generación de modelos.
9. *Automatización (Automation)*. Es una funcionalidad de *Enterprise Architect* que permite crear complementos propios, scripts y funcionalidades para dominios específicos.

En la figura 3.8 se presenta la estructura que maneja la herramienta *Enterprise Architect*. Esta figura se realizó acuerdo a la información mencionada.

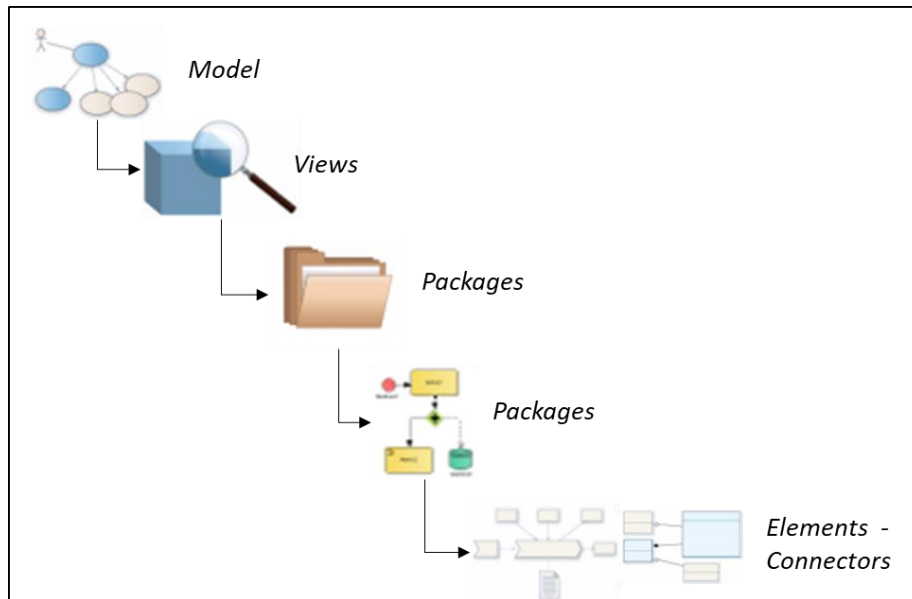


Figura 3.8 Estructura de un proyecto

En este capítulo se incluyen los conceptos claves que se utilizan para el desarrollo de esta investigación. Se presenta la información acerca de los perfiles UML y una introducción a UML con base a los Lenguajes de Dominio Específico (DSL), dado que esta investigación se enfoca en el desarrollo de un editor para la construcción de los diagramas de entidad.

En el siguiente capítulo se especifica la definición de los diagramas de entidades y el diseño utilizado para realizar el perfil UML..

CAPÍTULO 4. DIAGRAMAS DE ENTIDAD Y SU PERFIL UML

En este capítulo se presenta el proceso para diseñar el perfil UML, que contempla los elementos y la construcción de los diagramas de entidades. En la sección 4.1 se expone la identificación de los elementos y restricciones que se obtienen a partir de la metodología de Orr, que incluye: un breve análisis de la construcción de los diagramas, la identificación de los elementos, la clasificación de las entidades y una representación de los diagramas propuestos. En la sección 4.2 se realizan los pasos a seguir realizar el diseño del perfil UML con base a lo descrito en la sección 4.1., así mismo se define el meta-modelo, los estereotipos, las meta-clases y los valores etiquetados que contendrá dicho perfil.

4.1. Definición de los diagramas de entidades para un editor

Para cumplir con el objetivo de esta investigación, que es la generación de un editor para los diagramas de entidades, se procede a definir el proceso de construcción de los diagramas de entidades, así como identificar los elementos mencionados por (Orr, 1981) y las restricciones generales para su elaboración, dado que tres de los diagramas propuestos en la metodología AGD, poseen características similares. El análisis de la elaboración de los diagramas de entidades se requiere para poder realizar el diseño del perfil UML, que permitirá la generación del editor que soportará su construcción.

Es cierto que UML cuenta con una gama de diagramas que permite modelar etapas importantes de un sistema, sin embargo, existen aspectos de diseño, navegación y construcción que difícilmente pueden abordarse sin extender UML (Ricardo & Nibaldo, *New UML 2.0 based models to design WAP*, 2005). Dentro de los diagramas tradicionales de UML, no se cuenta con un proceso (o serie de diagramas) que represente los elementos de un modelo de dominio del problema, sin conocimiento previo del mismo.

Para solucionar esto, se propuso extender UML, mediante un perfil, con el propósito de modelar los diagramas de entidades, propuestos originalmente por Orr (Orr, 1981). El perfil UML se define conforme al procedimiento de cinco etapas propuesto por los autores (Fuentes & Vallecillo, 2004), descrito en la sección *3.2.1.2. Elaboración de un perfil UML*.

Uno de los requerimientos para construir el perfil es identificar el diagrama UML, que se va extender. En este caso, se tomó el diagrama de clases porque sus elementos (meta-clases) son los consideramos que mejor se adaptan a los diagramas de entidades. En la sección *3.3.2. Diagrama de Clases*, se describen los elementos de un diagrama de clases.

4.1.1. Identificación de elementos y restricciones

La metodología de Desarrollo Arquitectónico en Grupo (AGD, *Architectural and Group Development*) consiste en asignar mayor importancia al producto de software y su estructura, asociando un proceso a cada uno de los productos de trabajo. Dentro de los productos de trabajo,

en la categoría de Ingeniería, la metodología utiliza los modelos de la Arquitectura Dirigida por Modelo (*Model-Driven Architecture*, MDA): Modelo Independiente de Cómputo (CIM), Modelo Independiente de Plataforma (PIM) y Modelo de Plataforma Específica (*PSM*). Dentro del Nivel CIM, se emplea la metodología *Warnier/Orr* para mostrar una representación abstracta y estática de la organización, es decir, definir el modelo de dominio.


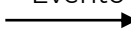

En esta metodología se incluyen tres diagramas: diagramas de entidades a nivel usuario, diagramas de entidades combinado y el diagrama de entidades de la aplicación, para mostrar la forma en que trabaja la organización.

La definición y construcción de estos diagramas se encuentran definidos en la sección 3.2. *Diagramas de entidades*. A partir esos criterios se obtuvieron las siguientes restricciones generales:

1. Un diagrama de entidades solo puede tener un encabezado.
2. Una entidad tiene un nombre único.
3. Existe una entidad principal, en el diagrama de entidad.
4. Entre dos entidades, se pueden enviar o recibir N transacciones.
5. Las flechas representan la dirección de la transacción (*evento*), entre las entidades.

En la tabla 4.1 se muestran los elementos, definición y símbolo visual, que pueden incluirse en los diagramas de entidad.

Tabla 4.1 Elementos del diagrama de entidades (Warnier/Orr)

Elemento	Definición	Notación
<i>Entidad</i>	Representa la persona u organización involucrada en el dominio de la aplicación.	
<i>Transacción</i>	Representa el proceso de una entidad a otra. No existe una secuencia entre los procesos.	Evento 
<i>Limite</i>	Permite delimitar la frontera de la aplicación en el diagrama de entidad combinado.	

4.1.1.1. Clasificación de las entidades

Adicionalmente, Orr (Orr, 1981) define, en los diagramas de entidades, dos tipos de entidades: internas (ubicadas dentro de la organización) y externas (ubicadas fuera de la organización). En diagramas de entidad a nivel usuario, una de las entidades internas se identifica como *entidad principal* y se coloca en el centro del área de dibujo; y las demás entidades internas y externas, serán entidades *secundarias*. Para un diagrama de entidades combinado, no existe la distinción entre entidad principal y secundarias (Ver tabla 4.2.)

Tabla 4.2 Clasificación de las entidades

Diagrama	Tipos de entidad en Wanier/Orr	Tipos entidad propuestas
Diagrama de entidades a nivel usuario	Entidad interna Entidad externa	Entidad principal Entidad secundaria
Diagrama de entidades combinado	Entidad interna Entidad externa	Entidad

Se realizó esta clasificación de las entidades se utiliza más adelante para definir el diseño del perfil UML.

4.1.1.2. Definición de los elementos de los diagramas

Los diagramas de entidades poseen un conjunto de elementos a utilizar en su construcción, como se define en la tabla 4.1. Donde las personas o elementos de las organizaciones se representan con una elipse (entidad) y la relación entre dos entidades se representa mediante una flecha, llamada *Transacción*. Así mismo, en cada transacción se define un *evento*, que representa el envío de algún documento o material de una entidad a otra, y dicho evento puede formar parte de un proceso.

Tal como lo especifica Orr (Orr, 1981) en su libro *Structured Requirements Definition*: “Los diagramas de entidades tienen el objetivo de ayudar a inicializar un proyecto. Para esto, se definen entidades que se representan con una elipse y los documentos o material que se envían/reciben por las entidades, se representa con flechas que indican la dirección del flujo de información”.

Sin embargo, siguiendo esta regla general de construcción se puede perder el detalle del flujo de información (*eventos/mensajes*), que puedan existir entre dos entidades y el proceso asociado a esos eventos/mensajes, como se muestra en la sección 4.2.1.3. *Representación de los diagramas*. Por tanto, fue necesario agregar elementos (o propiedades) a los ya definidos para representar y mejorar el entendimiento del flujo de la información y saber qué eventos/mensajes pertenecen a un proceso.

Las propiedades y/o elementos son:

- *Entidad*. Puede poseer: un nombre, un identificador único y un tipo de entidad (principal, secundaria, en caso de tratarse de un diagrama de entidades a nivel usuario).
- *Mensaje*. Anteriormente llamado evento, posee: un identificador único, pertenece a un proceso, el nombre del mensaje, dirección (enviado o recibido), un origen (entidad de donde sale) y destino (entidad que lo recibe).
- *Registro de Mensaje*. Tabla donde se agregan los mensajes que existan de una entidad a otra. Posee: un nombre (de preferencia de los identificadores de la entidad).

4.1.1.3. Representación de los diagramas de entidades combinados

Conforme a las reglas establecidas y a las restricciones anteriores, se presenta el proceso completo de construcción de un diagrama de entidades combinado, a partir de dos diagramas de entidades a nivel usuario. En el caso de estudio que se considera aquí, el primer diagrama de entidades a nivel usuario (*Figura 4.1*), contiene solo dos entidades, donde la *Entidad "Usuario 1"* es una entidad principal, porque se encuentra ubicada en la parte central del diagrama y la *Entidad 3* es una entidad secundaria colocada alrededor de la entidad "*Usuario 1*".

En el segundo diagrama de entidades a nivel usuario (*Figura 4.2*), la entidad principal es "*Usuario 3*", debido a que se crea un diagrama por cada usuario involucrado en el dominio de aplicación. Las entidades secundarias identificadas son: *Entidad 4* y *Entidad 2*. Aquí, la entidad "*Usuario 1*" pasa a ser una entidad secundaria (que no se incluye en la *Figura 4.2*), que está relacionada con la *Entidad 4*. Como se puede observar, los eventos no muestran un flujo coherente de la información, a menos que posean un número consecutivo, dentro del proceso al que pertenecen.

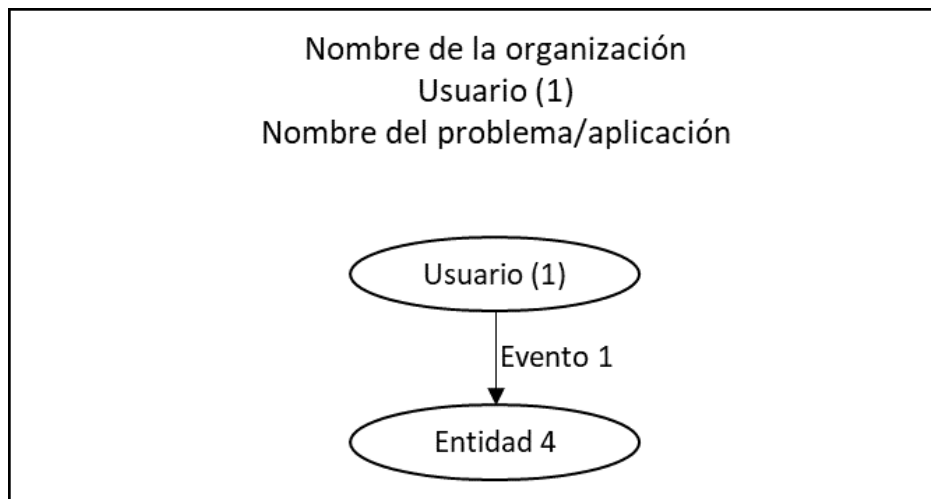


Figura 4.1 Diagrama de entidades a nivel usuario (1)

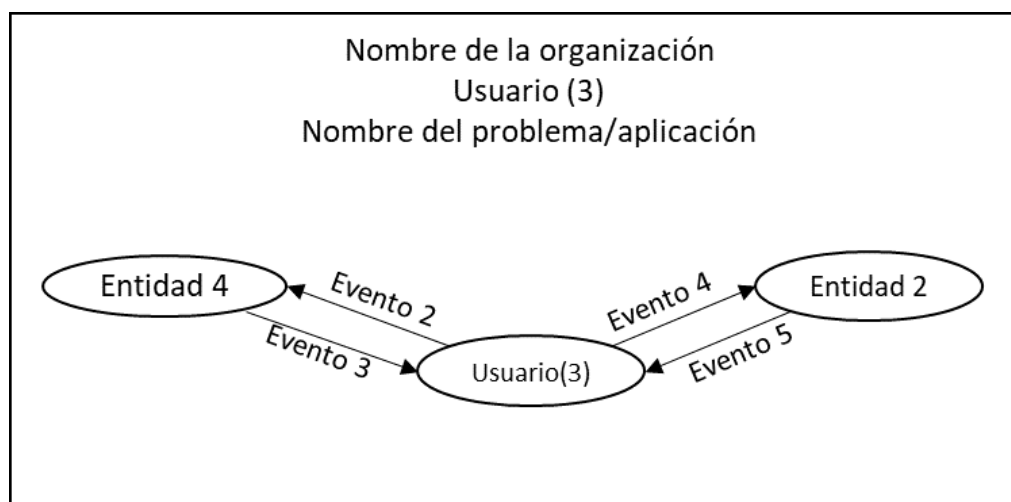


Figura 4.2 Diagrama de entidades a nivel usuario (3)

Deben existir mínimo dos diagramas de entidades a nivel usuario para construir el diagrama de entidades combinado. En este diagrama, las entidades se distribuyen en cualquier orden, se pierde el significado de una entidad principal o entidades secundarias, y todas las entidades pasan a ser solo entidades.

Para realizar la combinación de los diagramas, se identifican entidades en común de todos los diagramas de entidades individuales, y se van agregando en el diagrama de salida, respetando las relaciones que existen entre ellas. Por ejemplo, en los diagramas anteriores, la entidad en común es *Entidad 4*. El *diagrama de entidades del "Usuario 1"*, tiene una entidad

secundaria y el segundo diagrama del “Usuario 3” posee dos. Todas las entidades y las transacciones, con sus eventos asociados, se muestran en el diagrama de entidades combinado (*figura 4.3*).

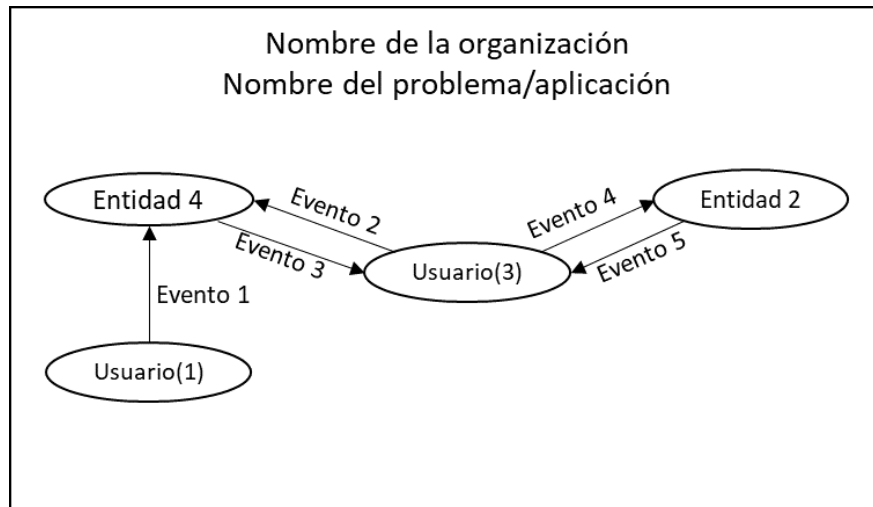


Figura 4.3 Diagrama de entidad a nivel combinado

El siguiente diagrama es el diagrama de entidades a nivel aplicación. En este diagrama, se decide qué entidades y transacciones (con sus eventos asociados), serán parte de la aplicación. En este caso, se seleccionan la entidad “Usuario 1”, la Entidad 2 y la entidad “Usuario 3” con sus respectivas transacciones y eventos asociados, para incluirse en una nueva entidad, que se nombrará “Nombre de Aplicación”, la cual se coloca al centro del área de dibujo, como la entidad principal. Se respetan las transacciones y eventos asociados, que existen entre Entidad 4 y nueva entidad “Aplicación”, porque no están considerados dentro de la aplicación.

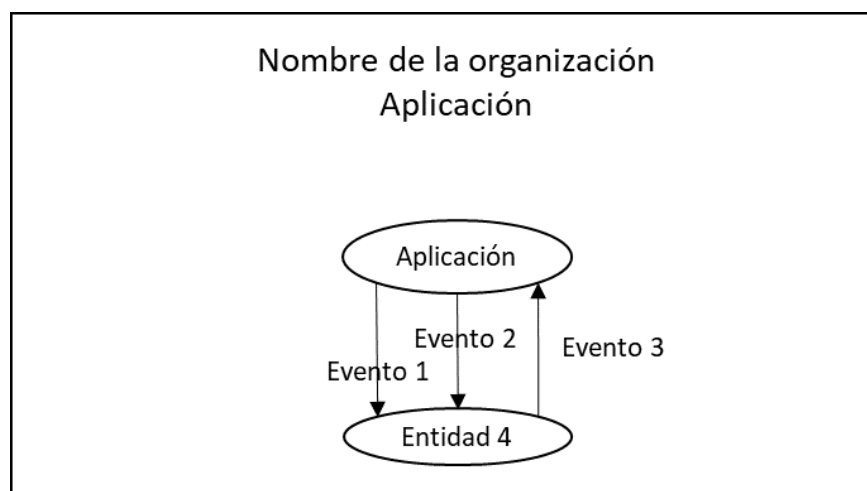


Figura 4.4 Diagrama de entidades a nivel aplicación

4.2. Diseño del perfil UML

En este apartado se presenta el diseño del perfil UML para modelar los diagramas de entidades con base en lo establecido por la OMG (*Object Management Group*) y el procedimiento descrito por los autores (Fuentes & Vallecillo, 2004), presentado en la sección 3.3.1. *Perfil UML*.

4.2.1 Creación del metamodelo

Para generar un meta-modelo (que contenga todos los elementos a utilizar en un modelo, que se ajuste a él), se debe conocer el dominio del problema/aplicación. Un dominio de problema/aplicación es un modelo donde se describe la estructura y el comportamiento de los elementos involucrados en la operación en un problema/aplicación, es decir, sus conceptos (o elementos), las relaciones entre ellos, y las posibles restricciones existentes.

En este caso, el dominio de aplicación es el modelado de los diagramas de entidades, y los elementos que conformarán el metamodelo, son aquellos que se utilizan en la elaboración de dichos diagramas. Estos elementos se encuentran descritos en el apartado 4.1.1.2. *Definición de los elementos de los diagramas*.

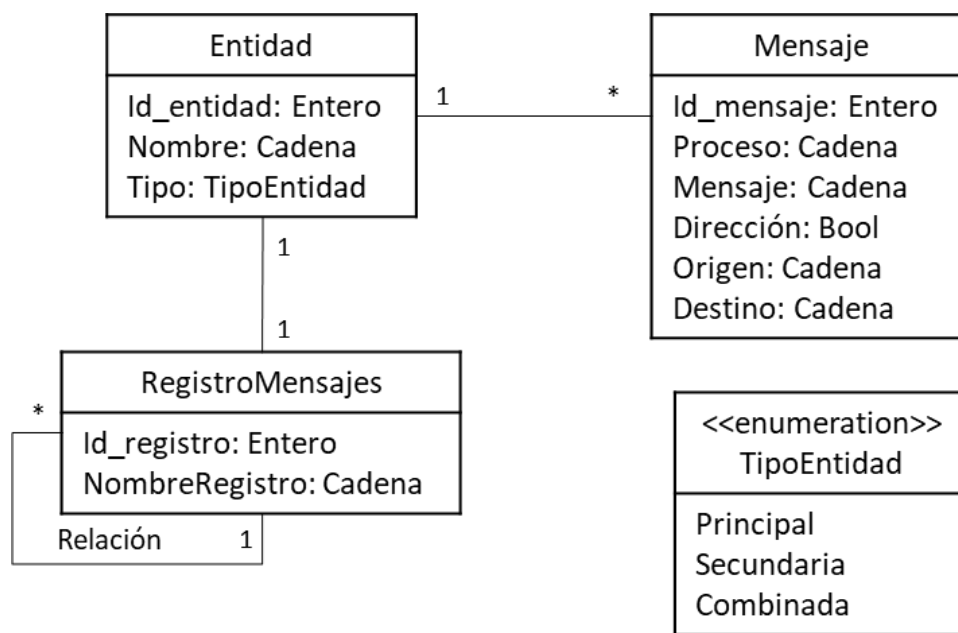


Figura 4.5 Modelo del dominio

En la figura 4.5 se presenta el modelo de dominio propuesto para el perfil UML, donde se declaran los elementos que conforman los diagramas de entidades, donde:

- Una entidad puede enviar N cantidad de mensajes, pero varios mensajes solo tendrán un propietario.
- La entidad principal puede relacionarse con una o varias entidades secundarias, pero una secundaria solo tendrá una entidad principal.
- Se genera una tabla de registro, asociada a una relación, que contiene todos los mensajes enviados entre dos entidades.

4.2.2. Definición del perfil UML

Los elementos presentados en el meta-modelo anterior pasarán a ser los *estereotipos* que se incluirán en el perfil UML. Un estereotipo es un nuevo tipo de elemento del modelo concebido por el modelador y basado en un tipo de elemento del modelo existente (en este caso un tipo de diagrama de UML).

Los estereotipos deben poseer el mismo nombre que los elementos del meta-modelo. Por tanto, los estereotipos definidos en esta etapa son:

- Entidad.
- RegistroMensaje
- Mensaje

Cabe mencionar que el estereotipo *RegistroMensaje*, representa también el elemento relación. En la tabla 4.3 se describe cada estereotipo y el elemento que representa. Los elementos mostrados en la tabla corresponden al meta-modelo presentado en la figura 4.5.

Tabla 4.3 Descripción de los estereotipos

Estereotipo	Elemento	Descripción
Entidad	Entidad	Representa una persona u organización involucrada con el dominio del problema/aplicación. Una entidad puede ser: principal, secundaria o combinada.

RegistroMensaje	RegistroMensaje	Representa las relaciones (o transacciones) entre las entidades y a la vez contendrá los mensajes/eventos enviados entre las entidades. Se agregarán manualmente.
Mensaje	Mensaje	Representa los mensajes que son enviados de una entidad a otra. Un mensaje tiene asignado un identificador único, una dirección (envío o recibido), un origen, un destino y un proceso asociado.

4.2.3. Identificación de los elementos UML a extender

Los estereotipos son los elementos que poseerá nuestro perfil UML. Sin embargo, el siguiente paso es identificar los elementos de UML, que se van a extender, es decir, con elementos nos referimos a las meta-clases propias de UML. Dado que un perfil UML es un mecanismo de extensión, es necesario definir las meta-clases a extender. Una meta-clase es un elemento definido dentro del UML, que posee una sintaxis y una semántica, que puede extenderse.

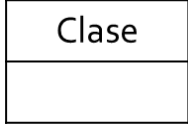
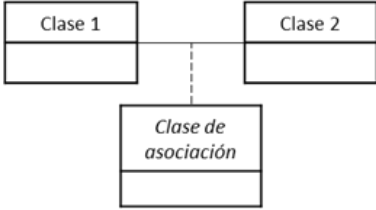
Para esta investigación, resultó más sencillo, verificar qué diagrama dentro de UML poseía los elementos necesarios para la construcción de los diagramas de entidades y permitían una mejor representación visual de dicho diagrama para mayor comodidad del usuario. Para esta verificación se utilizó el documento estándar de UML 2.5 establecido por la OMG.

Los elementos que se extenderán pertenecen al diagrama de clase que se encuentra descrito en la sección 3.3.2. *Diagrama de Clases*. Dichos elementos son meta-clases propias del metamodelo de un diagrama de clase a nivel general. Esto es así debido a que todos los diagramas de UML poseen una meta-modelo que los describe.

Se tomaron los elementos que se incluirán en el perfil y se utilizarán más adelante para realizar un diagrama de entidades. En la tabla 4.4 se

presentan los estereotipos (primera columna), las meta-clases (segunda columna) y la notación empleada para las meta-clases (tercera columna).

Tabla 4.4 Asociación de los estereotipos y meta-clases

Estereotipo	Metaclase ¹	Notación
Entidad	<i>Class</i>	
RegistroMensaje	<i>AssociationClass</i>	
Mensaje	<i>Attribute</i>	<i>attribute</i>

4.2.4. Definición de los valores etiquetados

En los pasos anteriores, se identifican los estereotipos y las meta-clases. Como se mencionó en el paso de Identificación de los elementos a extender, los estereotipos corresponden a los elementos definidos en el meta-modelo de los diagramas de entidades, y las meta-clases son los elementos propios de UML. Sin embargo, si el meta-modelo creado posee atributos, estos deben agregarse al perfil UML, como valores etiquetados. Un valor etiquetado es una propiedad adicional que corresponde a un estereotipo definido en un modelado adicional (es decir el perfil que se desea generar).

En este paso, los atributos definidos en el meta-modelo (ver Figura 4.5) se agregarán como valores etiquetados a los estereotipos, respetando la relación que existe entre los elementos del meta-modelo. Es decir, si el elemento Entidad, tiene como atributo *id_entidad*, este pasará como valor etiquetado al estereotipo *Entidad*, respetando el nombre. Así se hará para la mayoría de los atributos.

¹ Los nombres de las meta-clases mostrados en la tabla son tomados del documento estándar UML definido por (OMG Unified Modeling Language, 2015).

Otro punto importante y necesario es definir el tipo de dato que se le dará al valor etiquetado, ya que se requiere al momento de construir el perfil. Por tanto, en la tabla 4.5 se muestran los valores etiquetados, su tipo y una breve descripción para su mejor comprensión.

Tabla 4.5 Descripción de los elementos

Estereotipo	Valor etiquetado	Tipo	Descripción
Entidad	<i>id_entidad</i>	Entero	Identificador único para cada entidad presentada en el diagrama.
	<i>Nombre</i>	Cadena	Nombre que se le da a la entidad. Esta propiedad es heredada de la meta-clase <i>Class</i> .
	<i>Tipo</i>	Principal	Representa una organización, usuario o aplicación primordial del diagrama. Solo existe una entidad principal en el diagrama y se sitúa en la parte central del espacio de trabajo.
		Secundaria	Representa una organización, usuario o aplicación que esté relacionada con la entidad principal. Pueden existir de 1 a N entidades secundarias en el diagrama. Se colocan alrededor de la entidad principal.
Combinada		Representa una organización, usuario o aplicación. Se utiliza en el diagrama de entidades combinado donde no se toma en cuenta el concepto de entidad " <i>principal</i> " o " <i>secundaria</i> ".	
Registro Mensajes	<i>id_registro</i>	Entero	Identificador para cada tabla (T) de registro que se genera entre las entidades al enviar o recibir un mensaje.
	<i>nombre_relación</i>	Cadena	Nombre de la relación que existe entre dos entidades. Esta propiedad es heredada la meta-clase <i>Class</i> .
Mensaje	<i>id_Mensaje</i>	Entero	Identificador único para cada mensaje agregado al estereotipo <i>Registro</i> .

	<i>proceso</i>	Cadena	Nombre del proceso que indica el conjunto de mensajes existentes entre varias entidades.
	<i>mensaje</i>	Cadena	Mensaje enviado entre una entidad a otra. Esta propiedad será heredada de la meta-clase <i>Attribute</i> .
	<i>dirección</i>	Boolean	Representa el estado del mensaje, mediante 0 (o true) o 1 (false). Donde 0, representa mensaje enviado y 1 representa mensaje recibido. Esto depende del envío del mensaje en relación a la entidad <i>principal</i> .
	<i>origen</i>	Cadena	Nombre de la entidad origen, de la cual sale el mensaje.
	<i>destino</i>	Cadena	Nombre de la entidad destino, a la cual se le envía el mensaje.

Puede ser que algunos valores etiquetados mostrados en la tabla no se agreguen al diseño del perfil UML, debido a que las meta-clases tienen propiedades propias y al extenderlas, los estereotipos heredan dichas propiedades, así que, para no duplicarlas, se podría optar por no incluirlas en el diseño y hacer uso de las propiedades propias de las meta-clases.

4.2.5. Definición de las restricciones

Es necesario definir una serie de restricciones para el perfil UML conforme a lo expresado en el meta-modelo. Estas restricciones deben describirse en el lenguaje OCL (*Object Constraint Language*), que es un lenguaje definido por la OMG, y se considera el estándar para transcribir restricciones y consultas. En este caso, se consideró presentar las restricciones de manera textual, sin utilizar el lenguaje OCL, para evitar confusiones en el desarrollo.

Las restricciones que se implementaran en el perfil UML son:

- En un diagrama debe existir mínimo 2 entidades y una relación entre estas.
- Solo puede existir una entidad principal.

- Cada entidad (principal o secundaria) deberá contar con un identificador y nombre único.
- Cada mensaje debe tener un identificador numérico único.
- Pueden existir varias relaciones entre dos entidades.
- En la línea de relación se agrega solo el identificador del mensaje (es decir en el elemento *registroMensaje*).
- Cada identificador de mensaje debe coincidir con el que se encuentra en la tabla *RegistroMensajes*
- En la tabla *RegistroMensajes* para todos los mensajes que existan entre dos entidades.

4.2.6. Diseño del perfil UML

Finalmente el perfil UML para los diagramas de entidades se presenta en la figura 4.6. En dicho perfil se muestran los estereotipos con sus respectivos valores etiquetados y las meta-clases que se extienden.

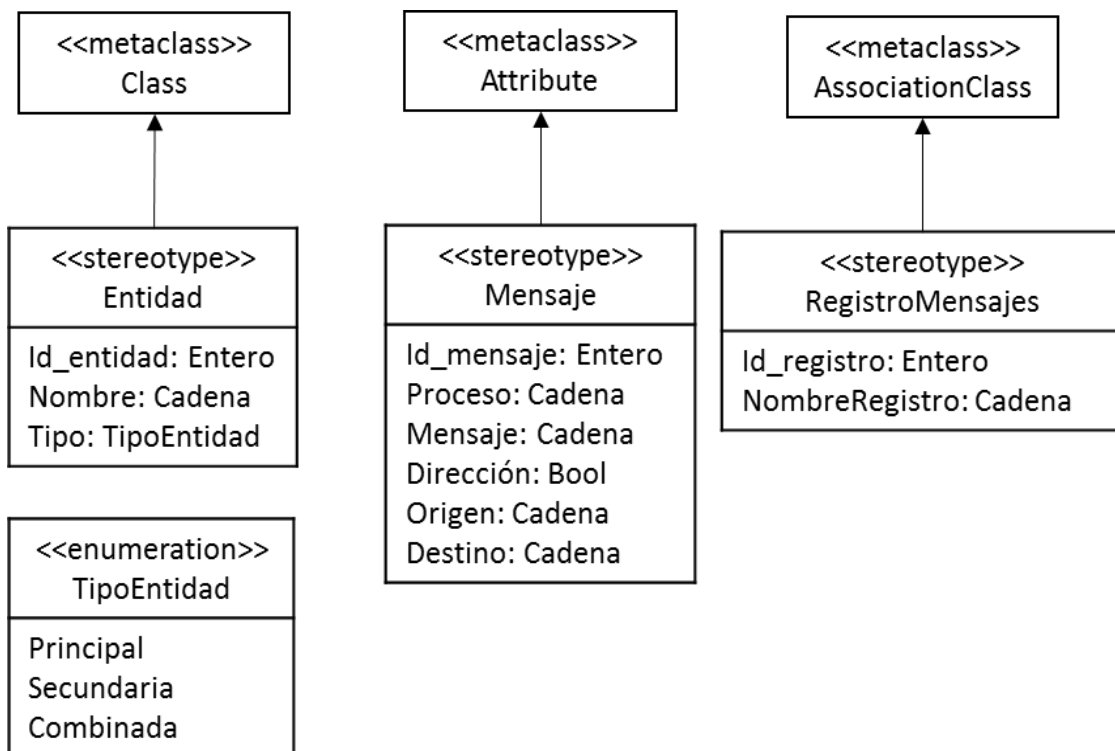


Figura 4.6 Perfil UML para los diagramas de entidades

4.3 Representación de los diagramas de entidades a partir del perfil UML

En la metodología de Warnier/Orr se tiene que, por cada mensaje enviado de entre dos entidades, se genera una línea, que indica la relación de comunicación y junto a esta, se coloca el nombre del mensaje. En los tres diagramas de entidades, se establece esa regla, para representar el flujo de los mensajes. Sin embargo, si existe una gran cantidad de información entre dos entidades, se puede llegar a una saturación de mensajes, es decir, visualmente el diagrama contendrá una gran cantidad de líneas, lo cual provoca que, al interpretar el diagrama, la información no sea percibida de forma adecuada.

Por lo que, fue necesario buscar un diagrama de UML que permitiera una mejor representación del flujo de los mensajes. Además, que es un requerimiento necesario para la elaboración del perfil UML, porque se deben identificar los elementos de UML a extender (las meta-clases), de los cuales se heredaran todas sus propiedades. En la sección *4.3.6 Diseño del Perfil UML*, se define la elaboración del perfil UML, partiendo de los elementos de un diagrama de entidades, hasta el diseño final del perfil.

Conforme a lo establecido anteriormente, la notación de los diagramas de entidades cambia (de una elipse al símbolo de una clase), pero las reglas de construcción se conservan. Por tanto, a continuación, se presenta una vista preliminar de los diagramas de entidades acorde a la notación establecida en el perfil UML y adecuando los ejemplos descritos en la sección *4.1.1.3. Representación de los diagramas de entidades combinados*, definidos de acuerdo a la metodología de Orr (Orr, 1981).

4.3.1. Diagramas de entidades conforme al perfil UML

Los diagramas de entidades que se presentan son:

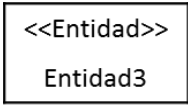
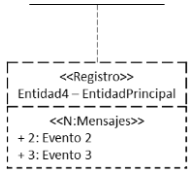
- Diagramas de entidades a nivel usuario.
- Diagrama de entidades combinado a nivel usuario.
- Diagrama de entidades a nivel aplicación.

Los tres diagramas poseen características similares y emplean los mismos elementos para su construcción, solo cambia el nivel en que se realizan. Los elementos identificados de estos diagramas son:

- **Entidad.** Representa una persona u organización involucrada con el dominio de aplicación.
- **RegistroMensajes.** Representa las transacciones que pueden existir entre las entidades y a su vez posee una tabla, que contendrá los mensajes enviados entre las entidades.
- **Mensaje.** Representa los mensajes que se envían de una entidad a otra. Un mensaje tiene asignado un identificador único y una dirección. El mensaje debe empezar con un número seguido del símbolo “:” seguido del mensaje. Tiene la siguiente nomenclatura: *N:Mensaje*. Donde:
 - **N:** Número del mensaje.
 - **Mensaje:** Mensaje enviado o recibido.

En la tabla 4.6 se muestran las propiedades de los elementos identificados del diagrama de entidades.

Tabla 4.6 Elementos de la adecuación de los diagramas de entidades

Elementos	Símbolo	Propiedades	Descripción
Entidad		<i>Identificador numérico</i>	Identificador único para cada entidad presentada en el diagrama.
		<i>Nombre</i>	Nombre que se le asigna a la entidad.
		<i>Tipo de entidad</i>	El tipo de entidad puede ser: Principal, secundaria o combinada.
Registro de mensajes		<i>Identificador numérico</i>	Identificador único para cada registro de mensajes.
		<i>Nombre del registro</i>	Representa el nombre que tendrá el registro de mensajes. Este nombre debe coincidir con las entidades que se relacionan entre sí.
		<i>Identificador numérico</i>	Identificador único del mensaje,

Mensaje		<i>Proceso</i>	Nombre del proceso, el cual contiene una conjunto de mensajes entre entidades.
		<i>Mensaje</i>	Mensaje que se envía de una entidad a otra.
		<i>Dirección</i>	Representa el estado del mensaje, este puede ser de enviado o recibido, respecto a la entidad principal.
		<i>Origen</i>	Nombre de la entidad origen, de la cual sale el mensaje.
		<i>Destino</i>	Nombre de la entidad destino, a la cual se le envía el mensaje.

4.3.2. Adecuación del diagrama de entidades a nivel usuario

En las figuras 4.7 y 4.8 se muestran los diagramas de entidades a nivel usuario, que están descritos en la sección 4.1.1.3. *Representación de los diagramas de entidades combinados*. En el primer diagrama, los mensajes que se envían entre *Entidad3* y la entidad "*Usuario 1*" (*entidad principal*), se colocan en la Tabla *Registro*, nombrada *Usuario1-Entidad3*. El mensaje entre las entidades es *1: Evento 1*, el número indica que es el primer mensaje enviado en el diagrama.

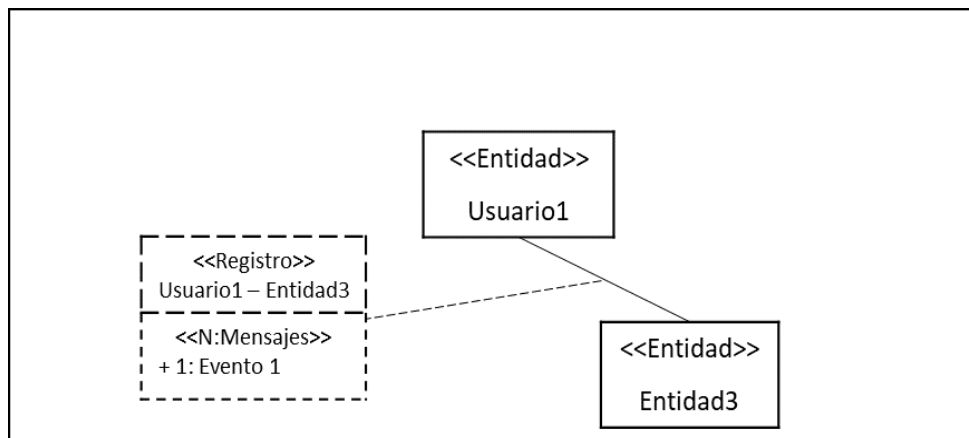


Figura 4.7 Adecuación del diagrama de entidades del usuario 1

En el segundo diagrama (ver figura 4.8), la entidad principal es *Usuario3*, y las entidades secundarias son *Entidad4* y *Entidad2*. Los mensajes que se envían entre *Usuario3* y *Entidad2* son: *4: Evento 4* y *5:Evento 5*.

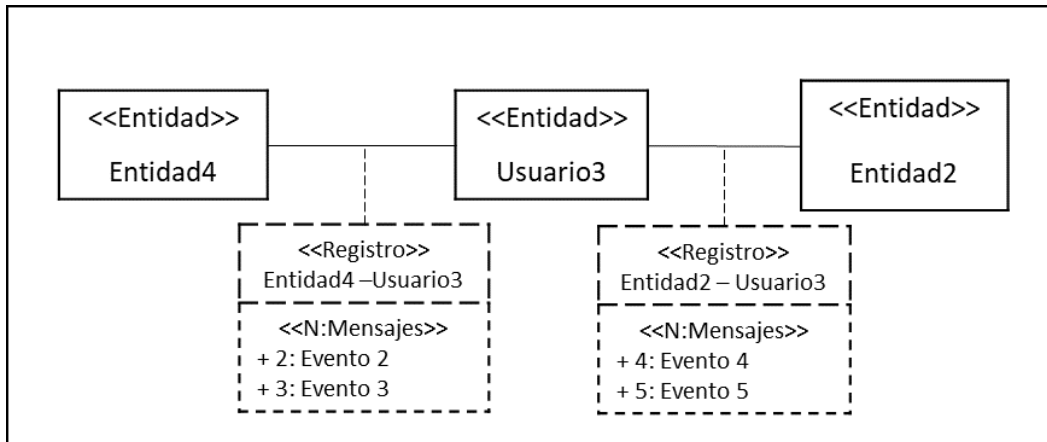


Figura 4.8 Adecuación del diagrama de entidad a nivel usuario (2)

4.3.3. Adecuación del diagrama de entidades combinado a nivel usuario

En la figura 4.9 se muestra el diagrama de entidades combinado, que es una adecuación del diagrama de la figura 4.3 presentado en la sección 4.1.1.3. *Representación de los diagramas de entidades combinado*. Se respetan las reglas de construcción, lo único que cambia es la notación definida en el perfil UML. Este diagrama es la combinación de los dos diagramas de entidades a nivel usuario mostrados anteriormente.

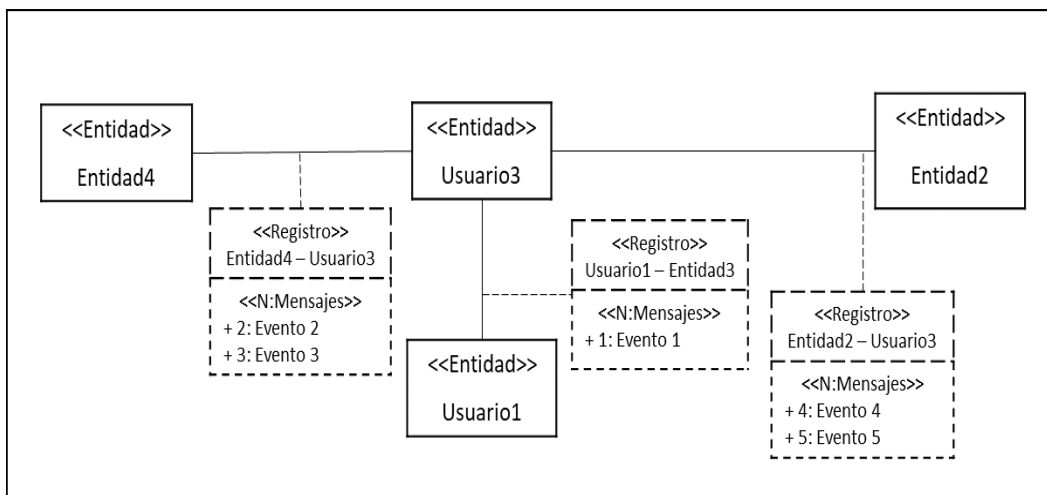


Figura 4.9 Adecuación del diagrama de entidades combinado

4.3.4. Adecuación del diagrama de entidades a nivel aplicación

En el último diagrama de entidades es a nivel aplicación (figura 4.10), se genera una nueva entidad, que es la *Entidad Aplicación*, como se muestra en la figura siguiente. En este caso, se consideran incluidas en la aplicación las entidades *Usuario1*, *Entidad2* y *Usuario3*, así como las transacciones y eventos asociados: *Usuario1-Usuario3* y *Entidad2-Usuario3*.

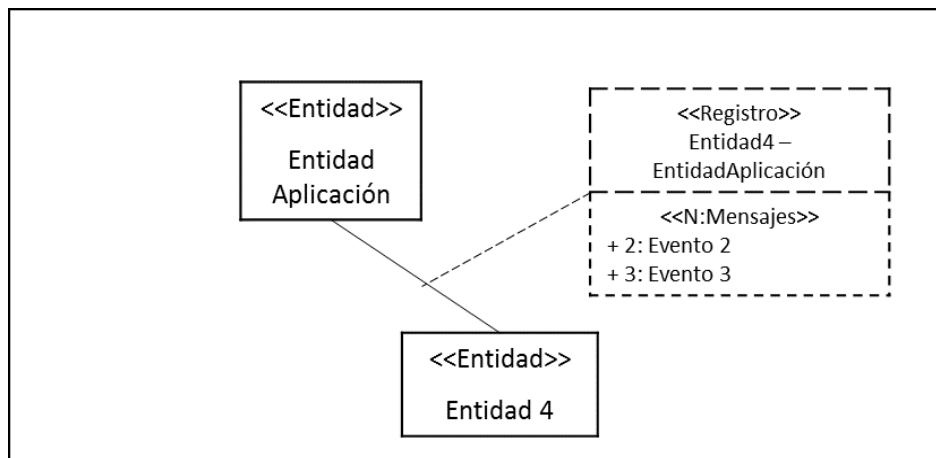


Figura 4.10 Adecuación del diagrama de entidad a nivel aplicación

En este capítulo se presentó la definición de los elementos de los diagramas de entidades, así como los pasos para diseñar un perfil UML, incluyendo los estereotipos, meta-clases y valores etiquetados. También se mostró una representación de los diagramas de entidades respecto al perfil elaborado.

En el siguiente capítulo se presenta un breve análisis y evaluación de las herramientas de modelado y el por qué se seleccionó *Enterprise Architect* para el desarrollo el perfil UML.

CAPÍTULO 5. CREACIÓN DEL PERFIL UML PARA EL DSL: DIAGRAMAS DE ENTIDAD Y GENERACIÓN DE HERRAMIENTA DE EDICIÓN

En este capítulo se exponen las herramientas de modelado que se evaluaron para crear el perfil UML y los pasos para generar el editor del DLS Diagramas de entidad en la herramienta de modelado *Enterprise Architect*.

En este capítulo se describen las características de las herramientas de modelado que se evaluaron con la finalidad seleccionar una que permitiera la creación e implementación de un perfil UML. En la sección 5.1. se exponen un resumen de las herramientas de modelado. En la sección 5.2 se explica el desarrollo del perfil UML con la herramienta seleccionada, *Enterprise Architect*, que incluye cinco sub-secciones: el desarrollo del perfil, la generación del editor y la aplicación del editor.

5.1. Análisis y evaluación de las herramientas de modelado

En este apartado se presentan las herramientas de modelado que pueden utilizarse para construir un perfil UML. Cabe mencionar que inicialmente se planeó utilizar la herramienta Papyrus para desarrollar los perfiles UML, pero conforme se fue diseñando y realizando las actividades para construir el perfil se encontraron una serie de inconvenientes. Por lo cual se evaluaron otras herramientas, que permitieran realizar todas las actividades, que es la aplicación del perfil. A continuación, se presenta una introducción acerca de las herramientas consultadas y posteriormente la elaboración del diagrama con los elementos del perfil. El diseño del perfil UML que se utiliza para configurar las herramientas se encuentra descrito en la sección 4.2 *Diseño del perfil UML*.

5.1.1. Papyrus

Papyrus ofrece un soporte muy avanzado para perfiles UML que permite a los usuarios definir editores para DSLs (*Domain Specific Language*) basados en el estándar UML2.

Con base en la documentación consultada en (Eclipse, 2016), el marco Papyrus permite la creación de editores, creando y aplicando perfiles UML. Es decir, permite extender el lenguaje UML, agregando algunos elementos y restricciones, pero respetando la sintaxis y semántica original. Los autores de (Acevedo, Gutiérrez, Silva, & Gallardo, 2016) mencionan que estas adaptaciones se definen en relación con los mecanismos de extensibilidad de UML, que son: estereotipos, valores etiquetados y restricciones.

5.1.1.1. Elaboración y aplicación del perfil UML

Para realizar el perfil se consultó la página de (Eclipse, 2016) y la guía oficial que proporciona Papyrus realizado por (Gérard, 2011). Cuando se comenzó a utilizar la herramienta, se creó el ejemplo presentado en el documento, pero se encontraron errores al aplicarlo, dado que la representación del perfil no era correcta. Por tanto, se consideró realizar el prototipo de perfil que se muestra en la figura 5.1.

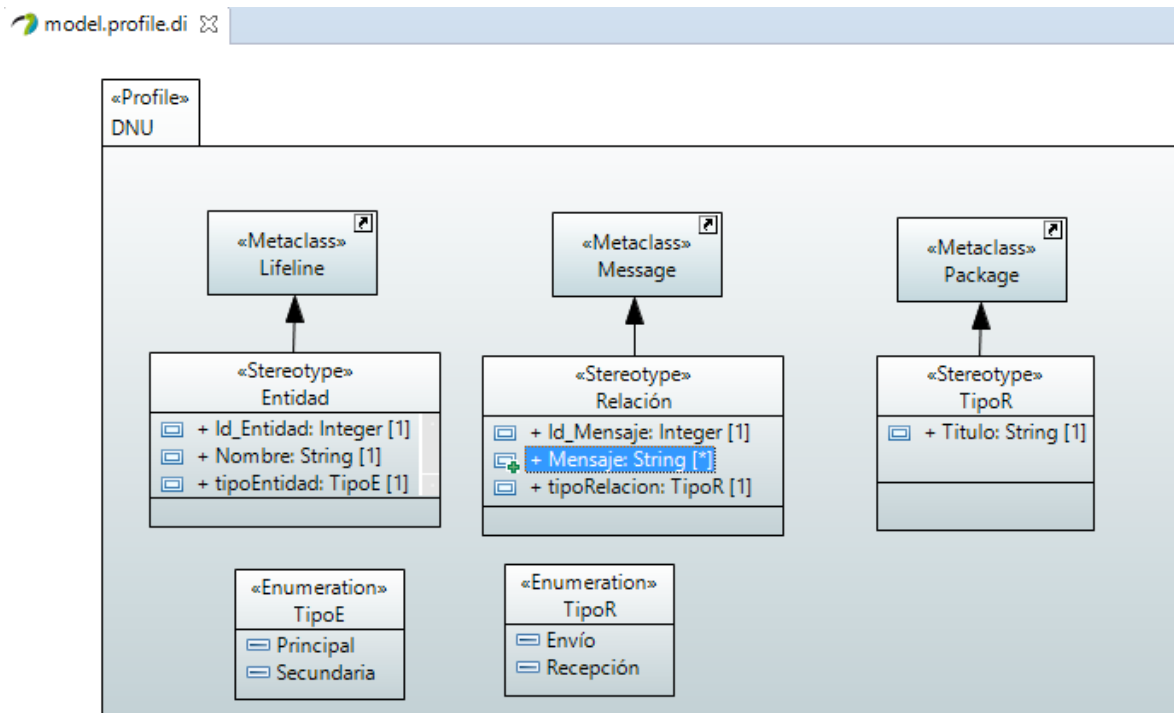


Figura 5.1 Perfil UML elaborado en Papyrus

Cuando se tiene el diagrama del perfil UML, se debe crear un nuevo proyecto y elegir el diagrama UML, que define los estereotipos. Cada elemento del perfil se agrega al modelo que se edita, seleccionándolo desde la ventana de propiedades. Solo se agregan estereotipos a los elementos que se definieron en el perfil.

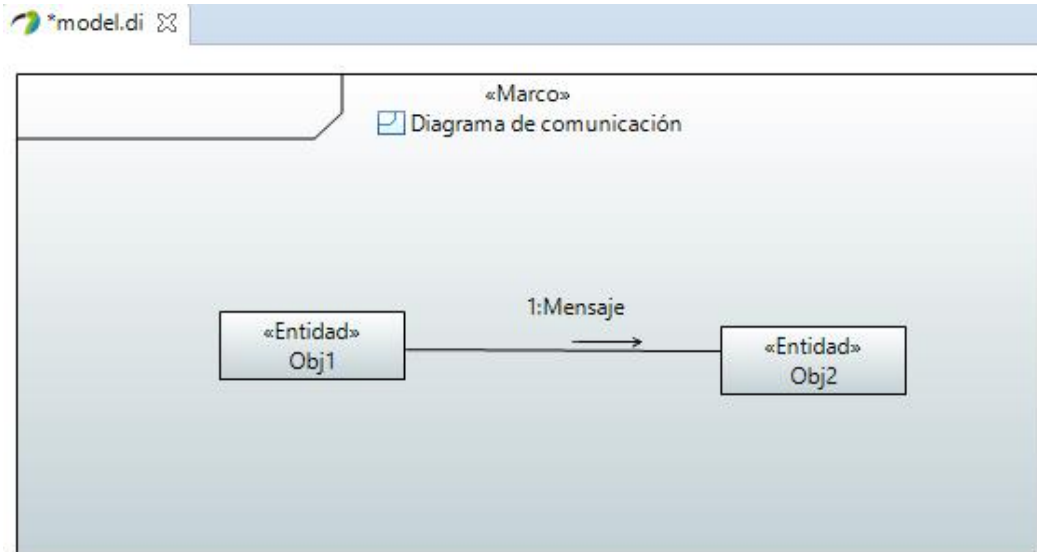
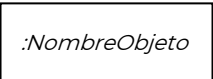

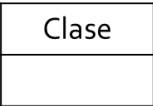


Figura 5.2 Aplicación del perfil UML (Papyrus)

En la figura 5.2 se muestra un diagrama creado a partir del perfil UML, los elementos que se incluyeron se describen en la tabla 5.1.

Tabla 5.1 Elementos definidos

Estereotipo	Metaclase	Notación	Descripción	Atributos
Entidad	<i>Lifeline</i>		Representa una persona u organización involucrada con el dominio de aplicación. Una entidad puede ser: principal o secundaria.	<i>Id_Entidad</i> <i>Tipo</i>
Relación	<i>Association</i>		Representa las interacciones que pueden existir entre las entidades.	-
Registro	<i>Class</i>		Contendrá los mensajes enviados entre las entidades. Se agregarán manualmente.	-
Mensaje	<i>Attribute</i>	<<N:mensaje>>	Representa los mensajes que son enviados de una entidad a otra. Un mensaje tiene asignado un identificador único y una dirección (envío o recibido).	<i>Id_Mensaje</i> <i>Dirección</i>

En Papyrus no se encontró la opción de importar el perfil como un archivo XML para utilizarse, en otra herramienta. Aunque cuenta con la opción de integrarlo a un editor creado con la herramienta, mediante la creación de *plugins*.

5.1.2. Enterprise Architect

Enterprise Architect (EA) es una herramienta de modelado que se basa en UML 2.5 para el diseño y construcción de sistemas de software, para el modelado de procesos de negocio y para fines de modelado general, tales como la visualización de los sistemas y procesos existentes.

EA tiene un mecanismo de perfil UML genérico para cargar y trabajar con diferentes perfiles. Los perfiles UML se presentan en archivos XML, con un formato específico. Estos archivos XML se pueden importar en la herramienta. Una vez importados, se genera un editor, en el que se pueden arrastrar y soltar los elementos del perfil en el diagrama que incluye los elementos del DSL. EA adjuntará el estereotipo, los valores y los valores predeterminados, notas etiquetadas e incluso un meta-archivo si se especifica uno, para el nuevo perfil (Sparx Systems Pty Ltd., 2017).

5.1.2.1. Elaboración y aplicación del perfil UML

Se realizó el diagrama de perfil en Enterprise Architect, con algunas variaciones en el nombre de las meta-clases. Analizando los diagramas de UML, se consideró extender el diagrama de comunicación, y se requirió que meta-clases corresponden a un elemento del diagrama de comunicación, se tuvo que generar uno y así conocer el nombre que la herramienta le asigna. En la figura 5.3, se muestran las meta-clases que se considera extender, para el diseño del perfil, para el DSL “diagrama de entidades”.

En la definición de los diagramas de entidades, se requiere definir reglas como (incluir un ejemplo). Así que se utilizó la asignación de las reglas OCL, que se basa en el estándar definido por la (OMG, 2006). Utilizando el OCL, fue posible establecer las reglas asociadas a cada elemento, donde se requiere. Cuando se tiene el diagrama del perfil, la herramienta EA facilita la exportación a un archivo XML 2.1, que se encuentra estructurado con base al estándar de UML soportado por la OMG (OMG Unified Modeling Language, 2015). El archivo XML se presenta en la figura 5.4.

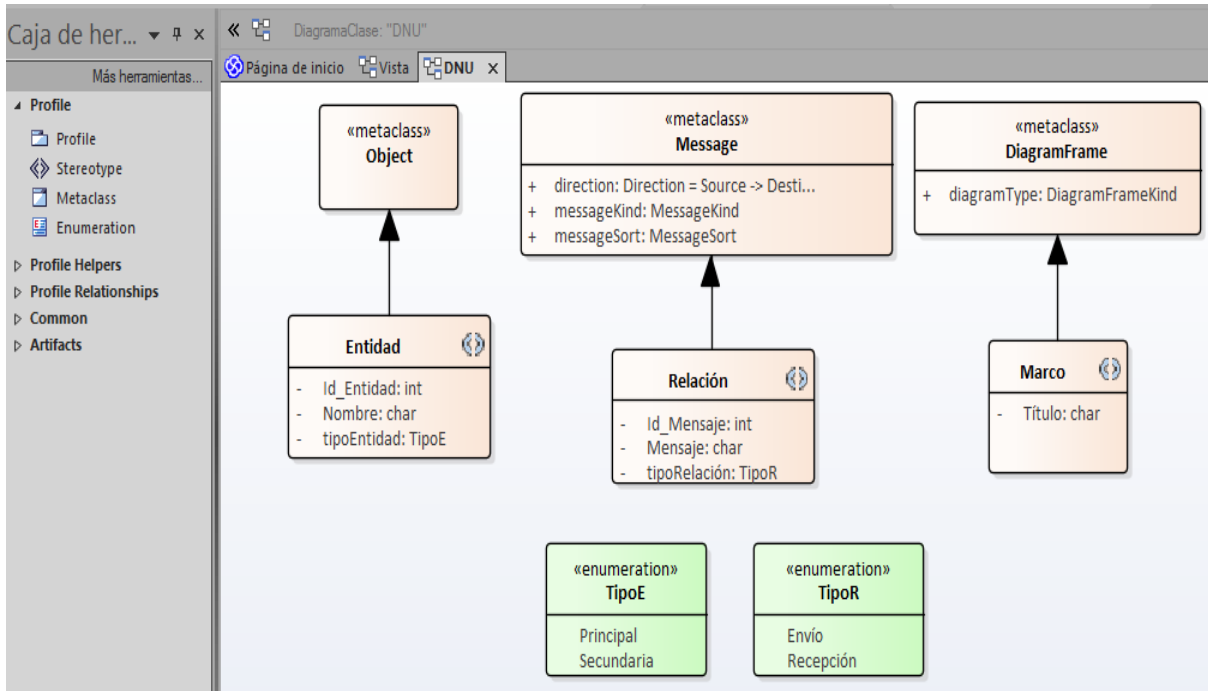


Figura 5.3 Perfil UML elaborado en Enterprise Architect

```

1 <?xml version="1.0" encoding="Windows-1252"?>
2 <UMLProfile profiletype="uml2">
3 <Documentation id="487B1P74-5" name="DNU" version="1.0" notes="DNU"/>
4 <Content>
5 <Stereotypes>
6 <Stereotype name="Entidad" notes="" cx="0" cy="0" bgcolor="-1" fontcolor="-1" bordercolor="-1" borderwidth="-1" hideicon="0">
7 <AppliesTo>
8 <Apply type="Object"/>
9 </AppliesTo>
10 <TaggedValues>
11 <Tag name="Id_Entidad" type="int" description="" unit="" values="" default=""/>
12 <Tag name="Nombre" type="char" description="" unit="" values="" default=""/>
13 <Tag name="tipoEntidad" type="enumeration" description="" unit="" values="Principal,Secundaria" default=""/>
14 </TaggedValues>
15 </Stereotype>
16 <Stereotype name="Marco" notes="" cx="0" cy="0" bgcolor="-1" fontcolor="-1" bordercolor="-1" borderwidth="-1" hideicon="0">
17 <AppliesTo>
18 <Apply type="DiagramFrame">
19 <Property name="diagramType" value=""/>
20 </Apply>
21 </AppliesTo>
22 <TaggedValues>
23 <Tag name="Titulo" type="char" description="" unit="" values="" default=""/>
24 </TaggedValues>
25 </Stereotype>
26 <Stereotype name="Relación" notes="" cx="0" cy="0" bgcolor="-1" fontcolor="-1" bordercolor="-1" borderwidth="-1" hideicon="0">
27 <AppliesTo>
28 <Apply type="Message">
29 <Property name="direction" value="Source &gt; Destination"/>
30 <Property name="messageKind" value=""/>
31 <Property name="messageSort" value=""/>
32 </Apply>
33 </AppliesTo>
  
```

Figura 5.4 Archivo XML generado de EA

El archivo XML generado se importa dentro de la herramienta, de manera que se habilita un editor, que permite hacer uso de los elementos definidos en el perfil. Así que EA, permite crear un editor, y soporta el Desarrollo Dirigido por Modelos (*Model Driven Development*), que permite añadir un perfil UML a la Vista de Recursos de la herramienta.

EA tiene tres paquetes para trabajar con perfiles: 1) la creación del perfil, donde se define sus estereotipos, meta-clases y restricciones; 2) la caja de herramientas personalizada, es para definir páginas de la Caja de herramientas; 3) los tipos de diagramas, para crear sus propios tipos de diagramas personalizados.

El editor, que utiliza el perfil creado, se muestra en la figura 5.5.

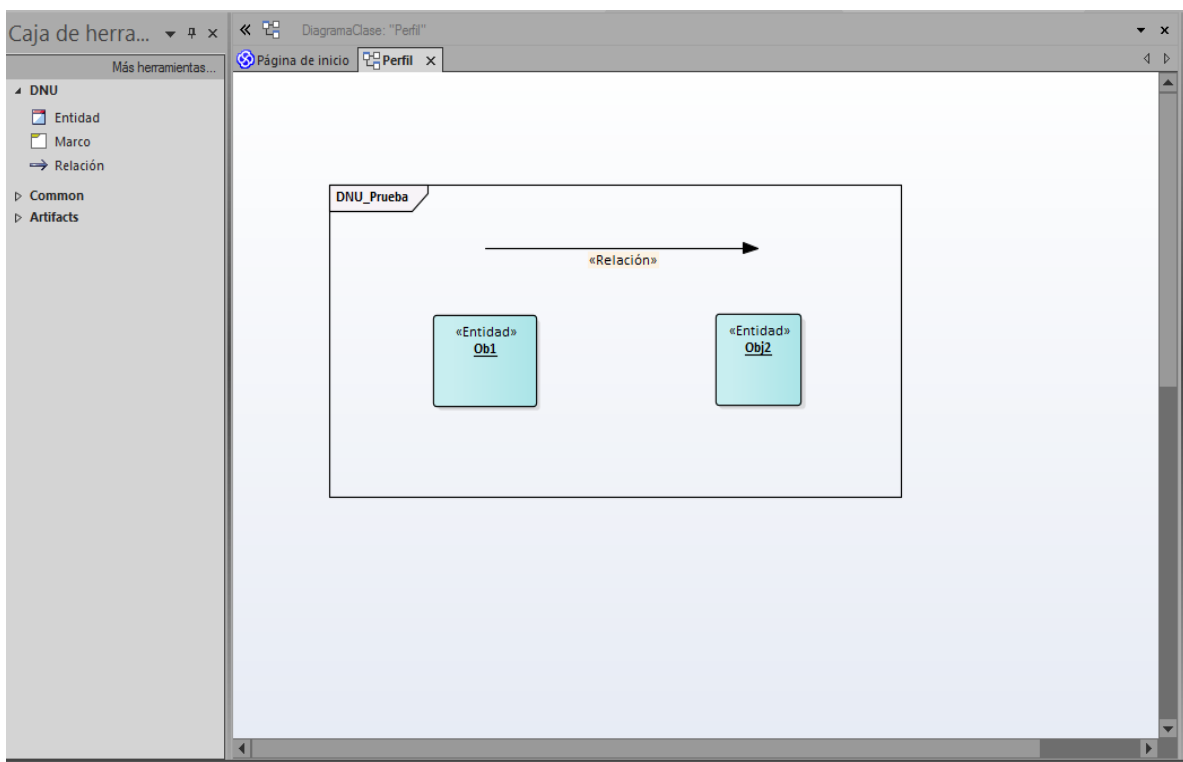


Figura 5.5 Aplicación del Perfil UML (Enterprise Architect)

Los pasos que se siguieron para la creación de un perfil UML en *Enterprise Architect* se tomaron de (Sparx Systems, 2016).

5.1.3. Análisis de otras herramientas de modelado

Otras herramientas de modelado que se analizaron para la elaboración de perfiles UML son:

- **StarUML.** De acuerdo a la página oficial (MKLab, 2014), la herramienta permite la construcción y utilización de perfiles. Sin embargo, al crear el perfil propuesto se notó que StartUML no permitía el uso de las meta-clases requeridas de UML. En cuanto a la aplicación del perfil, solo muestra la opción de aplicar un perfil que ya está pre-cargado, no aparecen los otros perfiles generados. A pesar de que permite exportar el diagrama de perfil en un archivo XML, este no contiene la estructura estándar definida por (OMG Unified Modeling Language, 2015). Esta herramienta solo permite crear el diagrama de perfil, más no utilizarlo.
- **Altova UModel.** Dentro de los diagramas que se pueden crear con la herramienta está el diagrama de perfil. Altova UModel permite definir estereotipos, definiciones de etiquetas y restricciones personalizadas en un diagrama UML especial. La personalización de estos estereotipos, etiquetas y restricciones se definen en un perfil, que después se aplica a un paquete (Altova, 2017). Aun cuando la herramienta permite crear este tipo de diagramas no admite la creación de perfiles UML y generación del mismo en un archivo XML.
- **Visual Paradigm.** La herramienta permite la generación de perfiles, sin embargo, sólo se puede definir y aplicar un estereotipo a un elemento (meta-clase) de un diagrama en específico, es decir, no cuenta con la opción de elegir otras meta-clases de distintos diagramas de UML y de crear otros estereotipos. Por tanto, no fue posible construir el perfil UML.

5.1.4. Tabla comparativa

Se compararon cinco herramientas de modelado para determinar sus facilidades para trabajar con los perfiles UML. La mayoría de las herramientas permiten la creación de los perfiles UML, sin embargo, solo se modela como un diagrama, pero no se pueden utilizarse para crear una aplicación para editar diagramas del DSL a soportar.

Los parámetros que se tomaron en cuenta para comparar las herramientas, se muestran en la tabla 5.1, y son:

- **Licencia.** Tipo de licencia que emplea el software, pueden ser: software libre o comercial.
- **Versión utilizada.** La versión de la herramienta que se utilizó para generar el perfil UML.
- **Versión de UML.** La versión de UML que emplea la herramienta. La versión de UML va desde la 1.0 a la 2.5.
- **Versión de XML soportada.** Si brindan el soporte al estándar XML conforme lo establecido por la OMG. Las versiones estándar de XML son 1.1 y 2.1.
- **Soporte XML para perfiles UML.** Si la herramienta permita la exportación del perfil en un archivo XML, con la estructura establecida por la OMG. En el documento oficial de UML, se presenta un etiquetado específico para perfil UML.
- **Transformaciones.** Si la herramienta soporta el mecanismo de transformaciones MDA (*Model Driven Architect*), de tipo Modelo a modelo o modelo a texto.
- **Permite la creación de transformaciones.** La herramienta permite la creación y aplicación de transformaciones. Algunas herramientas lo permiten, pero implementan diferentes mecanismos.
- **Existe documentación actualizada.** Si la herramienta cuenta con un soporte actual para la generación de perfiles UML.
- **Fácil de comprender.** La herramienta es fácil de comprender (de acuerdo a la documentación) para la generación de perfiles.
- **Grado de complejidad.** Si la herramienta es difícil de manejar y cuenta con documentación actualizada sobre la generación de perfiles UML.

Cabe mencionar que los parámetros anteriores están basados bajo el criterio de desarrollar un perfil UML en cada herramienta y no se contemplan todos los aspectos de modelado.

Tabla 5.2 Comparación de las herramientas de modelado

Herramientas Características	Papyrus	Enterprise Architect		StarUML	Altova UModel	Visual Paradigm
Licencia	Libre	Comercial		Comercial	Comercial	Comercial
Versión utilizada	2.0.1	8.0	13.0	2.8	2013	7.1
Versión de UML	2.5	1.3, 2.0-2.1	2.5	2.0	2.4	2.1
Versión de XML soportada	N/A	1.1, 2.1		2.1	XMI 2.4	-
Soporte para Perfiles UML	Si	Si		Parcialmente	Parcialmente	Parcialmente
Soporte XML para perfiles UML	No	Si		No	No	No
Transformaciones	Si	Si		Si	Si	Si
Permite la creación de transformaciones	Si	Si		No	No	No
¿Existe documentación actualizada?	Si	Si		Si	Si	Si
¿es fácil de comprender?	Parcialmente	Si		Parcialmente	Si	Si
¿grado de complejidad?	Medio	Medio		Fácil	Medio	Fácil

Las herramientas de modelado *StarUML*, *Altova UModel* y *Visual Paradig* permiten la diagramación del perfil UML, más no su implementación y aplicación. Mientras que *Papyrus* y *Enterprise Architect*, cuentan con mejor soporte que admite la creación, implementación y aplicación de perfiles UML. Todas admiten algún tipo de transformación, en su gran mayoría de modelo a texto.

Esta comparación surgió por la necesidad de seleccionar una herramienta que permitiera el mecanismo completo de perfiles UML. Sin embargo, con base a la información consultada y la experiencia que se obtuvo al generar un perfil en *Papyrus* y *Enterprise Architect*, se concluyó que, aunque ambas herramientas poseen una gama de opciones para el desarrollo y soporte de perfiles UML, el grado de complejidad y la dificultad para integrar un perfil UML en *Papyrus*, y la falta de documentación, resultó ser una elección menos viable para desarrollar el perfil UML. Por tanto, para esta investigación se optó por *Enterprise Architect*, dado que posee una documentación más completa y los mecanismos adecuados para generar un editor.

En el siguiente apartado se describe la creación del perfil UML para el DSL: diagramas de entidad y generación de la herramienta de edición en *Enterprise Architect*.

5.2. Desarrollo del perfil UML en *Enterprise Architect*

En este apartado se presenta el procedimiento que se llevó a cabo para desarrollar el perfil UML, y que se utilizará para construir el editor de los diagramas de entidades. Los elementos que se integrarán en el diagrama *Profile* se encuentran descritos en la sección 4.2.6. *Diseño del perfil UML*.

5.2.1. Generar proyecto

Para generar un perfil UML, es necesario crear un proyecto en *Enterprise Architect*, siguiendo los pasos:

1. Generar un nuevo proyecto en la herramienta. Dar clic en el icono de *Enterprise Architect*, seleccionar la opción *New Project*. (ver figura 5.7)

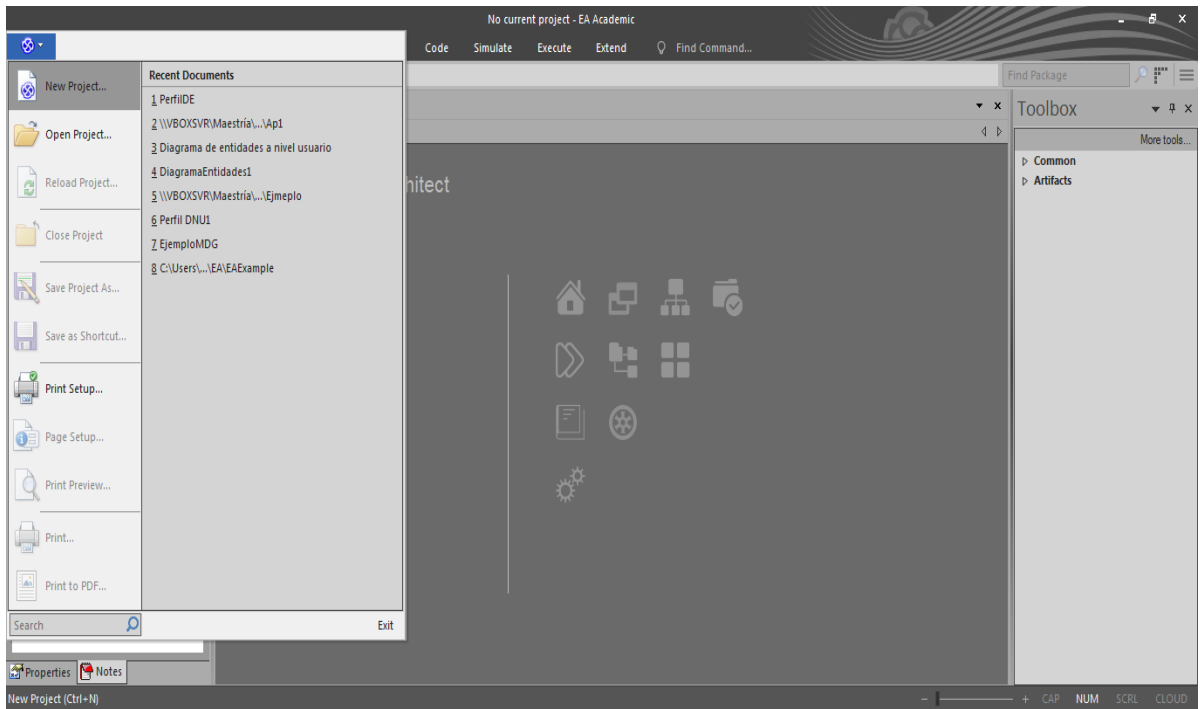


Figura 5.6 Seleccionar la opción New Project

2. En la nueva ventana, elegir la ruta donde se mostrará guardará el proyecto, ingresar el nombre del proyecto y después dar clic en *Guardar*. (ver figura 5.8)

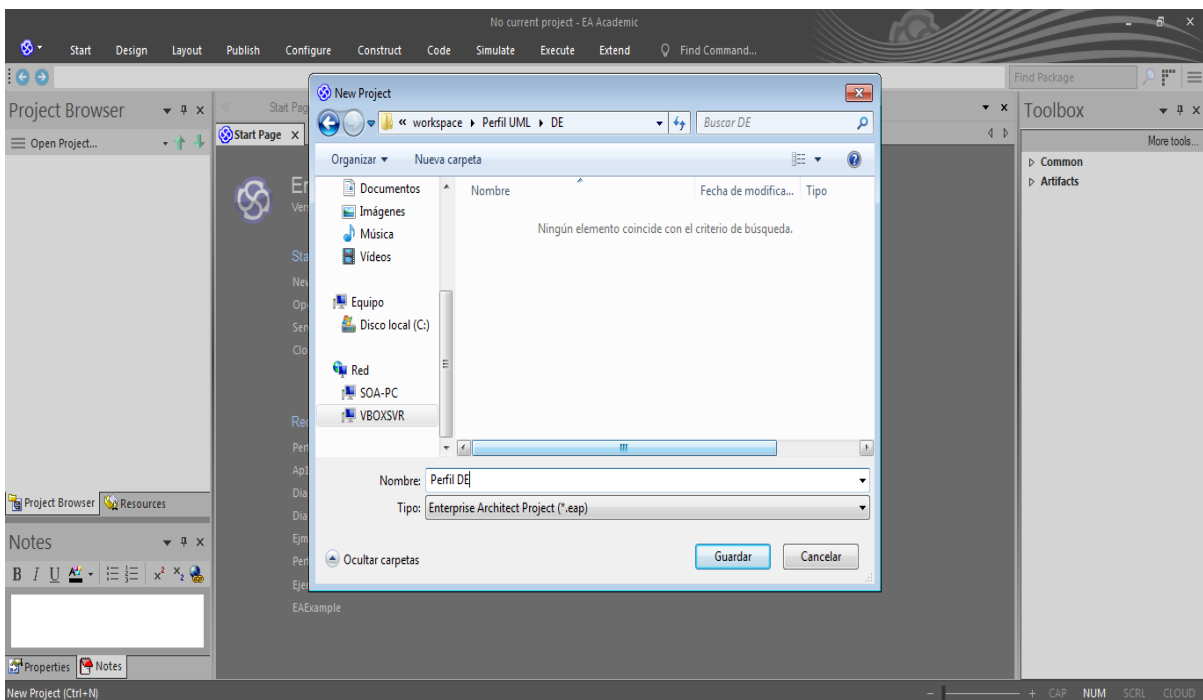


Figura 5.7 Guardar proyecto

3. Al crear el proyecto, la herramienta genera un modelo (*model*) por defecto, el cual se encuentra en el panel *Project Browser*, como se muestra en la figura 5.9.

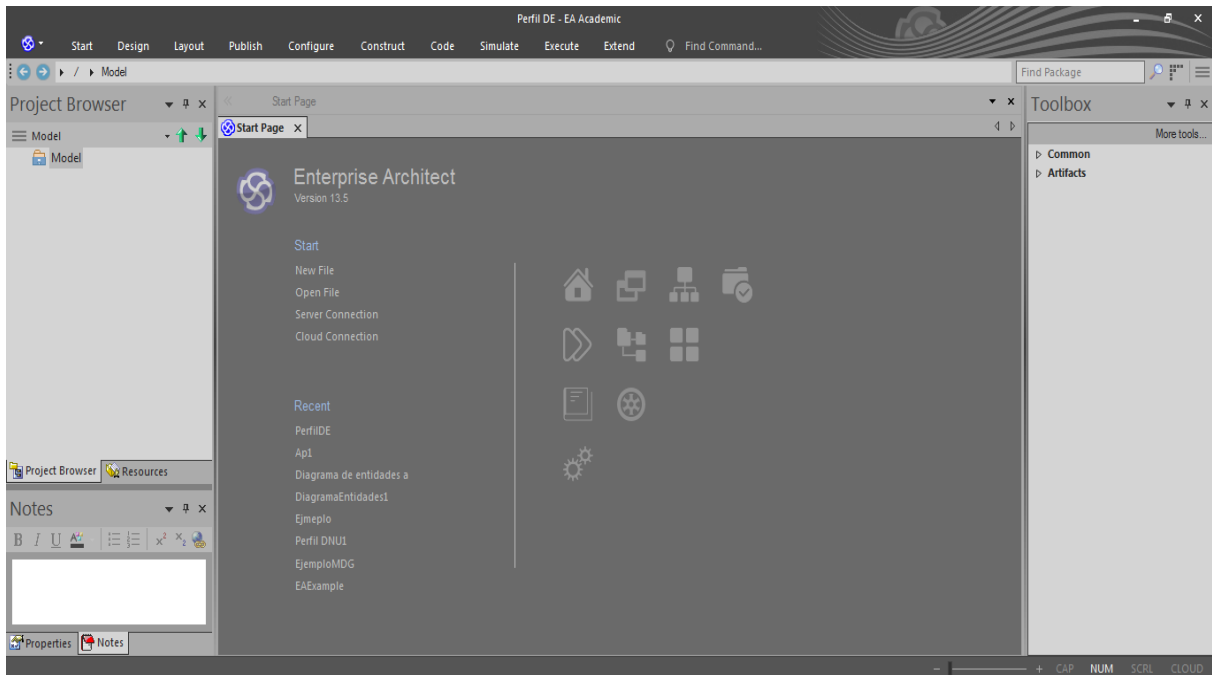


Figura 5.8 Model

4. Es necesario crear el siguiente nivel de un proyecto en EA, que es la vista (*Views*). Dar clic derecho en el Model, seleccionar *Add -> Add View*.

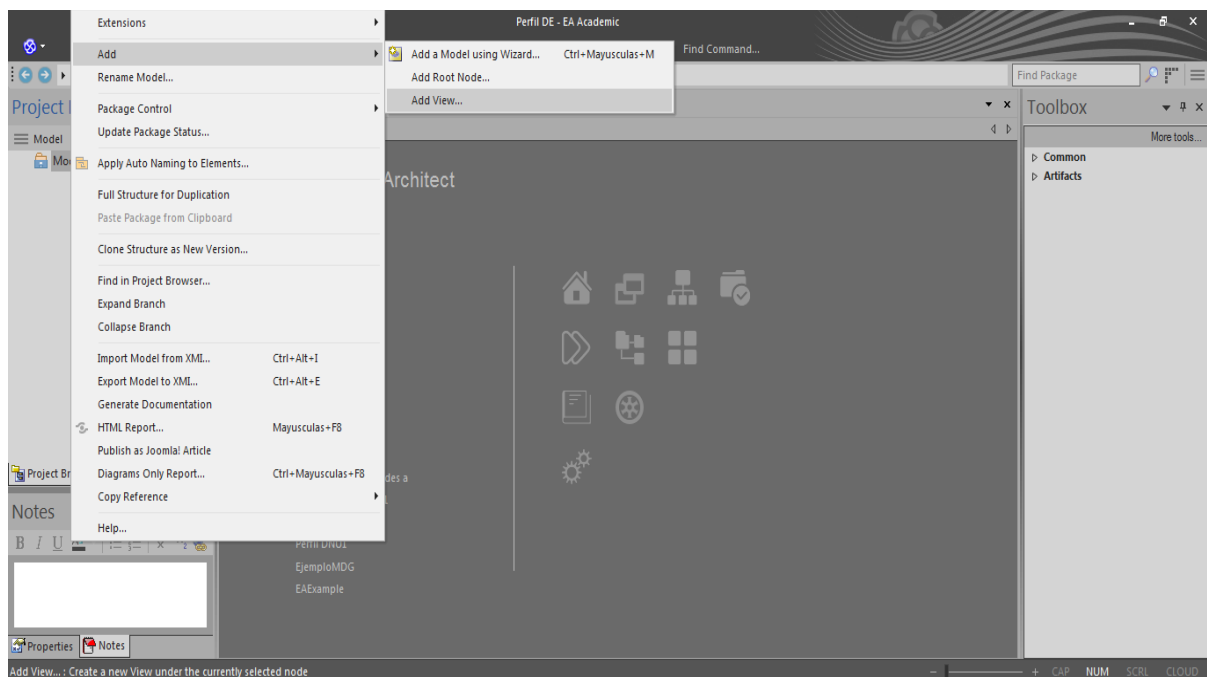


Figura 5.9 Agregar vista (View)

5. En la figura 5.11, se muestra una ventana, donde se debe introducir el nombre de la vista y después dar clic en el botón *Ok*.

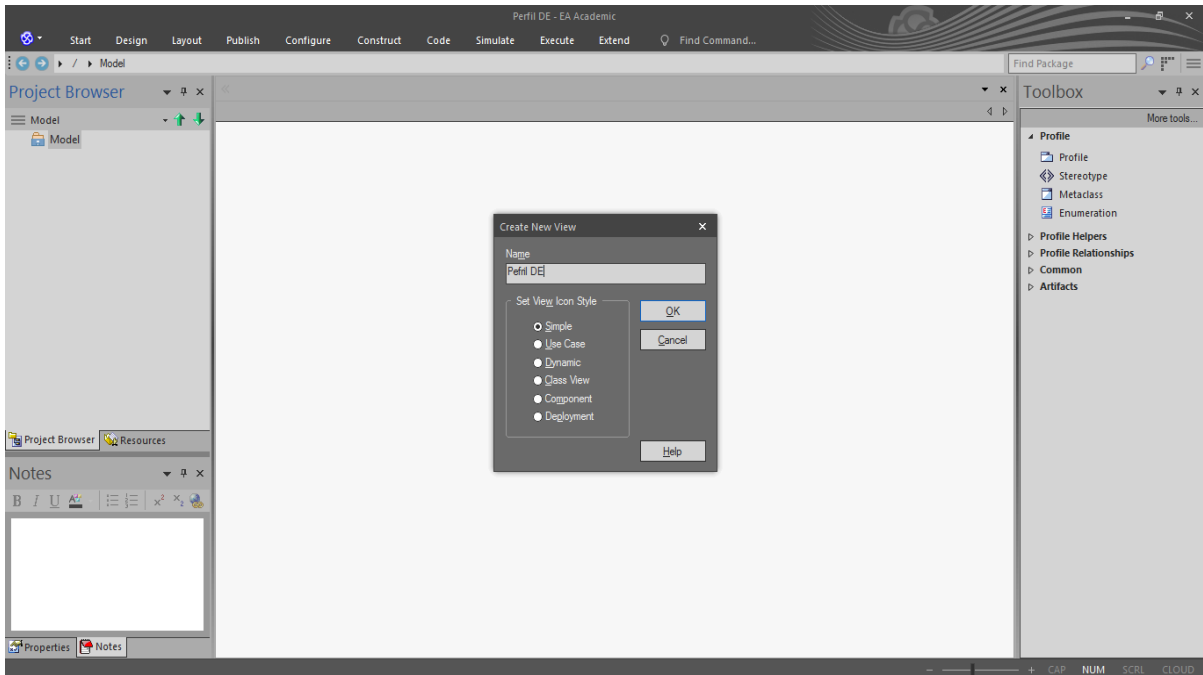


Figura 5.10 Generar vista (View)

6. Se puede visualizar la vista generada (View) en el panel *Project Browser* (figura 5.12).

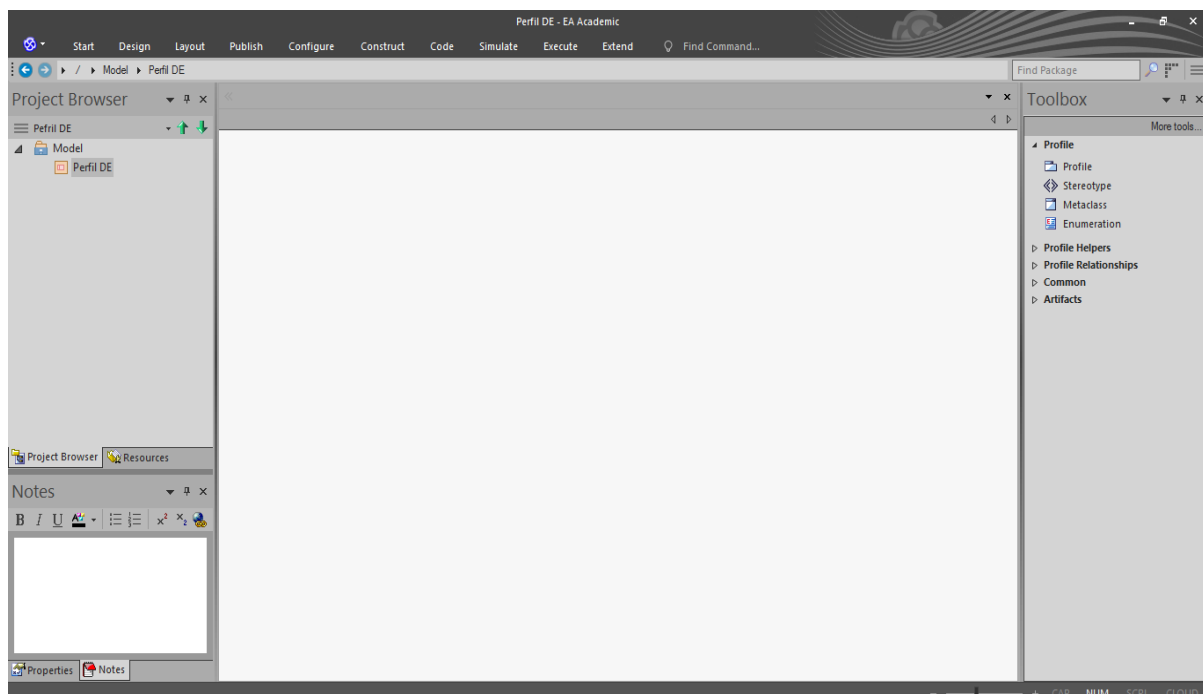


Figura 5.11 Vista (View)

7. Se procede a agregar un diagrama para realizar el perfil UML. Dar clic derecho en la vista (*View*) y seleccionar *Add diagram*, como se muestra en la figura 5.13.

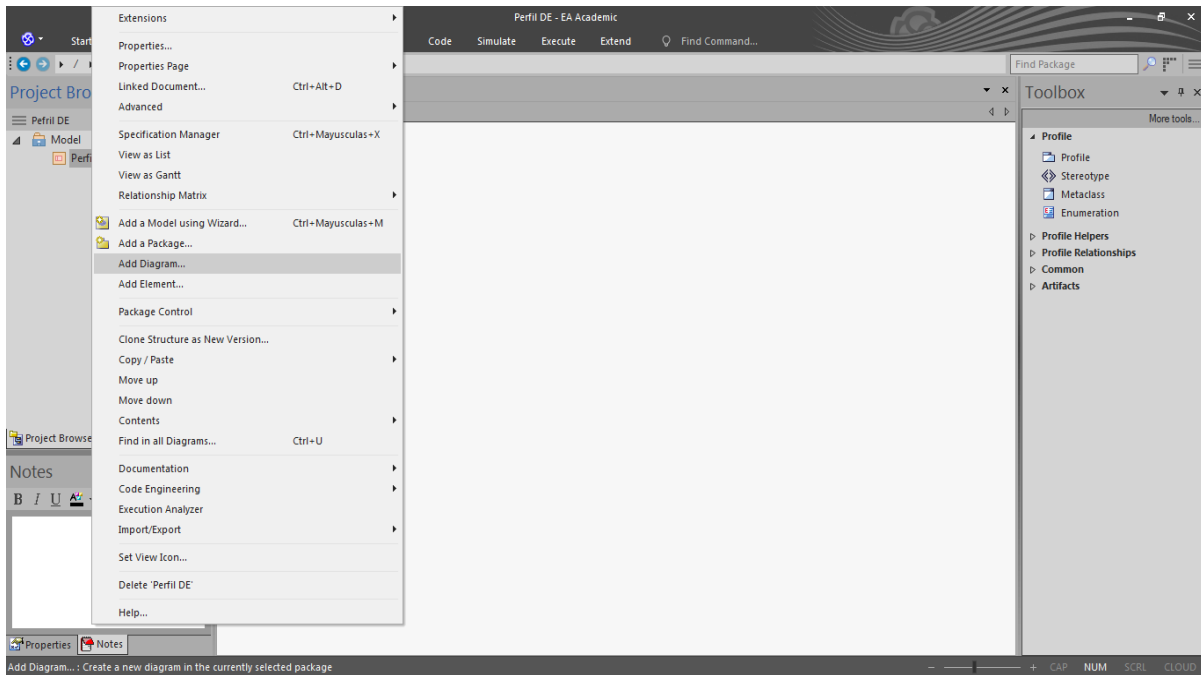


Figura 5.12 Agregar diagrama

8. En la figura 5.14 se muestra una ventana para agregar un diagrama, para eso seleccionar *UML Structural* -> *Class* y dar clic en el botón *Ok*.

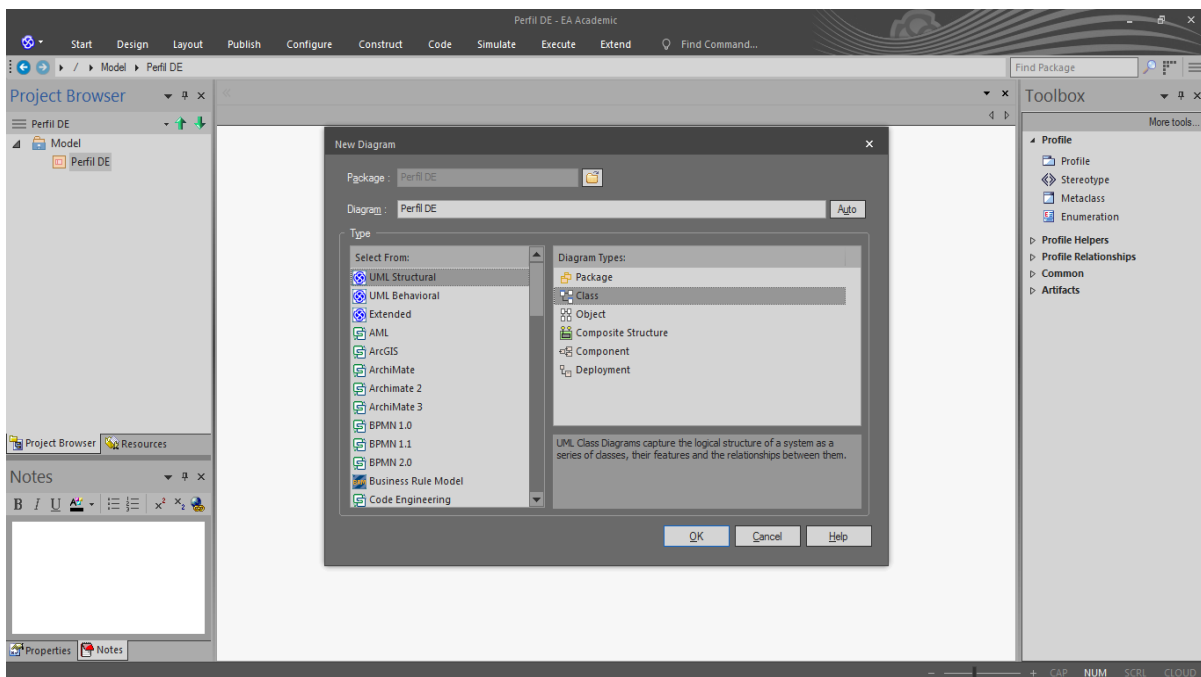


Figura 5.13 Seleccionar diagrama

9. Se genera el área de trabajo para crear el diagrama y se muestra en el panel Project Browser en el diagrama creado (ver figura 5.15).

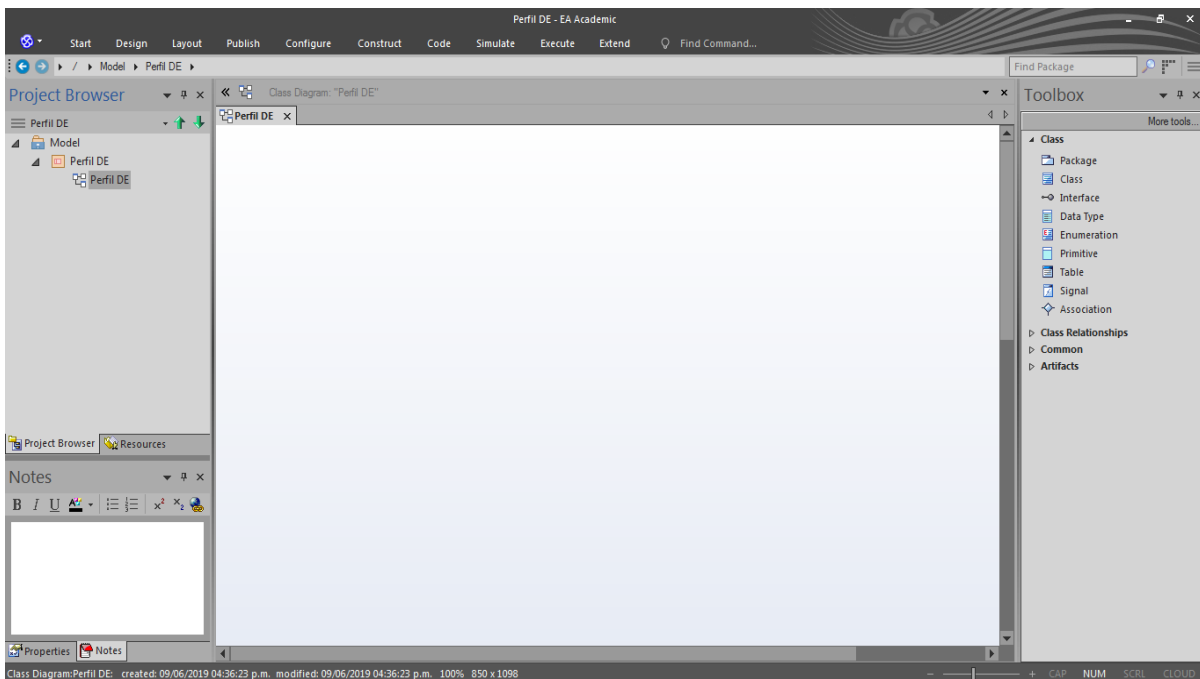


Figura 5.14 Área de trabajo para generar el diagrama

5.2.2. Desarrollo del perfil UML

Una vez creado el proyecto, se procede a desarrollar el perfil UML con el mecanismo *MDG Technologies* que pertenece a la herramienta EA. A continuación, se presentan los pasos para realizar el perfil UML.

1. Una vez generado el proyecto, se debe seleccionar en el panel *Toolbox*, la caja de herramientas para desarrollar un perfil UML. Dando clic en *More tools*, ubicado en el panel *Toolbox*, como se muestra en la figura 5.16.

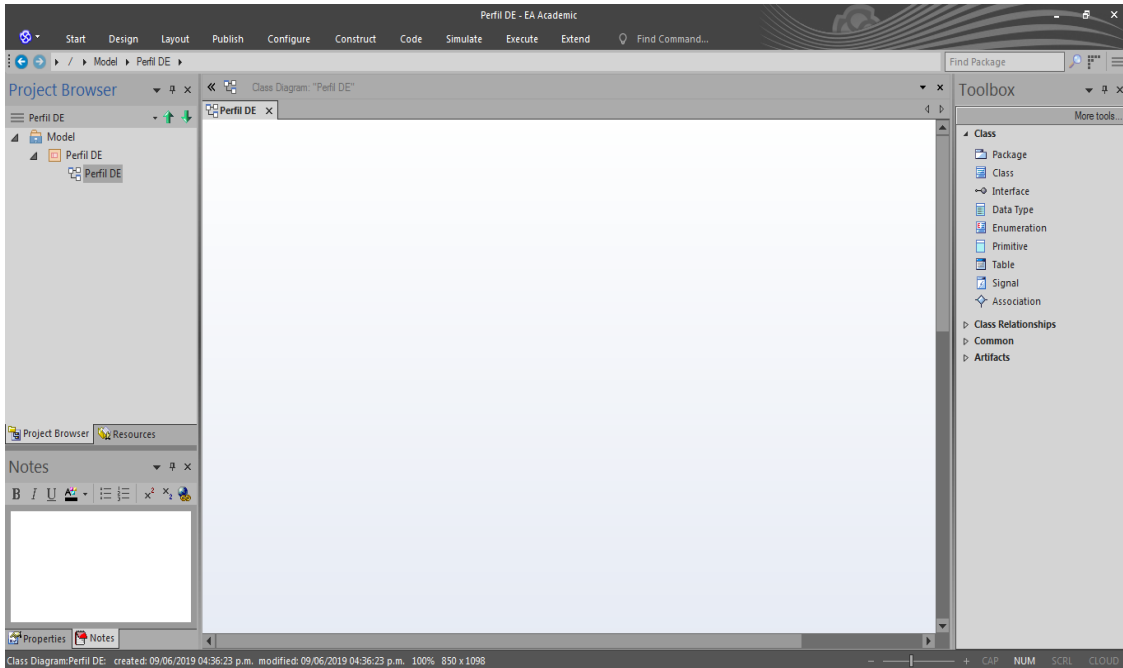


Figura 5.15 Panel Toolbox

2. En el menú desplegable, seleccionar la opción *Profile* (figura 5.17).

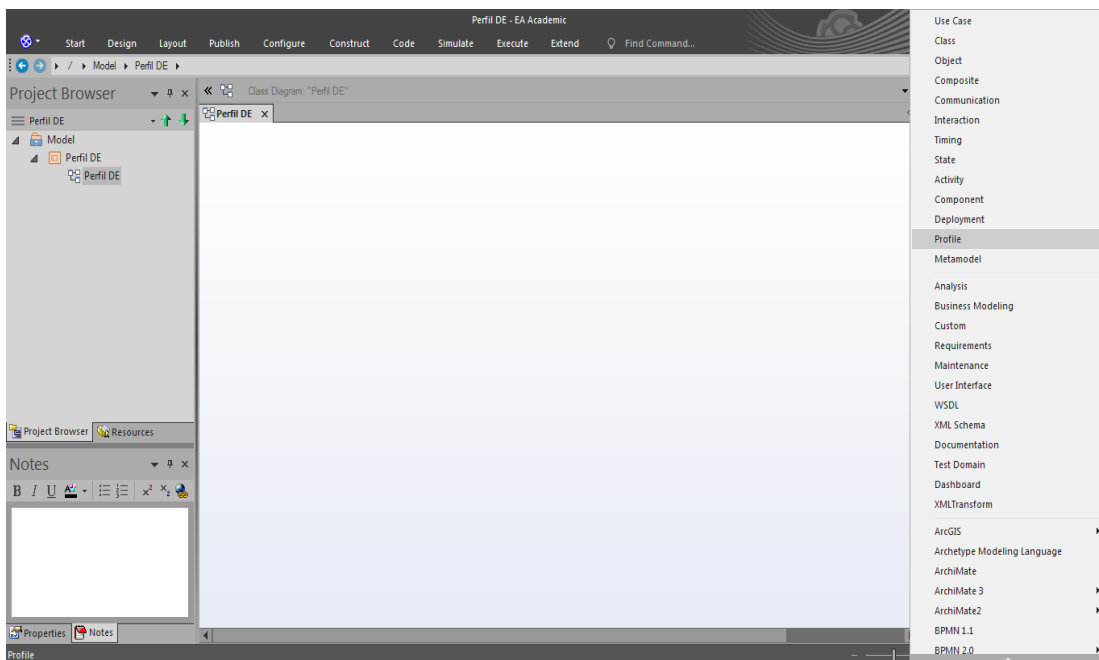


Figura 5.16 Seleccionar opción Profile

3. Se muestra en el panel Toolbox, los elementos que utilizan para construir un perfil UML. Seleccionar la opción *MDG Technology* ubicado en la sección *Profile Helpers*.

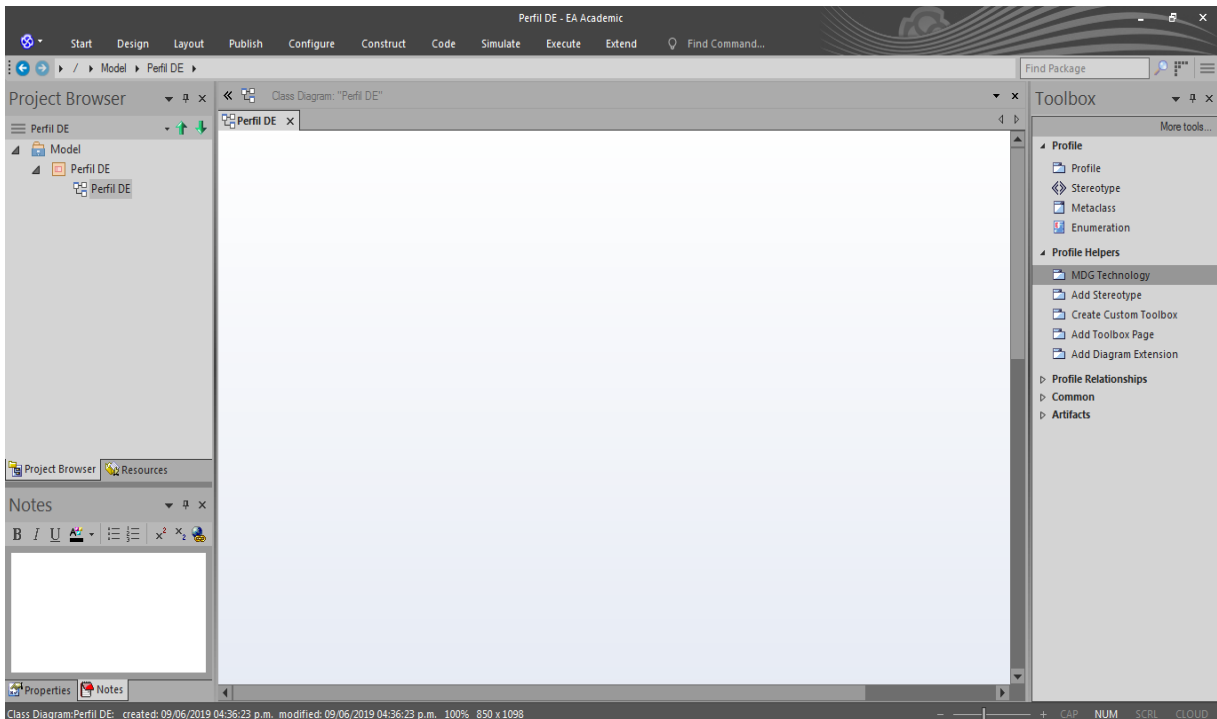


Figura 5.17 Elemento MDG Tecnología

4. En la ventana que aparece se debe ingresar el nombre de la tecnología a crear. En este caso es el perfil y se define como Perfil DE, DE por Diagrama de Entidades (figura 5.19).

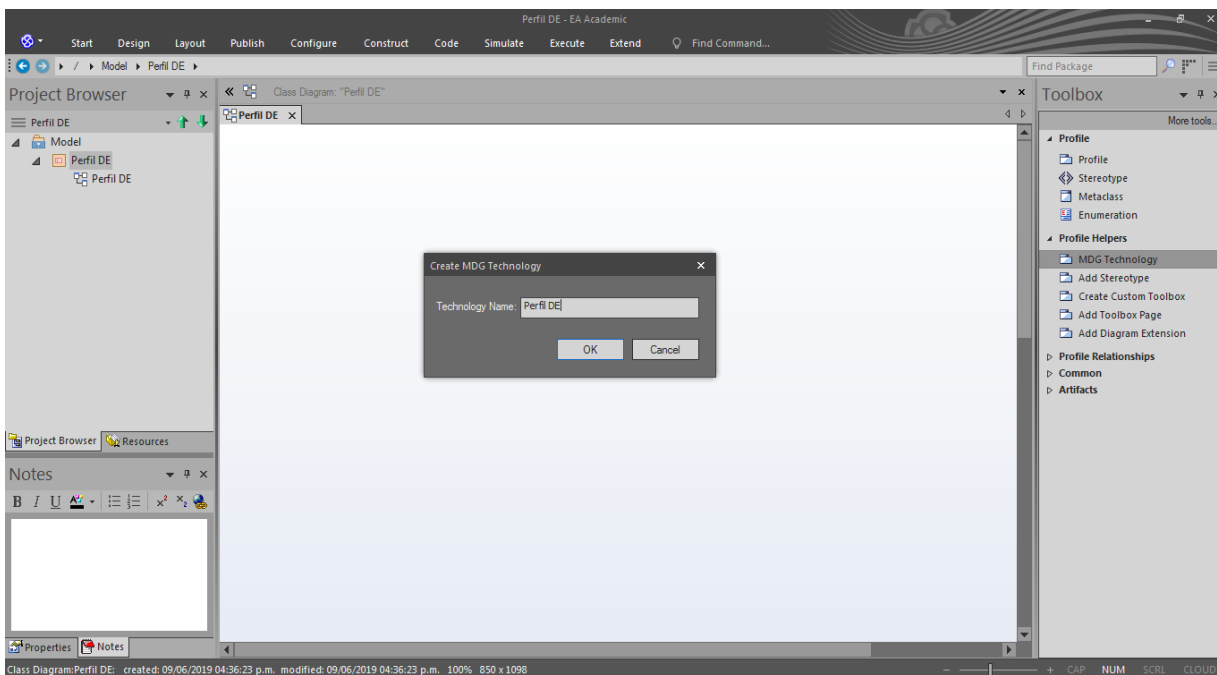


Figura 5.18 Nombre del perfil

5. El mecanismo *MDG Technology* genera tres plantillas para definir cada diagrama que formará parte del perfil UML (figura 5.20). Estos diagramas se definen en el capítulo 3. *Marco teórico*.

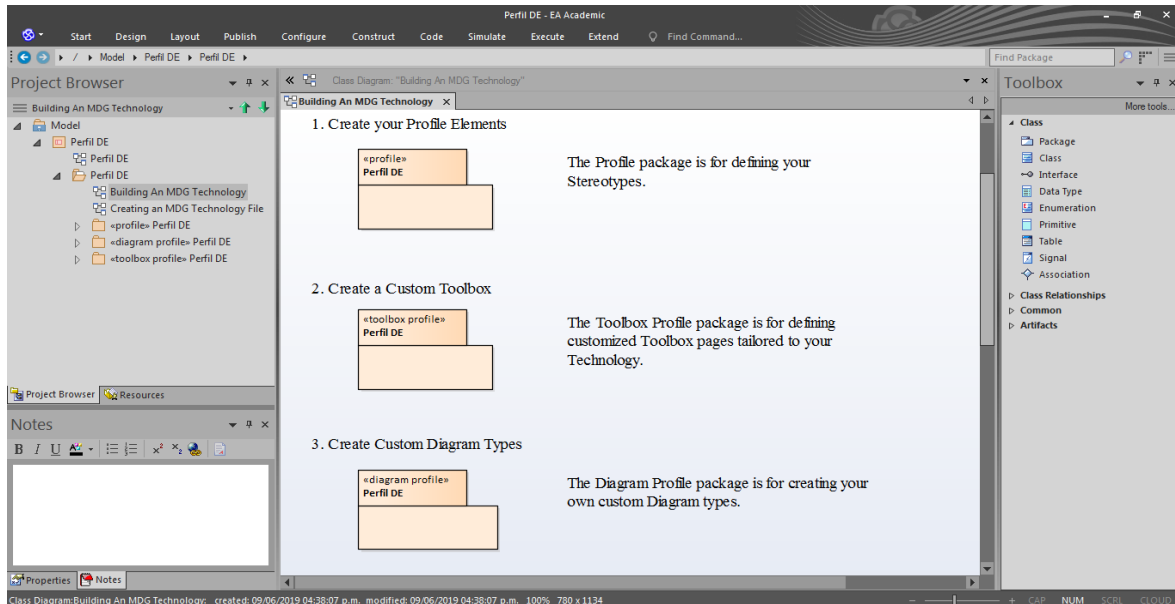


Figura 5.19 Diagramas para generar el perfil UML

5.2.2.1. Generar diagrama *Profile*

Cuando se han generado las tres plantillas, se construye el primer diagrama en el paquete `<<profile>>` que se muestra en la figura 5.21. Para construirlo se da doble clic en el paquete mencionado para abrir el área de trabajo.

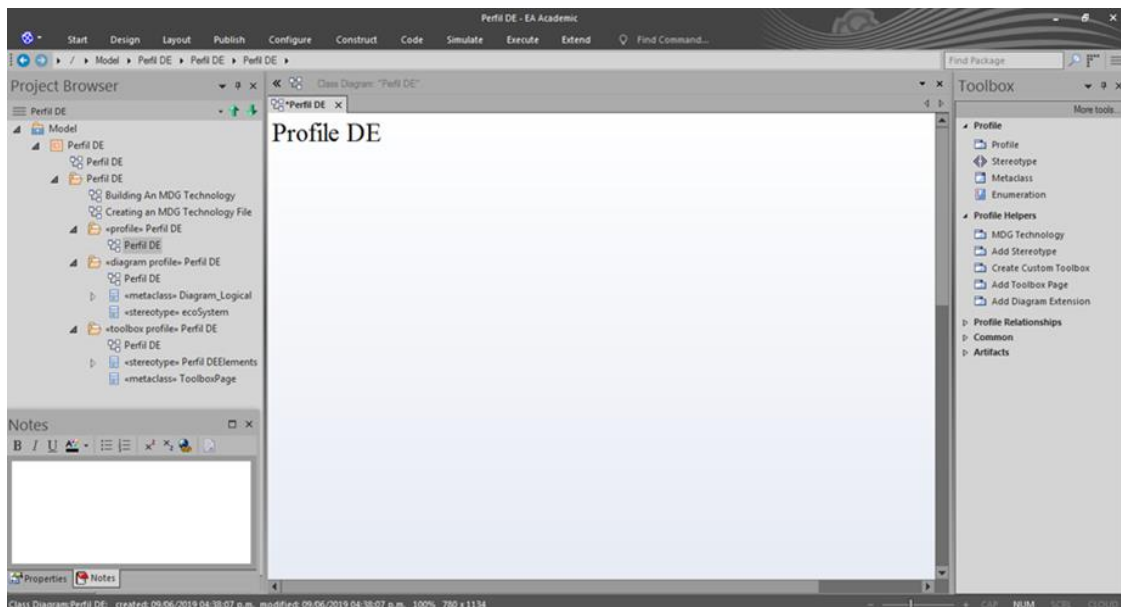


Figura 5.20 Diagrama Profile

Se agregan las meta-clases (*metaclass*), que son: *Class*, *Attribute* y *AssociationClass*, definidas en la sección *Diseño del perfil UML*.

1. Arrastrar el elemento *metaclass*, ubicado en el panel *Toolbox* para añadir una metaclase al área de trabajo. (figura 5.22)

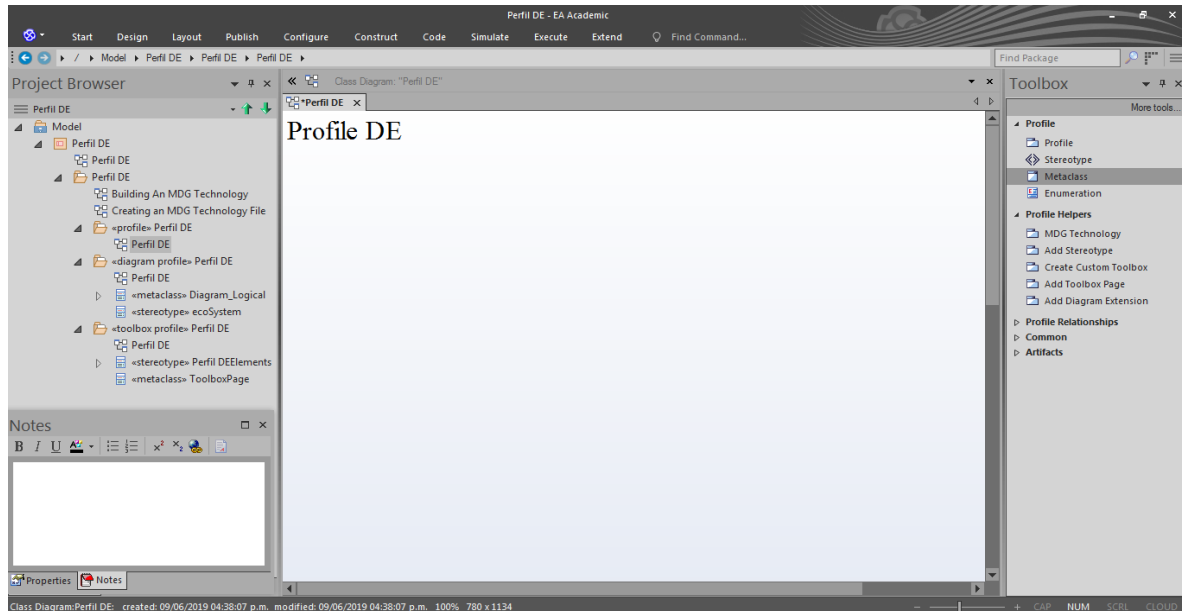


Figura 5.21 Seleccionar elemento Metaclass

2. Seleccionar las meta-clases que se utilizarán para el perfil UML, en la ventana *Extend metaclass* en la ruta *Core elements ->Classy Attribute*. (figura 5.23)

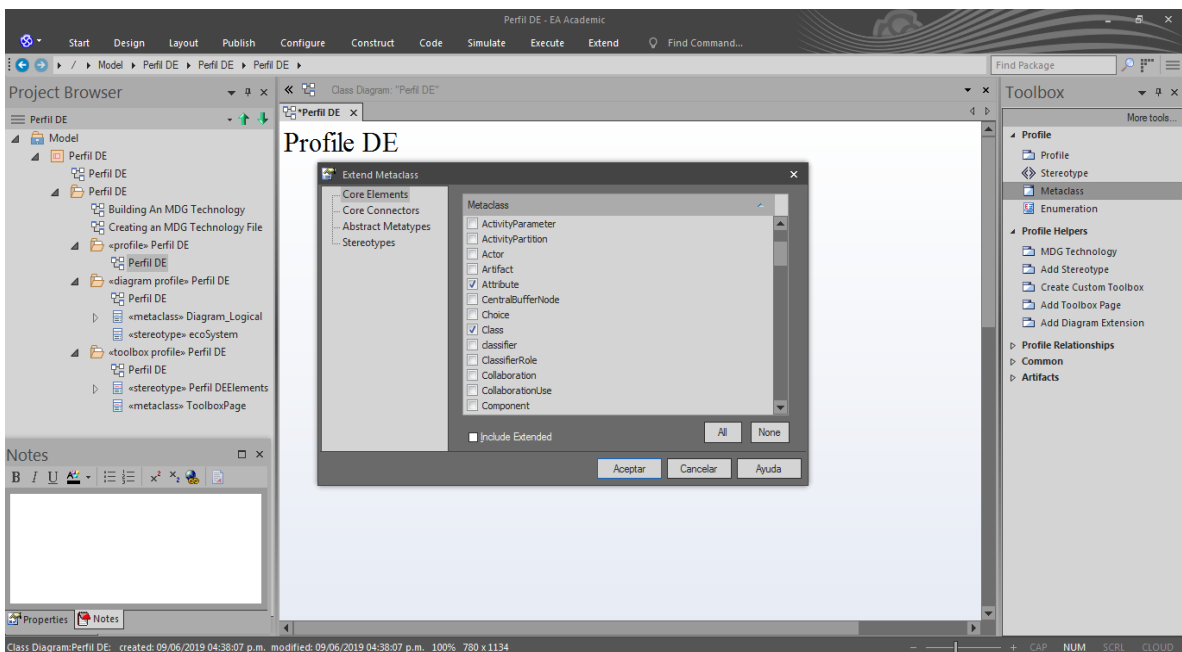


Figura 5.22 Agregar meta-clases (Class, Attribute y AssociationClass)

3. Las meta-clases seleccionadas se muestran en el área de trabajo (figura 5.24).

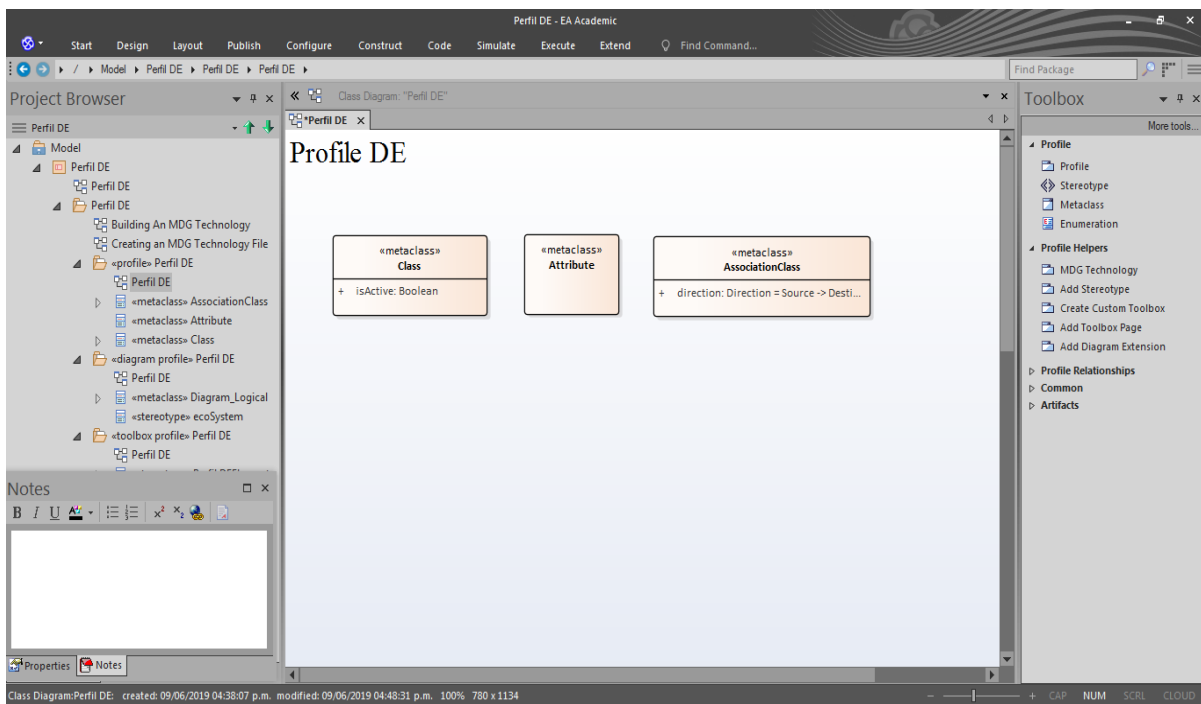


Figura 5.23 Metaclasses: Class, Attribute y AssociationClass

4. En este punto, se añaden los estereotipos: *Entidad*, *Mensaje* y *RegistroMensaje*. Arrastrar el elemento *Stereotype* al área de trabajo que se encuentra en el panel *Toolbox*. Ingresar el nombre del estereotipo *Entidad*.

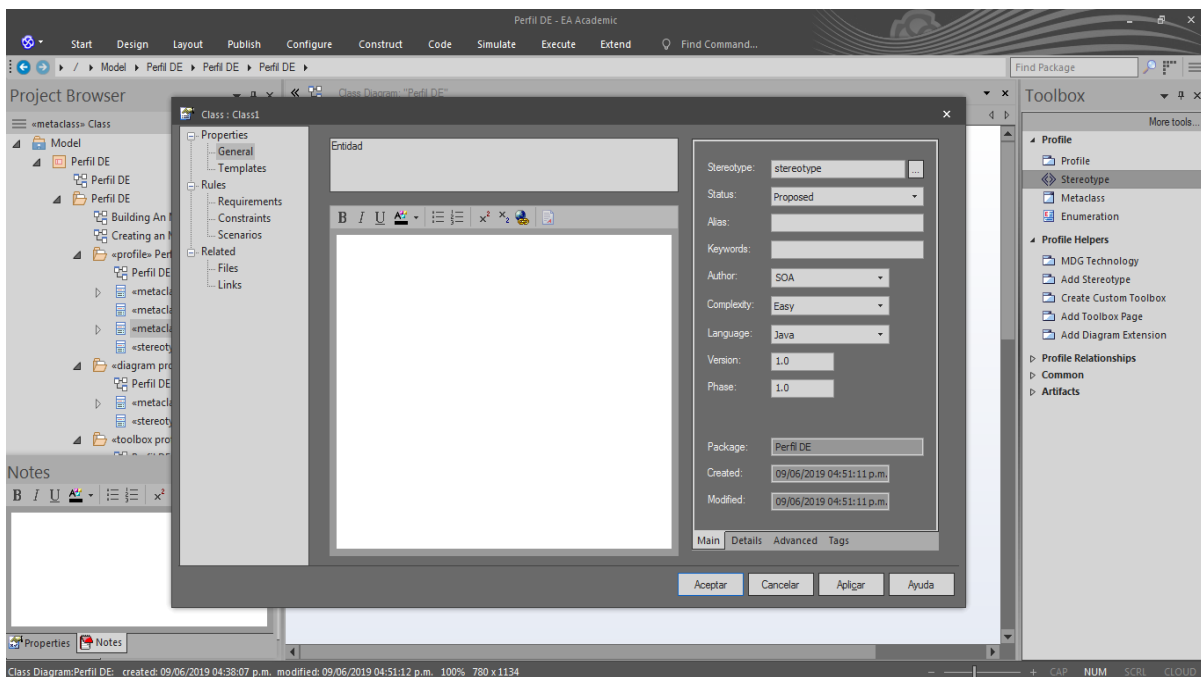


Figura 5.24 Agregar estereotipos

5. En la figura 5.26 se muestran los estereotipos añadidos. El paso 4 se repite hasta agregar los estereotipos faltantes.

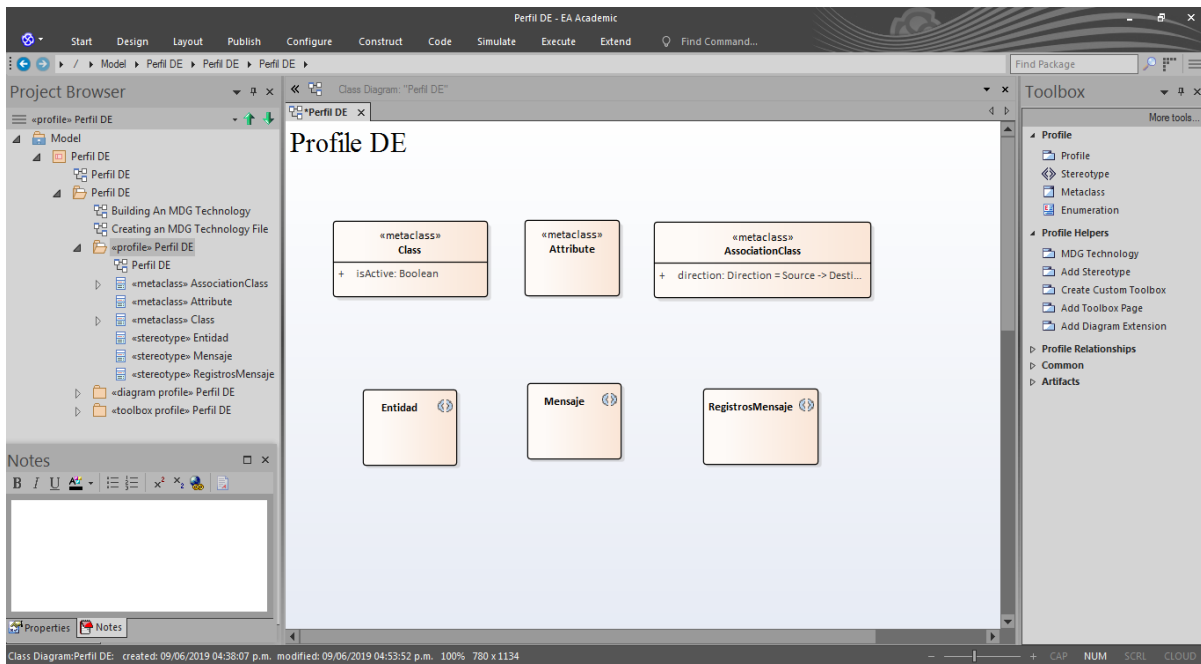


Figura 5.25 Definir estereotipos: Entidad, Mensaje y RegistroMensaje

6. Se agrega la relación entre las meta-classes y los estereotipos. Se selecciona la relación *Extension* de la sección *Profile Helpers*, y arrastrar la conexión desde el elemento Estereotipo (*Entidad*) hasta el elemento Metaclass (*Class*) que se extenderá.

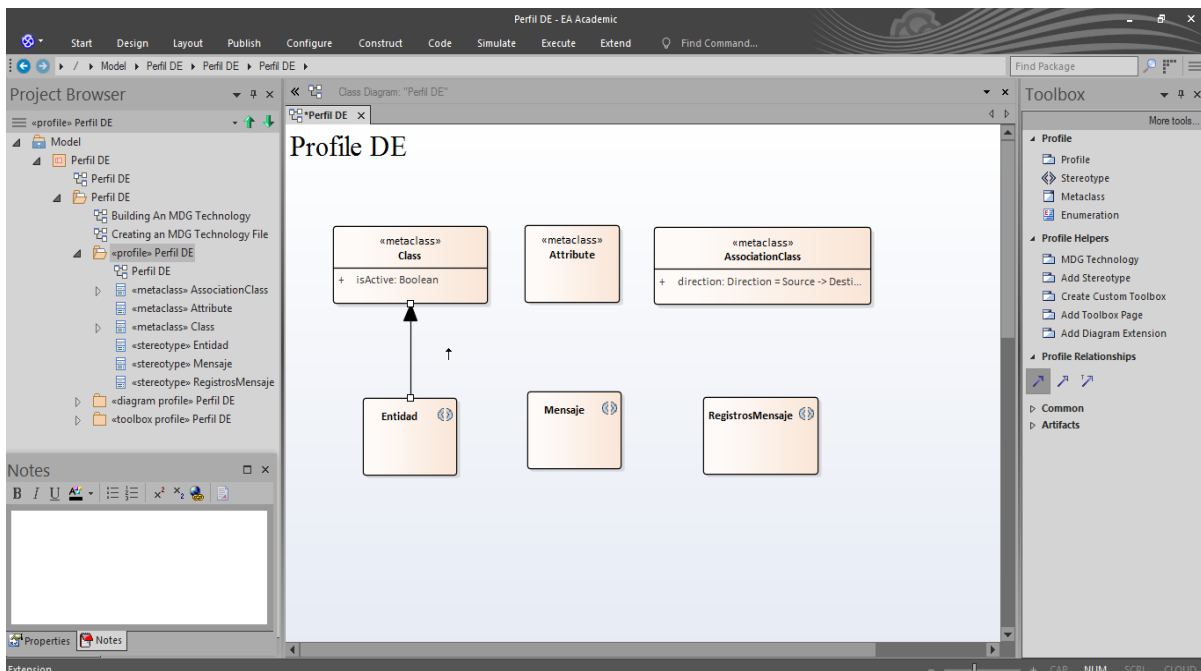


Figura 5.26 Agregar relación extensión entre estereotipos y metaclasses

7. Se agrega la relación del estereotipo Mensaje a la metaclass Attribute y del estereotipo RegistroMensaje a la meta-clase AssociationClass, repitiendo el paso 6. (Figura 5.28)

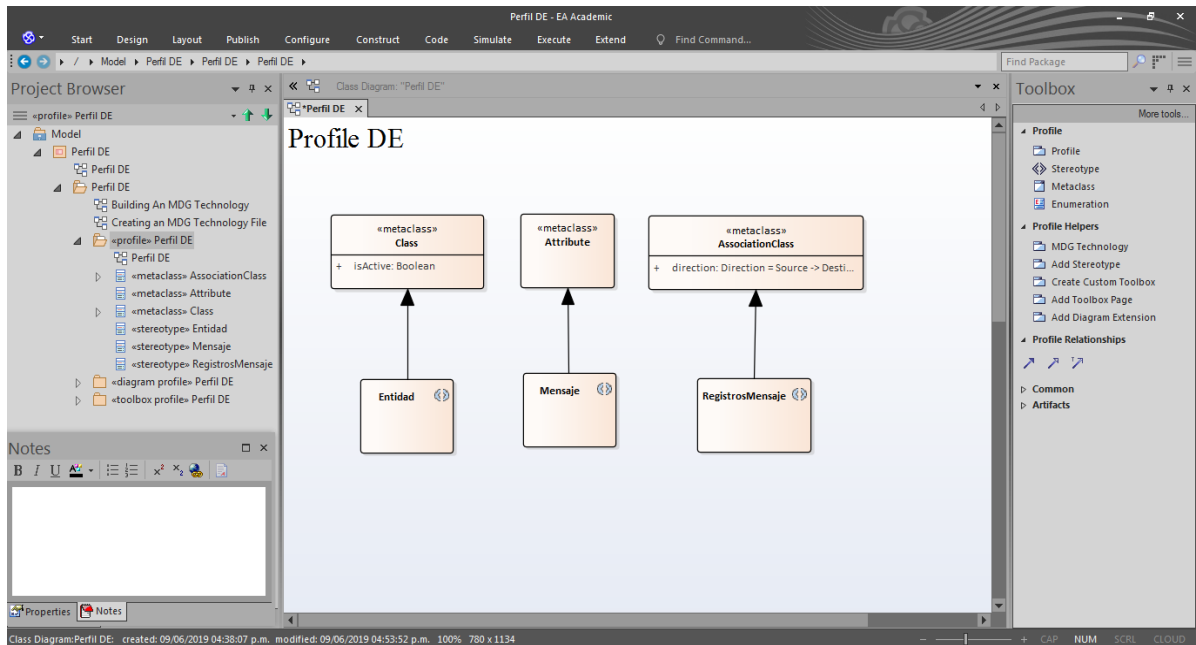


Figura 5.27 Relación Extensión

8. Se agregaron los valores etiquetados para cada uno de los estereotipos. Dando clic derecho en el estereotipo *Entidad*, en la ruta *Properties Page -> Tagged Values* (figura 5.29)

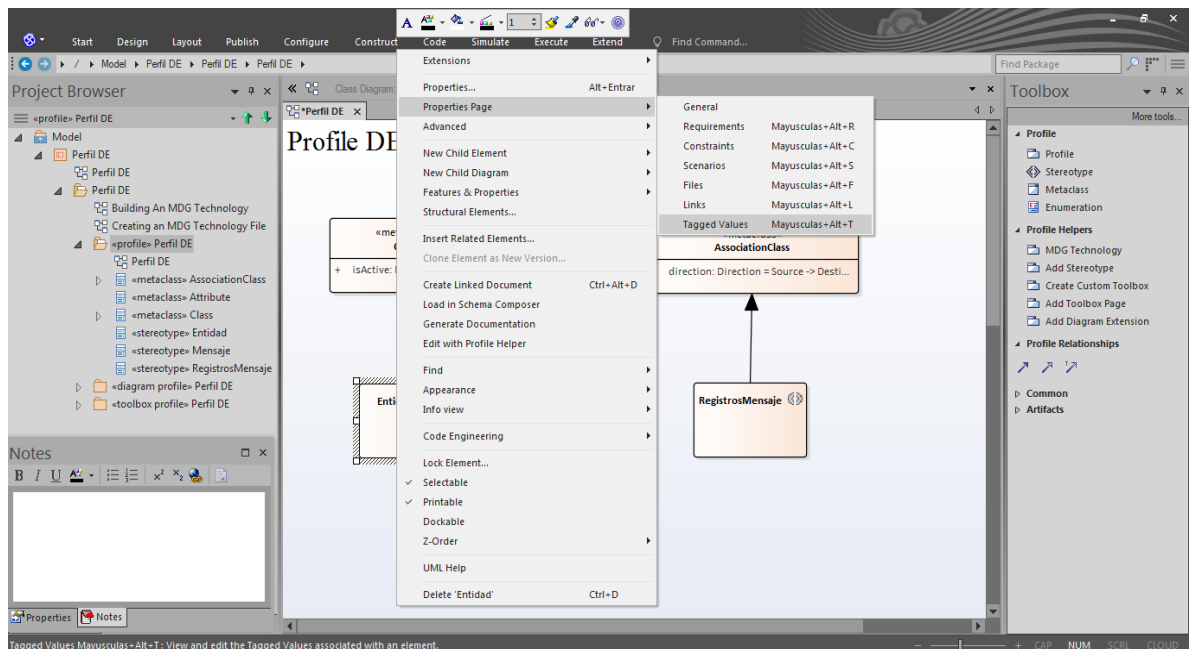


Figura 5.28 Valores etiquetados

9. En la ventana se selecciona *Attributes* para agregar los valores etiquetados. (ver figura 5.30)

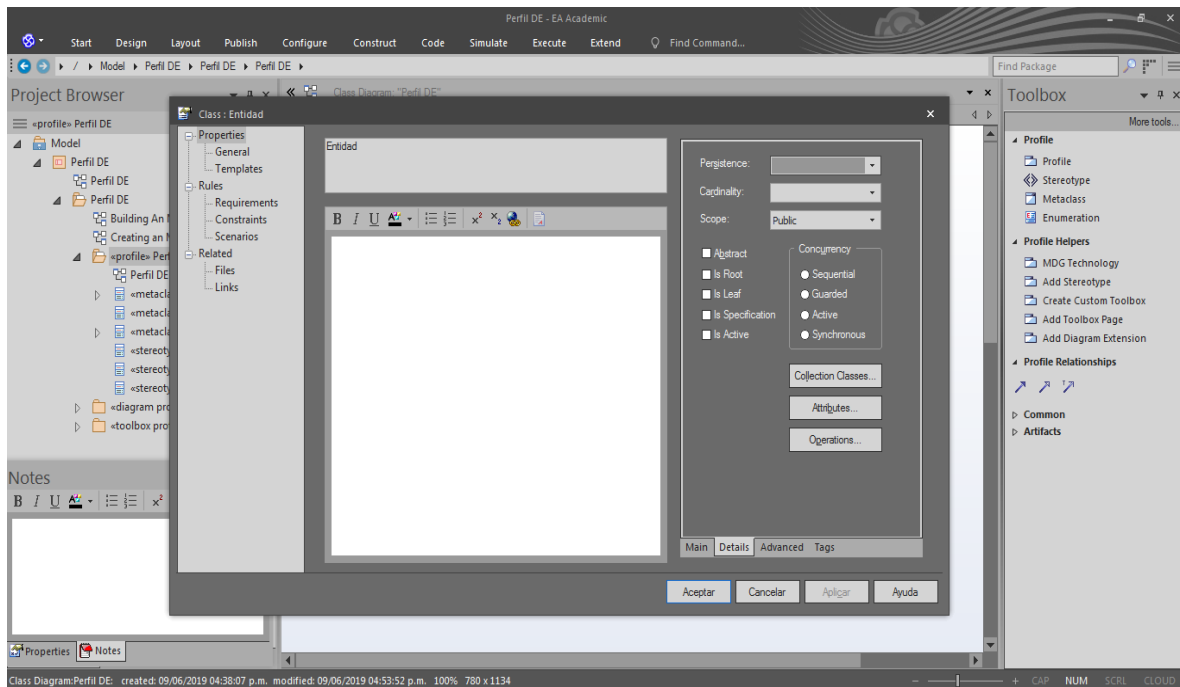


Figura 5.29 Seleccionar Attributes

10. Se ingresaron los valores etiquetados del estereotipo Entidad: *id_entidad* y *tipo*. (ver figura 5.31)

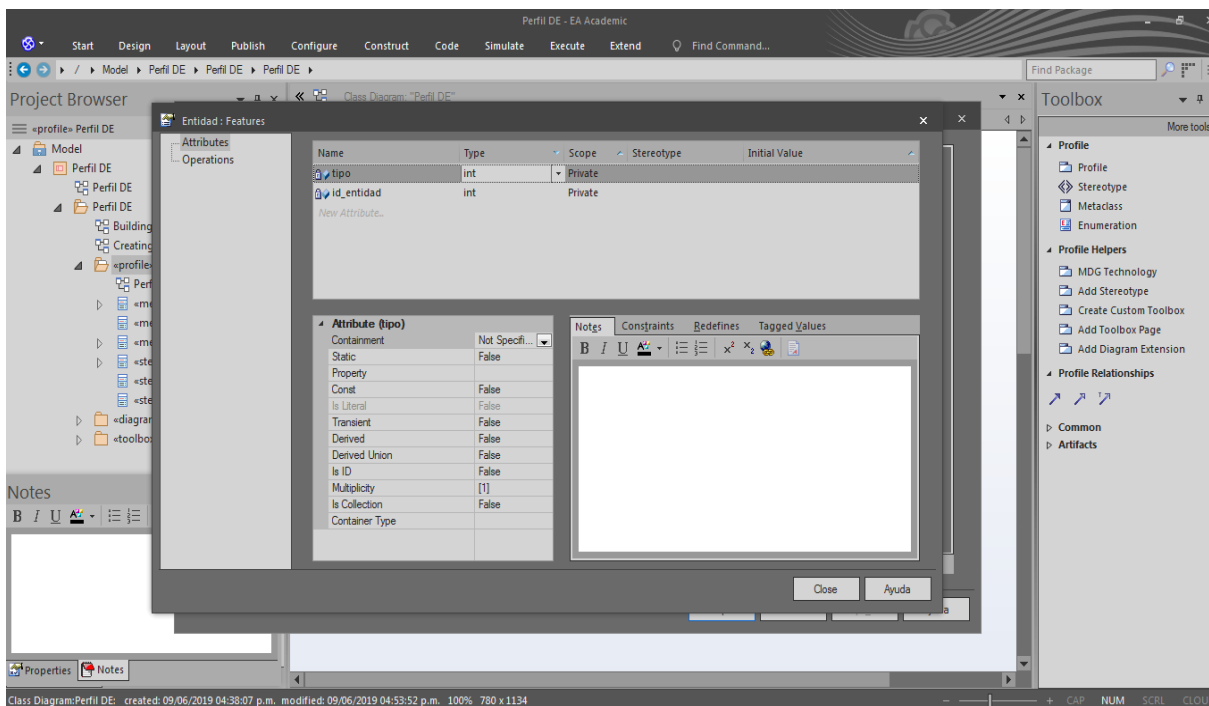


Figura 5.30 Agregar valores etiquetados

11. Se repiten los pasos del 8 al 10 para agregar los valores etiquetados de los estereotipos *Mensaje* y *RegistroMensaje*. (Figura 5.32)

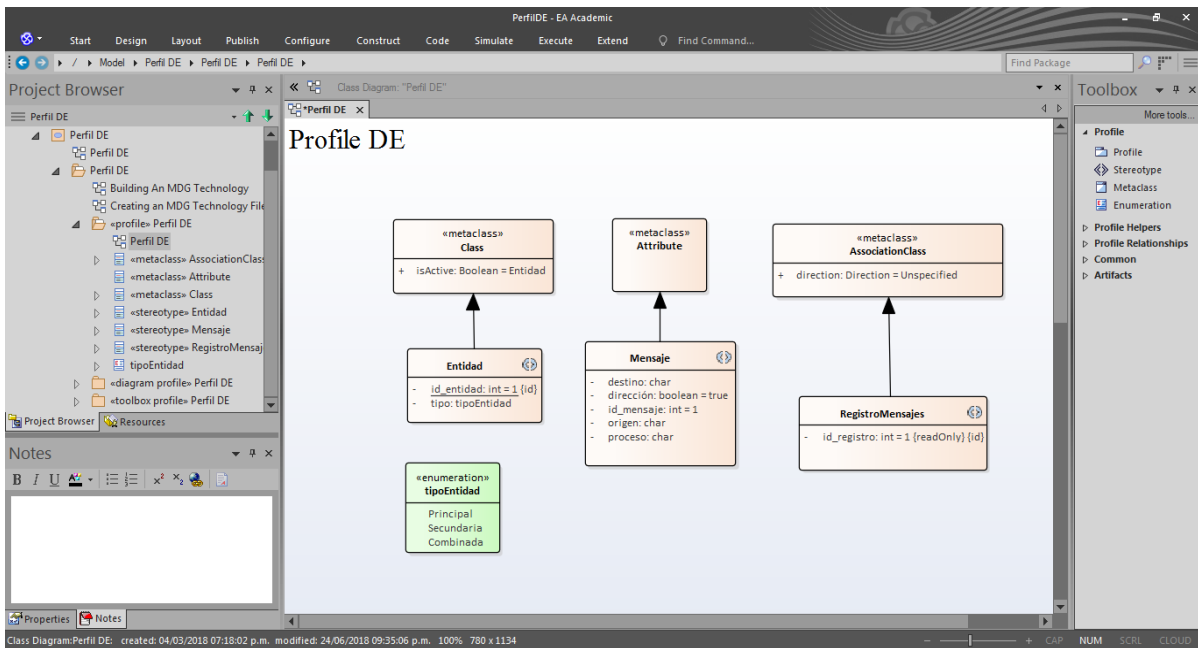


Figura 5.31 Profile DE

12. Se genera el archivo XML del diagrama Profile que se utilizará para construir el archivo MST. Se da clic derecho en el área de trabajo, se selecciona *Advanced* -> *Save as Profile*. (Figura 5.33)

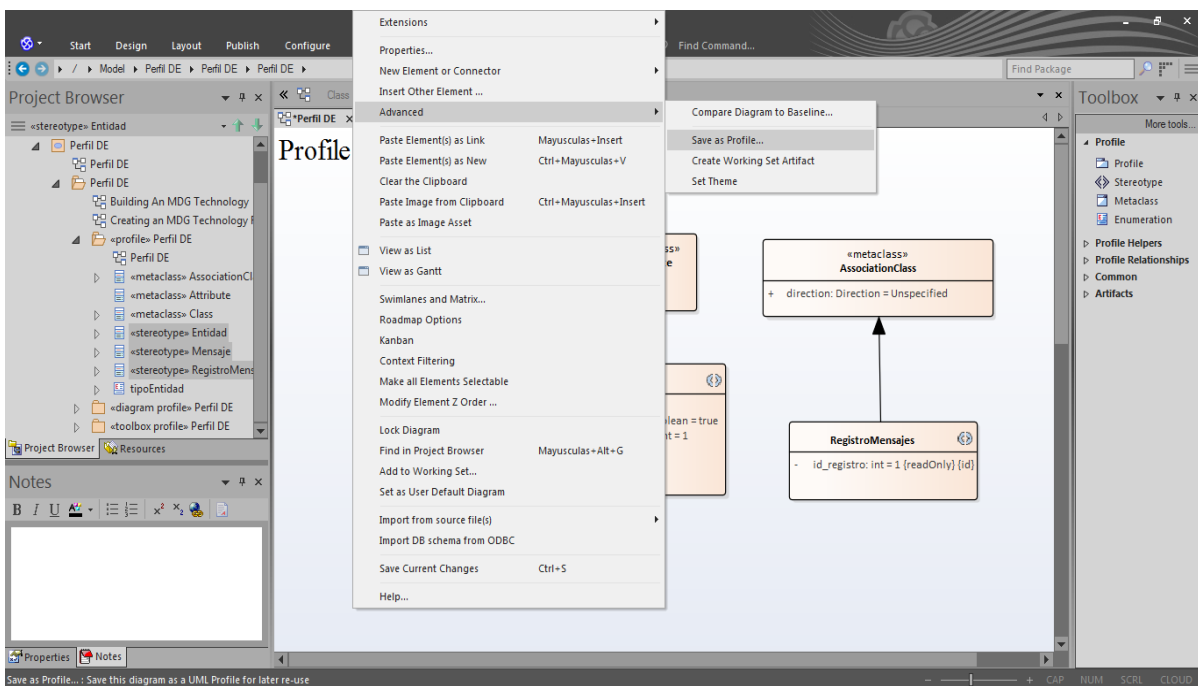


Figura 5.32 Generar archivo XML para el diagrama Profile

13. Se nombra al archivo XML como *Profile* y se da clic en *Guardar*. (Figura 5.34)

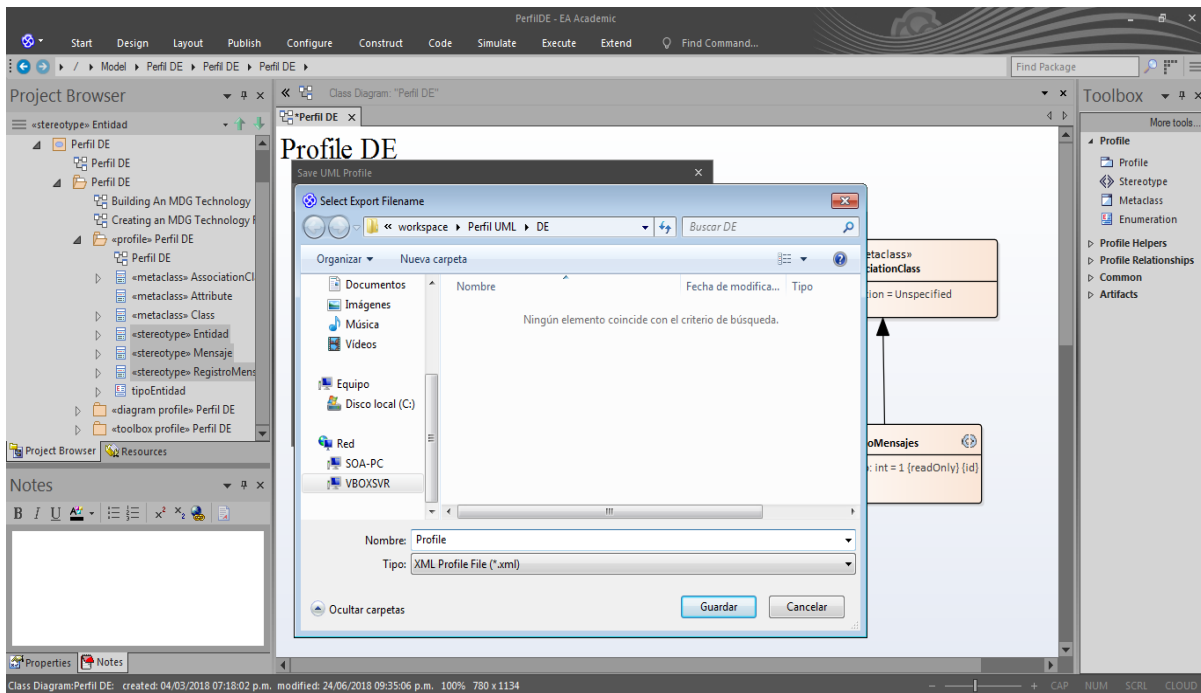


Figura 5.33 Guardar archivo Profile

14. En la figura 5.35 se muestra el archivo *Profile.xml* que se ha generado.

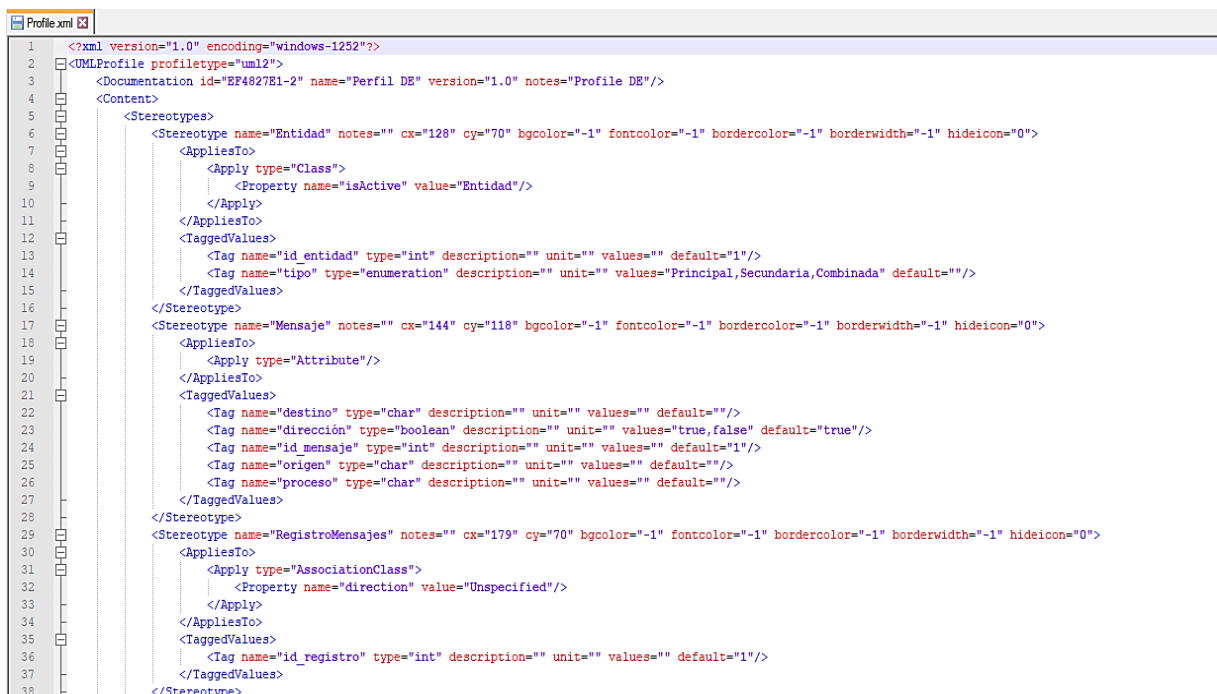


Figura 5.34 Archivo XML del diagrama Profile

5.2.2.2. Generar diagrama *Toolbox*

En esta sección se presentarán los pasos que se emplearon para construir el segundo diagrama, que es la caja de herramientas del perfil UML.

1. Se da doble clic en el paquete `<<Toolbox>>` para abrir el área de trabajo y construir el diagrama. (Ver figura 5.36)

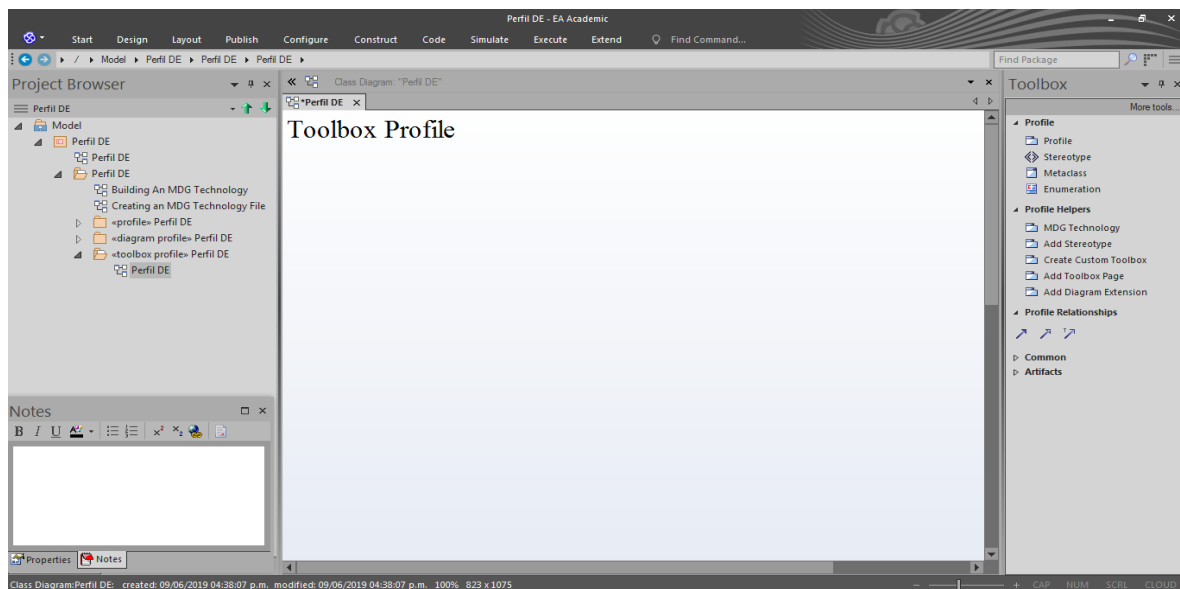


Figura 5.35 Diagrama *Toolbox*

2. Se selecciona la opción *Add Toolbox Page* que se encuentra en el panel *Toolbox*. Se agrega el nombre de la caja de herramientas en la ventana *Add ToolBox Page*, como se muestra en la figura 5.37.

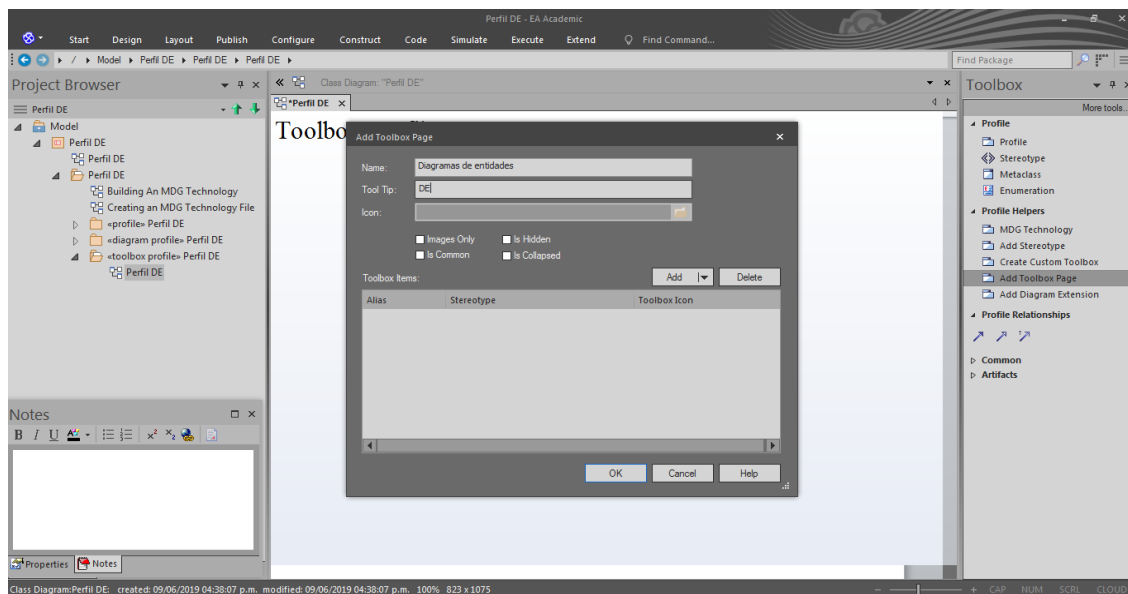


Figura 5.36 Ventana *Add ToolBox Page*

3. Se agregan los estereotipos definidos en el diagrama *Profile*, dando clic en el botón *Add*. Se muestra una ventana, donde se deben seleccionar los estereotipos: *Entidad*, *Mensaje* y *RegistroMensaje*. (Figura 5.38)

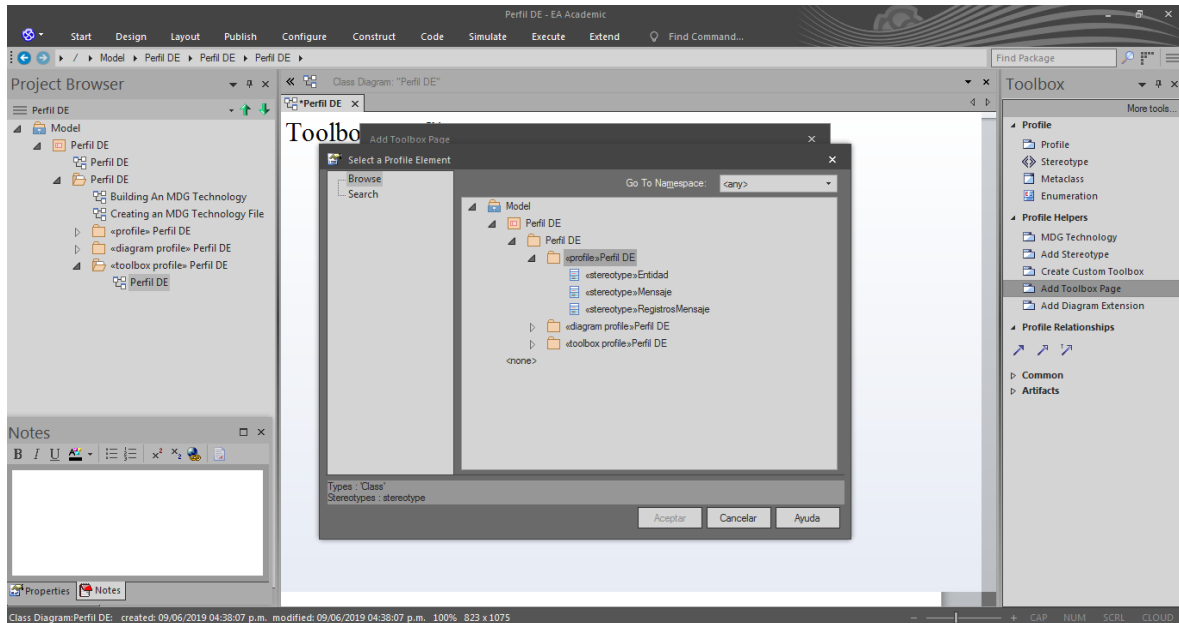


Figura 5.37 Agregar estereotipos (*Entidad*, *Mensaje* y *RegistroMensaje*)

4. Los estereotipos pasan a ser elementos en el diagrama Toolbox, y se muestran en la tabla *Toolbox items*. Una vez seleccionados, se da clic en el botón *Ok*. Ver figura 5.39

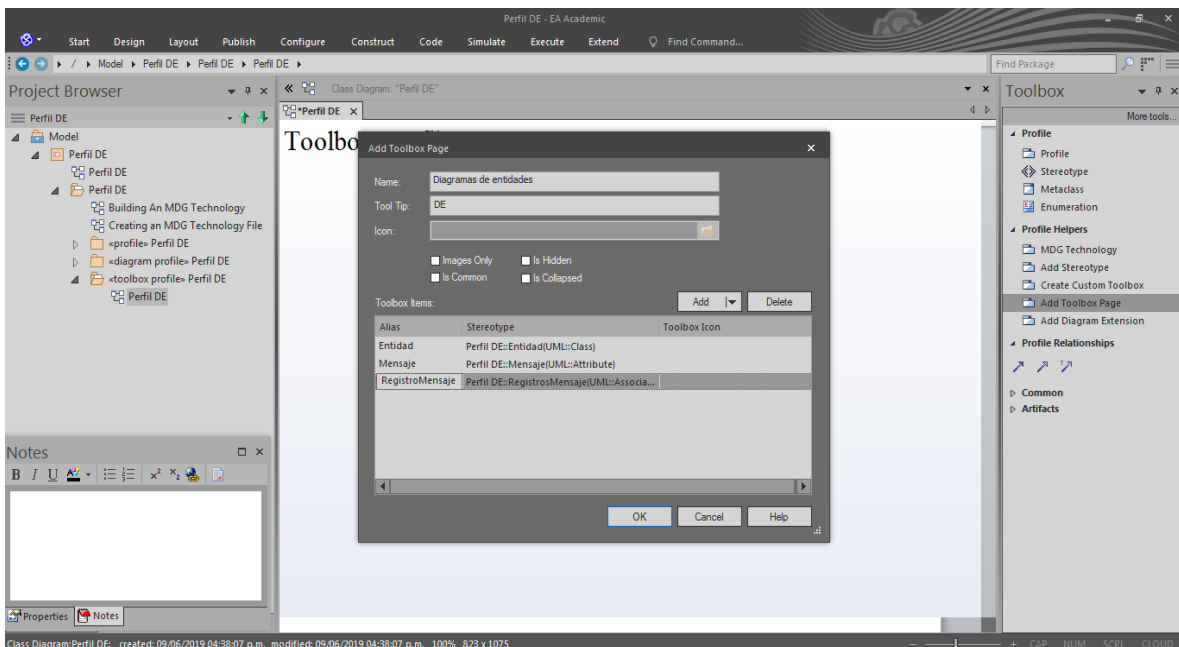


Figura 5.38 Elementos de la caja de herramientas

5. El diagrama *Toolbox*, es también un perfil donde se definen los elementos que poseerá el editor de los diagramas de entidades. (Figura 5.40)

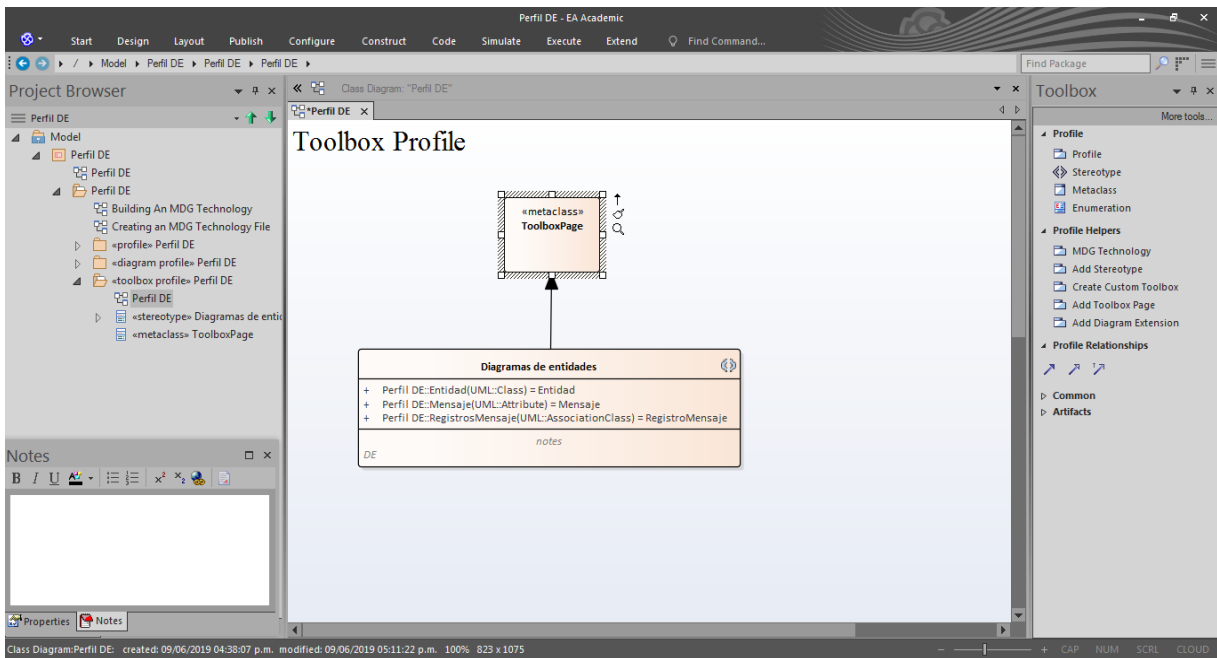


Figura 5.39 Diagrama Toolbox

6. Se guarda el diagrama *Toolbox* como un archivo XML, que se utiliza para construir el archivo MTS. El nombre de este archivo es: *Toolbox Profile*, como se muestra en la figura 5.41.

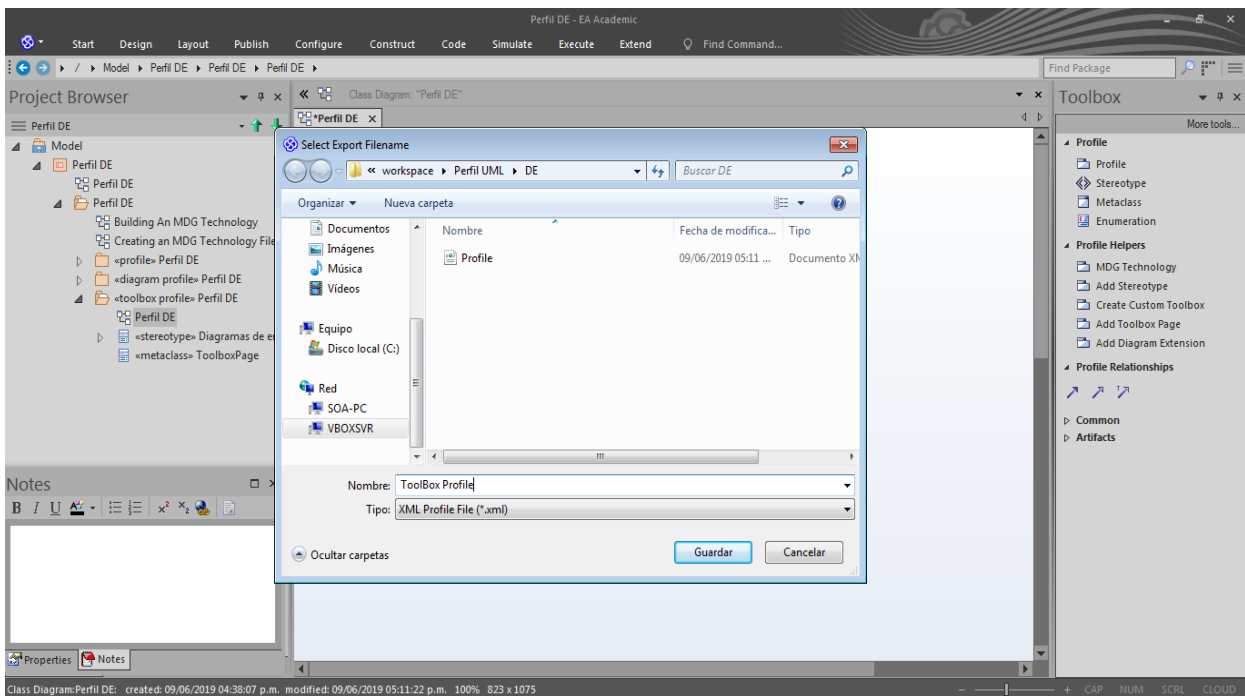


Figura 5.40 Guardar archivo XML del diagrama Toolbox

5.2.2.3. Generar diagrama *Diagram Profile*

En esta sección se muestra el proceso que se requirió para realizar el diagrama *Diagram Profile*, que reúne las meta-clases, los estereotipos y los elementos definidos tanto en el diagrama *Profile* y el diagrama *Toolbox*.

1. Dar clic en el paquete <<*Diagram Profile*>> que se encuentra en el panel *Project Brower*. (Ver figura 5.42)

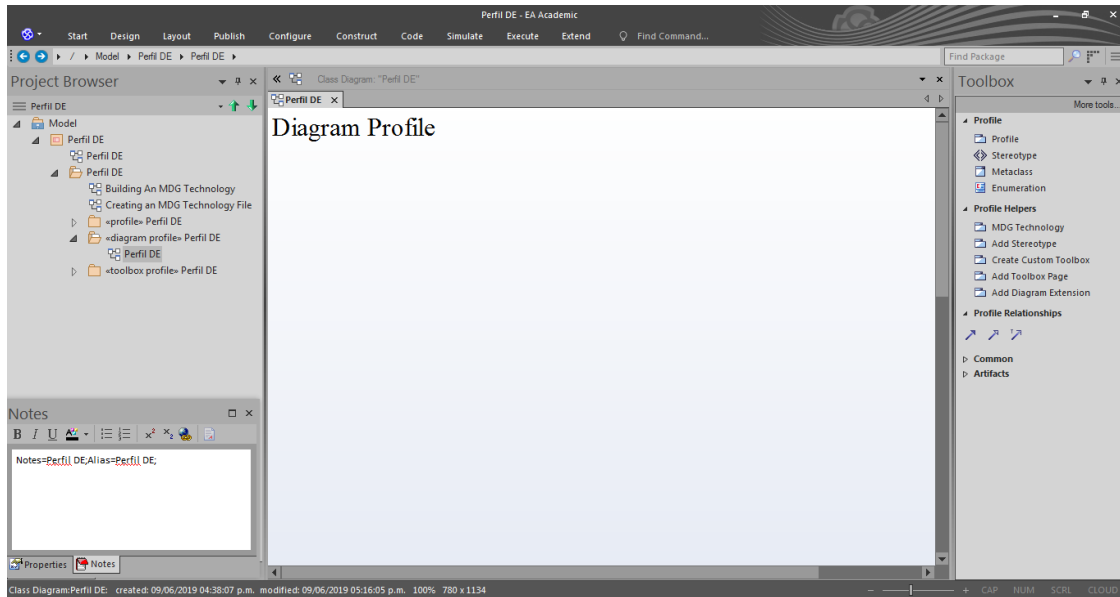


Figura 5.41 Área de trabajo para el diagrama *Diagram Profile*

2. Se arrastra el elemento *Metaclass* al área de trabajo y se selecciona la metaclasses que corresponde al diagrama a extender. (Ver figura 5.43)

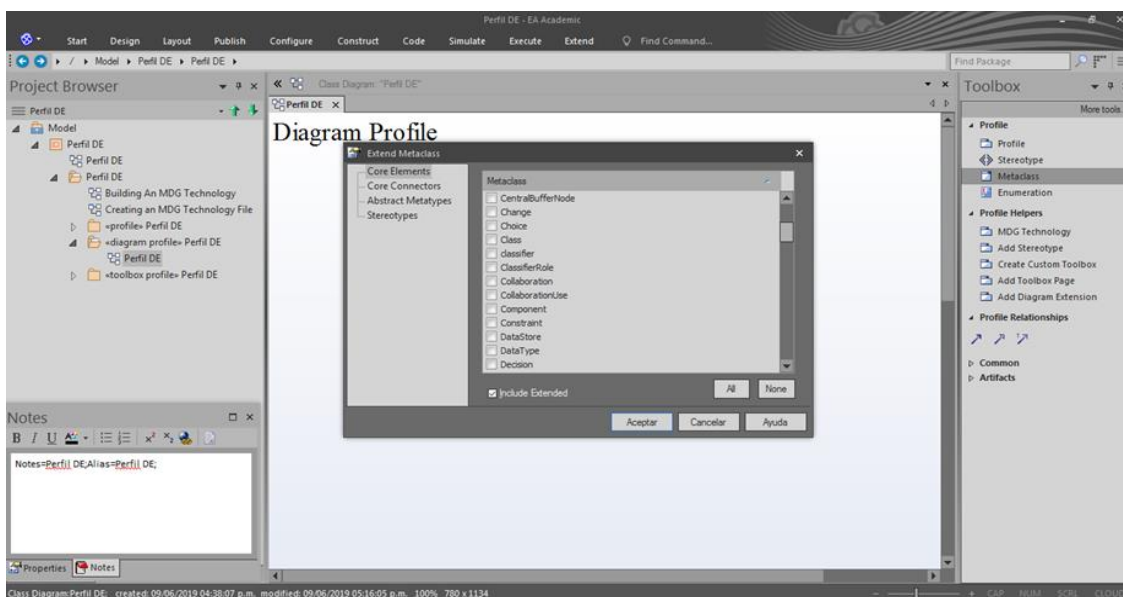


Figura 5.42 Seleccionar meta-clase

3. Se selecciona el elemento *Stereotype* y se agrega el primer diagrama definido. Además, se selecciona la relación *Extension* para unir ambos elementos agregados, como se presenta en la figura 5.45.

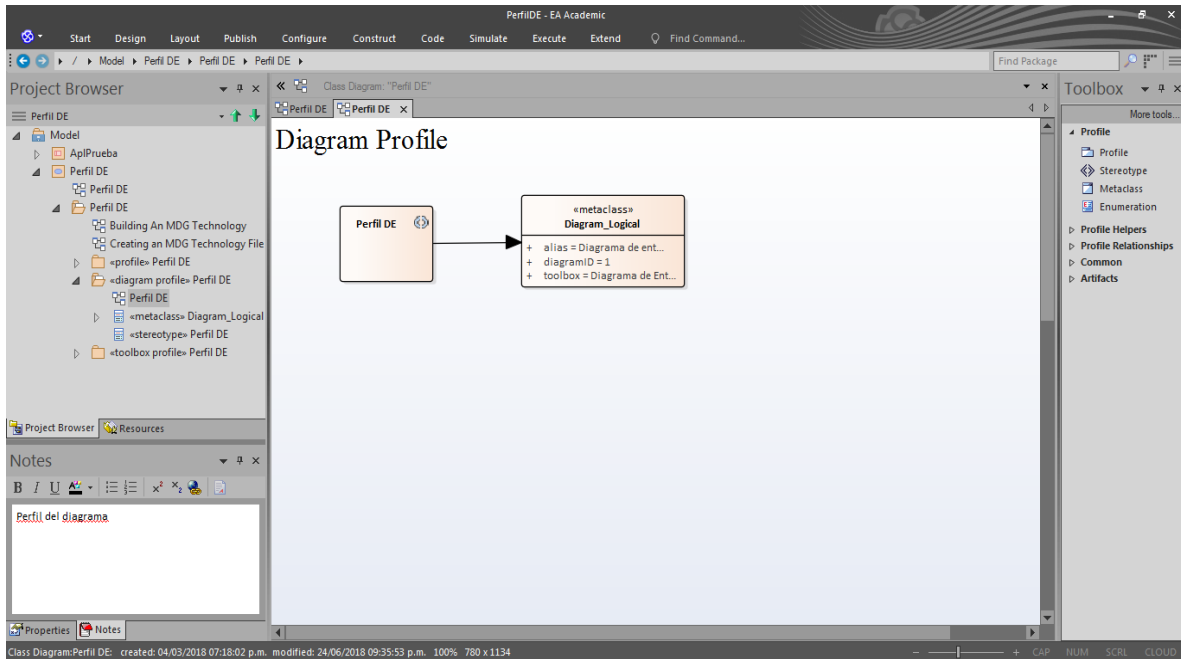


Figura 5.43 Diagram Profile

4. Se genera el archivo XML con el nombre de *Diagram Profile*, que utiliza también para generar el archivo MTS. Ver figura 5.45

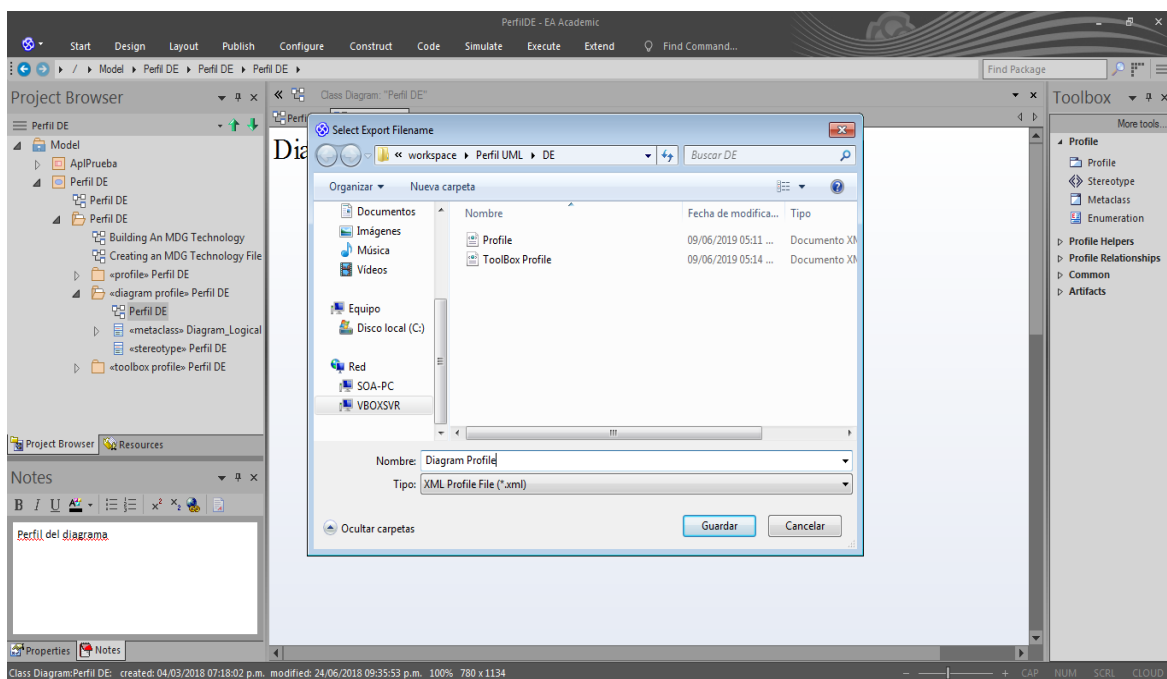


Figura 5.44 Archivo XML de Diagram Profile

5.2.2.4. Generar archivo MTS

Por último, se genera el archivo MTS que integra los tres diagramas descritos anteriormente. Este archivo contiene los elementos, meta-clases, estereotipos y valores etiquetados definidos en los archivos XML *Profile*, *Toolbox* y *Diagram Profile*.

A continuación, se describen los pasos que se llevaron a cabo para realizar el archivo MTS.

1. Se selecciona *Publish -> Generate MDG Technology*

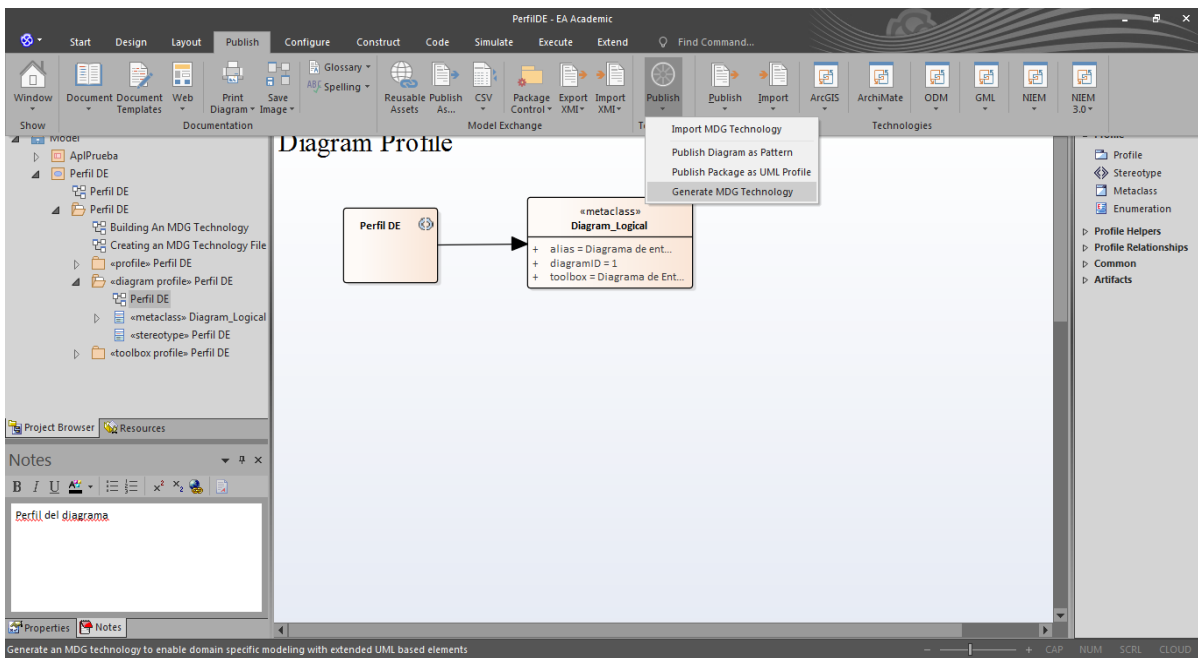


Figura 5.45 Generate MDG Technology

2. En la figura 5.47, se muestra una ventana, donde se da clic en *Siguiente*.

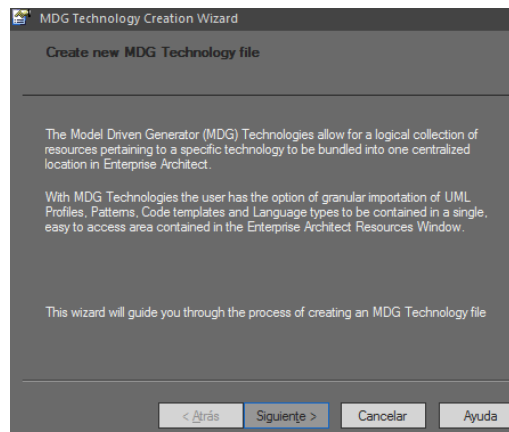


Figura 5.46 Winzard MDG Technology

3. Se selecciona la opción *Create a new MTS file* y clic en *Siguiente*. (Ver figura 5.48)

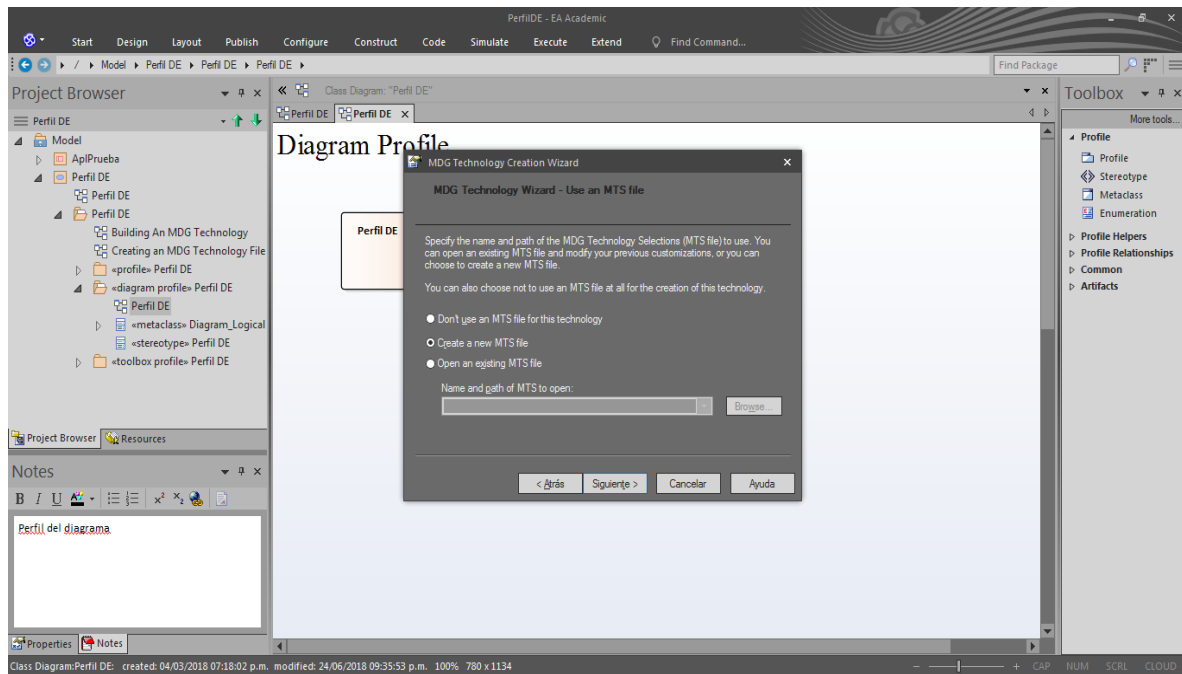


Figura 5.47 Crear archivo MST

4. Se llenan los campos Technology: *Diagrama de Entidades*, FileName: *Perfil DE*, ID: *DE*, Version: *1.0* y Notes: *Diagramas de entidades*. (Figura 5.49)

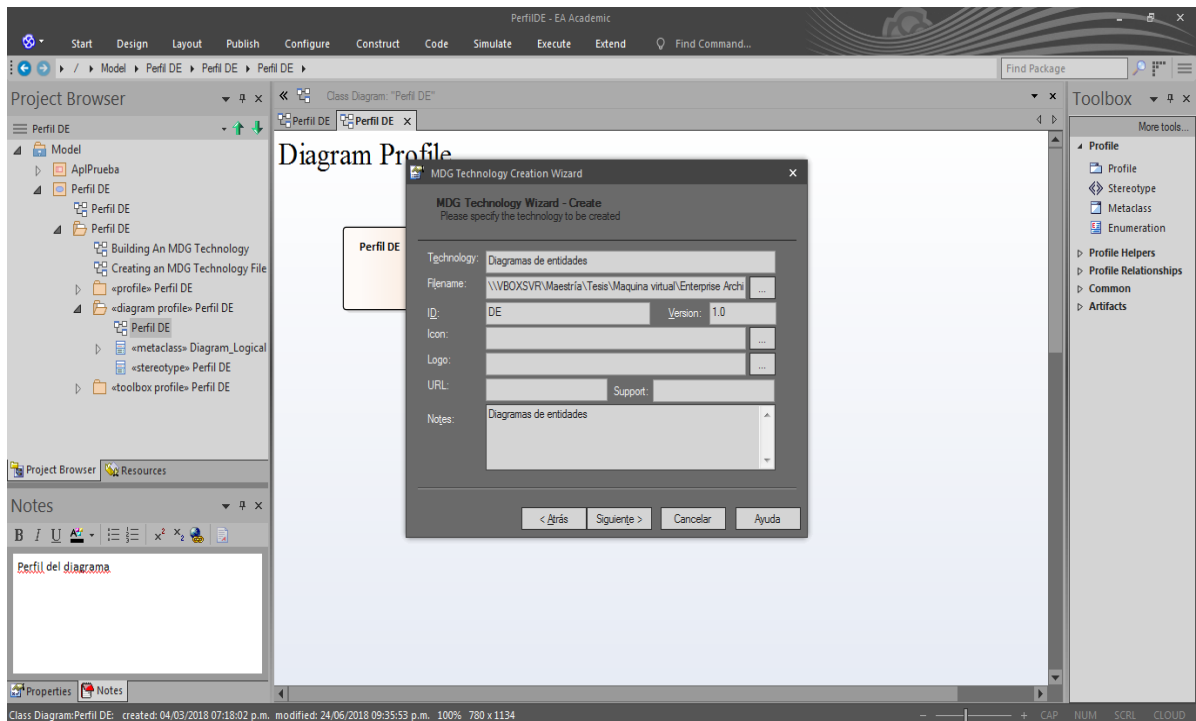


Figura 5.48 Especificaciones de la tecnología a crear

5. Se seleccionan los metamodelos *Profiles*, *Diagram Profile* y *Toolbox*, que son los archivos que se requieren para generar el archivo MTS. (Figura 5.50)

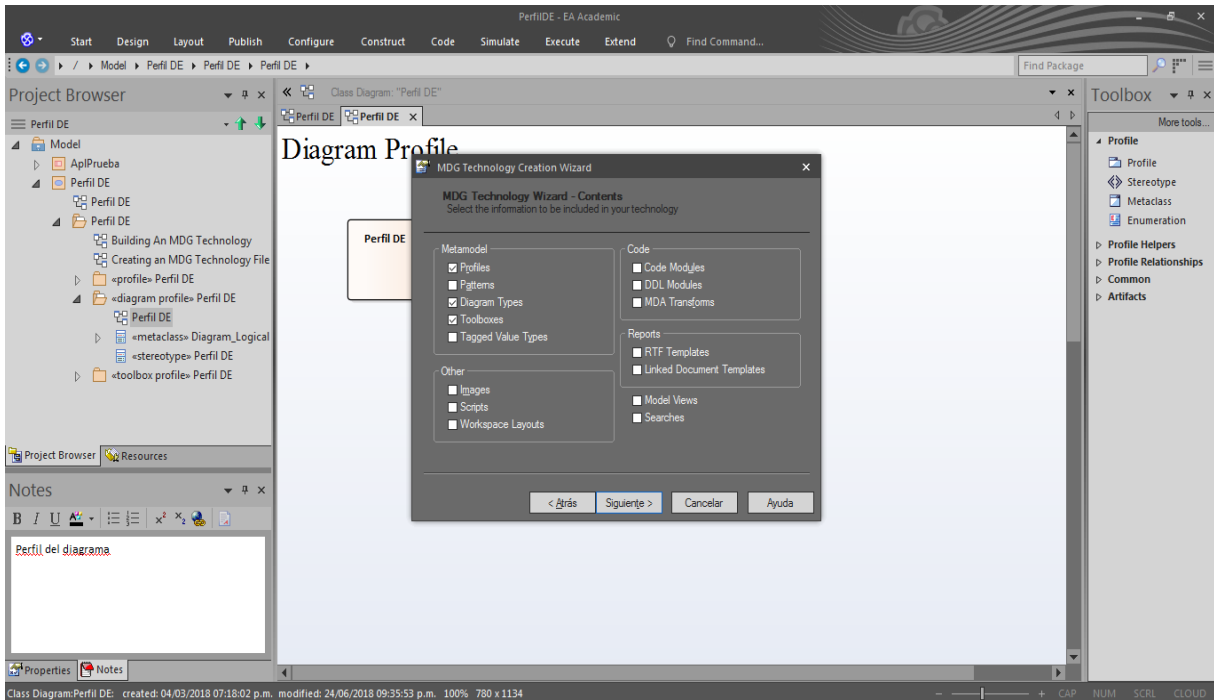


Figura 5.49 Seleccionar meta-modelos para el archivo MTS

6. Se selecciona la ruta donde se encuentran guardados los archivos XML (ver figura 5.51).

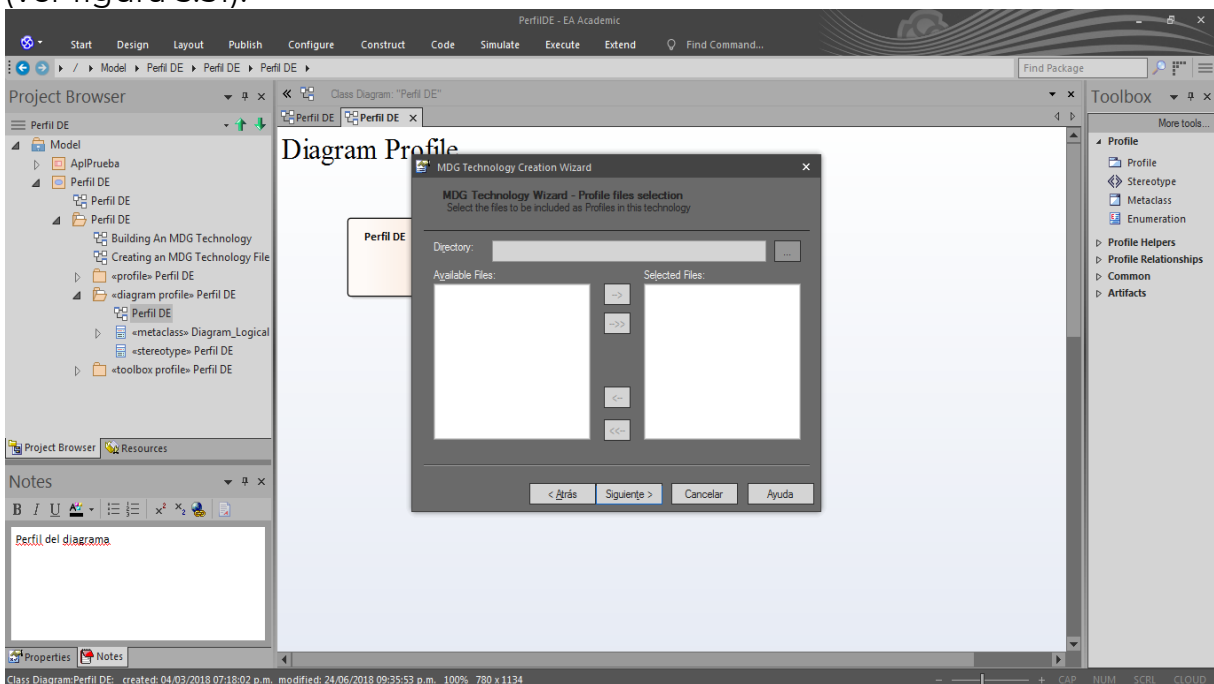


Figura 5.50 Seleccionar dirección de los archivos XML

7. Se elige el archivo *Profile* y se da clic en *Siguiente*. (Ver figura 5.52)

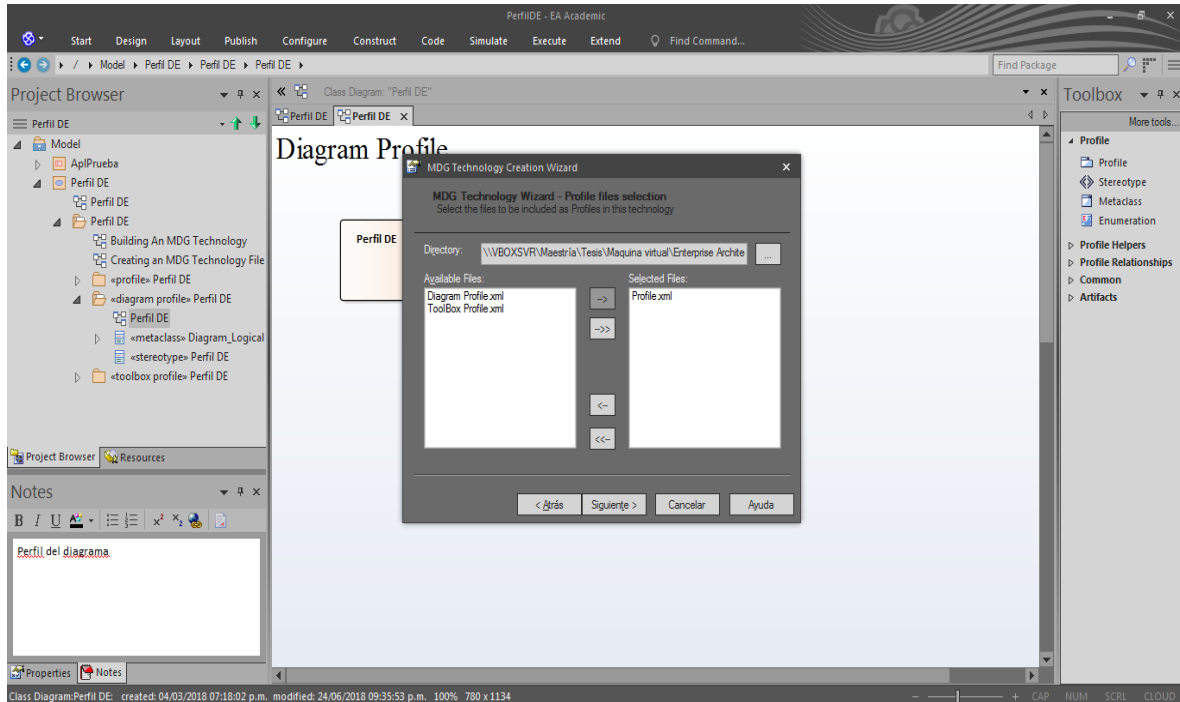


Figura 5.51 Elegir archivo Profle

8. Se elige el archivo *Diagram Profile* y se da clic en *Siguiente*. (Figura 5.53)

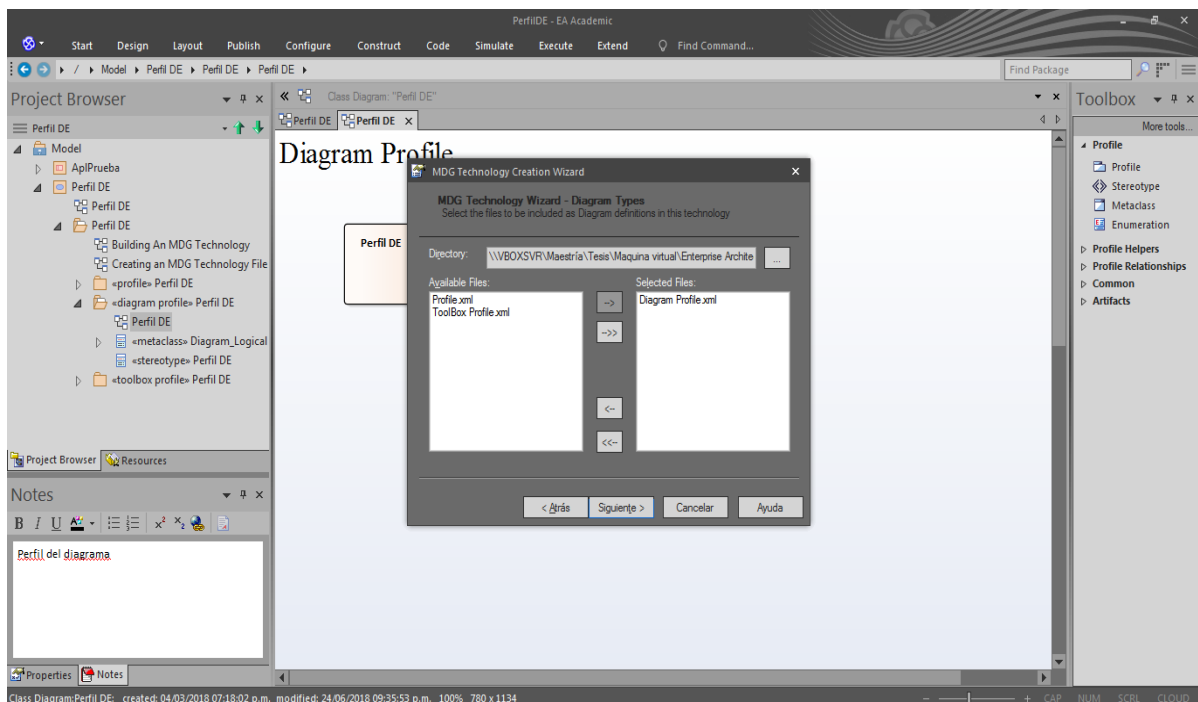


Figura 5.52 Elegir archivo Diagram Profile

9. Se elige el archivo *Toolbox* se da clic en *Siguiente*. (Ver figura 5.54)

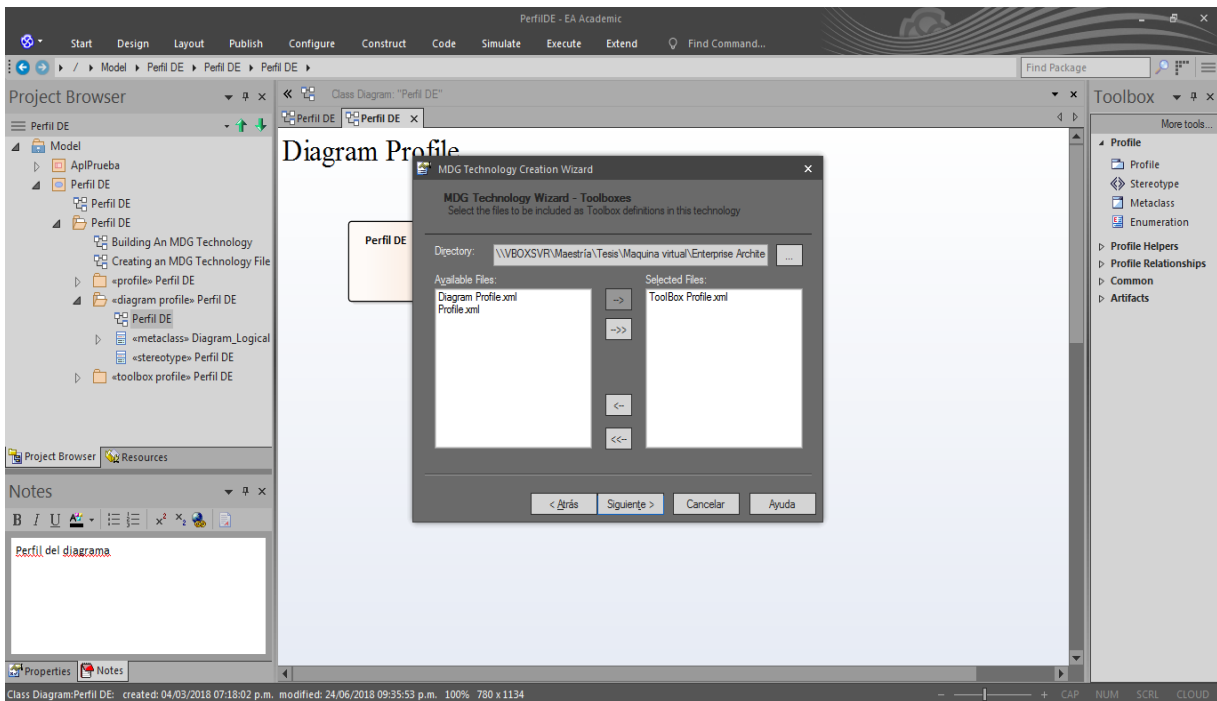


Figura 5.53 Elegir archivo Toolbox

10. Antes de generar el archivo MTS se muestra un resumen de los archivos seleccionados. Se da clic en *Finalizar*. (Ver figura 5.55)

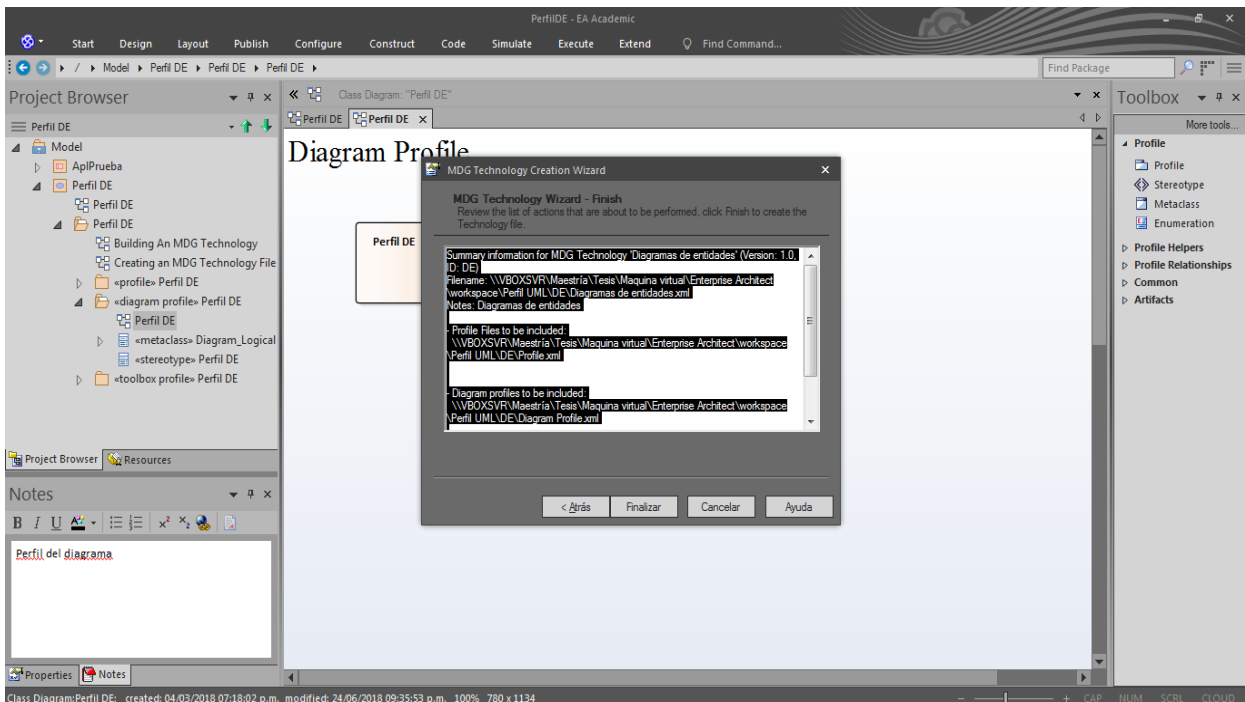


Figura 5.54 Resumen de los archivos seleccionados

11. Por último, se muestra un mensaje que el archivo fue creado con éxito. (Ver figura 5.56)

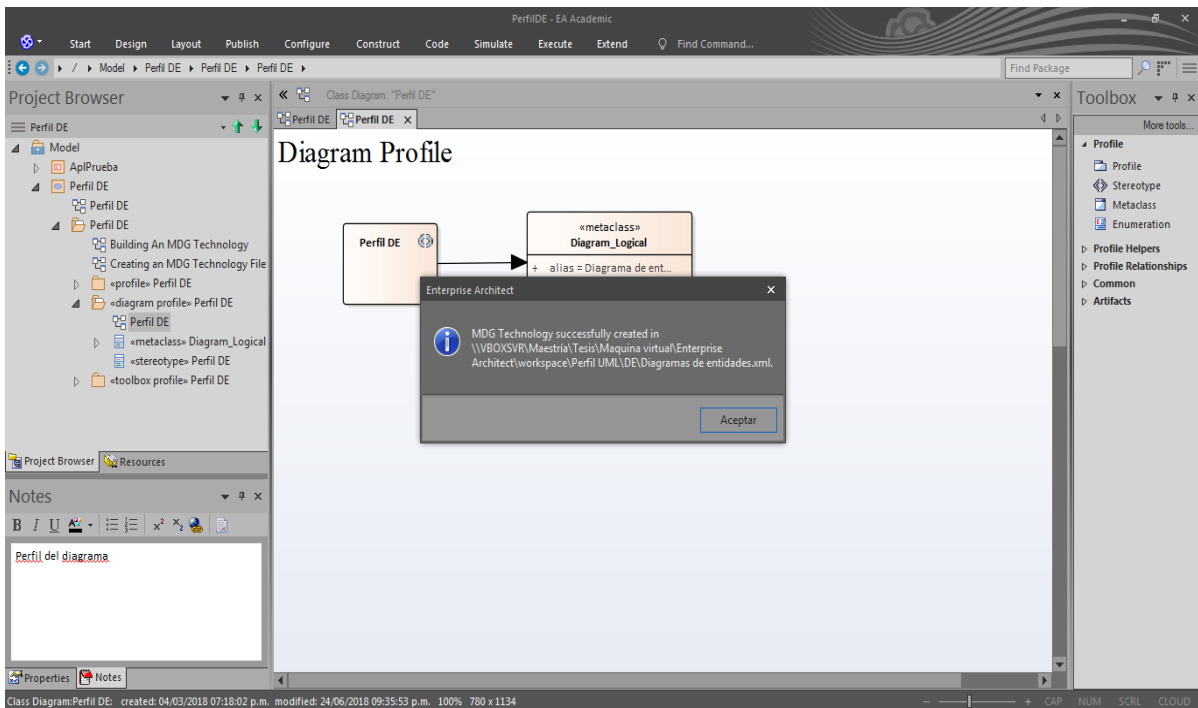


Figura 5.55 Mensaje del archivo MTS generado exitosamente

12. El archivo MTS se genera en un formato XML, como se muestra en la figura 5.57.

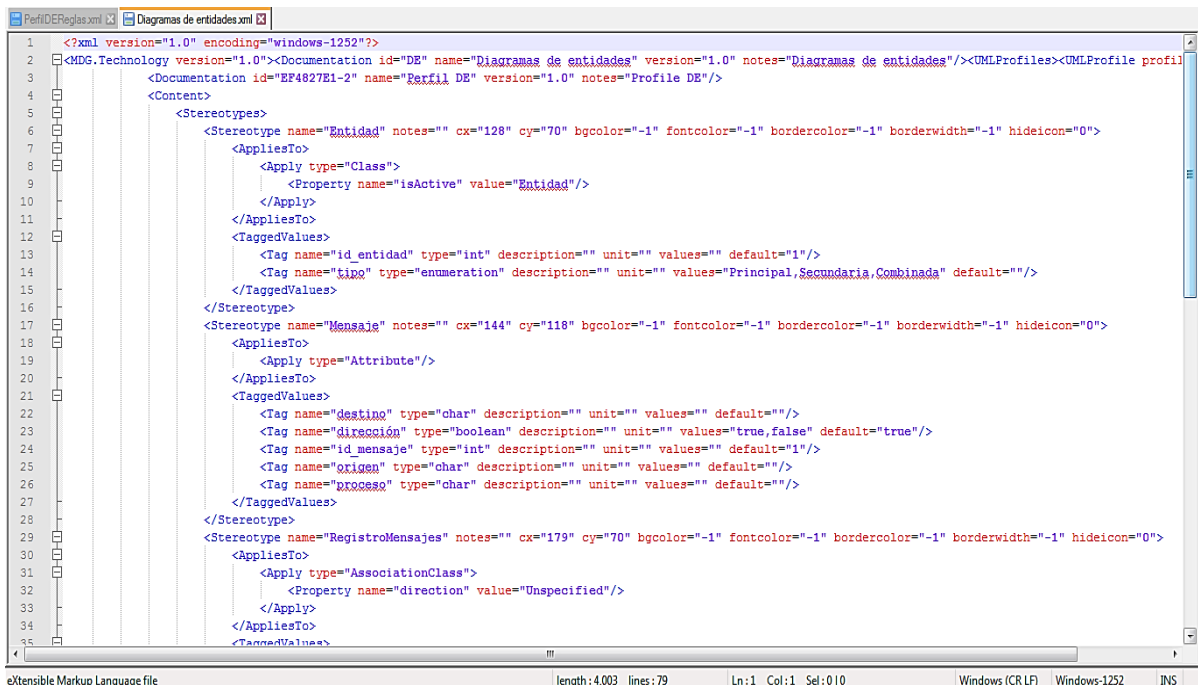


Figura 5.56 Archivo MTS

5.3. Instanciar el editor en Enterprise Architect

Para crear el editor de los diagramas de entidades, se debe instanciar el archivo MTS en *Enterprise Architect*.

1. En la pestaña *Publish*, se da clic en *Publish -> Import MDG Technology*, para importar el archivo MTS en *Enterprise Architect* (ver figura 5.58).

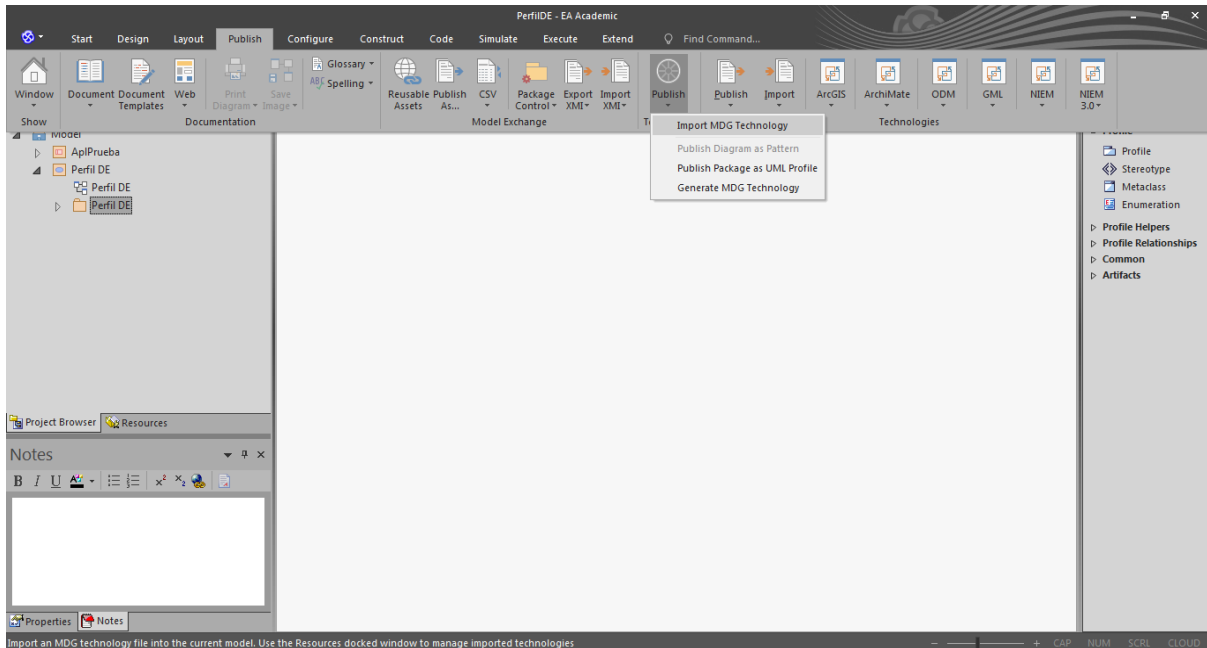


Figura 5.57 Instanciar archivo MTS

2. Se selecciona el archivo Diagrama de entidades. (ver figura 5.59)

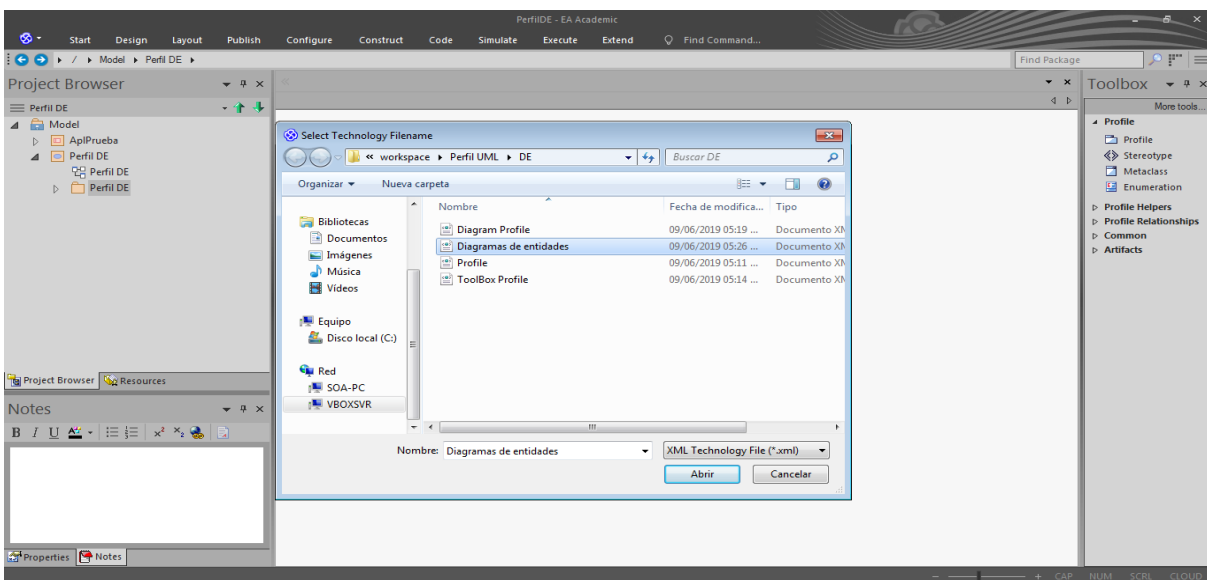


Figura 5.58 Elegir archivo

3. En la figura 5.60 se muestra una ventana con las características del editor que se creará.

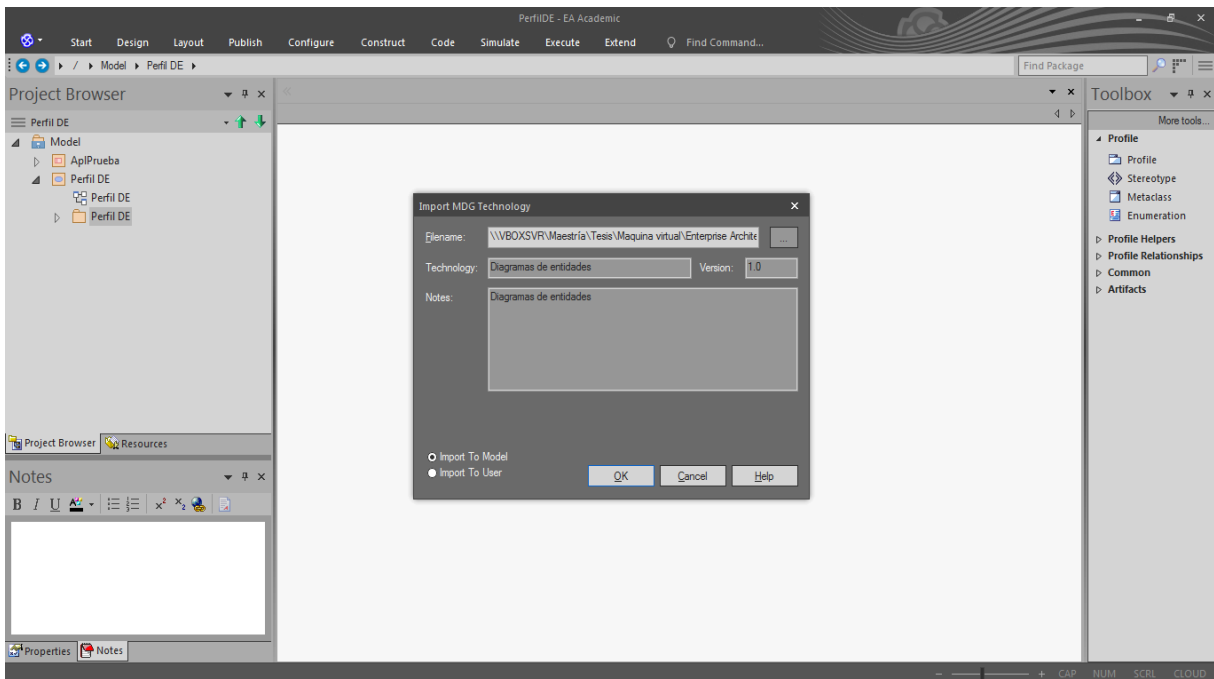


Figura 5.59 Propiedades del editor.

Con esto se concluyó la carga del editor para los diagramas de entidades.

5.3.1. Ejemplo del editor

A continuación, se muestra un ejemplo para crear un diagrama de entidades en *Enterprise Architect*.

1. Se genera un proyecto nuevo en EA (ver sección 5.2.1. *Generar proyecto*).
2. Se da clic derecho en *Viewy* se selecciona *Add Diagram*.
3. En el lado izquierdo se muestran los tipos de diagramas que tiene EA. En la figura 5.61, se presenta el diagrama que definimos anteriormente. Seleccionamos *Diagrama de entidades* y el diagrama a crear sería de tipo *Perfil DE*. Al seleccionar *Perfil DE*, en el recuadro de abajo se muestra la descripción que se definió en el perfil UML.

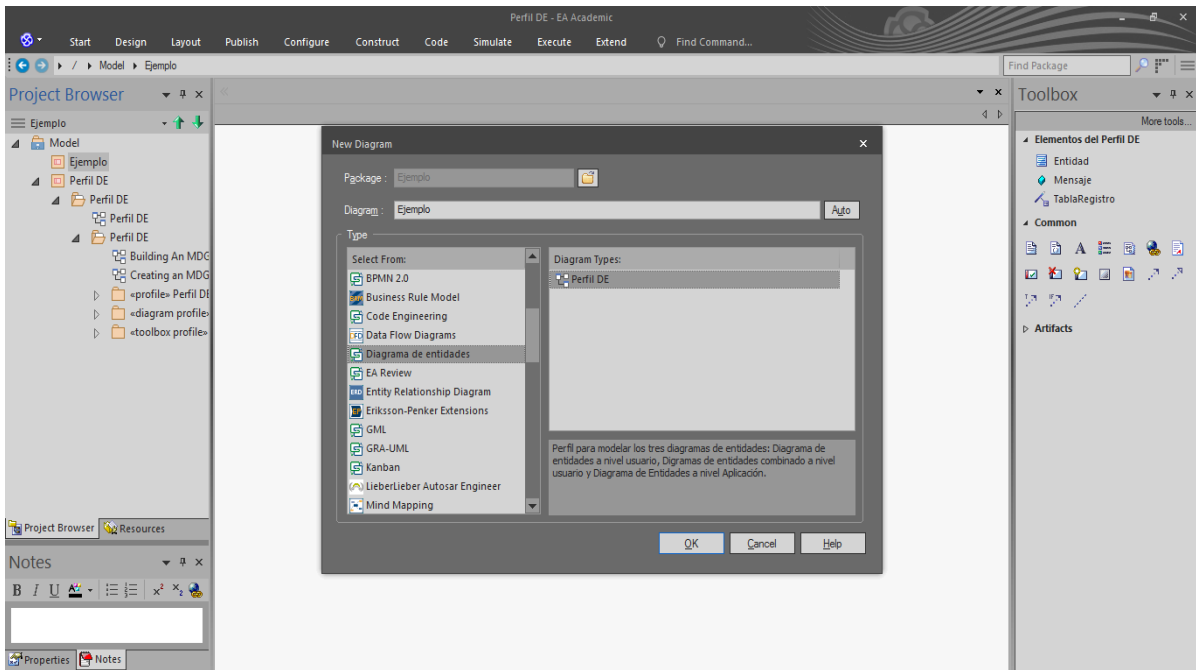


Figura 5.60 Generar un diagrama de entidades

4. Una vez seleccionado el tipo de diagrama se genera el área de trabajo para construir los diagramas de entidades, así como se muestran los elementos que se utilizarán para construir dicho diagrama. (Ver figura 5.62)

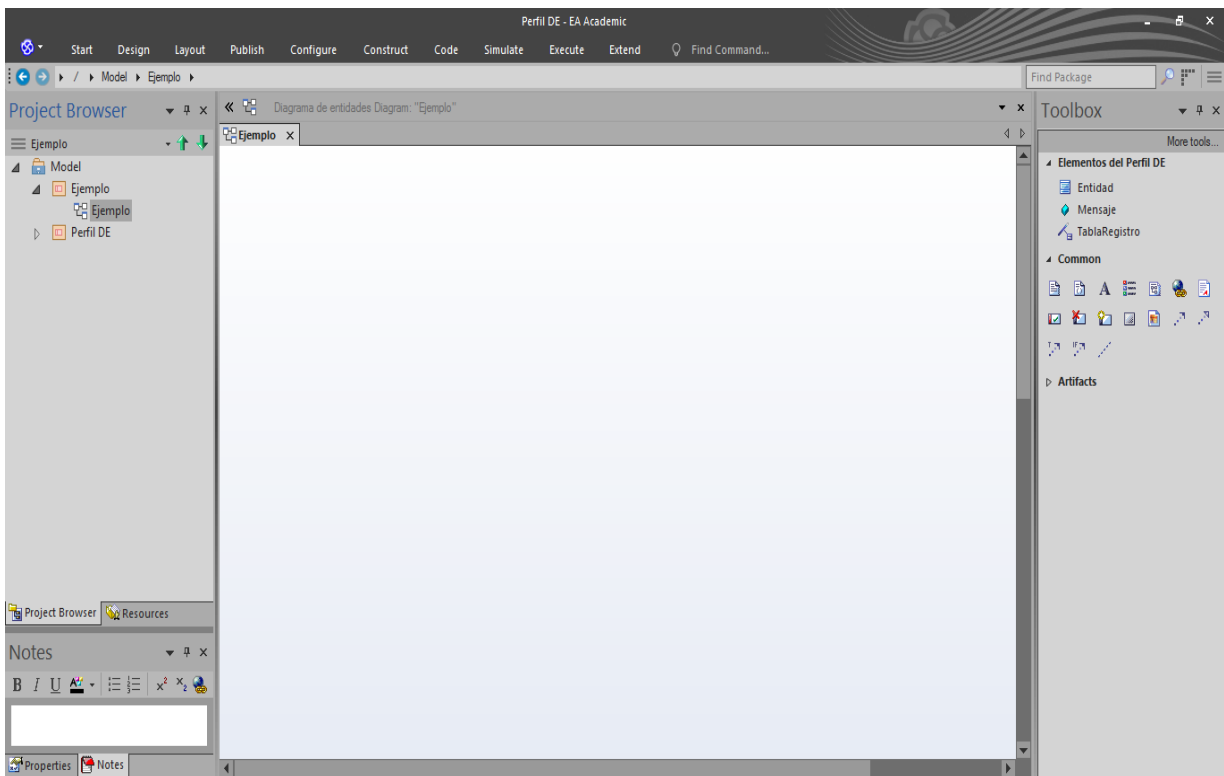


Figura 5.61 Área de trabajo para construir un diagrama de entidades

5.3.2. Pasos para construir un diagrama de entidades en Enterprise Architect.

En este apartado se presentan los pasos para construir un diagrama de entidades utilizando el editor en *Enterprise Architect*.

Para la generar cualquiera de los tres diagramas de entidad (a nivel usuario, combinado o aplicación) se deben realizar los siguientes pasos:

1. El usuario abre el software *Enterprise Architect* y genera un nuevo proyecto (le da un nombre).
2. El software presenta una ventana para elegir el diagrama desea crear.
3. El usuario da clic en la opción cancelar.
4. El usuario da clic derecho en *Model* ubicado en panel *Project Browser*.
5. El usuario genera una vista simple, da clic derecho en *Model* (ubicado en el panel *Project Browser*) y selecciona la opción *Add view*.
6. El software presenta una ventana donde solicita el nombre y tipo de vista.
7. El usuario ingresa el nombre de la vista y elige una vista simple.
8. El software genera la vista (paquete) y se muestra en el panel *Project Browser*.
9. El usuario da clic derecho en la vista generada, y selecciona *Add Diagram* para crear un diagrama.
10. El software muestra una ventana con los tipos de diagramas.
11. El usuario ingresa un nombre y selecciona *Entity diagram* en el recuadro *Type*.
12. El software genera el área de trabajo para crear el diagrama de entidades.
13. El usuario selecciona el elemento *Entidad* de la caja de herramienta y lo coloca en el área de trabajo.
14. El software muestra una ventana solicitando el nombre de la entidad y las propiedades correspondientes del elemento *Entidad* (ubicado en el panel derecho con el nombre *DE*).
15. El usuario ingresa el nombre de la entidad. En las propiedades, ingresa el identificador (por ejemplo, 1) y selecciona el tipo de entidad (sea principal, secundaria y combinada, dependiendo del tipo de diagrama).

- 15.1. Si usuario no llena las propiedades correspondientes, se mostrará un mensaje de error.
16. El usuario repite del 13 al 16 hasta colocar todas las entidades correspondientes. Tomando en cuenta que:
 - 16.1. Si se va realizar un diagrama de entidades a nivel usuario solo debe existir una entidad de tipo *principal* y n entidades de tipo *secundaria*. En caso, que ya exista una entidad de tipo principal, y el usuario desea crear otra entidad del mismo tipo, se mostrará un mensaje de error.
 - 16.2. Si se va realizar un diagrama de entidades combinado, todas las entidades deben ser de tipo *combinado*. En caso que, existan entidades creadas del tipo *principal* o *secundaria*, y el usuario desea colocar otra entidad del tipo *combinado*, se mostrará un mensaje de error.
17. El usuario selecciona el elemento *TablaRegistro* de la caja de herramientas y coloca la relación entre las entidades correspondientes (por ejemplo, de A B).
18. El software muestra una ventana solicitando el nombre de la tabla de registro y las propiedades correspondientes del elemento.
19. El usuario ingresa el nombre de la tabla (por ejemplo, Entidad1 – Entidad 2) y el identificador único (por ejemplo, si es la primera tabla creada, el identificador sería 1).
20. El usuario repite el paso del 17 al 19 hasta colocar las relaciones (con la respectiva tabla) entre las entidades. Tomando en cuenta que:
 - 20.1. Solo puede existir una relación entre dos entidades.
 - 20.2. Por cada relación que el usuario coloca, se genera una tabla de registro única.
 - 20.3. En la tabla de registro, el usuario colocará los mensajes.
 - 20.4. El identificar debe ser único, si hay un duplicado se mostrará un mensaje de error.
21. El usuario selecciona el elemento *Mensaje* de la caja de herramientas y lo coloca en la tabla de registro (*TablaRegistro*).
22. El software muestra una ventana solicitando el mensaje.
23. El usuario ingresa el mensaje (Por ejemplo, A-1 Mensaje).
24. El usuario debe ingresar las propiedades que le corresponde al mensaje. Para ver estás propiedades:

- 24.1. El usuario da doble clic en el mensaje.
- 24.2. El software muestra una ventana, donde en la parte inferior derecha, se encuentra la opción de *Tagged Values*.
- 24.3. El usuario selecciona esa opción, e ingresa los valores solicitados (identificador, proceso, dirección, origen y destino). El usuario no debe duplicar el identificador.
25. El usuario repite del paso 21 al 24 hasta colocar todos los mensajes pertinentes en la tabla de registro.
26. El usuario repite del paso 14 al 25 hasta construir el diagrama de entidades.
27. El usuario guarda el diagrama de entidades.

Estos pasos se utilizan en la en anexo *B. Caso de estudio* para realizar los diagramas con la herramienta de edición.

El editor que se define en este capítulo, solo permite la elaboración de los diagramas de entidades, más no se abarca el proceso de transformación entre los diagramas. Por ejemplo, de un conjunto de diagramas de entidades a nivel usuario a un diagrama de entidades combinado, o un diagrama de entidades combinado a un diagrama de entidades a nivel aplicación. Por lo tanto, en el siguiente capítulo se especifican los requerimientos para construir el diagrama de entidades combinado a partir de un conjunto de diagramas de entidades.

CAPÍTULO 6. ESPECIFICACIÓN DE REQUERIMIENTOS PARA LA CONSTRUCCIÓN DE DIAGRAMA DE ENTIDADES COMBINADO

En este capítulo se especifica un requerimiento general para la construcción de los diagramas de entidades a nivel usuario y los requerimientos para la construcción de un diagrama de entidades combinado. Además, se presenta un ejemplo aplicando el proceso definido.

En este capítulo se describen los requerimientos para el proceso de transformación de un conjunto de diagramas de entidades a nivel usuario a un diagrama de entidades combinado. En la sección 6.1 se exponen los conceptos de un diagrama de entidades a nivel usuario y un diagrama de entidades combinado para una mejor comprensión del proceso. En la sub-sección 6.1.1. se describe el requerimiento general para construir un diagrama de entidades a nivel usuario. En la sub-sección 6.1.2. se enlistan los requerimientos para construir un diagrama de entidades combinado. En la sub-sección se presenta un ejemplo aplicando el proceso de transformación de un conjunto de entidades a nivel usuario a un diagrama de entidades combinado.

6.1. Transformación de diagramas ENU a diagrama EC

Es importante mencionar los conceptos de un diagrama de entidades a nivel usuario y un diagrama de entidades combinado para poder comprender el proceso de transformación. Así como señalar los elementos que se utilizan para la construcción de los diagramas de entidades en general (para mayor detalle ver sección 3.2. *Diagramas de entidades*).

- *Diagrama de entidades a nivel usuario.* Es un medio para recopilar información inicial a través de investigaciones y entrevistas. Este diagrama se genera por cada usuario involucrado en el dominio de la aplicación a realizar. Se utilizan dos tipos de entidades: principal y secundarias.
- *Diagrama de entidades combinado.* Como su nombre lo indica este diagrama representa la vista general combinada de todas las entidades involucradas en el sistema.

A continuación, se presentan, en la tabla 6.1, los elementos utilizados para la creación de los diagramas (ver sección 4.2.6. *Diseño del perfil UML*):

Tabla 6.1 Elementos utilizados en los diagramas de entidades

Elemento	Descripción
Entidad	Representa una persona u organización involucrada con el dominio de aplicación. Una entidad puede ser: principal o secundaria.

Relación	Representa las interacciones que pueden existir entre las entidades.
Registro	Contendrá los mensajes enviados entre las entidades. Se agregarán manualmente.
Mensaje	Representa los mensajes que son enviados de una entidad a otra. Un mensaje tiene asigna un identificador único y una dirección (envío o recibido).

6.1.1. Requerimiento para la creación de un diagrama ENU

En esta sección se detalla en el requerimiento general para la construcción de un diagrama de entidades a nivel usuario.

RF01: *Generar un diagrama de entidades a nivel usuario.* En este requerimiento solo se menciona el escenario principal que se necesita para generar correctamente un diagrama de entidades a nivel usuario. Es importante especificar bien las entidades y sus relaciones en el diagrama porque serán tomados como datos de entrada para el diagrama de entidades combinado.

En la tabla 6.2 se detallan las precondiciones y el proceso que se requiere para crear un diagrama de entidades a nivel usuario.

Tabla 6.2 Requerimientos para crear un diagrama ENU

Nombre	RF01: Generar un diagrama de entidades a nivel usuario
Actores	Usuarios generales
Descripción	Permite realizar la construcción de un diagrama de entidades a nivel usuario de manera correcta.
Precondiciones	1. Identificar la entidad principal. 2. Identificar las entidades secundarias. 3. Identificar las relaciones. 4. Identificar los mensajes.
Post condiciones	Construcción correcta del diagrama de entidades a nivel usuario

Escenario de éxito principal	<ol style="list-style-type: none"> 1. Agregar la entidad principal en. 2. Agregar las entidades secundarias alrededor de la entidad principal. 3. Agregar las relaciones entre la entidad principal y las entidades secundarias. 4. Generar una tabla de registro para cada relación. 5. Ordenar los mensajes por proceso. (por ejemplo, E-1). 6. Agregar los mensajes que se envían de una entidad a otra en la tabla de registro. <ol style="list-style-type: none"> 6.1. El mensaje se considera de envío cuando sale de la entidad principal a la entidad secundaria. 6.2 El mensaje se considera de recepción cuando sale de una entidad secundaria a una entidad principal.
Escenario de fracaso	Las relaciones entre la entidad principal y las secundarias no son correctas. Los mensajes no son claros.
Suposiciones	El diagrama de entidades a nivel usuario contempla todas las entidades y sus relaciones.

En la figura 6.1 se muestra el proceso obtenido para generar el diagrama de entidades a nivel usuario.

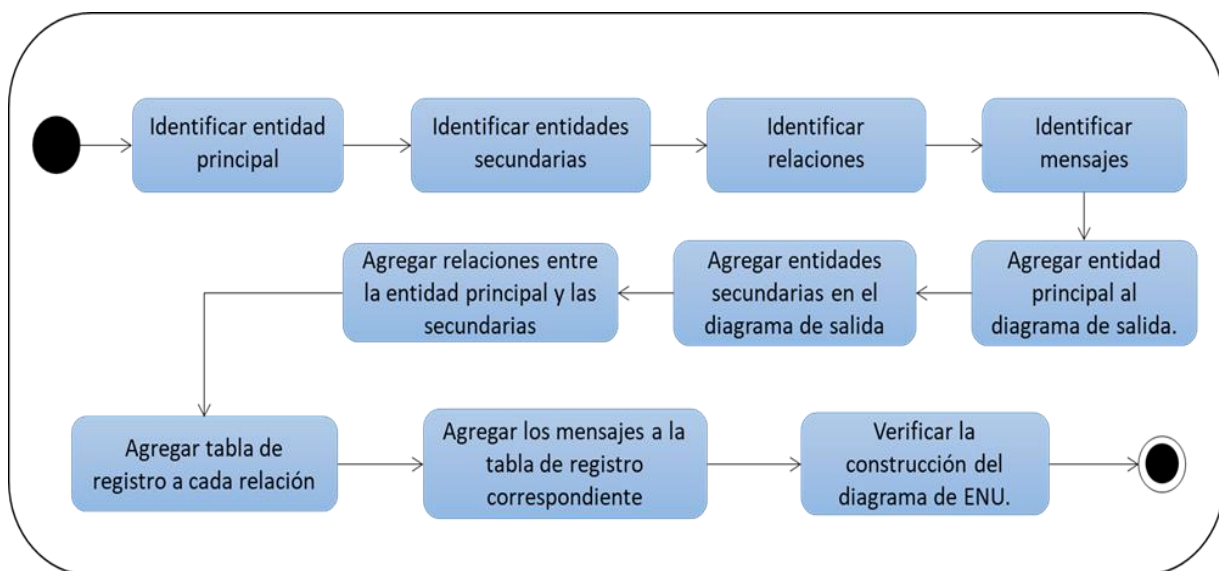


Figura 6.1 Proceso para la construcción de un diagrama de entidades a nivel usuario

6.1.2. Requerimientos para la transformación de Diagramas ENU a un Diagrama EC

En este apartado se describen los requerimientos específicos para realizar la construcción de un diagrama de entidades combinado a partir un conjunto de diagramas de entidades a nivel usuario, es decir, que permita la transformación entre los dos tipos de diagramas.

A continuación, se en listan los requerimientos específicos relacionados al requerimiento general *RF02: Generar un diagrama de entidades combinado*.

- **RF02:** *Generar un diagrama de entidades combinado*. Permite realizar la construcción correcta de un diagrama de entidades combinado.
 - **RF02-01:** *Leer de los diagramas de entidades*. Obtiene y almacena los elementos que involucran a cada uno de los diagramas leídos.
 - **RF02-01-01:** Obtener las entidades de cada diagrama.
 - **RF02-01-02:** Obtener las relaciones de cada diagrama.
 - **RF02-01-03:** Obtener los mensajes de cada relación.
 - **RF02-02:** *Comparar e identificar los elementos similares en los diagramas de entidades*. Permite procesar e identificar los elementos similares de los diagramas para su almacenamiento.
 - **RF02-02-01:** Comparar todas las entidades extraídas.
 - **RF02-02-02:** Comparar todas las relaciones extraídas.
 - **RF02-02-03:** Comparar los mensajes entre relaciones.
 - **RF02-03:** *Agregar los elementos identificados en el diagrama de salida*. Permite construir cada uno de los elementos identificados en el diagrama de salida.
 - **RF02-03-01:** Agregar las entidades identificadas, omitiendo duplicados.
 - **RF02-03-02:** Agregar las relaciones entre las entidades.
 - **RF02-03-03:** Agregar los mensajes correspondientes a las relaciones.

En las tablas 6.3 a 6.6 se describen los requerimientos especificados.

Tabla 6.3 Requerimiento RF02

Nombre	RF02: Generar un diagrama de entidades combinado
Actores	Usuario
Descripción	Permite realizar la construcción correctamente de un diagrama de entidades combinado.
Precondiciones	1. Dos o más diagramas de entidades a nivel usuario construidos correctamente (RF01).
Post condiciones	Construcción correcta del diagrama de entidades combinado.
Escenario de éxito principal	<ol style="list-style-type: none"> 1. Identificar todas las entidades similares que existen en los diagramas de entidades a nivel usuario. 2. Agregar todas las entidades en el diagrama de salida, omitiendo entidades duplicadas. 3. Validar las relaciones entre las entidades, evitando duplicadas y redundancias. 4. Agregar las relaciones entre las entidades en el diagrama de salida. 5. Validar los mensajes de cada relación, evitando duplicados y redundancias. 6. Agregar los mensajes a las relaciones en el diagrama de salida.
Escenario de fracaso	<p>Los nombres de las entidades no coinciden.</p> <p>Faltan relaciones entre los diagramas.</p> <p>Los mensajes no son iguales.</p>
Suposiciones	El diagrama de entidades combinado contempla todas las entidades y sus relaciones.

Tabla 6.4 Requerimiento RF02-01

Nombre	RF02-01: Leer los diagramas de entidades correctamente.
Actores	Usuario
Descripción	Permite obtener los elementos de cada uno de los diagramas ingresados.
Precondiciones	1. Dos o más diagramas de entidades a nivel usuario construidos correctamente.

Post condiciones	Extracción y almacenamiento de los elementos.
Escenario de éxito principal	<ol style="list-style-type: none"> 1. Leer un diagrama de entidades a nivel usuario. 2. Identificar las entidades, relaciones y mensajes del diagrama. 3. Almacenar las entidades, relaciones y mensajes obtenidos. 4. Mientras existan otros diagramas por leer, se repite del paso 1 al 3.
Escenario de fracaso	<p>Algunas entidades no poseen nombre. Las relaciones no poseen mensajes.</p>
Suposiciones	Se obtienen todos los elementos encontrados en el conjunto de diagramas de entidades a nivel usuario.

Tabla 6.5 Requerimiento RF02-02

Nombre	RF02-02: Comparar e identificar los elementos similares de los diagramas de entidades.
Actores	Usuarios
Descripción	Permite procesar los elementos obtenidos para posteriormente almacenarlos.
Precondiciones	1. Dos o más diagramas de entidades a nivel usuario construidos correctamente.
Post condiciones	Procesamiento y validación correcta de los elementos.
Escenario de éxito principal	<ol style="list-style-type: none"> 1. Leer una entidad. 2. Comparar la entidad con las entidades obtenidas. 3. Verificar si esta repetida, si está repetida eliminarla de las entidades almacenadas e ir paso 9, sino identificarla como entidad única y posterior ir al paso 9. 4. Almacenar como entidad única. 5. Revisar si existen otras entidades por leer, si es así, repetir del paso 1 al 5, sino, ir al paso 6. 6. Leer relaciones entre entidades. 7. Comparar relaciones.

	<p>8. Si la relación esta repetida, se elimina de las relaciones obtenidas y se almacena, sino solo se almacena como relación única.</p> <p>9. Verificar si existen otras relaciones, repetir paso 6 al 8.</p>
Escenario de fracaso	<p>Las relaciones no coinciden.</p> <p>No se encontraron entidades similares.</p>
Suposiciones	Conjunto de elementos procesados y validados.

Tabla 6.6 Requerimiento RF02-03

Nombre	RF02-03 Agregar los elementos identificados en el diagrama de salida.
Actores	Usuarios
Descripción	Permite obtener los elementos de cada uno de los diagramas ingresados.
Precondiciones	1. Dos o más diagramas de entidades a nivel usuario construidos correctamente.
Post condiciones	Construcción correcta del diagrama de entidades combinado.
Escenario de éxito principal	<p>Agregar las entidades identificadas al diagrama de salida.</p> <p>Agregar las relaciones entre las entidades correspondientes.</p> <p>Agregar los mensajes identificados a cada relación.</p>
Escenario de fracaso	Las relaciones no coinciden.
Suposiciones	Generación del diagrama de entidades combinado.

En la figura 6.2 se presentan la secuencia del proceso para construir un diagrama de entidades combinados.

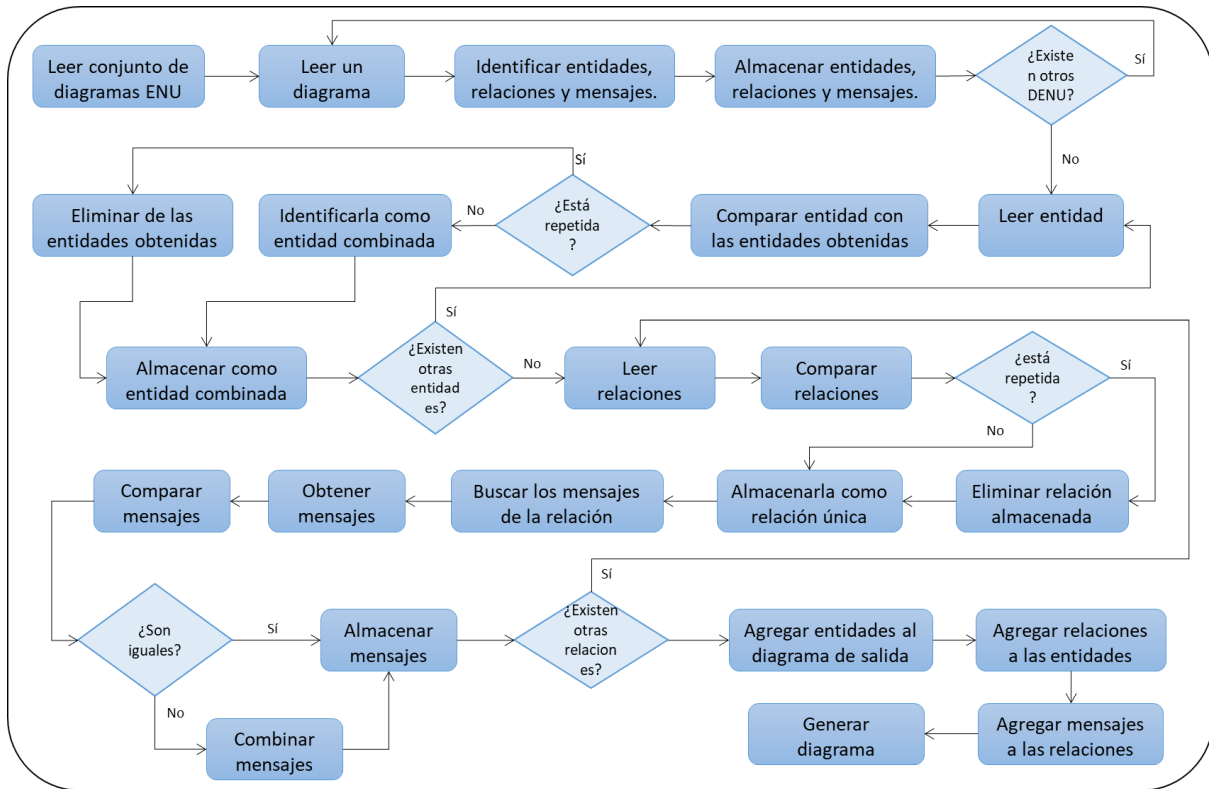


Figura 6.2 Proceso para la generación de un diagrama de entidades combinado

6.1.3. Resultado

En este apartado se muestra una representación del proceso de transformación de un conjunto de diagramas de entidades a nivel a un diagrama de entidades combinado.

1. Datos de entrada

Se elaboran dos diagramas a nivel usuario basado en el documento *Reporte Técnico* de la M. C. Edna López López.

Los diagramas creados son:

- Diagrama de entidades a nivel usuario (AB). Ver figura 6.3
- Diagrama de entidades a nivel usuario (JCI). Ver figura 6.4

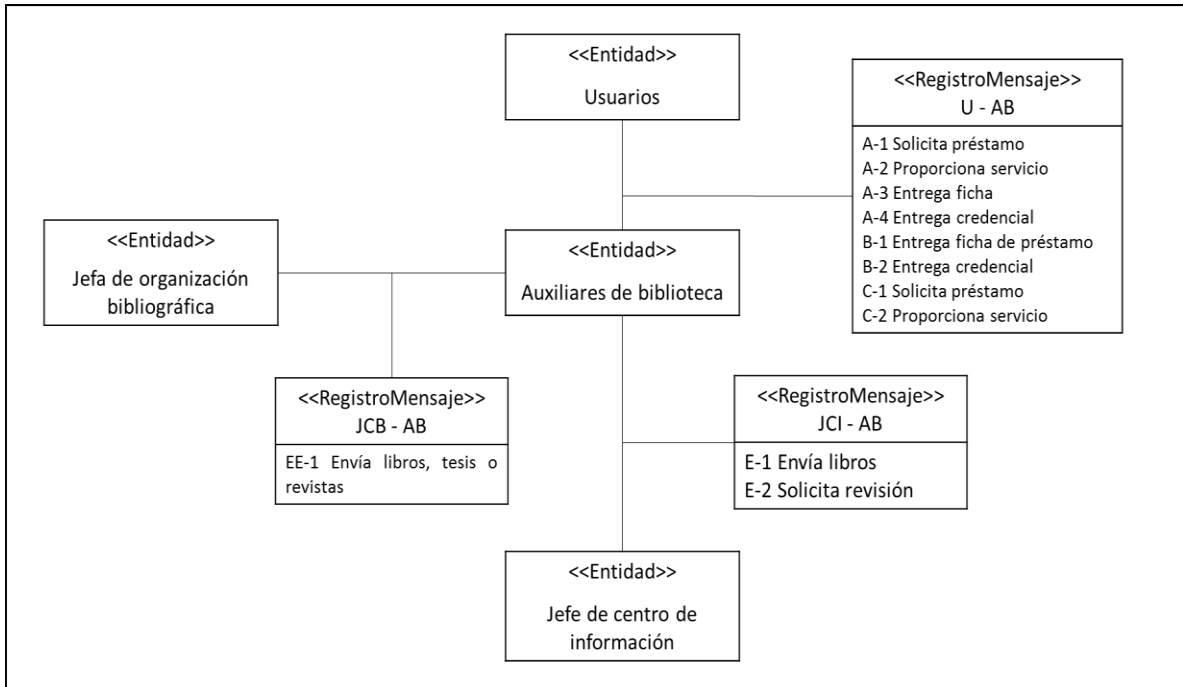


Figura 6.3 Diagrama de entidades a nivel Auxiliares de Biblioteca (AB)

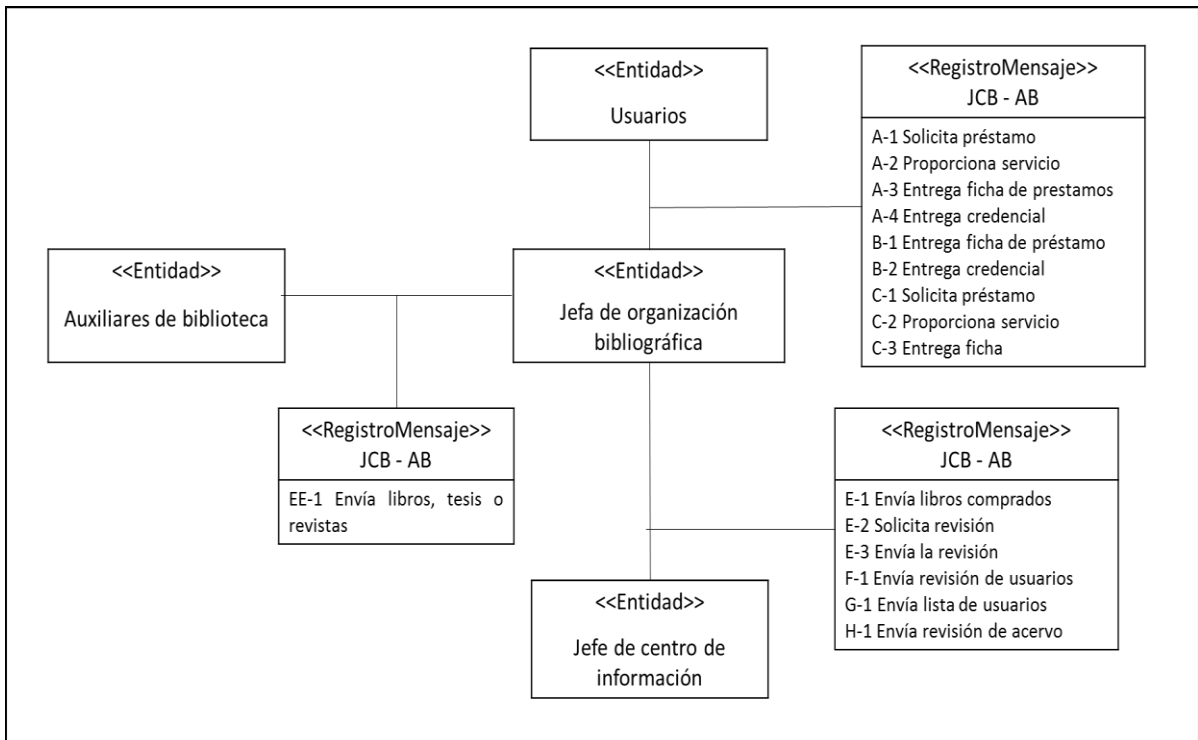


Figura 6.4 Diagrama de entidades a nivel Jefa de organización bibliográfica (JOB)

2. Identificación de entidades

De acuerdo a los diagramas presentados en la figura 6.3 y 6.4, se identifican un conjunto de entidades, evitando agregar duplicados como se muestra en la tabla 6.7. A cada entidad se le agrega un identificador que se utilizará más adelante para obtener las relaciones que existen entre ellas.

Tabla 6.7 Entidades

Nombre de la entidad	Identificador
Auxiliares de Biblioteca	AB
Usuarios	U
Jefa de Organización Bibliográfica	JOB
Jefe de Centro de Información	JCI

3. Identificación de relaciones

Cuando se obtienen las entidades, se procede a identificar las relaciones que existen entre estas. A cada relación se le asigna un identificador y una descripción para reconocer a qué par de entidades corresponden (ver tablas 6.8 y 6.9).

Tabla 6.8 Relaciones del diagrama de entidades a nivel AU

Relación	Descripción
JOB - AB	Relación entre Jefa de organización bibliográfica y Auxiliares de biblioteca
JCI - AB	Relación entre Jefe de centro de información y Auxiliares de biblioteca
U - AB	Relación entre Usuarios y Auxiliares de biblioteca

Tabla 6.9 Relaciones del diagrama de entidades a nivel JOB

Relación	Descripción
AB - JOB	Relación entre Auxiliares de biblioteca y Jefa de organización bibliográfica
U - JOB	Relación entre Usuarios y Jefe de centro de información
JCI - JOB	Relación entre Relación entre Jefe de centro de información y Jefa de organización bibliográfica

4. Identificación de mensajes

En este punto se registran los mensajes que tienen de una entidad a otra, para utilizarlo en la construcción del diagrama de entidades combinado. Se genera una tabla por relación, la cual contendrá el proceso, el mensaje, el origen y el destino. Las tablas 6.10, 6.11 y 6.12 corresponden al diagrama de entidades de nivel AB.

Tabla 6.10 Mensajes de la relación JOB a AB

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
EE-1	Enviar artículos registrados	Envía libros, tesis o revistas registradas para ser exhibidas	JOB	AB

Tabla 6.11 Mensajes de la relación JCI – AB

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
E-1	Envía libros	Envía libros comprados o tesis	JCI	AB
E-2	Solicita revisión	Solicita revisión de catalogación y clasificación de libros o tesis (fichas de acervo)	AB	JCI

Tabla 6.12 Mensajes de la relación U – AB

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
A-1	Solicita préstamo	Solicita préstamo bibliotecario de acervo (no único ni 1er ejemplar)	U	AB
A-2	Proporciona servicio	Proporciona servicio bibliotecario	AB	U
A-3	Entrega ficha	Entrega ficha de préstamos de acervo	AB	U

A-4	Entrega credencial	Entrega credencial	U	AB
B-1	Entrega ficha de préstamo	Entrega ficha de préstamo	U	AB
B-2	Entrega credencial	Entrega credencial	AB	U
C-1	Solicita préstamo	Solicita préstamo bibliotecario de acervo único o 1er ejemplar	AB	U
C-2	Proporciona servicio	Proporciona servicio bibliotecario	AB	U

Las tablas 6.13, 6.14 y 6.15 pertenecen al diagrama de entidades a nivel JOB.

Tabla 6.13 Mensajes de la relación U - JOB

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
EE-1	Enviar artículos registrados	Envía libros, tesis o revistas registradas para ser exhibidas	JOB	AB

Tabla 6.14 Mensajes de la relación de U – JOB

Proceso	Mensaje	Descripción de mensaje	Origen	Destino
A-1	Solicita préstamo	Solicita préstamo bibliotecario de acervo (no único ni 1er ejemplar)	U	JOB
A-2	Proporciona servicio	Proporciona servicio bibliotecario	JOB	U
A-3	Entrega ficha	Entrega ficha de préstamos de acervo	JOB	U
A-4	Entrega credencial	Entrega credencial	U	JOB
B-1	Entrega ficha de préstamo	Entrega ficha de préstamo	U	JOB
B-2	Entrega credencial	Entrega credencial	JOB	U

C-1	Solicita préstamo	Solicita préstamo bibliotecario de acervo único o 1er ejemplar	U	JOB
C-2	Proporciona servicio	Proporciona servicio bibliotecario	JOB	U
C-3	Entrega ficha	Entrega ficha para préstamo de acervo único o 1er ejemplar para firma	JOB	U
C-4	Entrega ficha	Entrega ficha de acervo único o 1er ejemplar firmada	U	JOB
C-5	Entrega credencial	Entrega credencial	JOB	U
D-1	Retorna acervo	Retorna acervo único o 1er ejemplar	U	JOB
D-2	Entrega credencial	Entrega credencial	JOB	U

Tabla 6.15 Mensajes de la relación JCI – JOB

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
E-1	Envía libros	Envía libros comprados	JCI	JOB
E-2	Solicita revisión	Solicita revisión de catalogación y clasificación de libros o tesis (fichas de acervo)	JOB	JCI
E-3	Envía la revisión	Envía la revisión de catalogación y clasificación para ser capturada en el sistema	JCI	JOB
F-1	Envía revisión de usuarios	Envía revisión de usuarios para ser modificados	JCI	JOB
G-1	Envía lista de usuarios	Envía lista de usuarios para dar de alta	JCI	JOB
H-1	Envía revisión de acervo	Envía revisión de acervo para ser modificado o eliminado en el sistema	JCI	JOB

5. Salida

Cuando se han identificado las entidades, las relaciones y el flujo de los mensajes que existen en cada diagrama, se procede a transformar el diagrama de entidades combinado. Se sigue paso a paso el proceso, agregando primero las entidades y después las relaciones que existan, así como el flujo de mensajes de cada relación. En la figura 6.5 se muestra el resultado final.

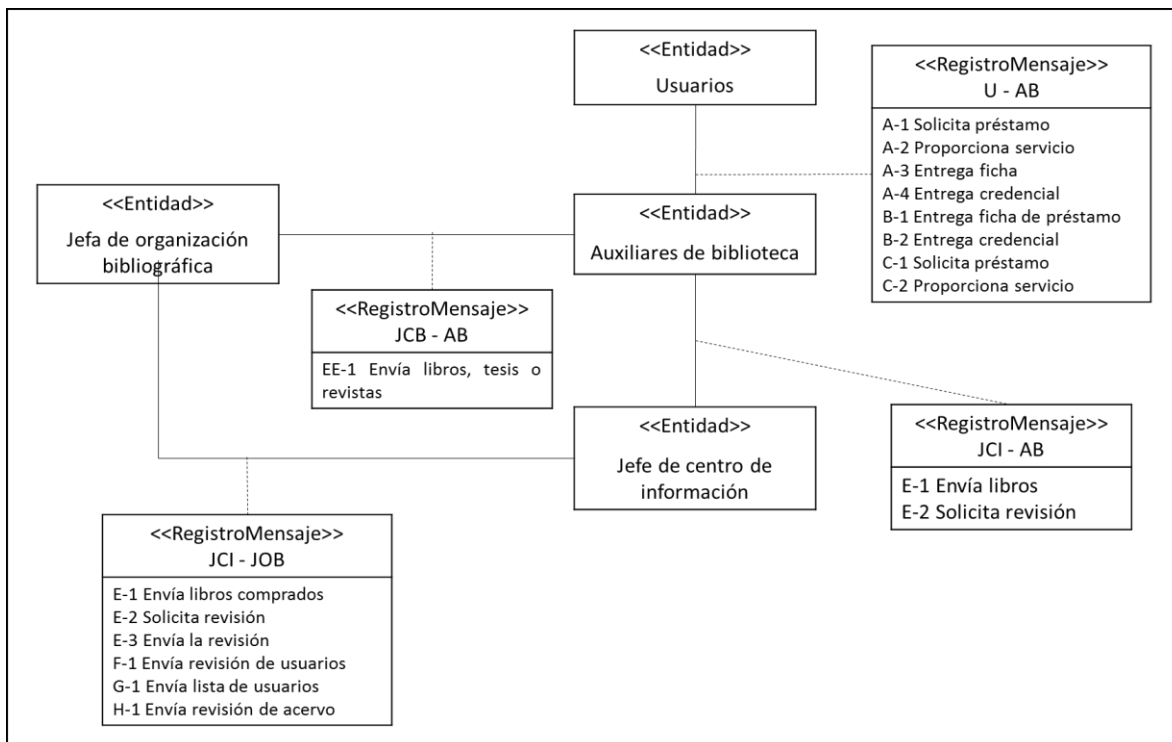


Figura 6.5 Diagrama de entidades combinado

En el siguiente capítulo se presentan las pruebas realizadas para probar las funcionalidades generales del editor y los resultados obtenidos a partir de dichas pruebas.

CAPÍTULO 7. PRUEBAS Y RESULTADOS

En este capítulo se presentan las pruebas realizadas acorde al diseño de pruebas bajo la norma de la IEEE para la documentación de pruebas de software y sistemas (*IEEE Computer Society, 2008*). Asimismo, se muestran los resultados obtenidos de dichas pruebas.

7.1. Pruebas

En esta sección se presenta un breve resumen del plan de pruebas que se realizó para generar los diagramas de entidades empleando el perfil UML descrito e implementado en la sección 4.4. *Instanciar el editor en Enterprise Architect*. El plan de pruebas fue diseñado con base al estándar IEE 829.

Para realizar la fase de pruebas se desarrollaron seis apartados:

1. **Plan de Pruebas.** En este apartado se define el alcance del plan de pruebas, los criterios de aceptación, los elementos y los casos de pruebas que se requieren para llevarlo a cabo.
2. **Diseño de Pruebas.** Se detallan las características a probar, y la forma para realizar las pruebas conforme a los casos de pruebas descritos en el punto 1.
3. **Casos de Pruebas.** Se especifica el avance, entrada y salida de cada de prueba que se establecen en el apartado *Plan de Pruebas* y acorde a los casos de pruebas.
4. **Procedimiento de Pruebas.** Se describen los pasos del procedimiento que se sigue para realizar el conjunto de pruebas descritas en el punto 2. *Casos de Prueba*. Este procedimiento se establece en la sección 4.5. *Proceso para construir un diagrama de entidades* para generar cada uno de los diagramas de entidades propuestos.
5. **Registro de Pruebas.** Se registran los detalles de la ejecución de las pruebas.
6. **Reporte de Pruebas.** Se detallan los resultados generales de las pruebas, conclusiones y recomendaciones.

En el anexo A. *Plan de Pruebas*, se muestra con mayor detalle cada una de las etapas mencionadas.

7.1.1. Plan de Pruebas

En esta apartado se define el alcance del plan de pruebas, los criterios de aceptación, los elementos y los casos de pruebas que se requieren para llevarlo a cabo (ver tabla 7.1).

Los casos de pruebas que se establecen son:

- Crear el diagrama entidad a nivel usuario.
- Crear el diagrama entidad combinado a nivel usuario.
- Crear el diagrama entidad a nivel aplicación.

Tabla 7.1 Casos de Pruebas

Funcionalidad	Caso de Prueba
Crear el diagrama entidad a nivel usuario.	TC-01
Crear el diagrama entidad combinado a nivel usuario.	TC-02
Crear el diagrama entidad a nivel aplicación.	TC-03

Para mayor detalle ver el anexo A.1. *Plan de Pruebas*.

7.1.2. Diseño de Pruebas

En esta sección se detallan las características a probar, y la forma para realizar las pruebas conforme a los casos de pruebas descritos en el punto anterior.

Las estrategias de pruebas propuestas se describen en la tabla 7.2.

Tabla 7.2 Pruebas a realizar

Prueba	Etapas	Descripción	Funcionalidades
Crear diagrama entidad a nivel usuario	Crear el diagrama entidad a nivel usuario	Verificar que el diagrama cuenta con los elementos necesarios apegando al siguiente diagrama: Diagrama de comunicación a nivel "Jefa de organización bibliográfica" a nivel CIM.	<ul style="list-style-type: none"> ▪ Agregar una entidad ▪ Nombrar una entidad ▪ Agregar una relación TablaRegistro ▪ Nombrar relación TablaRegistro ▪ Agregar un mensaje
2.Crear diagrama	Crear el diagrama	Verificar que el diagrama cuenta con	

entidad combinado a nivel usuario	entidad combinado a nivel usuario	los elementos necesarios apegando al siguiente diagrama: Diagrama de comunicación combinado de todos los usuarios a nivel CIM.	▪ Agregar propiedades al mensaje
3.Crear diagrama entidad a nivel aplicación	Crear el diagrama entidad a nivel aplicación	Verificar que el diagrama cuenta con los elementos necesarios apegando al siguiente diagrama: Diagrama de comunicación a nivel aplicación a nivel CIM.	

En cada una de las pruebas se evalúan 6 funcionalidades generales del editor.

7.1.3. Casos de Pruebas

En el caso de Pruebas se especifica el avance, entrada y salida de cada de prueba que se establecen en el documento *Plan de Pruebas* y acorde a los casos de pruebas.

Para cada caso de prueba se describe:

- Objetivo.
- Entrada del caso.
- Salida del caso.

Ver Anexo A.3. *Casos de Prueba*.

7.1.4. Procedimiento de Pruebas

En este apartado se describen los pasos del procedimiento que se sigue para realizar el conjunto de pruebas descritas en el Anexo A.2. *Casos de Prueba*. Este procedimiento es el que establece en la sección 5.3.2. *Pasos*

para construir un diagrama de entidades en Enterprise Architect. para generar cada uno de los diagramas de entidades propuestos.

Los puntos en general que contemplan para el proceso son:

- **Reporte** del caso a ejecutar. Breve descripción de la ejecución de la prueba, que se documenta en el reporte de pruebas.
- **Preparación.** Se debe tener implementado el perfil UML en la herramienta *Enterprise Architect*.
- **Inicio.** Se debe generar un nuevo proyecto y agregar el tipo de diagrama a realizar, en este caso, es Diagrama de entidades.
- **Proceso.** Se establece el proceso que se utiliza para generar un diagrama.
- **Medición.** El tiempo de elaboración depende del usuario.
- **Suspender.** Este caso, la prueba se puede suspender en el momento que se desee.
- **Reiniciar.** La prueba se puede volver a realizar siempre que se requiera.
- **Cierre.** La prueba termina con la generación exitosa de los diagramas de entidades.

Solo se muestra un breve resumen de cada punto que se realiza en los procedimientos de casos de prueba, la descripción detallada se encuentra en el Anexo A.4. *Procedimiento de Pruebas*.

7.1.5. Registro de Pruebas

En este apartado se registran los detalles de la ejecución de las pruebas.

Los puntos que se presentan son:

- El registro de la prueba.
- Las actividades y eventos que se realizan al ejecutar la prueba.
- Las anomalías presentadas.
- Un ejemplo de los datos de pruebas para cada caso.

A continuación, en la tabla 7.3 se presenta el registro de la prueba TPr-01 y en la tabla 6.4 los datos de entrada que se emplearon para dicha prueba. En

el anexo A.5. *Caso de estudio*, se presentó la ejecución de dicha prueba con respecto al perfil UML.

Tabla 7.3 Registro de la prueba TPr -01

Paso	Actor	Descripción	Verificado
1.	Usuario	Abrir la herramienta <i>Enterprise Architect</i>	Sí
2.	Usuario	Generar un nuevo proyecto.	Sí
3.	Usuario	Dar clic derecho en el paquete <i>Model</i> ubicado en el panel <i>Project Broser</i> .	Sí
4.	Usuario	Elije un vista simple.	Sí
5.	Herramienta	Genera la vista (paquete), que se muestra en el panel <i>Project Browser</i> .	Sí
6.	Usuario	Dar clic derecho en la vista generada, y seleccionar <i>Add Diagram</i> para crear un diagrama.	Sí
7.	Herramienta	Muestra una ventana con los tipos de diagramas.	Sí
8.	Usuario	Ingresar el nombre del diagrama y seleccionar <i>Entity Diagram</i> del recuadro <i>Type</i> .	Sí
9.	Herramienta	Genera el área de trabajo para crear el diagrama de entidades.	Sí
10.	Usuario	Seleccionar el elemento <i>Entidad</i> de la caja de herramientas y agrega al área de trabajo.	Sí
11.	Herramienta	Muestra un ventana solicitando el nombre de la entidad y sus propiedades correspondientes.	Sí
12.	Usuario	Ingresar el nombre de la entidad. En la propiedades ingresar el identificador, y seleccionar el tipo de entidad (principal o secundaria).	Sí

13.	Usuario	Repetir el paso 10 al 13, hasta colocar todas las entidades correspondientes.	Sí
14.	Usuario	Seleccionar el elemento <i>TablaRegistro</i> de la caja de herramientas y coloca la relación entre las entidades correspondientes.	Sí
15.	Herramienta	Muestra una ventana solicitando el nombre de la tabla de registro.	Sí
16.	Usuario	Ingresar el nombre de la tabla.	Sí
17.	Usuario	Repetir del paso 14 al 16, hasta agregar todas las relaciones entre las entidades.	Sí
18.	Usuario	Seleccionar el elemento <i>Mensaje</i> de la caja de herramientas y colocarlo en la tabla de registro (<i>TablaRegistro</i>).	Sí
19.	Herramienta	Muestra una ventana solicitando el mensaje.	Sí
20.	Usuario	Ingresar el mensaje.	Sí
21.	Usuario	Dar doble clic en el mensaje.	Sí
22.	Herramienta	Muestra una ventana, donde en la parte inferior derecha, se encuentra la opción <i>Tagged Values</i> .	Sí
23.	Usuario	Seleccionar la opción <i>Tagged Values</i> e ingresar los valores solicitados (identificador, proceso, dirección, origen y destino).	Sí
24.	Usuario	Repetir del paso 20 al 23 hasta añadir todos los mensajes que existen entre las entidades.	Sí
25.	Usuario	Guardar el diagrama de entidades a nivel usuario.	Sí

Los datos de ejemplo utilizados en las pruebas se muestran en la tabla 7.4

Tabla 7.4 Datos empleados en la prueba

Nombre del campo	Tipo	Valor
Entidad	Cadena	Jefe de organización bibliográfica
Mensajes	Cadena	2: Solicita préstamos. 1: Envía libros, tesis o revistas. 2: Enviar lista. 3: Envía libros.
RegistroMensajes	Cadena	Usuarios – JOB JOB – JCI

7.1.6. Reporte de Pruebas

En la tabla 7.5 se detallan los resultados generales de las pruebas, conclusiones y recomendaciones.

Tabla 7.5 Resultados obtenido de las pruebas

Requerimiento	Caso de Prueba	Resultado de la prueba
Crear el diagrama entidad a nivel usuario.	TC-01	Exitoso
Crear el diagrama entidad combinado a nivel usuario.	TC-02	Exitoso
Crear el diagrama entidad a nivel aplicación.	TC-03	Exitoso

En la tabla 7.6 se muestran los resultados de las pruebas realizadas a las funcionalidades generales del editor.

Tabla 7.6 Funcionalidades generales del editor

Funcionalidad	Resultado
1. Agregar un entidad	Exitoso
2. Nombrar un entidad	Exitoso
3. Agregar una relación <i>TablaRegistro</i>	Exitoso
4. Nombrar relación <i>TablaRegistro</i>	Exitoso
5. Agregar un mensaje	Exitoso
6. Agregar propiedades al mensaje	Exitoso

7.2. Resultados

En esta sección se presenta la ejecución del plan de pruebas y los resultados obtenidos para uno de los diagramas de entidades generado con el perfil UML propuesto.

Los resultados que se presentan son conforme a los casos de prueba planteados en el anexo *A.4. Pruebas*.

- Crear el diagrama entidad a nivel usuario.
- Crear el diagrama entidad combinado a nivel usuario.
- Crear el diagrama entidad a nivel aplicación.

En el *Anexo B. Caso de Estudio*, se describe el procedimiento general que se realiza para elaborar cualquiera de los diagramas de entidades mencionados anteriormente.

7.2.1. Creación de un diagrama de entidades

Siguiendo el procedimiento establecido en el *Anexo B. Caso de Estudio*, se presenta la generación de un diagrama de entidades a nivel usuario, empleado el perfil UML definido en *Enterprise Architect*. Antes de utilizar el perfil para desarrollar el diagrama de entidades, es necesario identificar los elementos que compondrán el diagrama, los cuales son:

- La entidad principal y las entidades secundarias con las que se relaciona.

- Relaciones que existen entre una entidad principal y una secundaria.
- Los procesos y mensajes que existen en cada relación.

7.2.1.1. Identificación de los elementos

En la figura 7.1 se muestra Diagrama de entidades a nivel “Jefa de organización bibliográfica” a nivel CIM (obtenido del *Reporte Técnico RT-DCC-1-2007*), el cual se realizará con el perfil UML para mapearlo como un diagrama de entidades a nivel usuario.

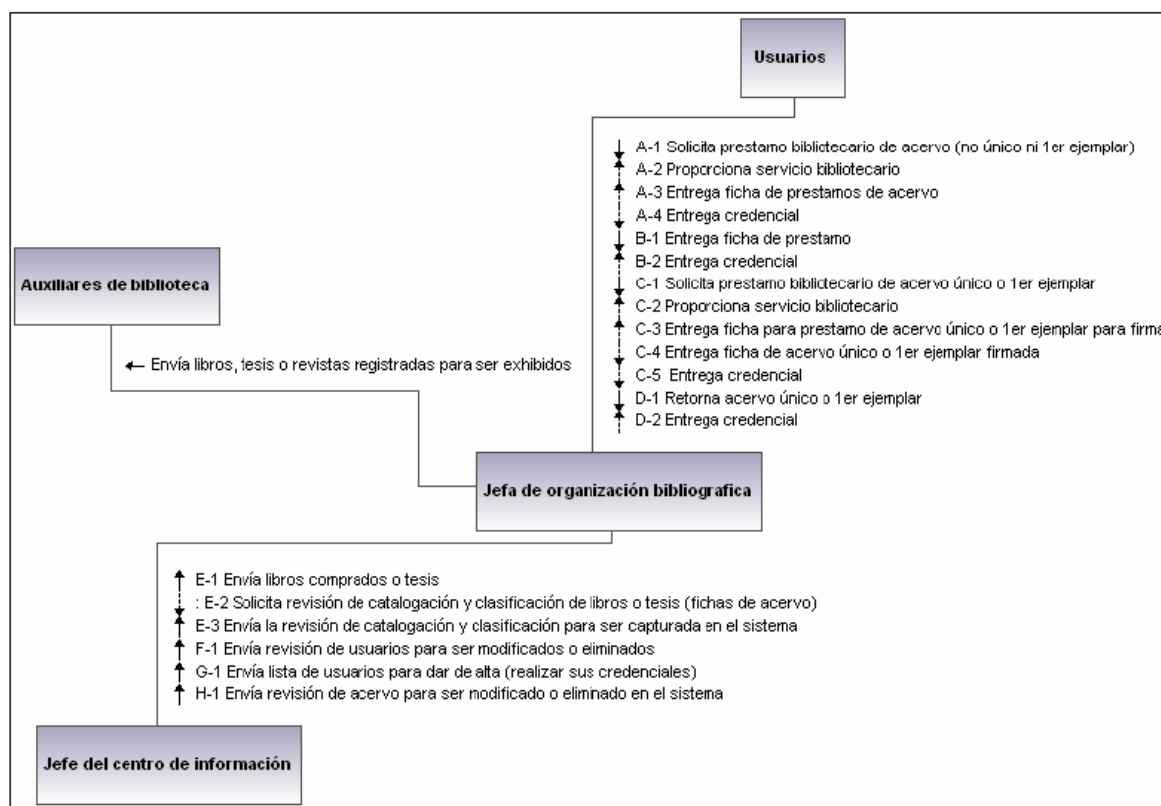


Figura 7.1 Diagrama de entidades a nivel “Jefa de organización bibliográfica” a nivel CIM

Para generar el diagrama de entidades a nivel usuario, es necesario identificar primero la entidad principal y las entidades secundarias, así como las relaciones y mensajes que existen entre ellas. De acuerdo al diagrama anterior (*Figura 7.1*), se identifican las entidades, a las cuales se les agrega un identificador que será utilizado más adelante (tabla 7.7).

Tabla 7.7 Entidades identificadas

Nombre de la entidad	Tipo de entidad	Identificador
Jefa de Organización Bibliográfica	Principal	JOB
Jefe de Centro de Información	Secundaria	JCI
Usuarios	Secundaria	U
Auxiliares de Biblioteca	Secundaria	AB

A su vez, los mensajes de recibido se muestran con el símbolo ←, y los mensajes de envío con el símbolo →. En las tablas siguientes se muestra un resumen de los mensajes y el proceso correspondiente de cada relación. El nombre del proceso se respeta conforme lo establecido en el diagrama original (tablas 7.8, 7.9 y 7.10). Sin embargo, el mensaje utilizado se corta para una mejor representación en el diagrama.

Tabla 7.8 Relación de Jefa de Organización Bibliográfica (JOB) y Auxiliares de Biblioteca (AB)

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
EE-1	Enviar artículos registrados	Envía libros, tesis o revistas registradas para ser exhibidas	JOB	AB

Tabla 7.9 Relación de Usuarios (U) y Auxiliares de Biblioteca (AB)

Proceso	Mensaje	Descripción de mensaje	Origen	Destino
A-1	Solicita préstamo	Solicita préstamo bibliotecario de acervo (no único ni 1er ejemplar)	U	AB
A-2	Proporciona servicio	Proporciona servicio bibliotecario	AB	U
A-3	Entrega ficha	Entrega ficha de préstamos de acervo	AB	U
A-4	Entrega credencial	Entrega credencial	U	AB

B-1	Entrega ficha de préstamo	Entrega ficha de préstamo	U	AB
B-2	Entrega credencial	Entrega credencial	AB	U
C-1	Solicita préstamo	Solicita préstamo bibliotecario de acervo único o 1er ejemplar	U	AB
C-2	Proporciona servicio	Proporciona servicio bibliotecario	AB	U
C-3	Entrega ficha	Entrega ficha para préstamo de acervo único o 1er ejemplar para firma	AB	U
C-4	Entrega ficha	Entrega ficha de acervo único o 1er ejemplar firmada	U	AB
C-5	Entrega	Entrega credencial	AB	U
D-1	Retorna acervo	Retorna acervo único o 1er ejemplar	U	AB
D-2	Entrega credencial	Entrega credencial	AB	U

Tabla 7.10 Relación de Jefa de Organización Bibliográfica (JCI) y Jefe del Centro de Información (JCI)

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
E-1	Envía libros	Envía libros comprados o tesis	JCI	JOB
E-2	Solicita revisión	Solicita revisión de catalogación y clasificación de libros.	JOB	JCI
E-3	Envía la revisión	Envía la revisión de catalogación y clasificación para ser capturada en el sistema	JCI	JOB
F-1	Envía revisión de usuarios	Envía revisión de usuarios para ser modificados	JCI	JOB

G-1	Envía lista de usuarios	Envía lista de usuarios para dar de alta (realizar sus credenciales)	JCI	JOB
H-1	Envía revisión de acervo	Envía revisión de acervo para ser modificado o eliminado en el sistema	JCI	JOB

7.2.1.2. Construcción

A continuación, se presenta paso a paso el desarrollo de un diagrama de entidades a nivel usuario utilizando el perfil UML previamente establecido en la herramienta *Enterprise Architect*. Los datos de entrada son los establecidos en el Anexo A.5. *Registro de Pruebas*.

1. Crear las entidades.

Para crear las entidades definidas en la tabla 7.7 se siguen los pasos del 13 al 16 establecidos en el procedimiento general. Cada entidad posee sus propias características, que se llenan conforme a los establecido en las tablas 7.8, 7.9 y 7.10 que se encuentran en la sección 7.2.1.1. *Identificación de los elementos*.

Las entidades se agregan en el área de trabajo, seleccionando el elemento *Entidad* del panel *Toolbox*, como se muestra en la figura 7.2.

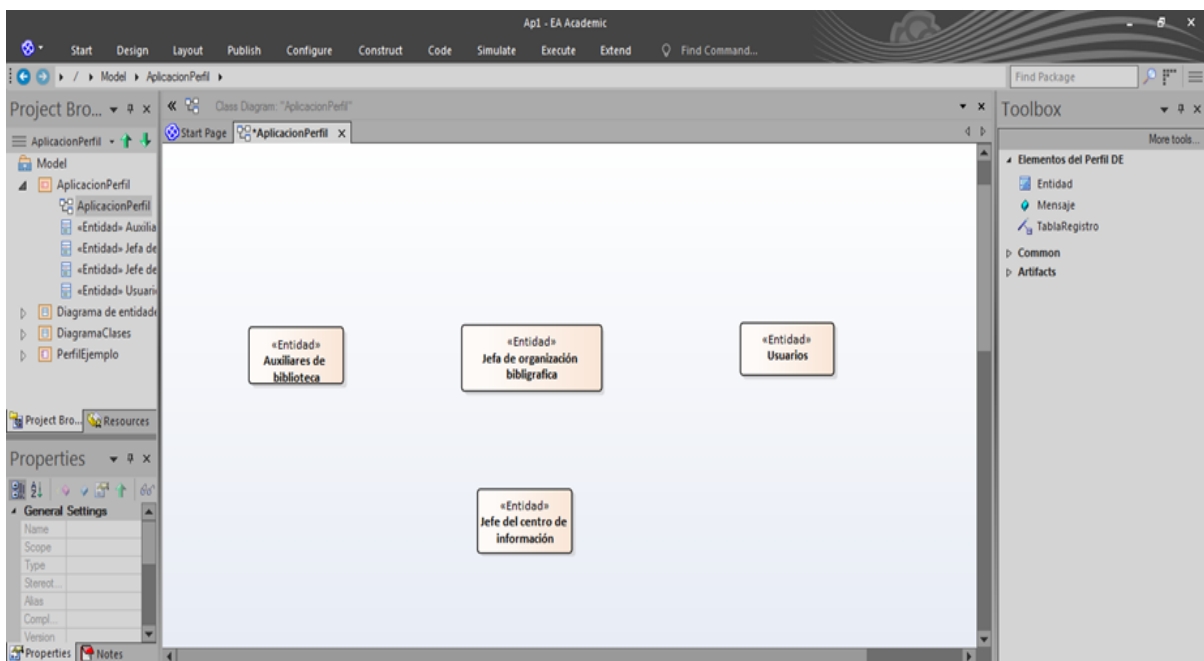


Figura 7.2 Agregar Entidades

En la figura 7.3 y 7.4 se presentan la definición del tipo de entidad que se debe establecer en las propiedades.

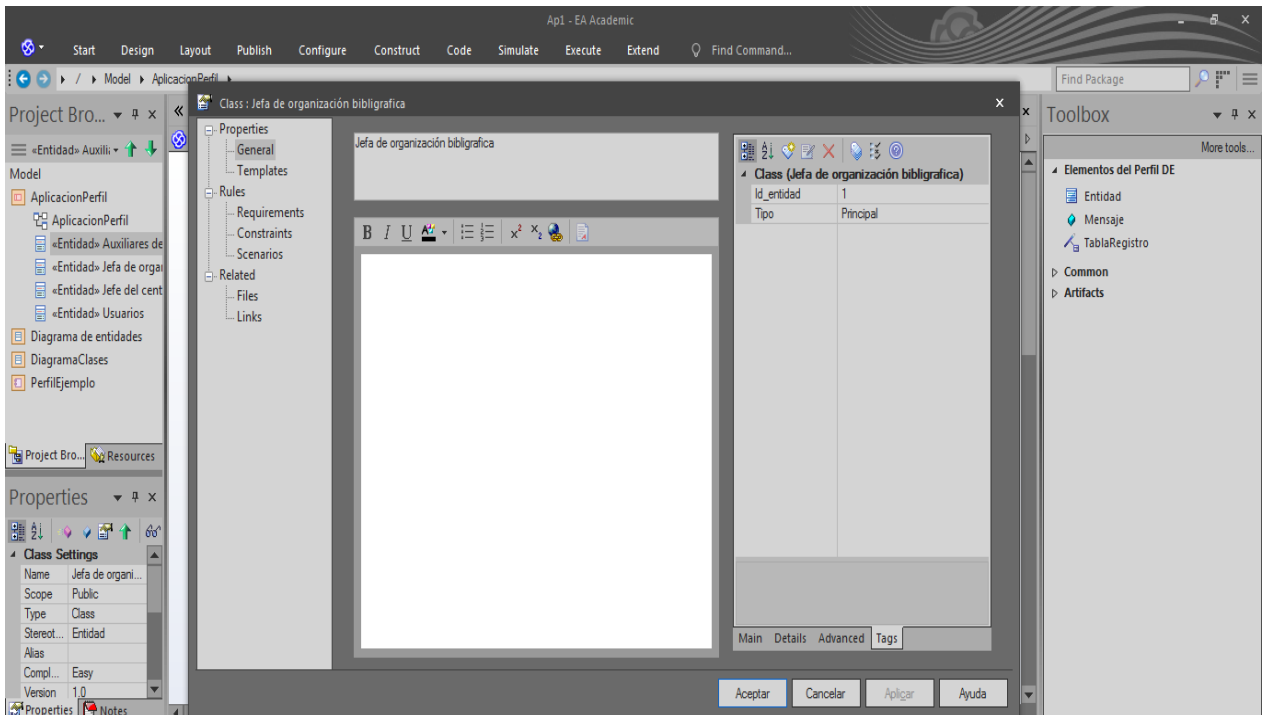


Figura 7.3 Definición de una entidad principal

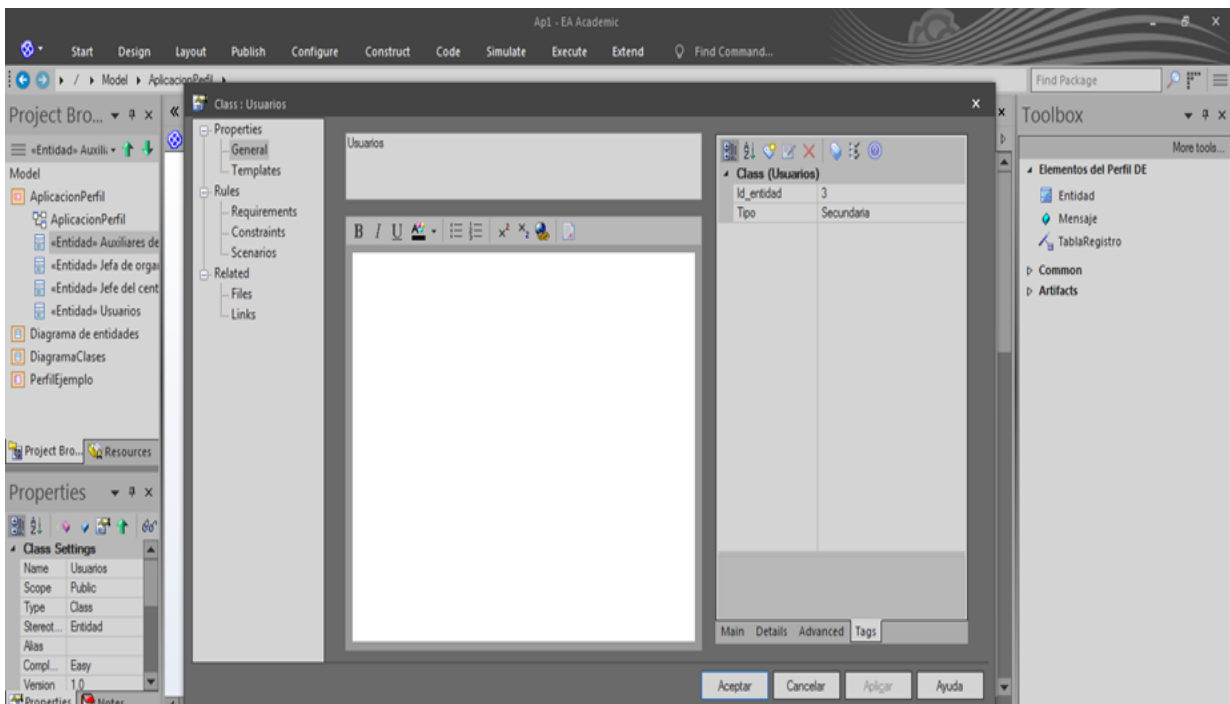


Figura 7.4 Definición de una entidad secundaria

2. Crear relaciones

Para agregar las relaciones entre entidades se selecciona el elemento *RegistroMensaje* de la caja de herramientas (ver pasos de 17 al 19 en la sección 5.3.2. *Pasos para construir un diagrama de entidades en Enterprise Architect*). Cada elemento *RegistroMensaje* que exista en el diagrama debe contener un nombre que indiqué la relación entre dos entidades como se puede observar en la figura 7.5.

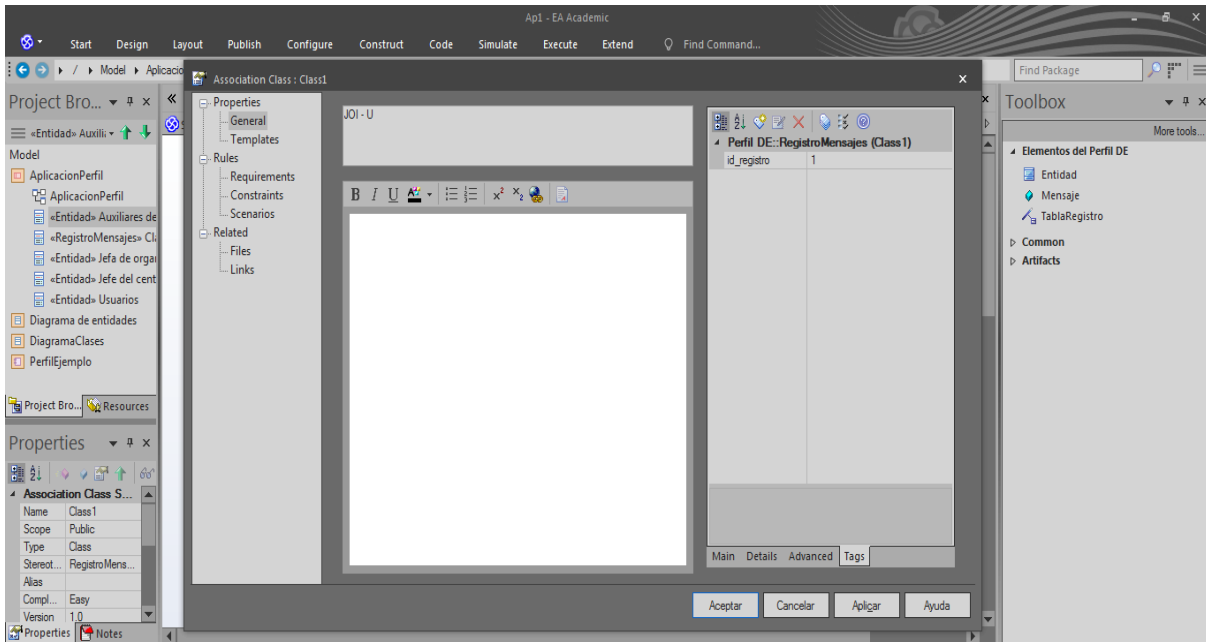


Figura 7.5 Nombre de la relación

En la figura 7.6 se muestra el diagrama con todas las relaciones que existen entre las entidades.

Los nombres de las relaciones se definen en las tablas 7.8, 7.9 y 7.10.

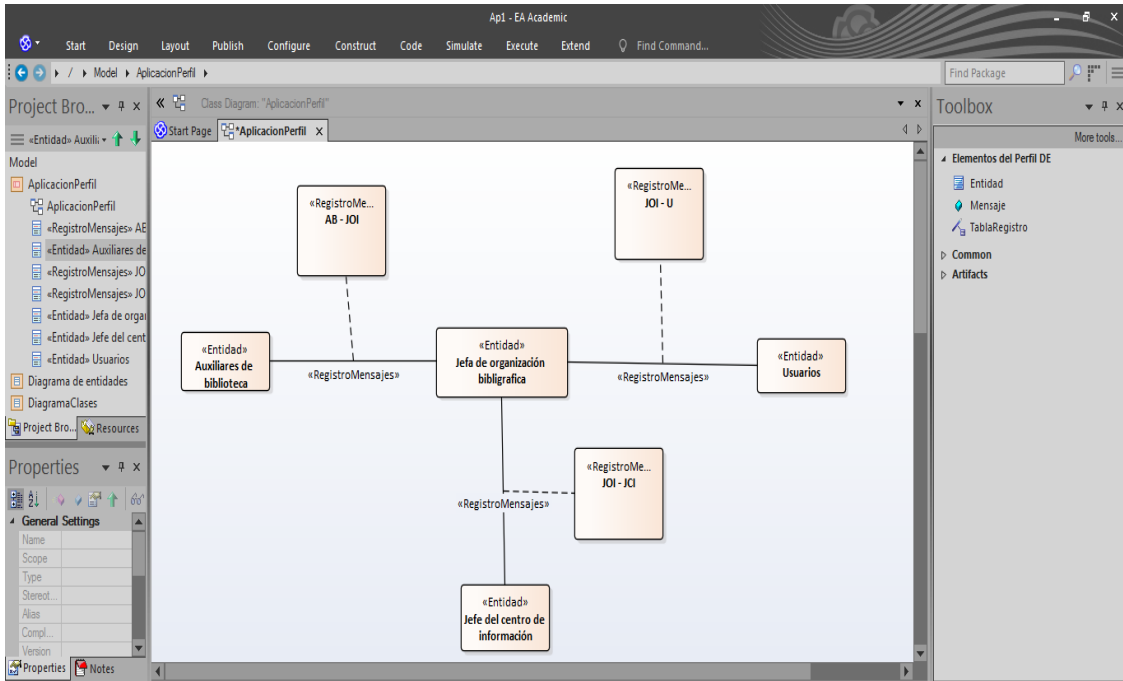


Figura 7.6 Relaciones entre entidades

3. Crear mensajes

En la caja de herramientas se selecciona el elemento Mensaje, y se agrega al elemento *RegistroMensaje* (repetir pasos del 21 al 25 del proceso general). Para cada mensaje se debe agregar: *id_mensaje*, destino, origen, dirección (envió o recibido) y proceso (figura 7.7).

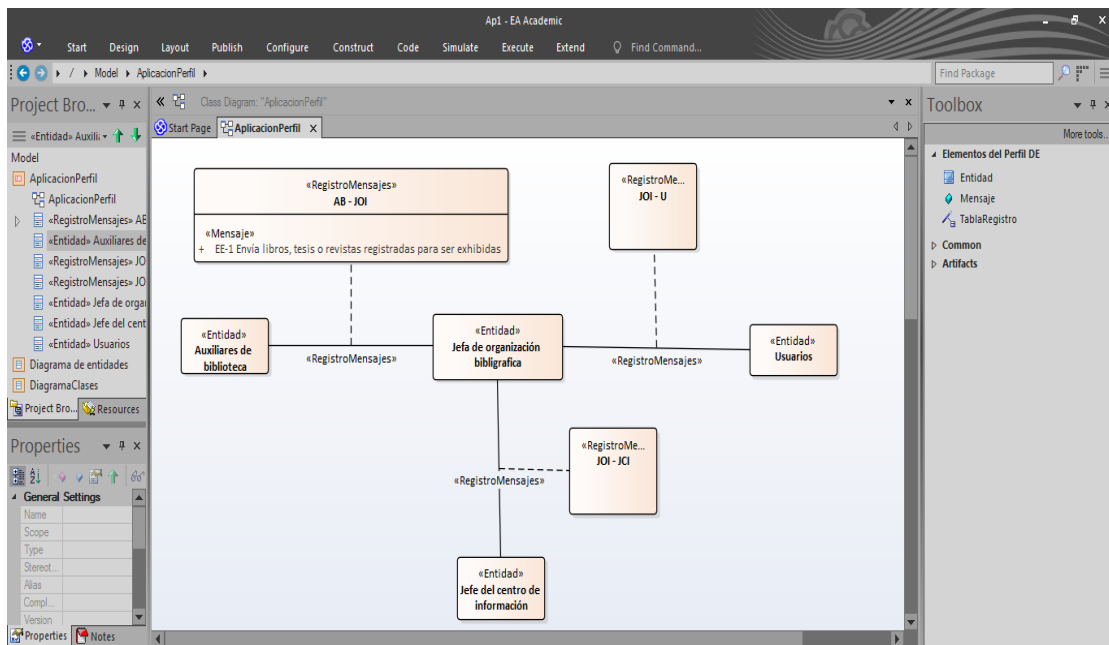


Figura 7.7 Agregar mensaje

Es recomendable que los mensajes sean cortos para una mejor representación en el diagrama, como lo establecido en las tablas de relación (figura 7.8).

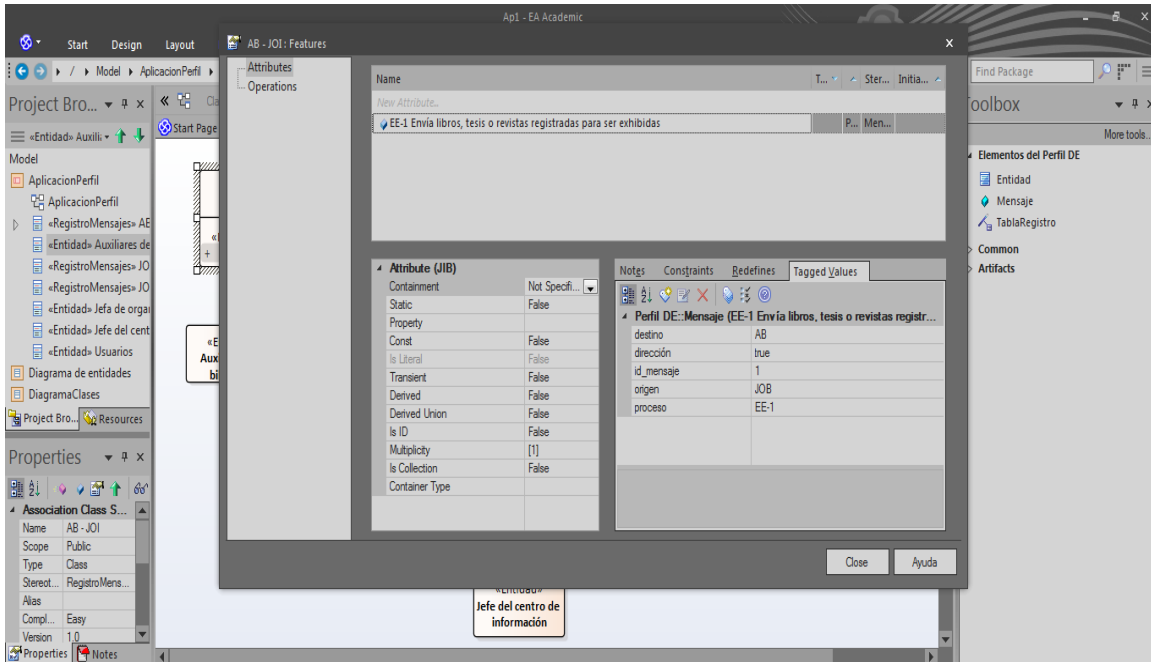


Figura 7.8 Propiedades del mensaje

7.2.1.3. Salida

El resultado obtenido es el diagrama de entidades a nivel usuario desarrollado con el perfil UML (ver figura 7.9).

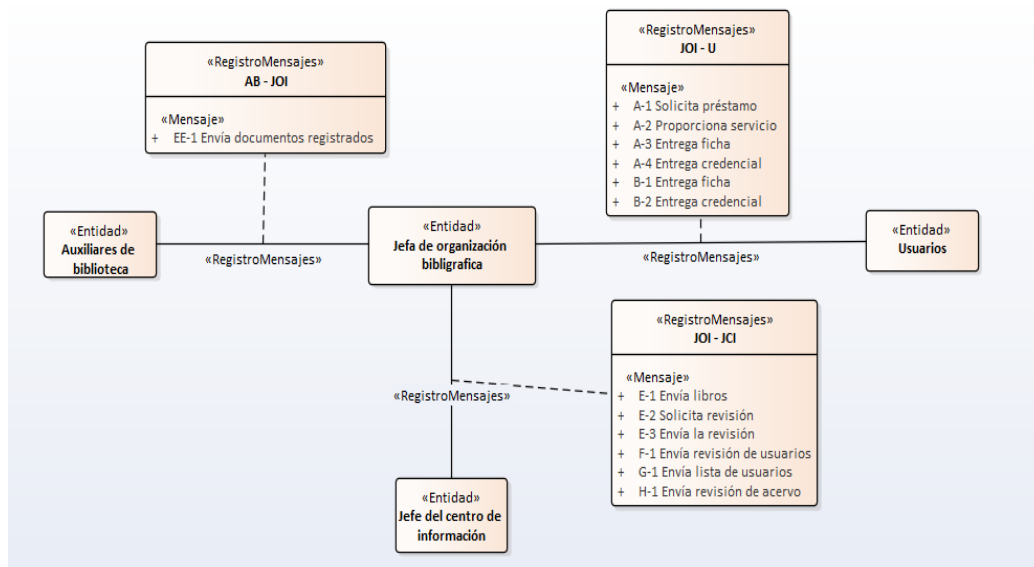


Figura 7.9 Diagrama de entidades a nivel usuario

En esta investigación, para generar la herramienta de edición del DSL: Diagramas de entidad, se analizaron las características fundamentales de los diagramas de entidades con el objetivo de diseñar un perfil UML, que sería la base para el editor, esto se expone el *capítulo 5. Creación del perfil UML para el DSL: diagramas de entidad y generación de herramienta de edición*.

El perfil UML que se desarrolló se implementó en la herramienta de modelado *Enterprise Architect*, seleccionada por los mecanismos con los que cuenta para la creación e instanciación de perfiles UML. El resultado obtenido de las pruebas realizadas a la herramienta de edición, son los requerimientos funcionales que se mencionan en la sección 7.1.2. Diseño de pruebas, y que resultaron exitosos en cada prueba. En el capítulo siguiente se plasman las conclusiones obtenidas después de realizar esta investigación, así como los trabajos futuros.

CAPÍTULO 8. CONCLUSIONES

En este capítulo se exponen las conclusiones obtenidas como resultado de la presente investigación. También se incluyen las aportaciones y los trabajos futuros.

8.1. Introducción

Se dificultó encontrar una herramienta de modelado que permitiera la generación de diagramas de dominio específico, a partir de perfiles UML. En algunos casos, no se contaban con las librerías actualizadas o con las opciones adecuadas para crear un editor propio, e inclusive la información estaba desactualizada.

Con la presente investigación se logró definir e instanciar un perfil UML para la construcción de diagramas de entidades, así como el uso de un editor propio para estos diagramas, mediante la herramienta de modelado *Enterprise Architect*. Contar con una herramienta de edición ayudará a generar los diagramas de entidades y definir mejor un modelo de dominio del problema para comprender el espacio del problema en el que se encuentra ubicado el dominio de aplicación. Además, ayudará a personas inexpertas en un dominio en particular a generar modelos, que permitan representar la problemática que se desea resolver.

8.2. Aportaciones

El proporcionar un perfil UML para la construcción de diagramas de entidades, y así como haber identificado una herramienta de modelado que permite obtener un editor mediante este perfil UML ayuda a los desarrolladores a definir un modelo de dominio del problema sin requerir la intervención constante de los expertos del dominio

Las aportaciones generadas en esta investigación son:

- Un proceso para la definición de un perfil UML, que incluye la identificación de las meta-clases a extender, los estereotipos, los valores etiquetados y las reglas y restricciones que son necesarias.
- Un documento de diseño e implementación del perfil UML que sirve como pauta para definir un editor para los diagramas de entidades.
- Un documento de los requerimientos para realizar el proceso de transformación de los diagramas de entidades.

8.3. Trabajos futuros

Esta investigación es una primera aproximación para la generación de modelos de dominio del problema mediante un perfil UML, haciendo uso de los diagramas de entidades propuestos por Orr. Solo se generó un perfil UML para realizar un editor que permitió la construcción de los diagramas de entidades, que ayuda a obtener una primera representación de las entidades y transacciones que realizan en el dominio del problema. Pero se requiere representar adecuadamente los procesos involucrados e implementar las transformaciones entre diagramas. Por lo tanto, como trabajos futuros se proponen los siguientes:

- Complementar el proceso de los diagramas para obtener un modelo del dominio del problema, que represente los procesos involucrados (diagrama de línea de producción).
- Definir y automatizar las transformaciones para obtener los diagramas de entidades combinados, de aplicación y el diagrama de línea de producción.

Referencias

- Sparx Systems Pty Ltd. (2017). *Sparx Systems*. Obtenido de http://www.sparxsystems.com.au/resources/developers/uml_profiles.html
- Acevedo, R. V., Gutiérrez, Y. G., Silva, C. V., & Gallardo, A. M. (2016). Un lenguaje de dominio específico con aspectos de seguridad para el modelado de almacenes de datos. *VII Taller Internacional de Calidad en las Tecnologías de la Información y las Comunicaciones Calidad 2016*.
- Albores Villatoro, L. A. (Agosto de 2007). Definición e implementación de un perfil de UML para la adquisición de requerimientos funcionales centrados en el usuario. Cuernavaca, Morelos, México: CENIDET.
- Altova. (2017). *ALTOVA*. Obtenido de <https://www.altova.com/es/umodel/profile-diagrams.html>
- Andreas Bender, S. B., & Kondov, I. (2014). An EMF-based Toolkit for Creation of Domain-specific Data Services. *Model-Driven Engineering and Software Development (MODELSWARD), 2014 2nd International Conference on*, 30-40.
- Andreas, B., Bozic, S., & Kondov, I. (2014). An EMF-based Toolkit for Creation of Domain-specific Data Services. *Model-Driven Engineering and Software Development (MODELSWARD), 2014 2nd International Conference on*, 30-40.
- Autili, M., Bertolino, A., Angelis, G. D., Ruscio, D. D., & Sandro, A. D. (2016). A Tool-Supported Methodology for Validation and Refinement of Early-Stage Domain Models. *IEEE Transactions on Software Engineering*, 2-25.
- Booch, G., Rumbaugh, J., & Jacobson, I. (2000). *El Lenguaje Unificado de Modelado*. Madrid: Addison Wesley.
- Boone, J. (3 de noviembre de 2014). *Cunningham & Cunningham, Inc*. Obtenido de <http://c2.com/cgi/wiki?ProblemDomain>
- Czarnecki, K., & Eisenecker, U. W. (2000). *Generative Programming—Methods, Tools, and Applications*. Addison-Wesley.
- Eclipse. (2016). *Eclipse documentation*. Obtenido de <http://help.eclipse.org/neon/index.jsp>
- Fuentes, L., & Antonio, V. (Marzo de 2004). Una Introducción a los Perfiles UML. España: Depto. de Lenguajes y Ciencias de la Computación, Universidad de Málaga. Obtenido de Depto. de Lenguajes y Ciencias de la Computación, Universidad de Málaga.
- Gérard, S. (25 de noviembre de 2011). *PAPYRUS USER GUIDE SERIES About UML profiling, version 1.0*. Obtenido de <https://eclipse.org/papyrus/documentation.html>
- González, G. M. (Agosto de 2006). Método de Desarrollo Arquitectónico en Grupo. D.F., México: Instituto Politécnico Nacional.

- Gronback, R. (2009). *Eclipse Modeling Project, A Domain-Specific Language Toolkit*. Addison-Wesley.
- Larman, C. (2003). *UML y Patronos*. Madrid: Prentice Hall.
- López, L. E. (noviembre de 2007). Caracterización de las fronteras de los modelos MDA mediante UML. Cuernavaca, Morelos, México: CENIDET.
- Ltd., S. S. (2019). *Sparx Systems*. Obtenido de https://sparxsystems.com/enterprise_architect_user_guide/14.0/modeling_tools/mdg_technologies.html
- MKLab. (2014). *StarUML 2 Documentación*. Obtenido de <http://docs.staruml.io>
- Object Management Group. (27 de Mayo de 2015). *Object Management Group. MDA Guide rev. 1.0*, de <http://www.omg.org/mda/>
- OMG. (mayo de 2006). Object Constraint Language (OCL) Versión 2.0. OMG Available Specification.
- OMG. (2019). Obtenido de The Object Management Group (OMG): <https://www.omg.org/>
- Ormeño, E., Lund, M., Aballay, L., & Aciar, S. (2012). An UML profile for modeling RESTful services. San Juan, Argentina: National University of San Juan.
- Orr, K. T. (1981). *Structured Requirements Definition*. Topeka: Orr Ken&Associates Inc. .
- Ricardo, S., & Nibaldo, R. (2005). New UML 2.0 based models to design WAP. *Seventh International Conference on UML Modeling Languages and Applications*.
- Rossel, P. O., Bastarrica, M. C., Hirschfeld-Kahler, N., Díaz, V., & Medina, M. (2014). Domain modeling as a basis for building a meshing tool software product line. *Advances in Engineering Software*, 77-89.
- Sparx Systems. (2016). *Sparx Systems*. Obtenido de <http://www.sparxsystems.com.ar/EAUserGuide/ea.html?addingelementsandmetaclass.htm>
- Taylor, R. N., Medvidovic, N., & Dashofy, E. M. (2008). *Software Architecture: Foundations, Theory, and Practice*. John Wiley & Sons.

ANEXO A. PLAN DE PRUEBAS

A.1. Plan de pruebas

A.1.1. Introducción

En el presente documento se expone el plan de pruebas a cubrir para asegurar el correcto funcionamiento del perfil UML resultado de la presente investigación, para obtener modelos de dominio. Se define el alcance del plan de pruebas, detalles del plan de pruebas, los criterios de aceptación, suspensión y reanudación; y se nombra los elementos que se toman en cuenta para este plan de pruebas.

A.1.1.1. Identificador del Documento

A continuación, en la Tabla A.1 se lista la nomenclatura utilizada para identificar los documentos en la etapa de pruebas.

Tabla A.1 Nomenclatura para identificar los documentos generados

Nomenclatura	Identificador	Descripción
AB	Tipo de documento	TP-Plan de pruebas (Test Plan). TD-Diseño de Pruebas (Test Desing). TC- Casos de Prueba (Test Case). TPr –Procedimiento de Pruebas (Test Log). TL – Registro de Pruebas (Test Report). TR – Reporte de Pruebas (Test Report).
NM	Numeración	Representa el número correspondiente al documento.

A.1.1.2. Alcance

Verificar que el perfil UML desarrollado en la presente investigación, genere el diagrama entidad a nivel usuario, el diagrama de entidad combinado a nivel usuario y el diagrama de entidad a nivel aplicación.

A.1.1.3. Referencias

IEE Computer Society. (2008) 829-2008-IEEE Estándar for Software and System Test Documentation.

A.1.2. Detalles del plan de pruebas del perfil UML

A.1.2.1. Funcionalidad del perfil

El plan de pruebas consiste en evidenciar el perfil UML desarrollado en la presente investigación, para la creación de los siguientes diagramas:

- **Crear el diagrama entidad a nivel usuario**

Colocar el elemento entidad, el elemento registro-mensaje y el elemento mensaje en el área de trabajo del software EA. A cada elemento se le asigna un nombre y sus propiedades correspondientes.

- **Crear el diagrama entidad combinado**

Colocar el elemento entidad, el elemento registro-mensaje y el elemento mensaje en el área de trabajo del software EA. A cada elemento se le asigna un nombre y sus propiedades correspondientes. Para este diagrama, el tipo de entidad debe ser: combinado.

- **Crear el diagrama entidad a nivel aplicación**

Colocar el elemento entidad, el elemento registro-mensaje y el elemento mensaje en el área de trabajo del software EA. A cada elemento se le asigna un nombre y sus propiedades correspondientes. Para este diagrama, los tipos de entidad deben ser: principal y secundaria, considerando que posee las mismas características que el diagrama de entidades a nivel usuario.

A.1.2.2. Matriz de trazabilidad de pruebas

Se provee las funcionalidades que se establecen para el perfil UML y el identificador del caso de prueba para cada requerimiento (ver tabla A.2).

Tabla A.2 Matriz de trazabilidad de prueba

Funcionalidad	Caso de Prueba
Crear el diagrama entidad a nivel usuario.	TC-01

Crear el diagrama entidad combinado a nivel usuario.	TC-02
Crear el diagrama entidad a nivel aplicación.	TC-03

A.1.2.3. Criterio de aceptación

Los casos de prueba son aceptados si cumplen con cada restricción especificada a partir del perfil UML y la metodología AGD.

A.1.2.4. Criterio para Suspensión y Reanudación de Pruebas

La suspensión de las pruebas se rige por la presencia de fallas en los elementos que impidan el cumplimiento de las restricciones. Al registrarse una falla se suspende la prueba y se reanuda al ser corregida.

A.1.2.5. Entregables

El resultado de las actividades correspondientes a las pruebas es la producción de los siguientes entregables:

1. Plan de Pruebas.
2. Diseño de Pruebas.
3. Casos de Pruebas.
4. Procedimiento de Pruebas.
5. Registro de Pruebas.
6. Reporte de Pruebas.

A.1.3. Administración de Pruebas

En esta sección se define las tareas de prueba a realizar. También se define la infraestructura utilizada para la etapa de pruebas.

A.1.3.1. Actividades Planificadas

Las actividades que son necesarias para preparar y realizar las pruebas son:

- Generación del plan de pruebas.
- Diseño de casos de pruebas.
- Ejecución de casos de pruebas.

- Generación de reporte de pruebas.

Las actividades son ejecutadas por el autor del proyecto.

A.1.3.2. Infraestructura

La infraestructura para las pruebas se especifica a continuación:

- Software:
 - Enterprise Architec.
 - Oracle VM VirtualBox.
 - La máquina virtual con las siguientes características: Sistema operativo Windows 7, memoria RAM 2 G.
- Hardware:
 - *Procesador:* AMD A10-8700P Radeon R6, 10 Compute Cores 4C+6G 1.80 GHz.
 - *Memoria RAM:* 16 .00 GB.
 - *Disco Duro:* 1 T.
 - *Sistema operativo:* Windows 10 de 64 bits.

A.2. Diseño de pruebas

A.2.1. Introducción

En el presente documento se identifican las características a probar, se determina la estrategia para realizar las pruebas y el criterio de aceptación de las pruebas.

A.2.1.1. Identificador de documento

El identificador del documento pactado en la nomenclatura especificada en el Plan de Pruebas es: *TD-01*.

A.2.1.2. Alcance

Enfocar las características que permiten el cumplimiento de la creación de los diagramas de acuerdo a cada restricción especificada a partir del perfil UML y la metodología AGD. Las pruebas utilizan el criterio de inspección visual para corroborar la existencia de los registros esperados.

A.2.1.3. Referencias

IEEE Computer Society. (2008) 829-2008-IEEE Standard for Software and System Test Documentation.

A.2.2. Detalles del diseño de la prueba

Esta sección describe las características a probar, los casos de pruebas, los criterios de aceptación y entregables.

A.2.2.1. Características a probar

El diseño de las pruebas engloba las características referentes a los siguientes diagramas a crear: el diagrama entidad a nivel usuario, el diagrama entidad combinado a nivel usuario y el diagrama entidad a nivel aplicación.

A continuación, se muestra en la tabla A.3 la matriz de trazabilidad de pruebas, en la cual se describen las características a probar, el nombre del caso y procedimiento de prueba correspondiente a cada requerimiento funcional.

Tabla A.3 Matriz de trazabilidad

Requerimiento	Funcionalidad	Caso de prueba	Procedimiento
1. Crear el diagrama entidad a nivel usuario	<ol style="list-style-type: none">Colocar al menos dos elementos entidad.Asignar nombre a cada elemento entidad.Asignar un identificador a cada elemento.Seleccionar el tipo de entidad (principal y secundaria).Establecer una relación entre las entidades.	TC-01	TPr-01

	<p>6. Agregar los mensajes correspondientes.</p> <p>7. Asignar a cada mensaje un identificador.</p> <p>8. Asignar el proceso al cual pertenece el mensaje.</p> <p>9. Definir el tipo de dirección (0,1).</p> <p>10. Especificar el origen y el destino del mensaje.</p>		
<p>2.Crear el diagrama de entidad combinado a nivel usuario</p>	<p>1. Colocar al menos dos elementos entidad.</p> <p>2. Asignar nombre a cada elemento entidad.</p> <p>3. Asignar un identificador a cada elemento.</p> <p>4. Seleccionar el tipo de entidad (combinado).</p> <p>5. Establecer una relación entre las entidades.</p> <p>6. Agregar los mensajes correspondientes.</p> <p>7. Asignar a cada mensaje un identificador.</p> <p>8. Asignar el proceso al cual pertenece el mensaje.</p> <p>9. Definir el tipo de dirección (0,1).</p> <p>10. Especificar el origen y el destino del mensaje.</p>	TC-02	TPr-02

<p>3. Crear el diagrama de entidad a nivel aplicación</p>	<ol style="list-style-type: none"> 1. Colocar al menos dos elementos entidad. 2. Asignar nombre a cada elemento entidad. 3. Asignar un identificador a cada elemento. 4. Seleccionar el tipo de entidad (principal o secundaria). 5. Establecer una relación entre las entidades. 6. Agregar los mensajes correspondientes. 7. Asignar a cada mensaje un identificador. 8. Asignar el proceso al cual pertenece el mensaje. 9. Definir el tipo de dirección (0,1). 10. Especificar el origen y el destino del mensaje. 	<p>TC-03</p>	<p>TPr-03</p>
---	--	--------------	---------------

A.2.2.2. Estrategia de pruebas

La realización de las pruebas será de la siguiente manera: en cada etapa se realiza varias actividades para probar la creación de los diagramas establecidos. Se hará uso de tres diagramas elaborados en el reporte técnico de la M.C. Edna Daniela López López como caso de estudio para comprobar actividad establecida. La Tabla 2 expone las pruebas a realizar.

Tabla A.4 Pruebas a realizar

Prueba	Etapas	Descripción
Crear diagrama entidad a nivel usuario	Crear el diagrama entidad a nivel usuario	Verificar que el diagrama cuenta con los elementos necesarios apegando al siguiente diagrama: Diagrama de comunicación a nivel "Jefa de organización bibliográfica" a nivel CIM.
2.Crear diagrama entidad combinado a nivel usuario	Crear el diagrama entidad combinado a nivel usuario	Verificar que el diagrama cuenta con los elementos necesarios apegando al siguiente diagrama: Diagrama de comunicación combinado de todos los usuarios a nivel CIM.
3.Crear diagrama entidad a nivel aplicación	Crear el diagrama entidad a nivel aplicación	Verificar que el diagrama cuenta con los elementos necesarios apegando al siguiente diagrama: Diagrama de comunicación a nivel aplicación a nivel CIM.

A.2.2.3. Criterio

Las pruebas serán aceptadas si cumplen con cada una de las funcionalidades especificadas en los requerimientos sin ningún contratiempo.

A.2.2.4 Entregables de pruebas

Los entregables resultados de las actividades de pruebas son las siguientes:

1. Plan de Pruebas
2. Diseño de Pruebas
3. Casos de Pruebas
4. Procedimiento de Pruebas
5. Registro de Pruebas
6. Reporte de Pruebas

A.3. Casos de pruebas

A.3.1. Introducción

En este documento se especifica el avance del presente documento, la entrada y salida respectiva de cada caso de prueba acorde al requerimiento expuesto en los puntos posteriores.

A.3.1.1. Alcance

Enfocar la realización de los casos de prueba especificados en el documento de diseño de pruebas, cumpliendo con los siguientes requerimientos: generar el diagrama entidad a nivel usuario, generar el diagrama entidad combinado a nivel usuario y generar el diagrama entidad a nivel aplicación.

A.3.1.1. Referencias

IEEE Computer Society. (2008) 829-2008 – IEEE Standard for Software and System Test Documentation.

A.3.2. Casos de Prueba

En la tabla A.5. se enlistan los requerimientos aprobar en el presente documento y su correspondiente identificador de caso de prueba.

Tabla A.5 Casos de Prueba

Identificador del caso de prueba	Requerimiento
TC-01	Crear el diagrama de entidad a nivel usuario.
TC-02	Crear el diagrama de entidad combinado a nivel usuario
TC-03	Crear el diagrama de entidad a nivel aplicación.

En cada caso de prueba se describirán los siguientes puntos:

- Objetivo.
- Entrada del caso.

- Salida del caso.

A.3.2.1. Caso de Prueba TC-01

En esta prueba se presentará la generación de un diagrama de entidades a nivel usuario.

- **Objetivo**

Verificar el cumplimiento de la creación del diagrama de entidades a nivel usuario.

- **Entrada**

Se describen a continuación los datos de entrada para cada elemento que compondrán el diagrama.

Para el elemento *Entidad* se requiere:

- El identificador de entidad.
- El nombre de la entidad.
- El tipo de entidad.

Para el elemento *Mensaje*:

- El identificador del mensaje.
- El proceso que corresponde a un mensaje.
- El mensaje que se envía de una entidad a otra.
- La dirección del mensaje.
- La entidad origen del mensaje.
- La entidad destino del mensaje.

Para el elemento *RegistroMensaje* es necesario:

- El identificador del registro.
- El nombre del registro.

- **Salida**

La salida es la generación del diagrama de manera satisfactoria, como se muestra en la figura A.1.

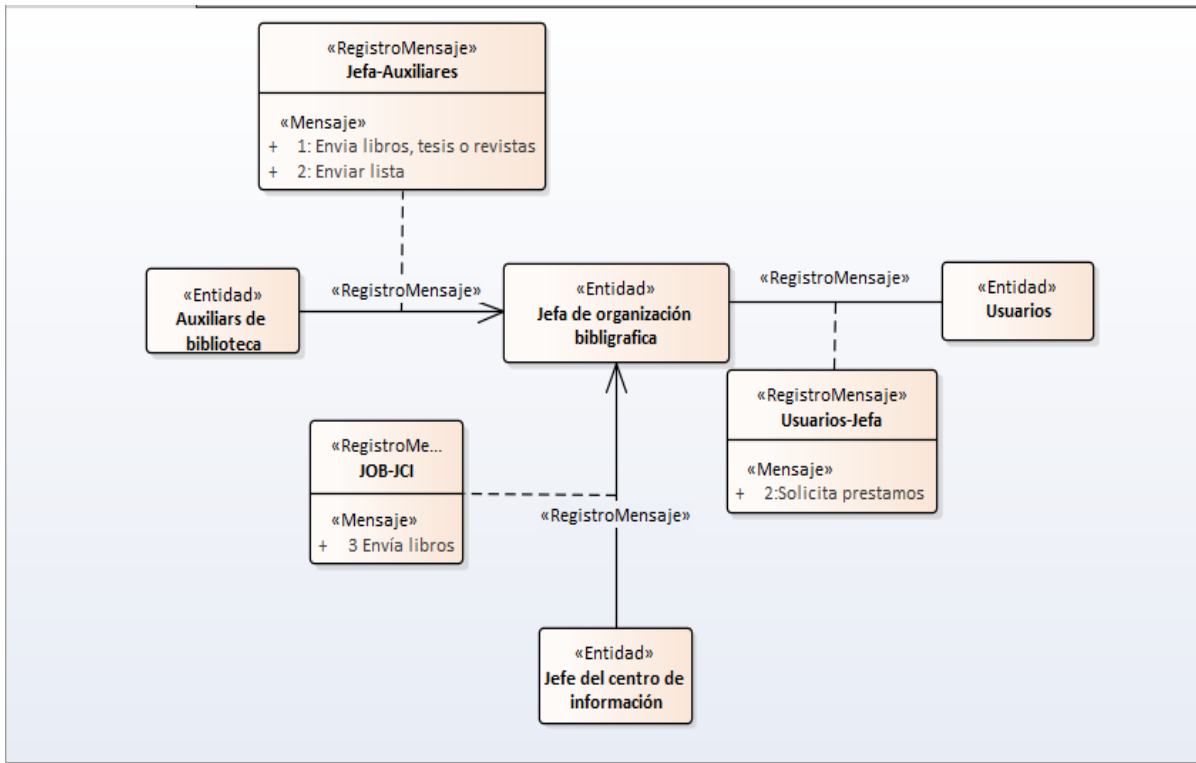


Figura A.1 Diagrama de entidades a nivel usuario

A.3.2.2. Caso de Prueba TC-02

En esta prueba se presentará la generación de un diagrama de entidades combinado a nivel usuario.

▪ Objetivo

Verificar el cumplimiento de la creación del diagrama de entidades combinado a nivel usuario.

▪ Entrada

Los elementos de entrada que se requieren para construir el segundo diagrama de entidades son:

- Entidad
- Mensaje
- RegistroMensaje

Cada elemento posee características propias, que son necesarias para completar el diagrama. Estas se describen en el caso de prueba TC-01 en la sección A.3.2.1. *Caso de Prueba TC-01*, dado que ambos diagramas utilizan los mismos elementos para su construcción, solo cambian las reglas de construcción.

▪ **Salida**

En la figura A.2 se muestra el diagrama de entidades combinado a nivel usuario.

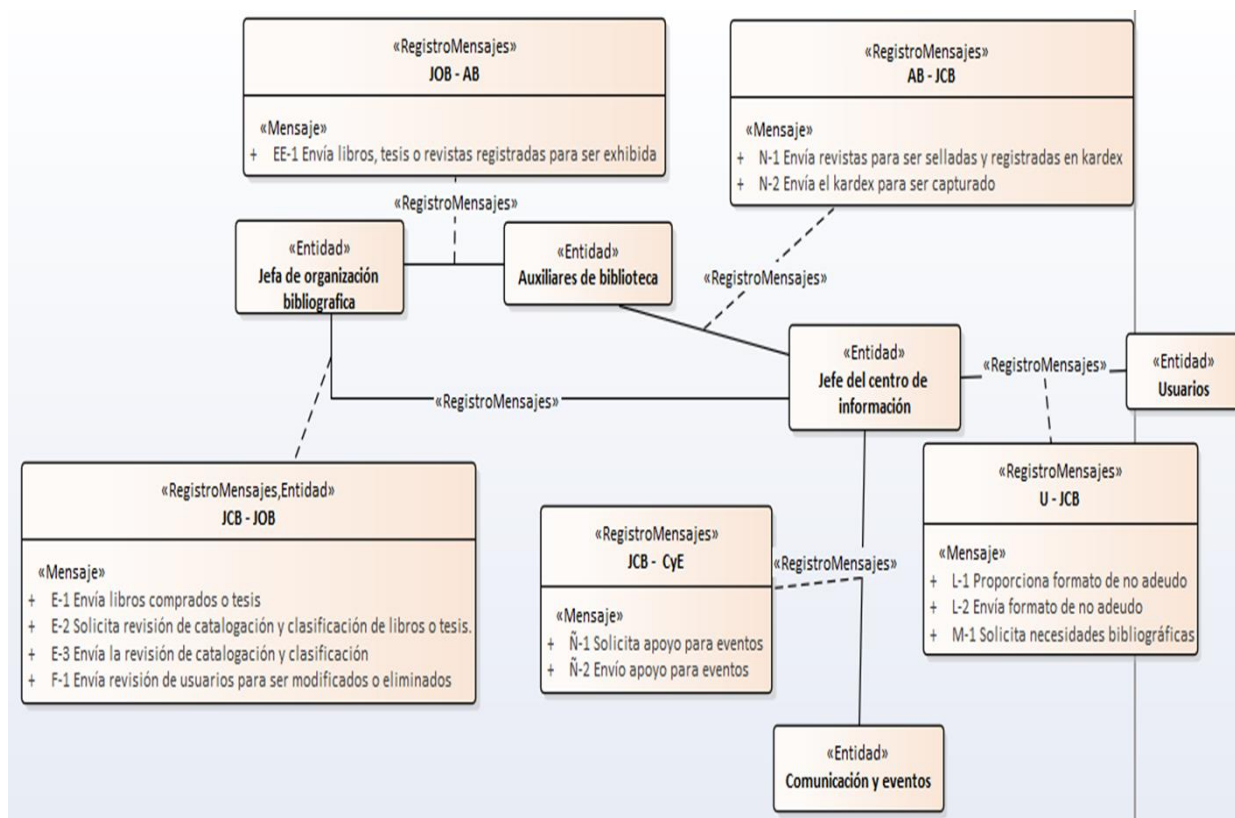


Figura A.2 Diagrama de entidades combinado a nivel usuario

A.3.2.3. Caso de Prueba TC-03

En esta prueba se presentará la generación de un diagrama de entidades a nivel aplicación.

▪ **Objetivo**

Verificar el cumplimiento de la creación del diagrama de entidades a nivel aplicación.

▪ **Entrada**

El diagrama de entidades a nivel aplicación pose características similares a un diagrama de entidades a nivel usuario, las únicas diferencias son las reglas que se aplican al construirlos, y que en un diagrama de entidades a nivel aplicación, solo se muestran las entidades externas que no se incluyen en la aplicación final.

Sus elementos entrada son *Entidad*, *Mensaje* y *RegistroMensaje*. Estos elementos se describen en el caso de prueba TC-02 en la sección A.3.2.1. *Caso de Prueba TC-01*, en el cual también se presentan cada una de sus propiedades.

▪ **Salida**

En la figura A.3 se muestra el diagrama de entidades combinado a nivel usuario.

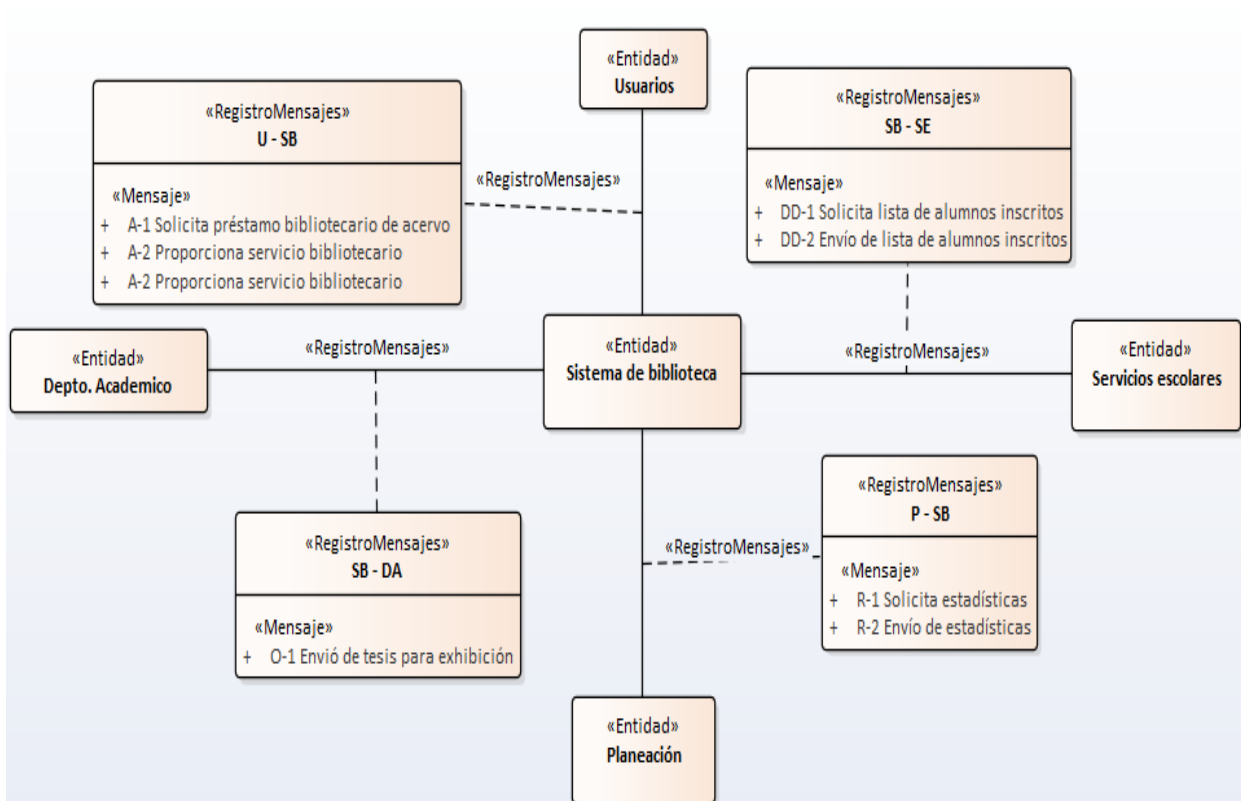


Figura A.3 Diagrama de entidades a nivel aplicación

A.4. Procedimiento de pruebas

A.4.1. Introducción

En el presente documento se describen los pasos a ejecutar para examinar el conjunto de pruebas especificadas en el documento Casos de prueba.

A.4.1.1. Alcance

Describir el procedimiento de los casos de prueba: TC-01, TC-02 y TC-03.

A.4.1.2. Referencias

IEEE Computer Society. (2008) 829-2008 – IEEE Standard for Software and System Test Documentation.

A.4.2. Procedimiento de Prueba TPr-01

En esta sección se mencionan los datos que se requieren como entrada y el procedimiento para realizar el caso de prueba TC-01.

A.4.2.1. Entradas, salidas y requerimientos especiales

Documentos:

- Plan de pruebas.
- Diseño de pruebas.

Aplicaciones:

- Enterprise Architect.

Datos:

- Nombre de las entidades.
- Tipo de entidad principal o secundaria.
- Identificador del mensaje.
- Mensaje.
- Entidad de origen.
- Entidad de destino.
- Nombre del registro.

A.4.2.2. Descripción de los pasos a seguir para ejecutar el caso de pruebas TC-01

Reporte: Su elaboración es manualmente observando los eventos que ocurren durante la ejecución de pruebas. El registro de los sucesos se realiza en el documento de reporte de pruebas.

Preparación: Antes de iniciar el procedimiento se debe verificar que el perfil UML este cargado en la aplicación de *Enterprise Architect*, que se encuentre la caja de herramientas propias del perfil y haber creado un nuevo proyecto sin elegir ningún diagrama.

Inicio: En la herramienta, se debe seleccionar la caja de herramientas que pertenece al perfil UML, en la cual se presentan los elementos necesarios para construir el diagrama.

Proceso: El usuario agregar las entidades y sus relaciones que considere necesarias al área de trabajo. A su vez, debe agregar el nombre de las entidades y los mensajes que interactúan entre ellas.

Medición: Depende del tiempo que el usuario se tome para agregar las entidades y sus relaciones. Así como las propiedades específicas de cada elemento.

Suspender: La operación se puede suspender el momento que el usuario lo desee, la herramienta no le afecta esta acción.

Reiniciar: La actividad se puede volver a realizar en el momento que el usuario lo requiera.

Cierre: La prueba termina con la creación del diagrama de entidades a nivel usuario.

Contingencias: La presencia de anomalías, obliga a suspender la prueba hasta que se corrija el error. Solucionado el problema se puede reiniciar la prueba.

A.4.3. Procedimiento de Prueba TPr-02

En esta sección se mencionan los datos que se requieren como entrada y el procedimiento para realizar el caso de prueba TC-02.

A.4.3.1. Entradas, salidas y requerimientos especiales

Documentos:

- Plan de pruebas.
- Diseño de pruebas.

Aplicaciones:

- Enterprise Architect.

Datos:

- Nombre de las entidades.
- Tipo de entidad combinada.
- Identificador del mensaje.
- Mensaje.
- Entidad de origen.
- Entidad de destino.
- Nombre del registro.

A.4.3.2. Descripción de los pasos a seguir para ejecutar el caso de pruebas TC-02

Reporte: Su elaboración es manualmente observando los eventos que ocurren durante la ejecución de pruebas. El registro de los sucesos se realiza en el documento de reporte de pruebas.

Preparación: Antes de iniciar el procedimiento se debe verificar que el perfil UML este cargado en la aplicación de *Enterprise Architect*, que se encuentre la caja de herramientas propias del perfil y haber creado un nuevo proyecto sin elegir ningún diagrama.

Inicio: En la herramienta, se debe seleccionar la caja de herramientas que pertenece al perfil UML, en la cual se presentan los elementos necesarios para construir el diagrama.

Proceso: El usuario agrega las entidades y sus relaciones que considere necesarias al área de trabajo. A su vez, debe agregar el nombre de las entidades y los mensajes que interactúan entre ellas.

Medición: Depende del tiempo que el usuario se tome para agregar las entidades y sus relaciones. Así como las propiedades específicas de cada elemento.

Suspender: La operación se puede suspender el momento que el usuario lo desee, la herramienta no le afecta esta acción.

Reiniciar: La actividad se puede volver a realizar en el momento que el usuario lo requiera.

Cierre: La prueba termina con la creación del diagrama de entidades combinado a nivel usuario.

Contingencias: La presencia de anomalías, obliga a suspender la prueba hasta que se corrija el error. Solucionado el problema se puede reiniciar la prueba.

A.4.4. Procedimiento de Prueba TPr-03

En esta sección se mencionan los datos que se requieren como entrada y el procedimiento para realizar el caso de prueba TC-03.

A.4.4.1. Entradas, salidas y requerimientos especiales

Documentos:

- Plan de pruebas.
- Diseño de pruebas.

Aplicaciones:

- Enterprise Architect.

Datos:

- Nombre de las entidades.
- Tipo de entidad (principal o secundaria).
- Identificador del mensaje.
- Mensaje.
- Entidad de origen.
- Entidad de destino.
- Nombre del registro.

A.4.4.2. Descripción de los pasos a seguir para ejecutar el caso de pruebas TC-03

Reporte: Su elaboración es manualmente observando los eventos que ocurren durante la ejecución de pruebas. El registro de los sucesos se realiza en el documento de reporte de pruebas.

Preparación: Antes de iniciar el procedimiento se debe verificar que el perfil UML este cargado en la aplicación de *Enterprise Architect*, que se encuentre la caja de herramientas propias del perfil y tener un diagrama de entidades combinado a nivel usuario en el área de trabajo.

Inicio: En la herramienta, dar clic el símbolo de Enterprise, seleccionar la opción *Open Project* y abrir el diagrama de entidades combinado a nivel usuario recién creado.

Proceso: El usuario debe guardar como un nuevo proyecto el diagrama. Después, debe seleccionar aquellas entidades que formarán parte del diagrama de entidades a nivel aplicación y modificar las propiedades correspondientes para este nivel.

Medición: Depende del tiempo que el usuario se tome para seleccionar las entidades y sus relaciones. Así como modificar las propiedades específicas de cada elemento.

Suspender: La operación se puede suspender el momento que el usuario lo desee, la herramienta no le afecta esta acción.

Reiniciar: La actividad se puede volver a realizar en el momento que el usuario lo requiera.

Cierre: La prueba termina con la creación del diagrama de entidades a nivel aplicación.

Contingencias: La presencia de anomalías, obliga a suspender la prueba hasta que se corrija el error. Solucionado el problema se puede reiniciar la prueba.

A.5. Registro de pruebas

A.5.1. Introducción

El presente documento tiene la función de proveer un registro cronológico acerca de detalles relevantes en la ejecución de las diferentes pruebas.

A.5.1.1. Alcance

Registrar las pruebas de los procedimientos de pruebas: TPr-01, TPr-02 y TPr-03 especificados en el documento de procedimiento de pruebas.

A.5.1.2. Referencias

IEEE Computer Society. (2008) 829-2008 – IEEE Standard for Software and System Test Document.

A.5.2. Registro de prueba TL -01

Se describe el registro de prueba para el procedimiento de prueba TPr-01, actividades y eventos de la prueba.

A.5.2.1. Descripción

El registro de la prueba TL-01 cubre el procedimiento de prueba TPr-01 correspondiente al caso de prueba TC-01. Los datos utilizados en esta prueba se obtuvieron del *Reporte Técnico RT-DCC-1-2007* elaborado por M. C. Edna Daniela López López.

A.5.2.2. Actividades y eventos de la prueba

En la tabla A.6 se describe el registro de eventos y actividades al ejecutar el procedimiento de prueba TPr-01.

Tabla A.6 Registro del evento TPr-01.

Paso	Actor	Descripción	Verificado
1.	Usuario	Abrir la herramienta <i>Enterprise Architect</i>	Sí
2.	Usuario	Generar un nuevo proyecto.	Sí
3.	Usuario	Dar clic derecho en el paquete <i>Model</i> ubicado en el panel <i>Project Broser</i> .	Sí
4.	Usuario	Elije un vista simple.	Sí
5.	Herramienta	Genera la vista (paquete), que se muestra en el panel <i>Project Browser</i> .	Sí
6.	Usuario	Dar clic derecho en la vista generada, y seleccionar <i>Add Diagram</i> para crear un diagrama.	Sí
7.	Herramienta	Muestra una ventana con los tipos de diagramas.	Sí
8.	Usuario	Ingresa el nombre del diagrama y seleccionar <i>Entity Diagram</i> del recuadro <i>Type</i> .	Sí
9.	Herramienta	Genera el área de trabajo para crear el diagrama.	Sí
10.	Usuario	Seleccionar el elemento <i>Entidad</i> de la caja de herramientas y agrega al área de trabajo.	Sí
11.	Herramienta	Muestra un ventana solicitando el nombre de la	Sí

		entidad y sus propiedades correspondientes.	
12.	Usuario	Ingresar el nombre de la entidad. En la propiedades ingresar el identificador, y seleccionar el tipo de entidad (principal o secundaria).	Sí
13.	Usuario	Repetir el paso 10 al 13, hasta colocar todas las entidades correspondientes.	Sí
14.	Usuario	Seleccionar el elemento <i>TablaRegistro</i> de la caja de herramientas y coloca la relación entre las entidades correspondientes.	Sí
15.	Herramienta	Muestra una ventana solicitando el nombre de la tabla de registro.	Sí
16.	Usuario	Ingresar el nombre de la tabla.	Sí
17.	Usuario	Repetir del paso 14 al 16, hasta agregar todas las relaciones entre las entidades.	Sí
18.	Usuario	Seleccionar el elemento <i>Mensaje</i> de la caja de herramientas y colocarlo en la tabla de registro (<i>TablaRegistro</i>).	Sí
19.	Herramienta	Muestra una ventana solicitando el mensaje.	Sí
20.	Usuario	Ingresar el mensaje.	Sí
21.	Usuario	Dar doble clic en el mensaje.	Sí
22.	Herramienta	Muestra una ventana, donde en la parte inferior derecha, se encuentra la opción <i>Tagged Values</i> .	Sí
23.	Usuario	Seleccionar la opción <i>Tagged Values</i> e ingresar los valores solicitados (identificador, proceso, dirección, origen y destino).	Sí
24.	Usuario	Repetir del paso 20 al 23 hasta añadir todos los mensajes que existen entre las entidades.	Sí

25.	Usuario	Guardar el diagrama de entidades a nivel usuario.	Sí
-----	---------	---	----

A.5.2.3. Anomalías

No se reportan anomalías.

A.5.2.4. Ejemplo de datos utilizados para la prueba.

En la Tabla A.7 se muestran los datos utilizados para probar el requerimiento funcional Crear el diagrama entidad a nivel usuario.

Tabla A.7 Datos empleados para probar el requerimiento funcional

Nombre del campo	Tipo	Valor
Entidad	Cadena	Jefe de organización bibliográfica
Mensajes	Cadena	2: Solicita préstamos. 1: Envía libros, tesis o revistas. 2: Enviar lista. 3: Envía libros.
RegistroMensajes	Cadena	Usuarios – JOB JOB – JCI

A.5.3. Registro de prueba TL - 02

Se describe el registro de prueba para el procedimiento de prueba TPr-02, actividades y eventos de la prueba.

A.5.3.1. Descripción

El registro de la prueba TL-02 cubre el procedimiento de prueba TPr-02 correspondiente al caso de prueba TC-02. Los datos utilizados en esta prueba se obtuvieron del *Reporte Técnico RT-DCC-1-2007* elaborado por M. C. Edna Daniela López López.

A.5.3.2. Actividades y eventos de la prueba

En la tabla A.8 se describe el registro de eventos y actividades al ejecutar el procedimiento de prueba TPr-02.

Tabla A.8 Registro del evento TPr-02.

Paso	Actor	Descripción	Verificado
1.	Usuario	Abrir la herramienta <i>Enterprise Architect</i>	Sí
2.	Usuario	Generar un nuevo proyecto.	Sí
3.	Usuario	Dar clic derecho en el paquete <i>Model</i> ubicado en el panel <i>Project Broser</i> .	Sí
4.	Usuario	Elije un vista simple.	Sí
5.	Herramienta	Genera la vista (paquete), que se muestra en el panel <i>Project Browser</i> .	Sí
6.	Usuario	Dar clic derecho en la vista generada, y seleccionar <i>Add Diagram</i> para crear un diagrama.	Sí
7.	Herramienta	Muestra una ventana con los tipos de diagramas.	Sí
8.	Usuario	Ingresar el nombre del diagrama y seleccionar <i>Entity Diagram</i> del recuadro <i>Type</i> .	Sí
9.	Herramienta	Genera el área de trabajo para crear el diagrama de entidades.	Sí
10.	Usuario	Seleccionar el elemento <i>Entidad</i> de la caja de herramientas y agrega al área de trabajo.	Sí
11.	Herramienta	Muestra un ventana solicitando el nombre de la entidad y sus propiedades correspondientes.	Sí
12.	Usuario	Ingresar el nombre de la entidad. En la propiedades ingresar el identificador, y seleccionar el tipo de entidad (combinado).	Sí

13.	Usuario	Repetir el paso 10 al 13, hasta colocar todas las entidades correspondientes.	Sí
14.	Usuario	Seleccionar el elemento <i>TablaRegistro</i> de la caja de herramientas y coloca la relación entre las entidades correspondientes.	Sí
15.	Herramienta	Muestra una ventana solicitando el nombre de la tabla de registro.	Sí
16.	Usuario	Ingresar el nombre de la tabla.	Sí
17.	Usuario	Repetir del paso 14 al 16, hasta agregar todas las relaciones entre las entidades.	Sí
18.	Usuario	Seleccionar el elemento <i>Mensaje</i> de la caja de herramientas y colocarlo en la tabla de registro (<i>TablaRegistro</i>).	Sí
19.	Herramienta	Muestra una ventana solicitando el mensaje.	Sí
20.	Usuario	Ingresar el mensaje.	Sí
21.	Usuario	Dar doble clic en el mensaje.	Sí
22.	Herramienta	Muestra una ventana, donde en la parte inferior derecha, se encuentra la opción <i>Tagged Values</i> .	Sí
23.	Usuario	Seleccionar la opción <i>Tagged Values</i> e ingresar los valores solicitados (identificador, proceso, dirección, origen y destino).	Sí
24.	Usuario	Repetir del paso 20 al 23 hasta añadir todos los mensajes que existen entre las entidades.	Sí
25.	Usuario	Guardar el diagrama de entidades combinado a nivel usuario.	Sí

A.5.3.3. Anomalías

No se reportan anomalías.

A.5.3.4. Ejemplo de datos utilizados para la prueba.

En la Tabla A.9 se muestran los datos utilizados para probar el requerimiento funcional Crear el diagrama de entidades combinado a nivel usuario.

Tabla A.9 Datos empleados para probar el requerimiento funcional

Nombre del campo	Tipo	Valor
Entidad	Cadena	Auxiliares de biblioteca, Jefe de centro de información
Mensajes	Cadena	N-1 Envía revistas para ser selladas y registradas en kardex N-2 Envía el kardex para ser capturado
RegistroMensajes	Cadena	AB - JCB

A.5.4. Registro de prueba TL - 03

Se describe el registro de prueba para el procedimiento de prueba TPr-03, actividades y eventos de la prueba.

A.5.4.1. Descripción

El registro de la prueba TL-03 cubre el procedimiento de prueba TPr-023 correspondiente al caso de prueba TC-03. Los datos utilizados en esta prueba se obtuvieron del *Reporte Técnico RT-DCC-1-2007* elaborado por M. C. Edna Daniela López López.

A.5.4.2. Actividades y eventos de la prueba

En la A.10 se describe el registro de eventos y actividades al ejecutar el procedimiento de prueba TPr-02.

Tabla A .10 Registro del evento TPr-03.

Paso	Actor	Descripción	Verificado
1.	Usuario	Abrir la herramienta <i>Enterprise Architect</i>	Sí
2.	Usuario	Generar un nuevo proyecto.	Sí

3.	Usuario	Dar clic derecho en el paquete <i>Model</i> ubicado en el panel <i>Project Broser</i> .	Sí
4.	Usuario	Elige un vista simple.	Sí
5.	Herramienta	Genera la vista (paquete), que se muestra en el panel <i>Project Browser</i> .	Sí
6.	Usuario	Dar clic derecho en la vista generada, y seleccionar <i>Add Diagram</i> para crear un diagrama.	Sí
7.	Herramienta	Muestra una ventana con los tipos de diagramas.	Sí
8.	Usuario	Ingresar el nombre del diagrama y seleccionar <i>Entity Diagram</i> del recuadro <i>Type</i> .	Sí
9.	Herramienta	Genera el área de trabajo para crear el diagrama de entidades.	Sí
10.	Usuario	Seleccionar el elemento <i>Entidad</i> de la caja de herramientas y agrega al área de trabajo.	Sí
11.	Herramienta	Muestra un ventana solicitando el nombre de la entidad y sus propiedades correspondientes.	Sí
12.	Usuario	Ingresar el nombre de la entidad. En la propiedades ingresar el identificador, y seleccionar el tipo de entidad (principal o secundario).	Sí
13.	Usuario	Repetir el paso 10 al 13, hasta colocar todas las entidades correspondientes.	Sí
14.	Usuario	Seleccionar el elemento <i>TablaRegistro</i> de la caja de herramientas y coloca la relación entre las entidades correspondientes.	Sí
15.	Herramienta	Muestra una ventana solicitando el nombre de la tabla de registro.	Sí

16.	Usuario	Ingresar el nombre de la tabla.	Sí
17.	Usuario	Repetir del paso 14 al 16, hasta agregar todas las relaciones entre las entidades.	Sí
18.	Usuario	Seleccionar el elemento <i>Mensaje</i> de la caja de herramientas y colocarlo en la tabla de registro (<i>TablaRegistro</i>).	Sí
19.	Herramienta	Muestra una ventana solicitando el mensaje.	Sí
20.	Usuario	Ingresar el mensaje.	Sí
21.	Usuario	Dar doble clic en el mensaje.	Sí
22.	Herramienta	Muestra una ventana, donde en la parte inferior derecha, se encuentra la opción <i>Tagged Values</i> .	Sí
23.	Usuario	Seleccionar la opción <i>Tagged Values</i> e ingresar los valores solicitados (identificador, proceso, dirección, origen y destino).	Sí
24.	Usuario	Repetir del paso 20 al 23 hasta añadir todos los mensajes que existen entre las entidades.	Sí
25.	Usuario	Guardar el diagrama de entidades combinado a nivel usuario.	Sí

A.5.4.3. Anomalías

No se reportan anomalías.

A.5.4.4. Ejemplo de datos utilizados para la prueba.

En la Tabla A.11 se muestran los datos utilizados para probar el requerimiento funcional Crear el diagrama de entidades a nivel aplicación.

Tabla ¡Error! No hay texto con el estilo especificado en el documento..11 Datos empleados para probar el requerimiento funcional

Nombre del campo	Tipo	Valor
Entidad	Cadena	Sistema de biblioteca, Planeación
Mensajes	Cadena	R-1 Solicita estadísticas R-2 Envío de estadísticas
RegistroMensajes	Cadena	P – SB

A.6. Reporte de pruebas

A.6.1. Introducción

El presente documento se detallan los resultados, se menciona conclusiones y recomendaciones.

A.6.1.1. Alcance

Reunir los casos de prueba descrito en el documento de casos de prueba, los cuales son: TC-01, TC-02 y TC-03.

A.6.1.2. Referencias

IEEE Computer Society. (2008) 829-2008 – IEEE Standard for Software and System Test Document.

A.6.2. Detalles

A continuación, se muestran los resultados de las pruebas, conclusiones y recomendaciones.

A.6.2.1. Resultados generales

En la tabla A.12 se muestran una descripción general de los resultados obtenidos.

Tabla A.12 Resultados obtenidos

Requerimiento	Caso de Prueba	Resultado de la prueba
Crear el diagrama entidad a nivel usuario.	TC-01	Exitoso
Crear el diagrama entidad combinado a nivel usuario.	TC-02	Exitoso
Crear el diagrama entidad a nivel aplicación.	TC-03	Exitoso

En la tabla A.13 se muestran los resultados de las pruebas realizadas a las funcionalidades generales del editor.

Tabla A.13 Funcionalidades

Funcionalidad	Resultado
1. Agregar un entidad	Exitoso
2. Nombrar un entidad	Exitoso
3. Agregar una relación <i>TablaRegistro</i>	Exitoso
4. Nombrar relación <i>TablaRegistro</i>	Exitoso
5. Agregar un mensaje	Exitoso
6. Agregar propiedades al mensaje	Exitoso

ANEXO B. CASO DE ESTUDIO

B.1. Introducción

En este anexo se detalla el desarrollo de un caso de estudio para demostrar la generación de los diagramas de entidades a través del perfil UML propuesto. En la sección B.1 se describe la construcción de un diagrama de entidades a nivel usuario. En la sección B.2 se expone la construcción de un diagrama de entidades combinado.

El desarrollo del caso de estudio consiste en la elaboración de cada uno de los diagramas de entidades (diagrama de entidades a nivel usuario, diagramas de entidades combinado y diagrama de entidades a nivel aplicación) propuestos en la metodología AGD. Para esto, se utilizan tres diagramas de entidades desarrollados en el *Reporte Técnico RT-DCC-1-2007* elaborado por la M. C. Edna Daniela López López. Cabe mencionar, que dichos diagramas están diseñados para un problema clásico de biblioteca y forman parte de un procedimiento más completo, y qué para este caso de estudio, solo se requieren para demostrar la funcionalidad del perfil UML.

Los diagramas que se elaborarán son:

- Diagrama de comunicación combinado de todos los usuarios a nivel CIM.
- Diagrama de comunicación a nivel aplicación a nivel CIM.

B.2. Diagrama de entidades combinado

La construcción del diagrama de entidades combinado a nivel usuario es muy similar al diagrama de entidades a nivel usuario. En este diagrama se reúnen todos los diagramas de entidades a nivel para ver las interacciones entre todas las entidades involucradas.

B.2.1. Identificación de los elementos

El diagrama que se mapea a un diagrama de entidades combinado a nivel usuario es el Diagrama de comunicación combinado de todos los usuarios a nivel CIM (ver figura B.1) generado en el reporte técnico.

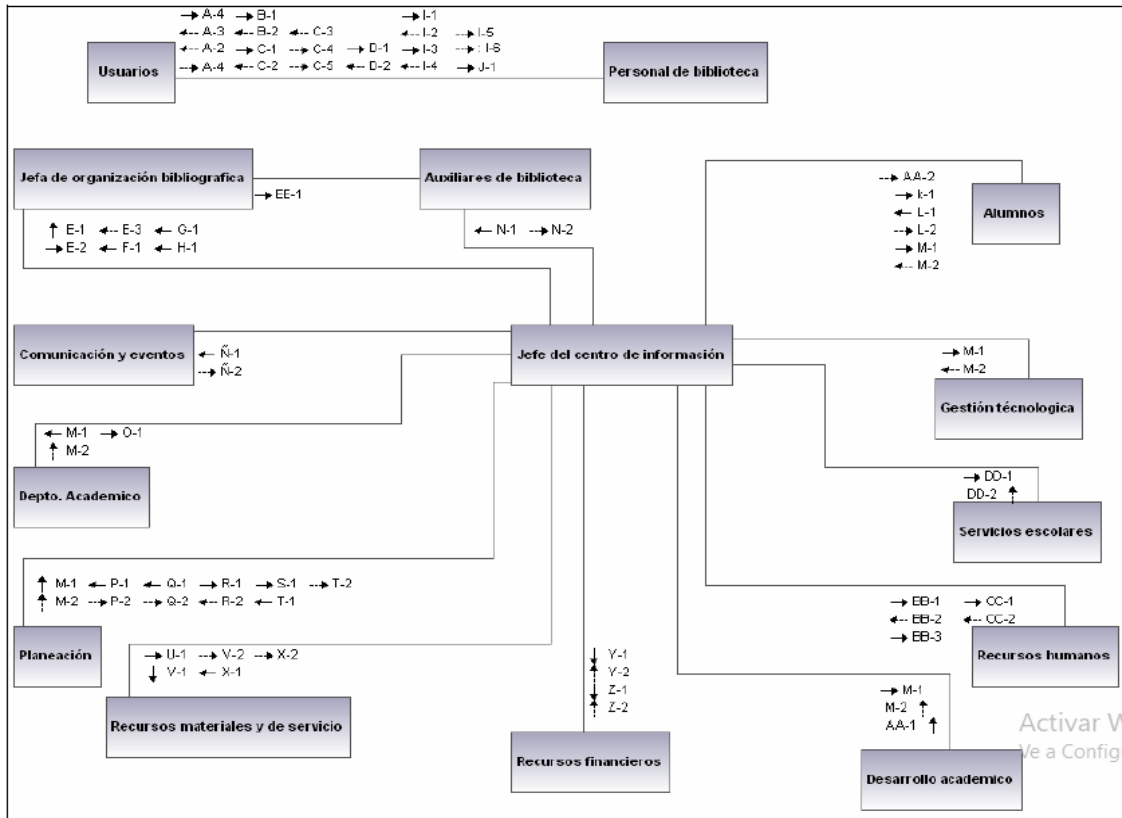


Figura B.1 Diagrama de comunicación combinado de todos los usuarios a nivel CIM

En este tipo de diagrama, las entidades se convierten en una entidad combinada, dado que no se toma en cuenta la interacción entre una entidad principal y una entidad secundaria. Conforme a esto, se identificaron las siguientes entidades (tabla B.1):

Tabla B.1 Entidades

Nombre de la entidad	Identificador
Usuarios	U
Personal de biblioteca	PB
Jefa de organización bibliográfica	JOB
Auxiliares de biblioteca	AB
Jefe del centro de información	JCI
Comunicación y eventos	CE
Depto. Académico	DA
Planeación	P
Recursos materiales y servicios	RMS
Recursos humanos	RH

Alumnos	A
Gestión tecnológica	GT
Servicios escolares	SE
Desarrollo académico	DE

Los mensajes de recibido se muestran con el símbolo ←, y los mensajes de envío con el símbolo →. En la tabla B.2 se muestran las interacciones entre la entidad *Usuarios* y la entidad *Personal de biblioteca*.

Tabla B.2 Relaciones entre las entidades Usuario y Personal de biblioteca

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
A-2	Proporciona servicio	Proporciona servicio bibliotecario	PB	U
A-3	Entrega ficha	Entrega ficha de préstamos de acervo	PB	U
A-4	Entrega credencial	Entrega credencial	U	PB
B-1	Entrega ficha	Entrega ficha de préstamo	U	PB
B-2	Entrega credencial	Entrega credencial	PB	U
C-1	Solicita préstamo	Solicita préstamo bibliotecario de acervo único o 1er ejemplar	U	PB
C-2	Proporciona servicio	Proporciona servicio bibliotecario	PB	U
C-3	Entrega ficha	Entrega ficha para préstamo de acervo único o 1er ejemplar para firma	PB	U
C-4	Entrega ficha	Entrega ficha de acervo único o 1er ejemplar firmada	U	PB
C-5	Entrega credencial	Entrega credencial	U	PB
D-1	Retorna acervo	Retorna acervo único o 1er ejemplar	U	PB
D-2	Entrega credencial	Entrega credencial	PB	U

En la tabla B.3 se presentan los mensajes y las relaciones entre la entidad Jefa de organización bibliográfica y Auxiliares de biblioteca.

Tabla B.3 Relación entre JOB y AB

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
EE-1	Envío de documentos	Envía libros, tesis o revistas registradas para ser exhibidas	JOB	AB

En las tablas B.4 a B.7 se muestran las relaciones y los mensajes que existe entre las entidades. La información que se presentan en las tablas es necesaria para facilitar la construcción de las entidades.

Tabla B.4 Relación entre JCI y JOB

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
E-1	Envía libros	Envía libros comprados o tesis.	JCI	JOB
E-2	Solicita revisión	Solicita revisión de catalogación y clasificación de libros o tesis.	JOB	JCI
E-3	Envía revisión	Envía la revisión de catalogación y clasificación para ser capturada en el sistema.	JCI	JOB
F-1	Envía revisión	Envía revisión de usuarios para ser modificados o eliminados.	JCI	JOB
G-1	Envía lista de usuarios	Envía lista de usuarios para dar de alta (realizar sus credenciales)	JCI	JOB

Tabla B.5 Relación entre JCB y AB

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
N-1	Envía revistas	Envía revistas para ser selladas y registradas en kardex.	JCB	AB
N-2	Envía el kardex	Envía el kardex para ser capturado.	AB	JCB

Tabla B.6 Relación entre JCI y CE

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
Ñ-1	Solicita apoyo	Solicita apoyo para eventos	JCI	CE
Ñ-2	Envío apoyo	Envío apoyo para eventos	CE	JCI

Tabla B.7 Relación entre JCE y DA

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
M-1	Solicita bibliografías	Solicita necesidades bibliográficas	JCE	DA
M-2	Envía bibliografías	Envía necesidades bibliográficas	DA	JCI

Cada tabla representa una relación entre dos entidades. Las tablas B.8 a B.13 muestran los mensajes entre P – JCI, RF – JCI, RH – JCI y SE – JCI.

Tabla B.8 Relación entre P – JCI

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
M-1	Solicita bibliografía	Solicita necesidades bibliográficas.	P	JCI
M-2	Envía bibliografía	Envía necesidades bibliográficas.	P	JCI
P-1	Envío programa	Envío de programa de trabajo anual.	JCI	P
P-2	Envío aceptación	Envío de aceptación del plan de trabajo anual.	P	JCI

Q-1	Solicita equipo	Solicita equipo e instalaciones	JCI	P
Q-2	Envío equipo	Envío de equipo e instalaciones.	P	JCI
R-1	Solicita estadísticas.	Solicita estadísticas.	P	JCI
R-2	Envío estadísticas.	Envío de estadísticas.	JCI	P

Tabla B.9 Relación entre RMS y JCI

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
V-1	Solicita personal	Solicita personal para mantenimiento.	JCI	RMS
V-2	Envío personal	Envío de personal para mantenimiento.	RMS	JCI
X-1	Solicita papelería	Solicita papelería.	JCI	RMS
X-2	Envío papelería	Envío de papelería.	RMS	JCI

Tabla B.10 Relación RF y JCI

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
Y-1	Solicita viáticos	Solicita viáticos y transporte	JCI	RF
Y-2	Envío recursos	Envío de recursos solicitados	RF	JCI
Z-1	Envío solicitud	Envío de solicitud de gastos	JCI	RF
Z-2	Cheque	Elaboración del cheque	RF	JCI

Tabla B.1.1 Relación JCI y DA

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
M-1	Solicita bibliografía	Solicita necesidades bibliográficas.	JCI	DA
M-2	Envía bibliografía	Envía necesidades bibliográficas.	DA	JCI
AA-1	Solicita seminario	Solicita la impartición de seminario (Técnicas para búsqueda de información).	DA	JCI

Tabla.B.1 2 Relación JCI y RH

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
BB-1	Solicita personal	Solicita personal para biblioteca	JCI	RH
BB-2	Proporciona personal	Proporciona personal para biblioteca	RH	JCI
BB-3	Acepta personal	Acepta personal para biblioteca	JCI	RH
CC-1	Solicita pago	Solicita pago	JCI	RH
CC-2	Recibe pago	Recibe pago	RH	JCI

Tabla B.1 3 Relación SE y JCI

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
DD-1	Solicita lista	Solicita lista de alumnos inscritos	JCI	SE
DD-2	Envío de lista	Envío de lista de alumnos inscritos	SE	JCI

B.2.2. Construcción

Conforme a lo establecido en la sección 5.3.2. *Pasos para construir un diagrama de entidades en Enterprise Architect*, se desarrolla el diagrama de entidades combinado.

1. Crear las entidades.

Las entidades que se colocaran en el área de trabajo son las definidas en la Tabla B.1. Estas entidades son consideradas de tipo *combinada*, dado que se trata de un diagrama general a nivel usuario, como se muestra en la figura 11.

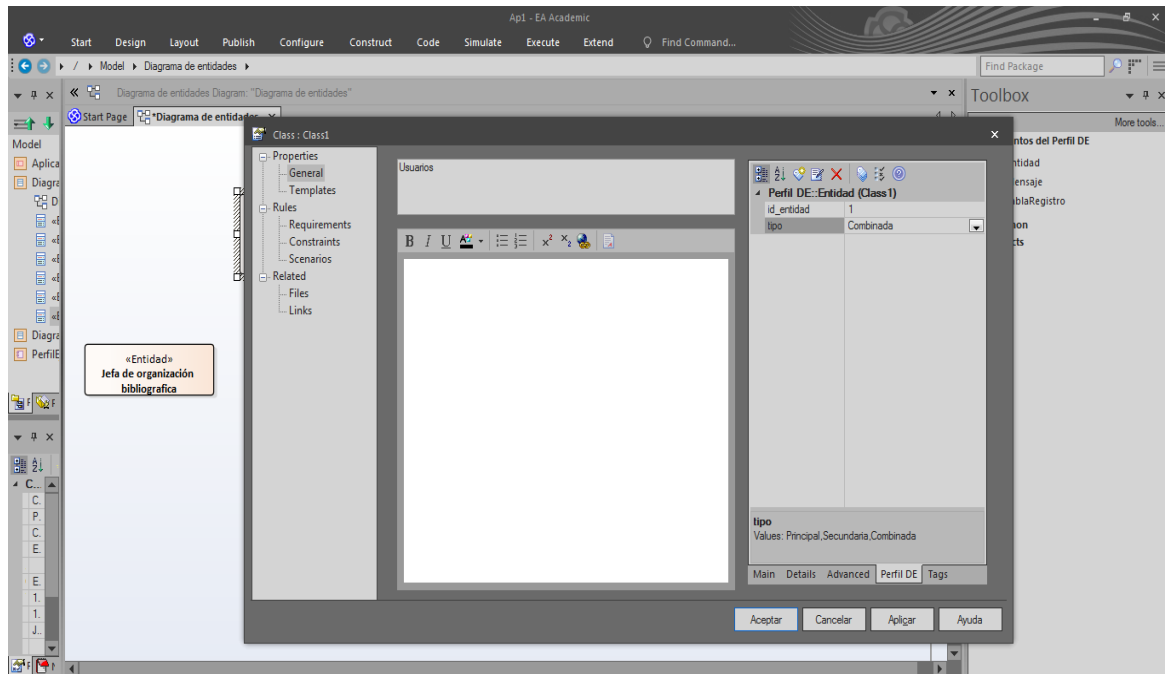


Figura B.2 Crear de una entidad combinada

En la figura B.3, se presentan todas las entidades para construir el diagrama de entidades combinado.

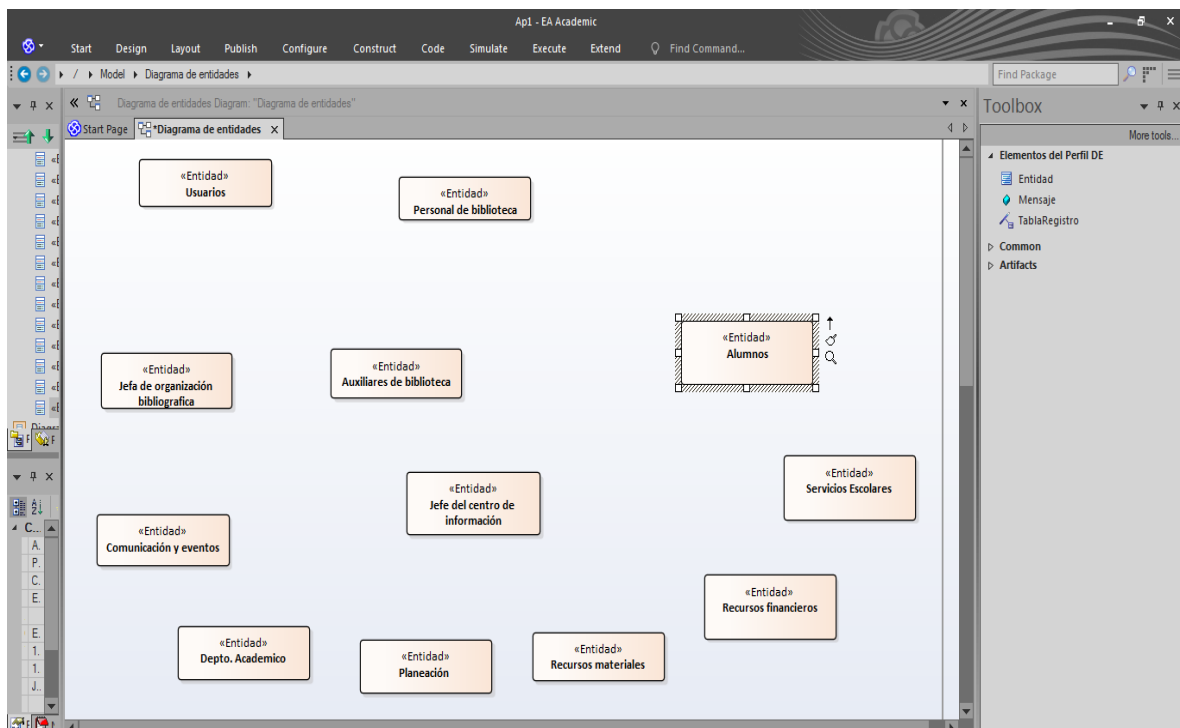


Figura B.3 Entidades combinadas

2. Crear relaciones

Se generan las relaciones que existen entre entidades, utilizando el elemento *TablaRegistro*. (Figura B.4)

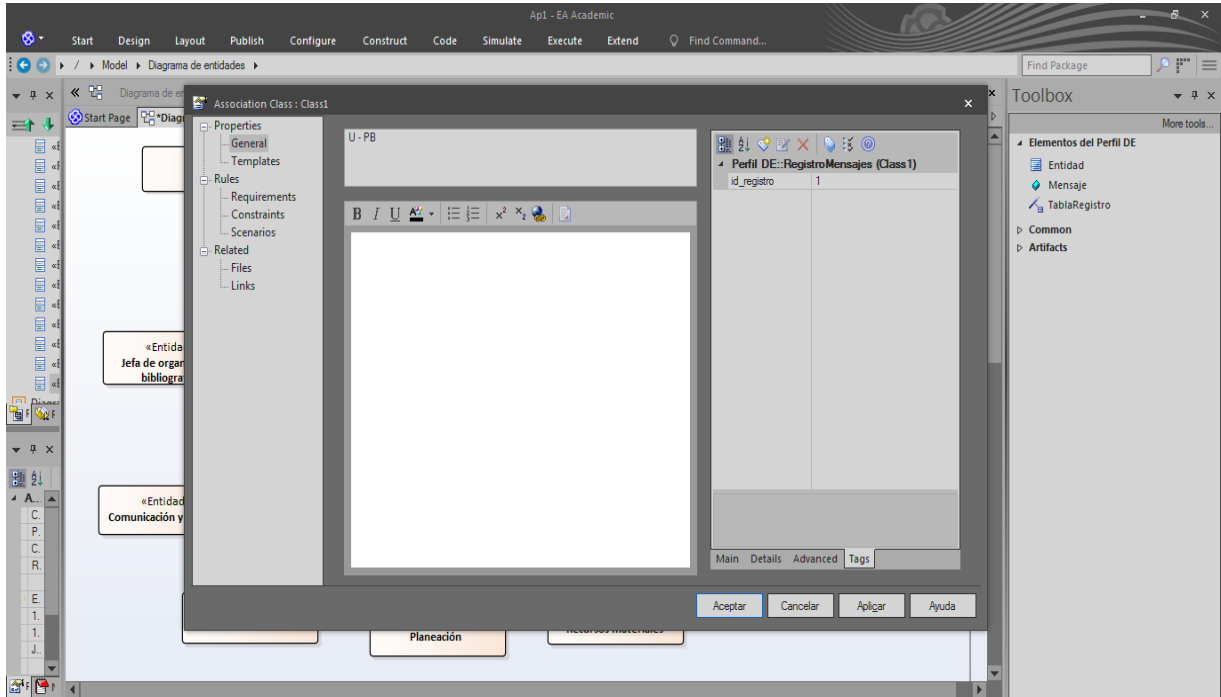


Figura B.4 Generar relaciones

En la figura B.5 se presentan el diagrama con todas las relaciones.

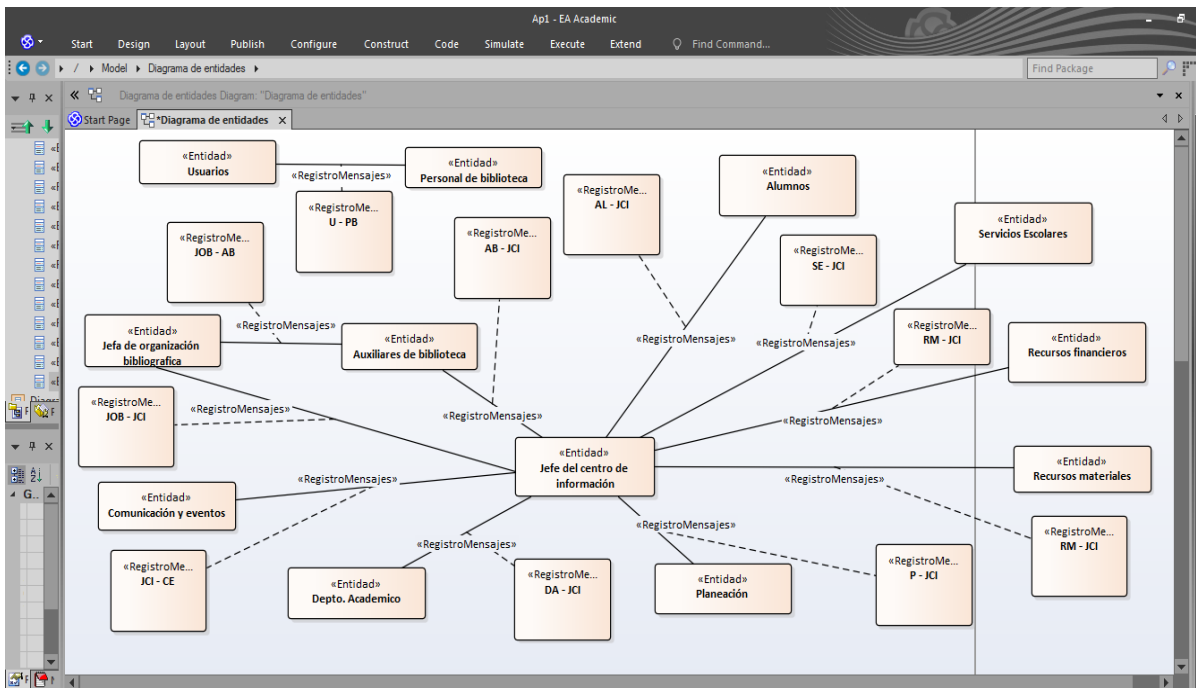


Figura B.5 Relaciones entre entidades combinadas

3. Crear mensajes

Una vez colocadas las entidades y sus respectivas relaciones, se agregan los mensajes que existen entre cada entidad. (Figura B.6)

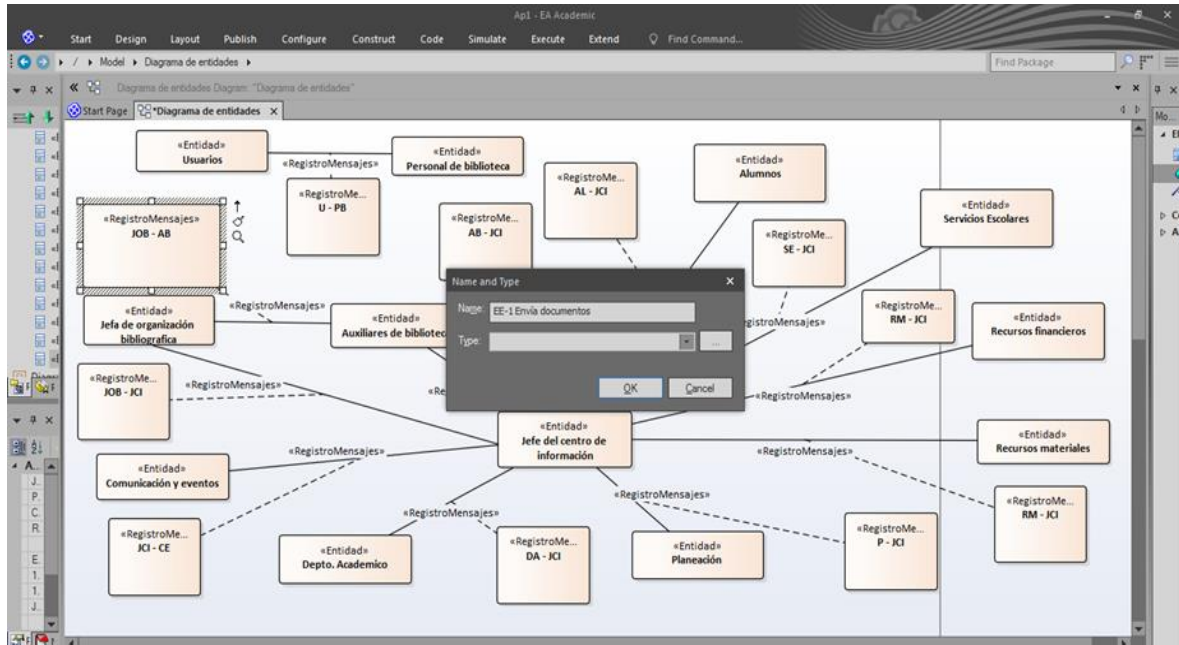


Figura B.6 Generar mensajes (combinada)

Cada mensaje se le agregan las propiedades definidas en las tablas de la sección 3.2.1. Identificación de elementos. (Figura B.7)

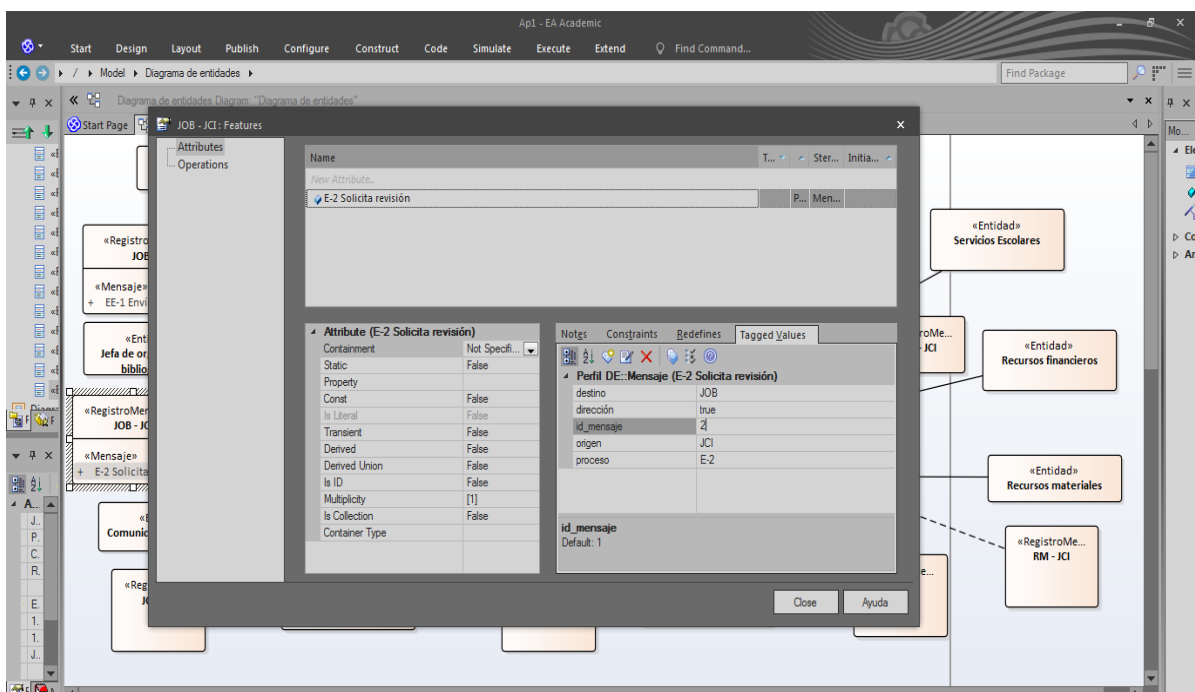


Figura B.7 Propiedades del mensaje

B.2.3. Resultado

El resultado obtenido es el diagrama de entidades combinado a nivel usuario (ver figura B.8).

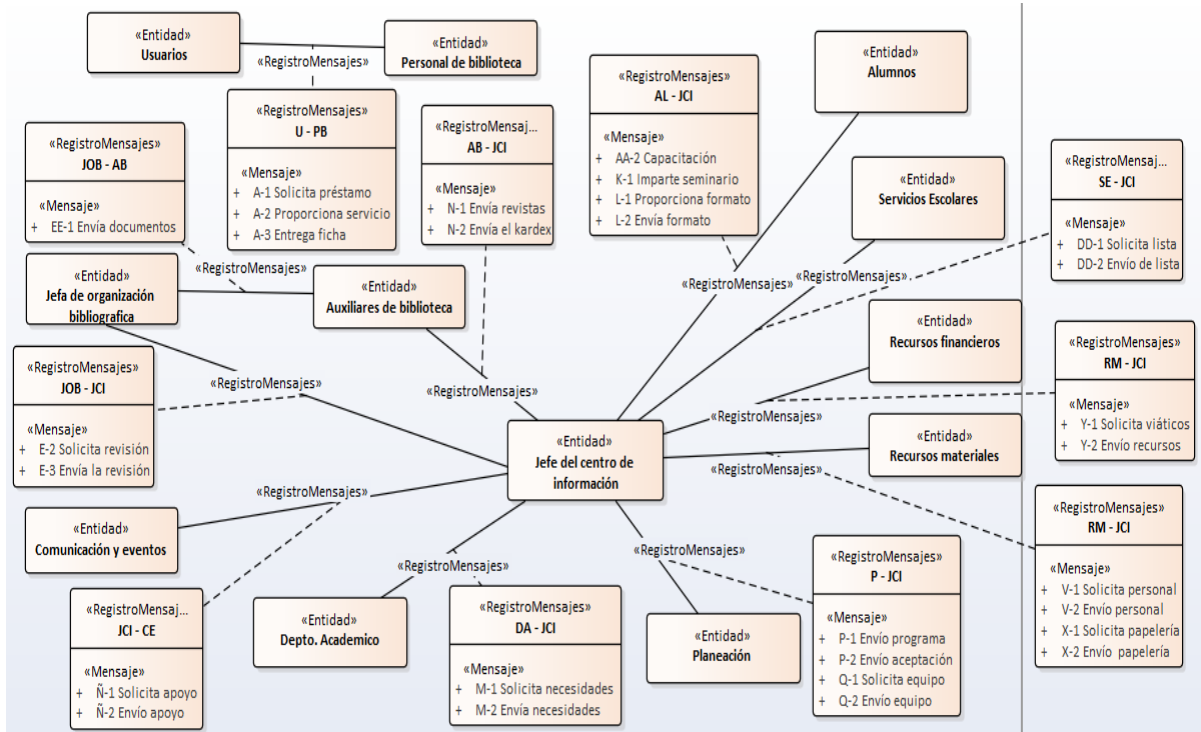


Figura B.8 Diagrama de entidades combinado a nivel usuario

B.3. Diagrama de entidades a nivel aplicación

El diagrama de entidades a nivel aplicación es obtenido a partir del segundo diagrama, y se construye con las mismas características de un diagrama de entidades a nivel usuario.

B.3.1. Identificación de los elementos

En la siguiente figura se muestra el Diagrama de comunicación a nivel aplicación a nivel CIM (obtenido del *Reporte Técnico RT-DCC-1-2007*), el cual se realizará con el perfil UML para mapearlo a un diagrama de entidades a nivel aplicación.

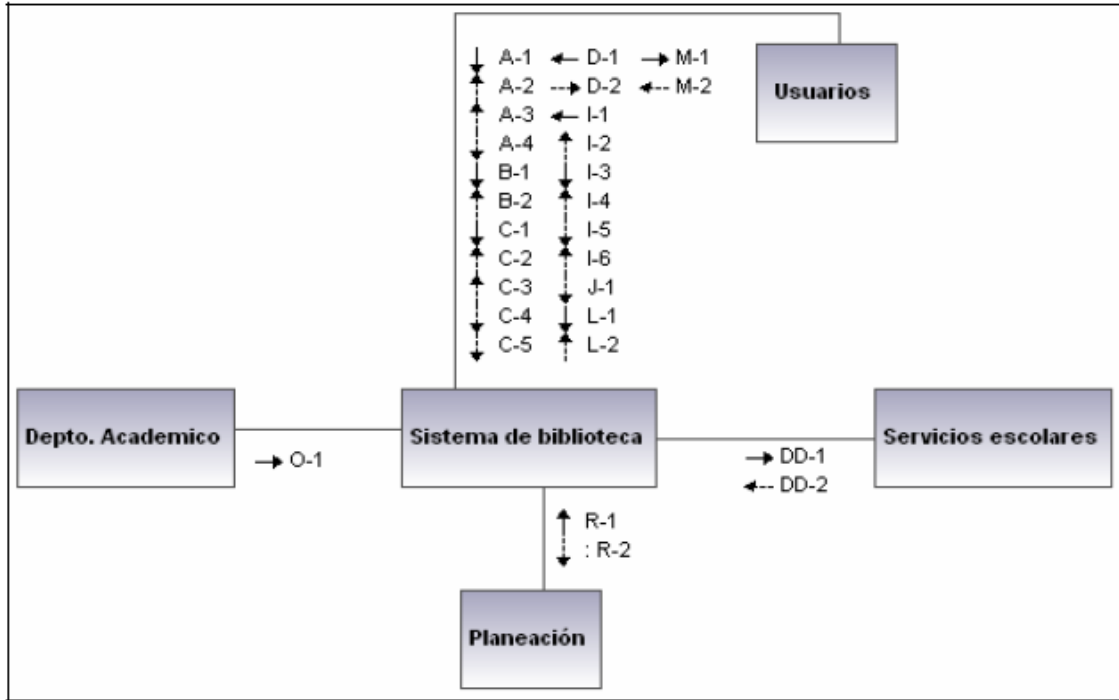


Figura B.8 Diagrama de comunicación a nivel aplicación a nivel CIM

En la tabla B.14 se presentan la entidad principal y las entidades secundarias. El identificador asignado a las entidades se utiliza más adelante para el nombre del elemento *TablaRegistro*.

Tabla B.14 Entidades definidas (nivel aplicación)

Nombre de la entidad	Tipo de entidad	Identificador
Sistema de biblioteca	Principal	SB
Usuarios	Secundaria	U
Depto. Académico	Secundaria	DA
Planeación	Secundaria	P
Servicios Escolares	Secundaria	SE

Las relaciones y mensajes que se presentan en la figura B.8, se definen a detalle en la tabla B.14 para una identificación apropiada de los elementos. Como en los anteriores diagramas, se acota el mensaje utilizado en el diagrama original para mejor visualización.

En las tablas B.15 a B.18 se muestran los procesos y los mensajes enviados o recibidos entre dos entidades.

Tabla B .15 Relación entre DA y SB

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
O-1	Envió de tesis	Envió de tesis para exhibición	DA	SB

Tabla B.16 Relación entre P y SB

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
R-1	Solicita estadísticas	Solicita estadísticas	P	SB
R-2	Envío de estadísticas	Envío de estadísticas	SB	P

Tabla B.17 Relación SB y SE

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
DD-1	Solicita lista	Solicita lista de alumnos inscritos	SB	SE
DD-2	Envío de lista	Envío de lista de alumnos inscritos	SE	SB

Tabla B.18 Relación U y SB

Proceso	Mensaje	Descripción del mensaje	Origen	Destino
A-1	Solicita préstamo	Solicita préstamo bibliotecario de acervo	U	SB
A-2	Proporciona servicio	Proporciona servicio bibliotecario	SB	U
A-3	Entrega ficha	Entrega ficha de préstamos de acervo	SB	U
A-4	Entrega credencial	Entrega credencial	U	SB
B-1	Entrega ficha	Entrega ficha de préstamo	U	SB
B-2	Entrega credencial	Entrega credencial	SB	U

C-1	Solicita préstamo	Solicita préstamo bibliotecario de acervo único o 1er ejemplar	U	SB
C-2	Proporciona servicio	Proporciona servicio bibliotecario	SB	U
C-3	Entrega ficha	Entrega ficha para préstamo de acervo único o 1er ejemplar para firma	SB	U
C-4	Entrega ficha	Entrega ficha de acervo único o 1er ejemplar firmada	U	SB
C-5	Entrega credencial	Entrega credencial	U	SB
D-1	Retorna acervo	Retorna acervo único o 1er	U	SB
D-2	Entrega credencial	Entrega credencial	SB	U
I-1	Solicita catalogo	Solicita el catalogo interbibliotecario	U	SB
I-2	Envía catalogo	Envía catalogo interbibliotecario	SB	U
I-3	Solicita formato	Solicita formato de préstamo interbibliotecario	U	SB
I-4	Proporciona formato	Proporciona el formato de préstamo interbibliotecario	SB	U
I-5	Envía formato	Envía formato de préstamo interbibliotecario llenado	U	SB
I-6	Recibe formato	Recibe formato de préstamo interbibliotecario firmado y sellado	SB	U

B.3.2. Construcción

En la sección anterior se definieron las entidades y la dirección de los mensajes entre ellas, con base al procedimiento marcado en la sección 2. Para la construcción del último diagrama se toma como datos de entrada las relaciones y mensajes definidos en las tablas,

1. Crear entidades

Para crear una entidad se selecciona el elemento Entidad de la caja de herramientas. A cada entidad se le asigna un nombre y un identificador. En las figura B.9 y B.10 se muestran las entidades definidas de la tabla B.14.

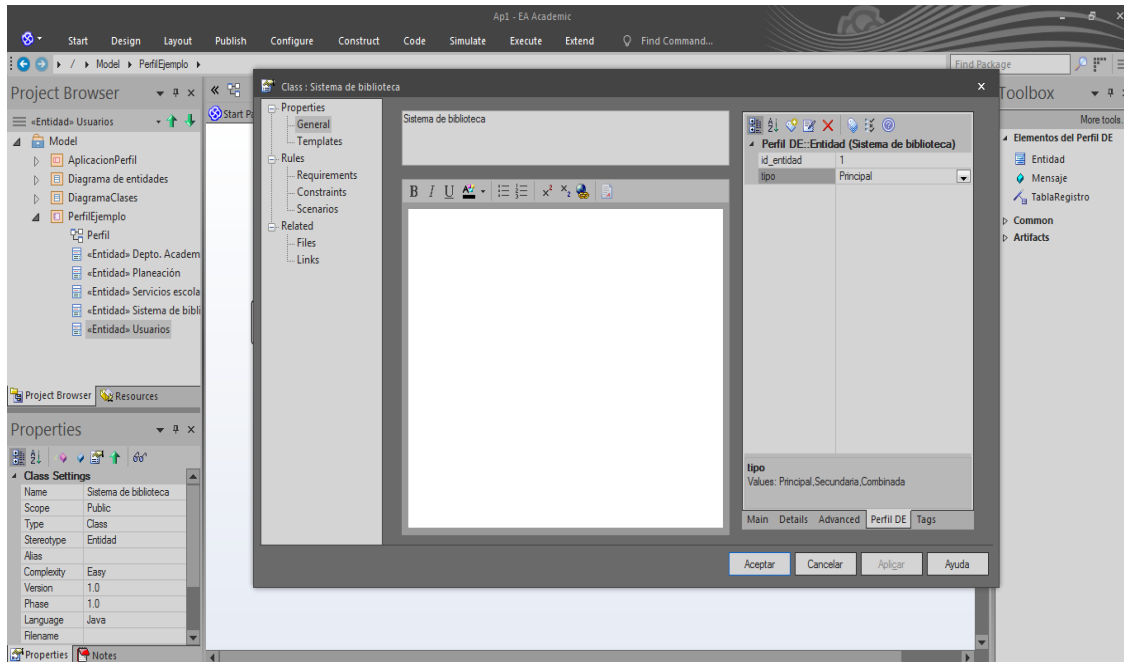


Figura B.9 Nombre de la entidad

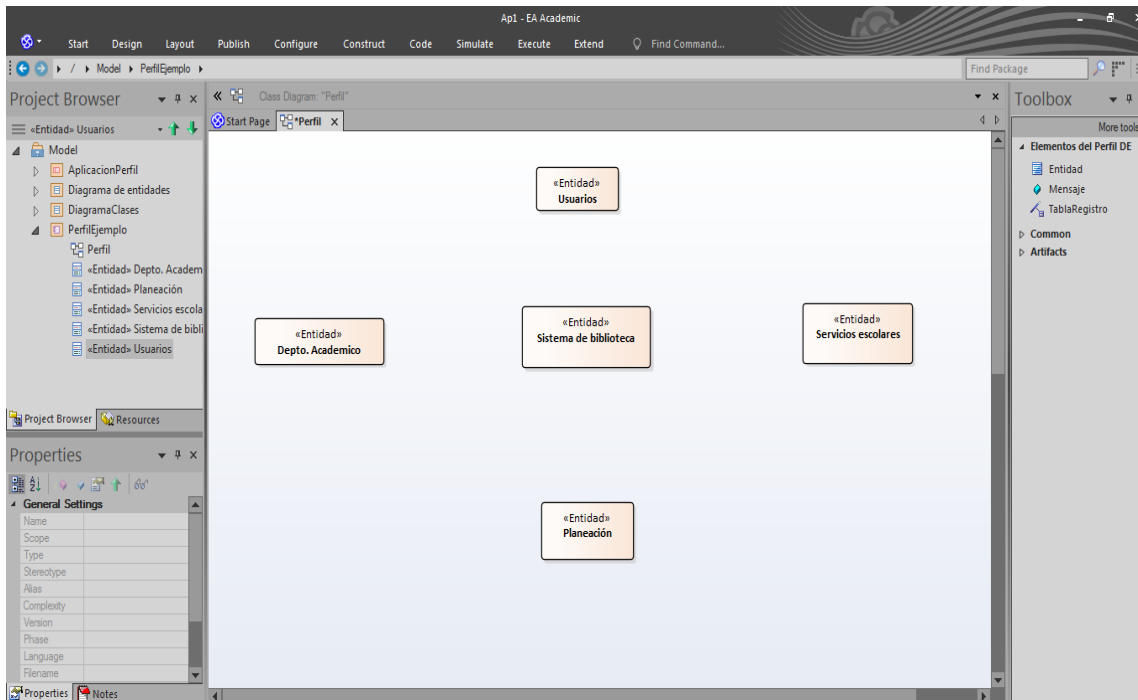


Figura B.10 Entidades

2. Crear relaciones

Las relaciones para las entidades se generan con el elemento *TablaRegistro*. Cada relación contiene una tabla que le permitirá ir agregando los mensajes existan. En la figura B.11 se muestran las entidades con respectivas relaciones. Cabe mencionar que el nombre de la relación es indicado por el identificador asociado a una entidad.

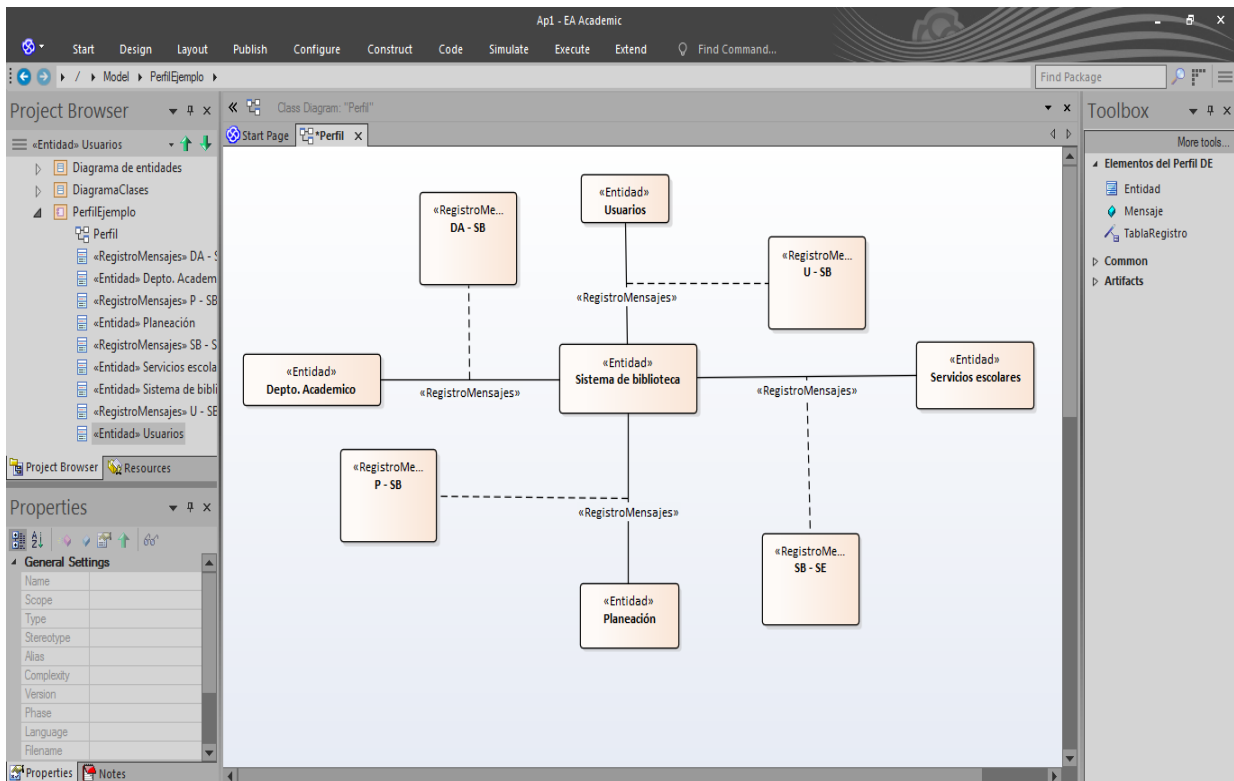


Figura B.1 1 Generación de relaciones (nivel aplicación)

3. Crear mensajes

Para agregar los mensajes se utiliza el elemento Mensaje definido en la caja de herramientas del Perfil UML. Cada mensaje posee propiedades propias, que fueron definidas previamente en la sección B.3.1. Identificación de los elementos y sirven como datos de entrada para esas propiedades.

En la figura B.12 y B.13 se presenta la definición de los mensajes.

B.2.3. Resultado

El resultado obtenido es el diagrama de entidades a nivel aplicación (ver figura B.14).

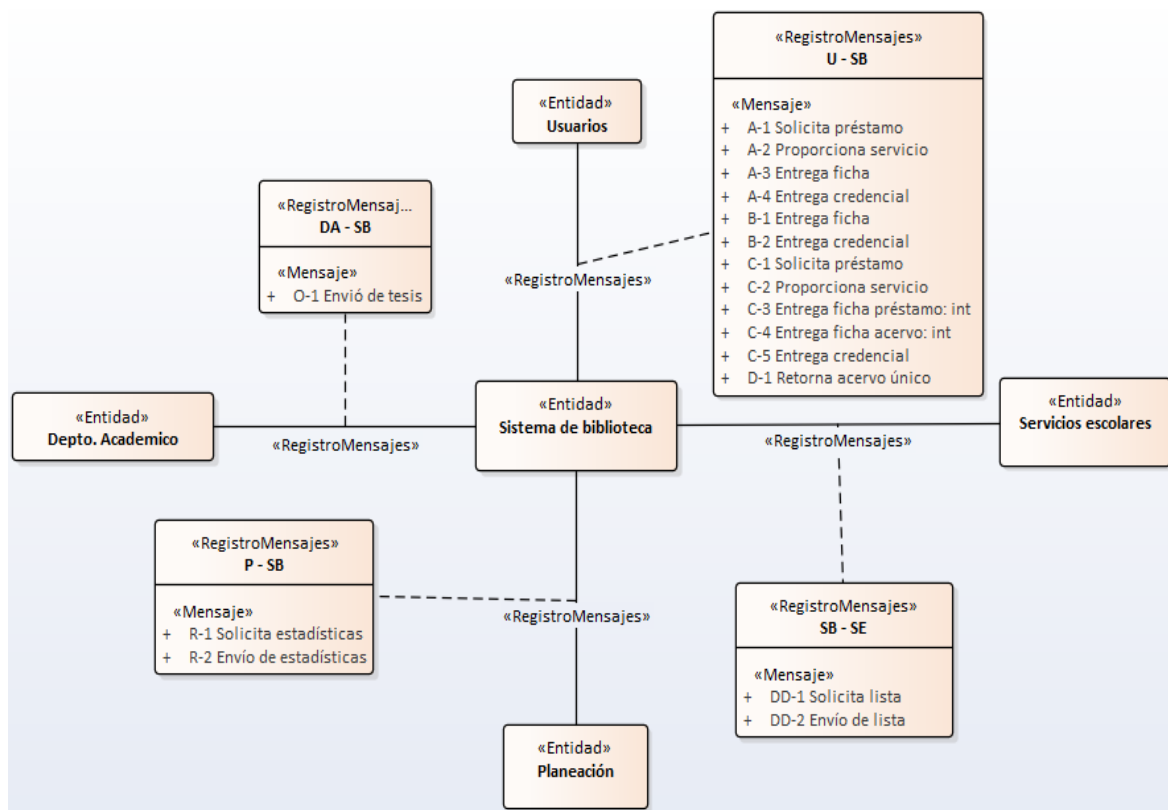


Figura B.14 Diagrama de entidades a nivel aplicación