



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN SISTEMAS E INFORMÁTICA**

**TEMA: ESTUDIO DE LA HERRAMIENTA ACCELEO Y SUS
BENEFICIOS RESPECTO A LA PROGRAMACIÓN TRADICIONAL**

AUTOR: SALVADOR MAYORGA, JULIA STEPHANIE

DIRECTORA: ING. HINOJOSA RAZA, CECILIA MILENA

CODIRECTOR: ING. CAMPAÑA ORTEGA, EDUARDO MAURICIO

SANGOLQUÍ, DICIEMBRE 2015



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

CERTIFICACIÓN

Certifico que el trabajo de titulación, “ESTUDIO DE LA HERRAMIENTA ACCELEO Y SUS BENEFICIOS RESPECTO A LA PROGRAMACIÓN TRADICIONAL” realizado por la señorita **Julia Stephanie Salvador Mayorga**, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos. Técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autoriza a la señorita **Julia Stephanie Salvador Mayorga** para que lo sustente públicamente.

Sangolquí, 09 de Diciembre del 2015

ING. CECILIA HINOJOSA

DIRECTORA

ING. MAURICIO CAMPAÑA

CODIRECTOR



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

AUTORÍA DE RESPONSABILIDAD

Yo, **Julia Stephanie Salvador Mayorga**, con cédula de identidad N° 2000069506, declaro que este trabajo de titulación **“ESTUDIO DE LA HERRAMIENTA ACCELEO Y SUS BENEFICIOS RESPECTO A LA PROGRAMACIÓN TRADICIONAL”** ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, 09 de Diciembre 2015

Julia Stephanie Salvador Mayorga

C.C. 2000069506



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

AUTORIZACIÓN

Yo, **Julia Stephanie Salvador Mayorga**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la Institución el presente trabajo de titulación **“ESTUDIO DE LA HERRAMIENTA ACCELEO Y SUS BENEFICIOS RESPECTO A LA PROGRAMACIÓN TRADICIONAL”** cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, 09 de Diciembre del 2015

Julia Stephanie Salvador Mayorga

C.C. 2000069506

DEDICATORIA

El presente trabajo se lo dedico a mi familia.

A mis padres Germán Salvador y Mónica Mayorga por todo su amor, su ayuda, comprensión y apoyo a lo largo de mi vida, por siempre estar conmigo, por todos sus consejos y enseñanzas por nunca dejarme sola, por enseñarme a levantarme, por ser mi guía, por ser la luz en mi camino, por ser las personas con las que siempre puedo contar.

A mi hermano Edwin Salvador por su apoyo y comprensión, por estar a mi lado, por darme ánimos cuando más lo necesito, por cuidarme, por ser un gran hermano y un gran amigo con el que puedo contar toda la vida.

A mis abuelitos Carlos Mayorga y Juana Mayorga, que me enseñaron a luchar, a seguir adelante siendo ellos un gran ejemplo de superación, de persistencia, aunque ya no están a mi lado; sus enseñanzas, consejos, apoyo y todo su amor siempre estarán conmigo, y siempre vivirán en mi corazón.

A todos mis maestros que por sus enseñanzas he logrado alcanzar el sitio en que me encuentro

A todos ellos muchas gracias los quiero mucho.

AGRADECIMIENTO

Quiero agradecer a:

Dios por guiar mi camino y darme la oportunidad de llegar hasta donde he llegado y por darme salud.

A mi Directora de Tesis la Ing. Cecilia Hinojosa que me ha brindado un gran apoyo para alcanzar esta meta.

A mi Director de Carrera el Ing. Mauricio Campaña por todo el apoyo brindado a lo largo de mi carrera.

A mi familia y amigos por apoyarme y darme ánimos para continuar y alcanzar mis metas, por estar siempre pendientes de mí y brindarme todo su cariño.

A mi familia Amarum, en especial a mi Sensei Víctor Hugo Proaño por todos sus consejos, por las enseñanzas impartidas y el apoyo brindado, por enseñarme a dar más de mí.

A todos muchas gracias.

“Cinco minutos más, un kilómetro más, un round más eso es lo que marca la diferencia entre las personas extra -ordinarias de las personas ordinarias”

Amarum

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN	I
AUTORÍA DE RESPONSABILIDAD	II
AUTORIZACIÓN.....	III
DEDICATORIA	IV
AGRADECIMIENTO	V
ÍNDICE DE CONTENIDOS	VI
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS.....	X
RESUMEN.....	XI
ABSTRACT.....	XII
CAPÍTULO 1 INTRODUCCIÓN	1
1.1. TEMA.....	1
1.2. ANTECEDENTES	1
1.3. PLANTEAMIENTO DEL PROBLEMA.....	2
1.4. JUSTIFICACIÓN	2
1.5. OBJETIVOS	3
1.5.1. GENERAL	3
1.5.2. ESPECÍFICOS.....	3
1.6. ALCANCE	3
CAPÍTULO 2 MARCO TEÓRICO DE REFERENCIA	5
2.1. METODOLOGÍA	5
2.1.1. METODOLOGÍA DE INVESTIGACIÓN	5
2.1.2. DESARROLLO DE SOFTWARE	6
2.1.2.1. ARQUITECTURA DIRIGIDA POR MODELOS – MDA.....	6

2.1.2.2. MODELOS MDA	8
2.1.2.2.1. MODELO INDEPENDIENTE DE LA COMPUTACIÓN – CIM.....	9
2.1.2.2.2. MODELO INDEPENDIENTE DE LA PLATAFORMA – PIM	9
2.1.2.2.3. MODELO ESPECÍFICO DE PLATAFORMA – PSM	9
2.2. PLATAFORMA DE DESARROLLO	10
2.2.1. ACCELEO	10
2.1.1. NETBEANS IDE	12
2.1.2. MYSQL.....	13
2.1.3. STARUML.....	13
2.1. ORGANIZACIÓN CORAZONES UNIDOS	14
2.2. TRABAJOS RELACIONADOS (BUSCAR TRABAJOS CON ACCELEO).....	15
CAPÍTULO 3 ESTUDIO COMPARATIVO.....	17
3.1. DESARROLLO DEL MÓDULO DEL SISTEMA.....	17
3.2. RESULTADOS OBTENIDOS.....	17
3.2.1. ACCELEO	17
3.2.2. NETBEANS – JAVA	30
3.3. CUADRO COMPARATIVO	37
CAPÍTULO 4 DESARROLLO DEL SISTEMA	40
4.1. MODELO INDEPENDIENTE DE LA COMPUTACIÓN – CIM	40
4.1.1. DIAGRAMA DE CASOS DE USO.....	41
4.1.2. REQUISITOS FUNCIONALES	42
4.2. MODELO INDEPENDIENTE DE LA PLATAFORMA – PIM.....	56
4.3. MODELO ESPECÍFICO DE PLATAFORMA – PSM.....	56
4.4. IMPLEMENTACIÓN DEL SISTEMA	58
4.5. PRUEBAS DEL SISTEMA	59
CAPÍTULO 5 CONCLUSIONES Y RECOMENDACIONES	61
5.1. CONCLUSIONES.....	61
5.2. RECOMENDACIONES	63
BIBLIOGRAFÍA.	64

ÍNDICE DE FIGURAS

Figura 1. Investigación Aplicada.....	5
Figura 2. Mapa Mental de MDA.....	6
Figura 3. Modelos MDA.....	8
Figura 4. Logotipo Acceleo. Fuente: Eclipse	11
Figura 5. Resumen de la evolución Acceleo.	12
Figura 6. Objetivos de la Asociación “Corazones Unidos”	15
Figura 7. Proyecto Acceleo.	18
Figura 8. Contenido del Proyecto Acceleo.	19
Figura 9. Proyecto Java en Eclipse.....	19
Figura 10. Modelo.	20
Figura 11. Código para crear Get y Set de la clase automáticamente.	20
Figura 12. Configuración de Acceleo.	21
Figura 13. Archivo Generado.	22
Figura 14. Organización de Paquetes del proyecto Eclipse.....	23
Figura 15. Código de conexión a la Base de Datos.	24
Figura 16. Código Get y Set generados por Acceleo.	25
Figura 17. Diseño de la ventana.....	26
Figura 18. Programación del botón Insertar.	27
Figura 19. Ingreso al módulo/Ventana después de dar clic en Limpiar.	27
Figura 20. Ingreso de datos.....	28
Figura 21. Mensaje después de Insertar.	28
Figura 22. Datos insertados en tabla Representante.	28
Figura 23. Datos buscados por número de cédula y activación del botón Modificar.	29
Figura 24. Modificación de datos.....	29
Figura 25. Mensaje de modificación de datos.	30
Figura 26. Datos modificados en tabla Representante.....	30
Figura 27. Organización de Paquetes del proyecto NetBeans.....	31
Figura 28. Clase de conexión a la Base de Datos.....	31
Figura 29. Clase Get y Set de la tabla representante.....	32
Figura 30. Diseño de la Ventana en NetBeans.	33
Figura 31. Fragmento de la codificación del botón Buscar.....	34
Figura 32. Ingreso al módulo/Ventana después de dar clic en Limpiar – NetBeans.....	34
Figura 33. Ingreso de datos Ventana NetBeans.....	35
Figura 34. Mensaje de Insertado Exitoso Ventana NetBeans.	35
Figura 35. Datos insertados en tabla Representante mediante NetBeans.	35
Figura 36. Datos buscados por número de cédula y activación del botón Modificar Ventana NetBeans.	36
Figura 37. Modificación de datos Ventana NetBeans.....	36
Figura 38. Mensaje de modificación de datos Ventana NetBeans.	37
Figura 39. Datos modificados en tabla Representante.....	37
Figura 40. Diagrama de Casos de Uso.	41
Figura 41. Diagrama de Clases.....	56
Figura 42. Diagrama Entidad – Relación.....	57

Figura 43. Modelo UML – Acceleo. 58

ÍNDICE DE TABLAS

Tabla 1 Cuadro comparativo Acceleo / Java (NetBeans)	38
Tabla 2 CU01 Ingresar al Sistema	42
Tabla 3 CU02 Crear Usuario	43
Tabla 4 CU03 Cambiar contraseña	44
Tabla 5 CU04 Crear Representante	45
Tabla 6 CU05 Buscar Representante	46
Tabla 7 CU06 Modificar Representante	47
Tabla 8 CU07 Consultar Representante	48
Tabla 9 CU08 Crear Paciente	49
Tabla 10 CU09 Buscar Paciente	50
Tabla 11 CU10 Modificar Paciente	51
Tabla 12 CU11 Consultar Paciente	52
Tabla 13 CU12 Crear Evaluación	53
Tabla 14 CU13 Buscar Historial	54
Tabla 15 CU14 Modificar Evaluación	55
Tabla 16 Prueba de Casos de Uso	59
Tabla 17 Prueba de Caja Negra	60

RESUMEN

La Ingeniería de Software procura mejorar la productividad de los equipos de desarrollo y la calidad del producto software resultante. Con estos objetivos se propone la Arquitectura Dirigida por Modelos – MDA, la que se fundamenta en el modelado y la transformación de modelos, hasta el nivel de abstracción requerido y dirigen de forma automatizada el desarrollo de código, mediante el uso de herramientas MDA. Los componentes de MDA son: Modelo Independiente de la Computación (CIM), Modelo Independiente de la Plataforma (PIM), y Modelo Específico de la Plataforma (PSM). El presente trabajo se planteó como objetivo realizar el estudio de la herramienta Acceleo, mediante el desarrollo del caso práctico “Sistema de Gestión de Fichas Médicas para la Asociación Corazones Unidos” a fin de comprobar los beneficios ofrecidos por la misma. Para el efecto se desarrolló un módulo del Sistema utilizando Acceleo y JAVA y se realizó un estudio comparativo para obtener las conclusiones acerca de la herramienta. Con la ayuda de la investigación aplicada se obtuvo la información necesaria acerca de MDA y sus herramientas y para el desarrollo del software se utilizaron los modelos de MDA. Los resultados obtenidos permitieron evidenciar las ventajas el uso de la herramienta Acceleo, entre otros: el software se desarrolló en menos tiempo, el código autogenerado puede ser utilizado en cualquier plataforma que utilice como lenguaje JAVA, y en el caso de haber actualizaciones en los modelos el código se actualizará automáticamente. Por lo que se puede concluir que la herramienta de MDA cumple con los beneficios ofrecidos.

PALABRAS CLAVE:

**ARQUITECTURA DIRIGIDA POR
MODELOS – MDA**

**MODELO INDEPENDIENTE DE LA
PLATAFORMA – PIM**

**MODELO INDEPENDIENTE DE LA
COMPUTACIÓN – CIM**

**MODELO ESPECÍFICO DE LA
PLATAFORMA – PSM**

ACCELEO

ABSTRACT

Software Engineering aims to improve the productivity of development teams and the quality of the resulting software product. With these objectives the Model Driven Architecture aims - MDA, which is based on modeling and model transformation to the level of abstraction required and run automated code development, using MDA tools. MDA components are: Computer Independent Model (CIM), Platform Independent Model (PIM) and Platform Specific Model (PSM). This paper was presented as objective the study of the Acceleo tool by developing the case "Records Management System for the United Hearts Medical Association" in order to ascertain the benefits offered by it. For this purpose a system module using Acceleo and JAVA developed and a comparative study was conducted to draw conclusions about the tool. With the help of applied research necessary information about MDA and was obtained and tools for software development MDA models were used. The results obtained allowed to demonstrate the advantages the use of Acceleo tool include: the software was developed in less time, the auto-generated code can be used on any platform that uses language like Java, and in the case of updates models the code is automatically updated. As it can be concluded that MDA tool meets the benefits offered.

KEYWORDS:

ARCHITECTURE DRIVEN MODEL – MDA

COMPUTER INDEPENDENT MODEL – CIM

PLATFORM INDEPENDENT MODEL – PIM

PLATFORM SPECIFIC MODEL – PSM

ACCELEO

CAPÍTULO 1

INTRODUCCIÓN

1.1. Tema

Estudio de la herramienta Acceleo y sus beneficios respecto a la programación tradicional

1.2. Antecedentes

Entre las tendencias dominantes en las herramientas tecnológicas, se encuentran aquellas que apoyan el desarrollo impulsado por modelos con énfasis en el diseño impulsado por arquitecturas. Esto se debe a que los modelos son los actores principales en el desarrollo del software. (Pressman, 2010)

La Arquitectura Dirigida por Modelos MDA propone que el desarrollo de software se debe basar en modelos, los cuales guiarán todo el proceso del desarrollo que serán transformados en código de manera automática. El objetivo de esta arquitectura es obtener beneficios importantes en aspectos como: productividad, portabilidad, mantenimiento, documentación y reutilización.

Para MDA se han desarrollado herramientas que tienen la capacidad de transformar los modelos de negocios en aplicaciones capaces de ser ejecutadas. Entre las herramientas MDA se encuentra Acceleo.

Acceleo es una aplicación pragmática basado en el estándar MOF Model to Text Transformation to Lenguaje (MOFM2T) del Grupo de Gestión de Objetos (Object Management Group – OMG).

Acceleo es un generador de código que pertenece a la Organización Eclipse, él cual es de fácil utilización, ofrece a los desarrolladores ayuda en el manejo del ciclo de vida de los generadores de código, posee una alta capacidad de personalizar. (The Eclipse Foundation, 2015)

1.3. Planteamiento del Problema

La programación tradicional es dependiente de la tecnología y la plataforma utilizada, lo cual ha ocasionado que el código no se pueda migrar de una plataforma a otra, pero en el ámbito de la informática existe una acelerada evolución tecnológica, lo que ha dado lugar a la aparición de diversas plataformas y tecnologías; provocando que al momento de cambiar de plataforma o de tecnología, la programación se vea afectada en su totalidad.

Los modelos son utilizados únicamente como parte de la documentación, pero no son parte fundamental de la programación ya que cuando hay un cambio en la programación los modelos no son actualizados inmediatamente, y en algunos casos los modelos no son modificados de acuerdo a los cambios realizados.

Todo esto provoca que el software presente problemas de productividad, portabilidad, reutilización, logrando que su mantenimiento y documentación no sea la adecuada.

1.4. Justificación

Las herramientas MDA buscan obtener beneficios en los aspectos de: productividad, portabilidad, mantenimiento y documentación. En este caso se utilizará la herramienta Acceleo para demostrar que brinda los beneficios mencionados.

Acceleo es una herramienta que genera código a través de modelos, logrando así que si existe un cambio en los modelos, la programación también cambia.

Mediante el estudio comparativo se podrá corroborar el verdadero nivel de mejora en los factores arriba indicados. Lo cual será un referente para futuros desarrollos.

1.5. Objetivos

1.5.1. General

Realizar el estudio de la herramienta Acceleo y desarrollar el caso práctico “Sistema de Gestión de Fichas Médicas para la Asociación Corazones Unidos” a fin de comprobar los beneficios ofrecidos por la herramienta.

1.5.2. Específicos

- Estudiar el marco teórico de MDA
- Analizar la herramienta Acceleo
- Desarrollar el Sistema de Gestión de Fichas Médicas para la Asociación Corazones Unidos utilizando JAVA, en donde se obtendrá datos para la comparación
- Desarrollar el Sistema de Gestión de Fichas Médicas para la Asociación Corazones Unidos utilizando Acceleo
- Realizar el estudio comparativo entre los aplicativos desarrollados JAVA y Acceleo y obtener las respectivas conclusiones

1.6. Alcance

Se realizará un estudio de MDA y su herramienta Acceleo, se desarrollará el caso práctico del Sistema de Gestión de Fichas Médicas para la Asociación

Corazones Unidos usando Acceleo y JAVA, con lo que se obtendrán datos para realizar la comparación entre los desarrollos, y se representará los resultados en un cuadro comparativo de las herramientas, en el cual se evidenciará el tiempo de programación empleado en el desarrollo, las líneas de código generadas, los errores y fundamentalmente se realizará el análisis de productividad, portabilidad, mantenimiento y documentación.

El software se desarrollará tanto en Acceleo como en JAVA utilizando los modelos de MDA:

- Modelo Independiente de la Computación – CIM
- Modelo Independiente de la Plataforma – PIM
- Modelo Específico de Plataforma – PSM

CAPÍTULO 2

MARCO TEÓRICO DE REFERENCIA

2.1. Metodología

Para el desarrollo de la tesis se utilizará como metodología de investigación la investigación científica y para el desarrollo del Software se utilizará la Arquitectura Dirigida por Modelos – MDA.

2.1.1. Metodología de Investigación

Se aplicará la Investigación Aplicada para el estudio de Acceleo, con el objetivo de investigar el marco teórico, los fundamentos, su aplicación y utilización. En la Figura 1. se explica acerca del concepto de la Investigación Aplicada y las formas de realizar la investigación.

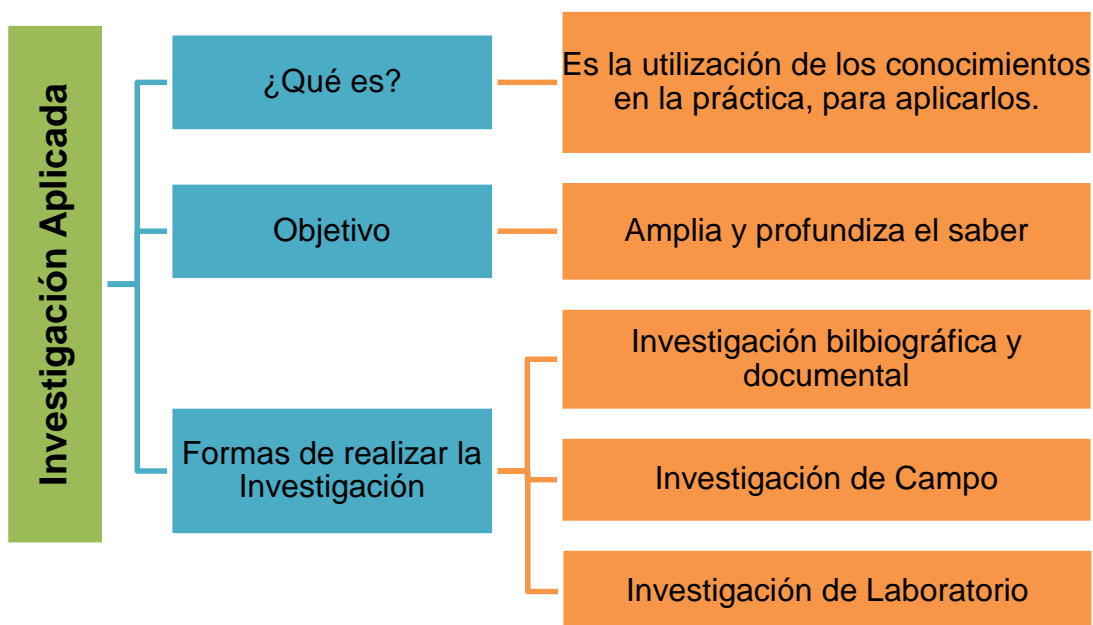


Figura 1. Investigación Aplicada.

2.1.2. Desarrollo de Software

2.1.2.1. Arquitectura Dirigida por Modelos – MDA

En la Figura 2 se observa un resumen de MDA

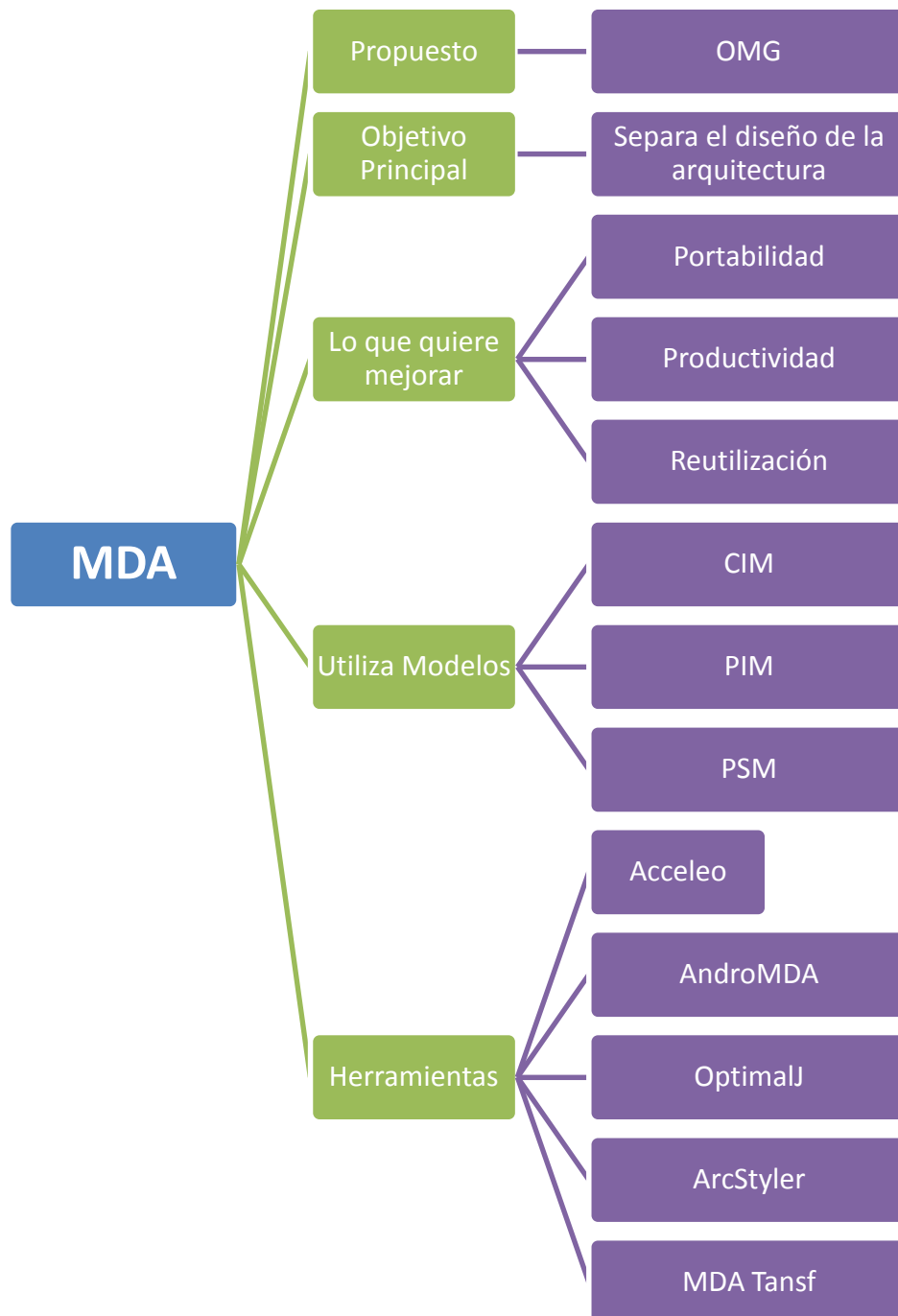


Figura 2. Mapa Mental de MDA.

La Arquitectura Dirigida por Modelos MDA por sus siglas en Inglés Model Driven Architecture, es un acercamiento al diseño de software, propuesto y patrocinado por el Object Management Group (OMG), en donde su principal objetivo es separar el diseño de la arquitectura y la tecnología de construcción, y realizar el desarrollo en diferentes lenguajes específicos como Java, C#, Python, etc. (Colaboradores de Object Management Group, 2015)

MDA se asegura de que el modelo sea independiente de la plataforma y que éste sea el encargado de guiar todo el proceso de desarrollo, representa un diseño conceptual en donde se especifica los requerimientos funcionales, el modelo se mantiene a pesar de los cambios que se produzcan en las tecnologías de fabricación y en las arquitecturas software. (Colaboradores OMG, 2015)

“MDA es una propuesta de la OMG para el desarrollo de software. La idea principal de MDA es especificar sistemas con modelos independientes de la plataforma, separando la especificación de la implementación 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. “ (Carrera Sepúlveda, Varas Contreras, & Urrutia Sepúlveda, 2010)

MDA tiene como objetivo mejorar:

- Portabilidad
- Productividad
- Reutilización

MDA propone tres modelos para el desarrollo, los cuales son:

- Modelo Independiente de la Computación – CIM
- Modelo Independiente de la Plataforma – PIM
- Modelo Específico de Plataforma – PSM

2.1.2.2. Modelos MDA

MDA se basa en la utilización de tres modelos que son: Modelo Independiente de la Computación – CIM, Modelo Independiente de la Plataforma – PIM, y, Modelo Específico de Plataforma – PSM; como se observa en la figura 3.

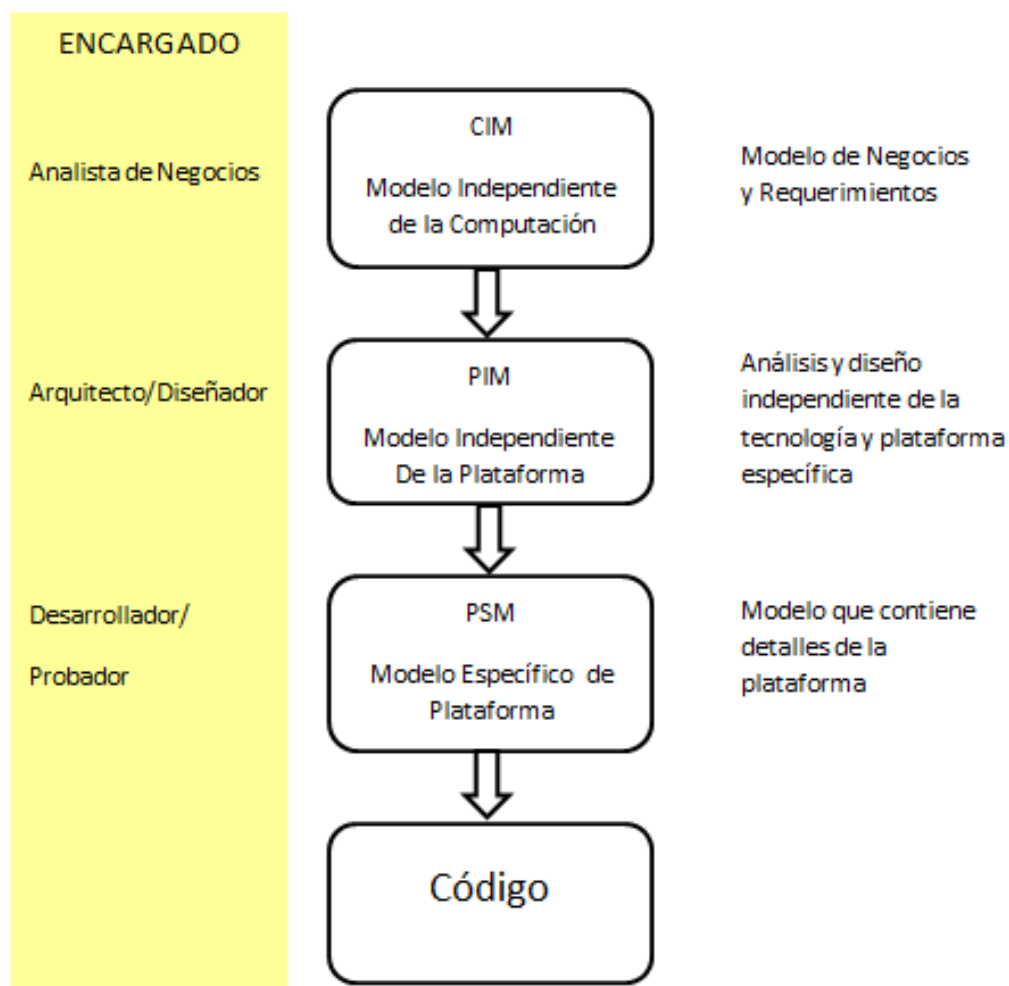


Figura 3. Modelos MDA.

2.1.2.2.1. Modelo Independiente de la Computación – CIM

El Modelo Independiente de la Computación es creado por expertos en el dominio y son aquellos que no muestran detalles de la estructura del sistema y representan el dominio del problema.

El CIM es un modelado de negocios y requerimientos, útil para comprender el problema y como fuente para los otros modelos. (Colaborador guest83f0d26 de Slide Share, 2009)

2.1.2.2.2. Modelo Independiente de la Plataforma – PIM

Los Modelos Independientes de la Plataforma son creados por arquitectos de software y describen una solución de software sin detalles de la plataforma y representa la estructura, funcionalidad y restricciones del sistema, teniendo en cuenta que el nivel de abstracción del modelo debe ser bueno de tal manera que no cambie con ninguna plataforma.

El PIM surge como resultado del análisis y diseño, y representa la lógica del sistema y las interacciones con el mundo exterior, sin detalles de la tecnología a implementar y como se adapta a una plataforma específica. (Colaborador guest83f0d26 de Slide Share, 2009)

2.1.2.2.3. Modelo Específico de Plataforma – PSM

Los Modelos Específico de Plataforma son creados por Diseñadores/Programadores y son aquellos modelos derivados de PIM que contienen detalles de la plataforma que se usará, especificando los detalles que el sistema implementará en una plataforma específica. El PSM se construye entre el diseño y la codificación. (Colaborador guest83f0d26 de Slide Share, 2009)

2.2. Plataforma de desarrollo

Se han desarrollado varias herramientas para implementar MDA, esto es realizar la transformación automatizada de los modelos CIM, PIM y PSM. Algunas herramientas, a pesar de haber presentado varias ventajas como sencillez y buen rendimiento, no tuvieron la acogida que se esperaba en el mercado, como por ejemplo se pueden citar: OptimalJ, ArcStyler, AndroMDA. Actualmente algunas herramientas que se encuentran vigentes son Acceleo, Sparx System.

Para el desarrollo del presente trabajo se utilizará: Acceleo, NetBeans; como base de datos: MySQL; para el desarrollo de los modelos: StarUML.

2.2.1. Acceleo

Acceleo actualmente es un Plug-in de Eclipse de código abierto, sus inicios se dieron en la empresa francesa Obeo con un crecimiento constante, este Plug-in es un sistema de generación de código basado en el estándar MOF Model to Text Transformation to Language (MOFM2T) del Grupo de Gestión de Objetos (Object Management Group – OMG).

“MOF es un estándar de OMG donde los modelos se pueden exportar desde una aplicación, importado a otro, transportados a través de una red, almacenada en un repositorio y luego recuperados, dictada en diferentes formatos (incluyendo XMI, formato estándar basado en XML de OMG para la transmisión y almacenamiento de modelo), transformado, y utilizado para generar el código de aplicación.” (Colaboradores OMG, 2015)

Acceleo al igual que Eclipse poseen una Licencia Pública de Eclipse (EPL) la cual no tiene costo alguno, es compatible con las plataformas Linux, Mac OS y Microsoft Windows (XP, Vista y 7). La marca Acceleo y su logotipo (Figura 4) eran la marca de la empresa Obeo, empresa creadora de Acceleo,

al momento que Acceleo fue aceptado en la Fundación Eclipse, su marca y logotipo fue transferido a la misma. (The Eclipse Foundation, 2015)

Acceleo se encuentra muy bien integrado en Eclipse e incluye las características habituales como el coloreado de sintaxis, control de trazabilidad, la cual permite encontrar los elementos del modelo, las partes utilizadas del generador y el código finalmente generado. (Ruiz Rube, Procesadores de Lenguaje II: Universidad de Cádiz, 2013)



Figura 4. Logotipo Acceleo. Fuente: Eclipse

Historia de Acceleo:

Acceleo nace en la empresa francesa Obeo el 01 de Abril del 2006 bajo la Licencia Pública GNU siendo éste compatible con Eclipse 3.0 y 3.1, se encontraba disponible en su página web Acceleo.org, tanto la página web como el Plug-in fueron lanzado conjuntamente; Acceleo 1.1 fue lanzado semanas más tarde, pero éste era compatible con Eclipse 3.2; Acceleo 1.2 fue lanzado bajo una nueva licencia, la cual era la Licencia Pública de Eclipse – EPL, ésta licencia era utilizada por los proyectos de la Fundación Eclipse; Acceleo 2 fue liberada el 5 de Junio de 2007, junto a la página web planet.acceleo.org, la cual contenía artículos que fueron creados por los miembros de la comunidad Acceleo y módulos de repositorios con generadores de código basado en Acceleo 2, ésta versión conjuntamente con sus subversiones fue la última versión liberada por la empresa Obeo, en el año 2009 el proyecto Acceleo es aceptado en la Fundación Eclipse.

Acceleo 3 utiliza el nuevo estándar de la OMG para la transformación de modelo a texto (MOFM2T), Acceleo 3.6 última versión liberada en la página de Eclipse el 08 de Junio del 2015.

En la figura 5 se observa un resumen de la evolución de Acceleo.

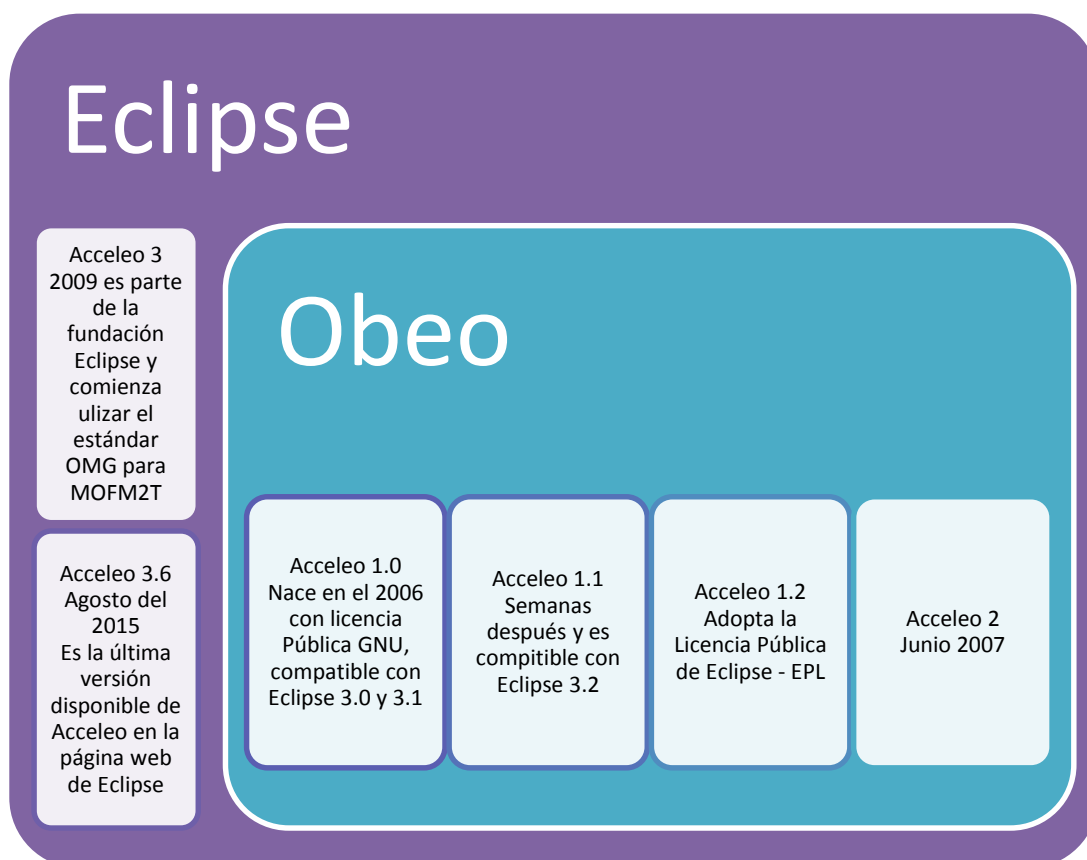


Figura 5. Resumen de la evolución Acceleo.

2.1.1. NetBeans IDE

Es un proyecto de entorno de desarrollo, de código abierto que fue fundado por Sun Microsystems en Junio del 2000, ésta herramienta tiene como objetivo permitir a los programadores escribir, compilar depurar y ejecutar programas, se encuentra escrito en Java, pero sirve para cualquier otro tipo de lenguaje de programación. Es un entorno multiplataforma de licencia GPL y es multilenguaje. (Oracle Corporation, 2015)

Actualmente se encuentra disponible la versión NetBeans 8.0.1, el cual puede ser descargado de manera gratuita de la página oficial de NetBeans netbeans.org/downloads.

2.1.2. MySQL

MySQL es software de Gestión de Bases de Datos de código abierto, fue fundada y desarrollada en Suecia por dos suecos y un finlandés : David Axmark , Allan Larsson , y Michael " Monty " Widenius desarrollado en su mayoría en ANSI C, recientemente adquirido por Oracle, puede utilizarse con productos compatibles con la licencia GNU GLP, y aquellas empresas que deseen incorporarlo en productos privativos deben comprar a la empresas una licencia para su uso.

MySQL es un software de gestión de base de datos más utilizados, ya que las organizaciones lo utilizan por su velocidad, fiabilidad y facilidad de uso. (MySQL, 2013)

Cuando se utiliza MySQL Worbench se puede realizar una base de datos a partir del diagrama Entidad – Relación, cumpliendo así el objetivo MDA que se basa en modelos para generar código, que en éste caso se transforma el diagrama Entidad – Relación a código SQL.

2.1.3. StarUML

StarUML es una herramienta de modelado UML/MDA, con las siguientes características: es de código abierto, rápido, flexible, extensible, de fácil utilización y desarrollado principalmente en Delphi.

“StarUML apoya a MDA y proporciona muchas variables de personalización como el perfil, Enfoque, Modelo Marco, NX (extensión de la notación), código de MDA, documento de plantilla y etc.” (StarUML, 2013)

Con StarUML a partir de los modelos se puede generar: el documento de Especificación de Casos de Uso, y las clases en .java, en las clases autogeneradas se puede observar los atributos declarados y las operaciones declaradas.

2.1. Organización Corazones Unidos

La Asociación “Corazones Unidos”, es una asociación de padres con hijos de capacidades especiales, la cual está constituida legalmente y en sus estatutos constan los fines específicos de la misma:

“Art. 5.- FINES ESPECÍFICOS.- La Asociación tendrá como fines específicos los siguientes:

- a. Ayuda para los asociados y sus familiares con capacidades diferentes
- b. Desarrollar programas de sensibilización, difusión, concientización sobre las personas con capacidades diferentes
- c. Promover o presentar actos culturales, científicos, deportivos o sociales pidiendo las exoneraciones y demás favorecimientos a que haya lugar
- d. Coordinar con otras instituciones públicas o privadas afines, sean estas locales, nacionales o internacionales, a fin de realizar actividades que les permitan alcanzar los objetivos de la organización
- e. Velar por el cumplimiento de la legislación que ampara a las personas con capacidades diferentes y sus familias
- f. Suscribir toda clase de contratos, convenios con instituciones públicas o privadas, con el fin de fomentar actividades que desarrollen la recreación, motivación, solidaridad, inclusión laboral, difusión de los valores y derechos e intercambios interculturales dentro de la sociedad,

de las personas con capacidades diferentes.” (Asociación Corazones Unidos , 2010)

En la figura 9 se puede observar un resumen de los objetivos que tiene la asociación.



Figura 6. Objetivos de la Asociación “Corazones Unidos”.

2.2. Trabajos Relacionados (buscar trabajos con Acceleo)

En el trabajo de aplicación de MDA a la Ingeniería del Conocimiento (Cañadas, Túnez, & Palma, 2004) se utiliza MDA con el objetivo de transformar los modelos generados en la Ingeniería del Conocimiento a

modelos UML, para conseguir que el desarrollo del software sea lo más sistemático posible, evidenciando la ventaja de los modelos MDA.

En el proyecto de análisis del enfoque MDA empleando AndroMDA para una caso de estudio (Román Alcaraz, 2012), en el cual se analiza la herramienta AndroMDA, realizando un caso de estudio con el cual se realiza su proceso de modelado y obtiene resultados experimentales entre AndroMDA y programación tradicional.

En el estudio comparativo de dos herramientas MDA: OptimalJ y ArcStyler (García Molina, Rodríguez, Menárguez, Ortín, & Sánchez, 2004); se realiza un análisis de propiedades extraídas de la especificación de MDA, desarrollando la aplicación Pet Store con cada herramienta y se realiza un estudio comparativo a partir de las mismas propiedades.

En el trabajo de Translating UML State Machines to Coloured Petri Nets Using Acceleo: A Report (Etienne, Mohamed Mahdi, & Choppy, 2014); se realiza la transformación de Modelo a Texto de Acceleo para redes Petri coloreadas, pero al utilizar la herramienta es limitada a ciertas características, pero la facilidad para agregar variables, funciones es una ventaja de Acceleo y las desventajas son la falta de documentación y ejemplos en línea, por lo que los autores no quedaron totalmente convencido de la utilización del mismo.

CAPÍTULO 3

ESTUDIO COMPARATIVO

3.1. Desarrollo del módulo del sistema

Para el desarrollo de este capítulo se desarrollará un módulo tanto en Eclipse – Acceleo como en NetBeans – Java.

El módulo seleccionado es: **Módulo Representante**.

3.2. Resultados obtenidos

3.2.1. Acceleo

Requisitos:

Para utilizar Acceleo se debe cumplir con los siguientes paso: descargar Eclipse de la página oficial: <http://www.eclipse.org/downloads/>; descomprimir el archivo y acceder en el programa.

Descargar los Plug-in: UML y Acceleo; utilizando la opción Install New Software que es proporcionado por Eclipse.

Una vez realizado esta preparación se puede comenzar a trabajar con Acceleo.

Creación del Módulo Representante en Acceleo:

Primero se crea un proyecto Acceleo (figura 7), el cual contendrá el modelo, las reglas y el código generado como se puede observar en la figura 8, y un proyecto Java en Eclipse en dónde se desarrollará el módulo y tendrá la conexión a la base de datos (figura 9).

En el Anexo 1, se encuentra los pasos para la creación del módulo.

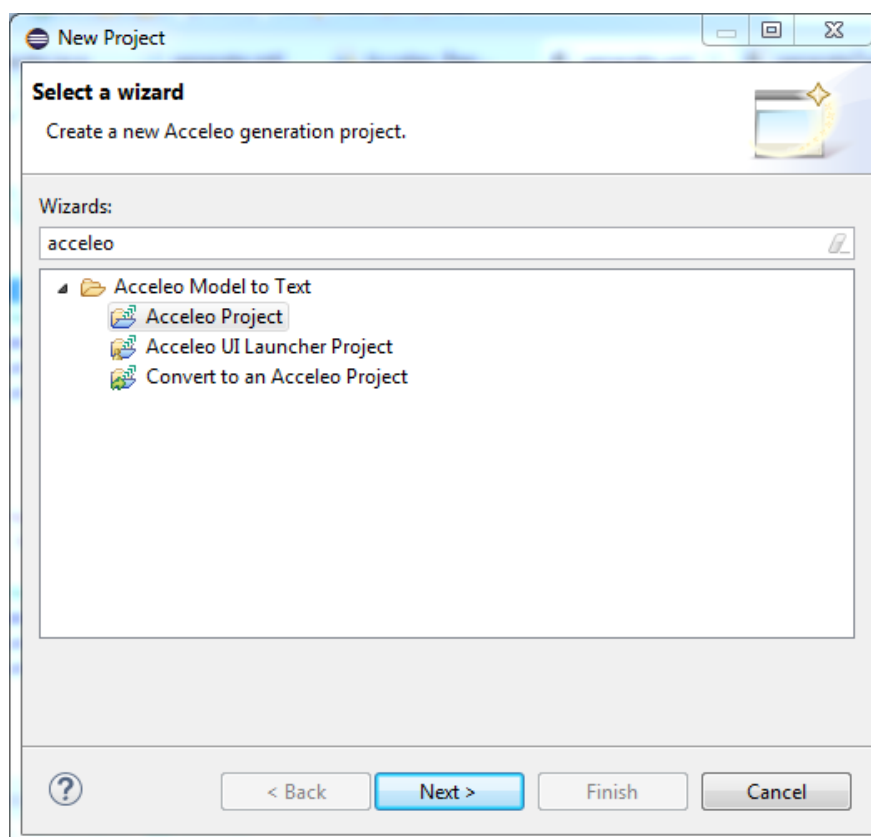


Figura 7. Proyecto Acceleo.

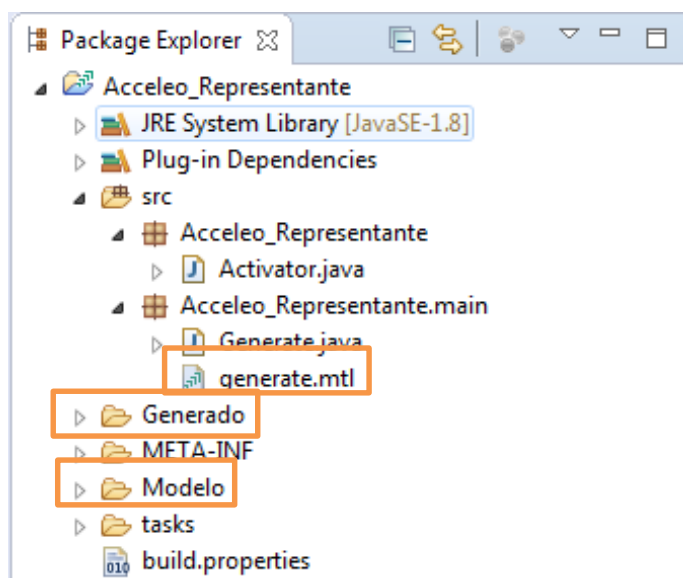


Figura 8. Contenido del Proyecto Acceleo.

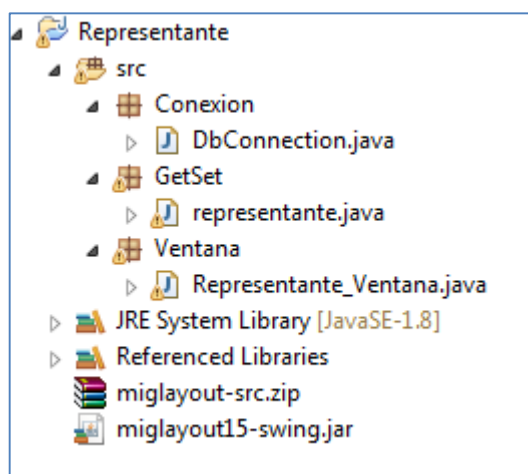


Figura 9. Proyecto Java en Eclipse.

Creación del Modelo:

Para generar el código con la ayuda de Acceleo, primero se debe crear el modelo, en donde deben constar las clases, los atributos y el tipo de dato de los atributos (figura 10).

En el Anexo A1 se encuentra detallado los pasos a seguir para la creación del Modelo.

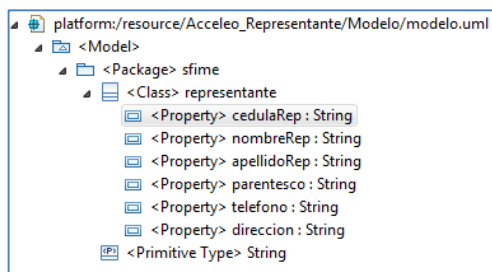


Figura 10. Modelo.

Generar Código:

Una vez que se ha creado el modelo UML que se utilizará para el proyecto, hay que dirigirse a la carpeta src del proyecto al paquete .main y al archivo generate.mtl, el cual es una plantilla en donde se procederá a agregar las reglas de generación de Acceleo, éste archivo tiene como objetivo generar el código automático para los Get y Set de los atributos de la clase (Fig. 11); éste archivo lo podemos modificar las veces que sea necesario y volver a generar el código con los cambios realizados.

```

1 [comment encoding = UTF-8 /]
2 [module generate('http://www.eclipse.org/uml2/5.0.0/UML')]
3
4
5 [template public generateElement(aClass : Class)]
6 [comment @main/]
7 [file (aClass.name, false, 'UTF-8')]
8   public class [aClass.name/]{
9     [generarDefAtributoa(aClass)/]
10    public void [aClass.name/]() {}
11    [generarGetSetAtributos(aClass)/]
12  }
13 [/file]
14 [/template]
15
16 [template public generarDefAtributoa(aClass : Class)]
17   [for (atributo : Property | aClass.attribute) separator('\n')]
18   private [atributo.type.name/] [atributo.name/];
19 [/for]
20 [/template]
21
22 [template public generarGetSetAtributos(aClass : Class)]
23   [for (atributo : Property | aClass.attribute) separator('\n')]
24   public [atributo.type.name/] get[atributo.name.toUpperFirst()/]() {
25     return this.[atributo.name/];
26   }
27 [/for]
28
29   [for (atributo : Property | aClass.attribute) separator('\n')]
30   public void set[atributo.name.toUpperFirst()/] ([atributo.type.name/] [atributo.name/]) {
31     this.[atributo.name/] = [atributo.name/];
32   }
33 [/for]
34 [/template]

```

Figura 11. Código para crear Get y Set de la clase automáticamente.

En ésta plantilla se puede observar que el texto que se encuentra entre “[” “]” son comandos de generación y el texto que se encuentra fuera de ellos es texto que se copiará directamente en el archivo final.

Los [template] son los que realmente hacen el trabajo de generar el código, se puede definir el número de templates que necesitemos, los template public podemos invocarlos desde cualquier otro template, pero el template donde encontramos el comentario [coment @main] es el template principal, el que será el primer template que se invocará. El editor de Acceleo tiene sugerencias incorporadas, lo que facilita el uso de los atributos del objeto.

Cuando se encuentre listo el código que se desea generar se procede a crear el Launcher de Acceleo (configuración de Acceleo, figura 12), éste es el encargado de crear el ambiente de generación e invocar al template principal (template con el comentario @mail), y generar el código según lo especificado en el template (figura 13).

En el Anexo A1 se encuentra detallado los pasos para generar el código.

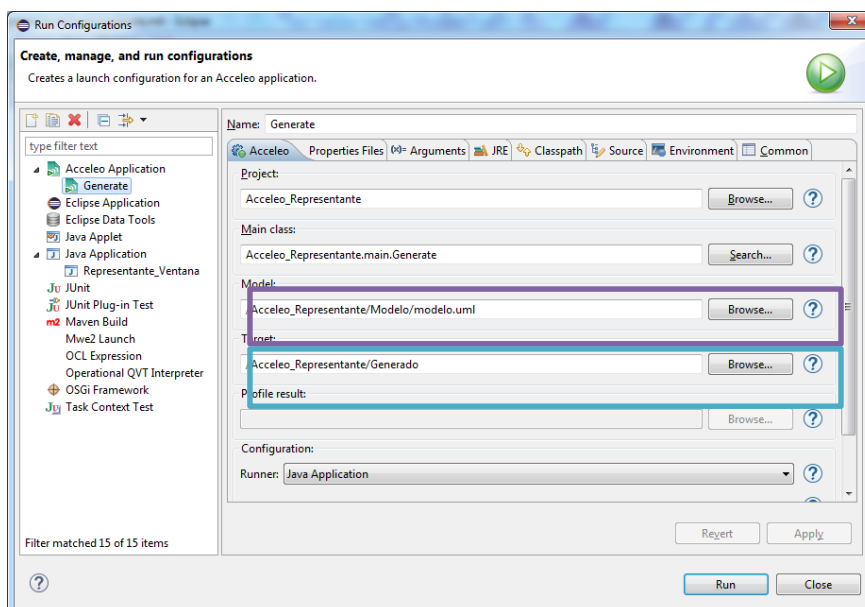


Figura 12. Configuración de Acceleo.

```
1 public class representante{
2     private String cedulaRep;
3
4     private String nombreRep;
5
6     private String apellidoRep;
7
8     private String parentesco;
9
10    private String telefono;
11
12    private String direccion;
13
14    public void representante() {}
15    public String getCedulaRep() {
16        return this.cedulaRep;
17    }
18
19    public String getNombreRep() {
20        return this.nombreRep;
21    }
22
23    public String getApellidoRep() {
24        return this.apellidoRep;
25    }
26
27    public String getParentesco() {
28        return this.parentesco;
29    }
30
31    public String getTelefono() {
32        return this.telefono;
33    }
34
35    public String getDireccion() {
36        return this.direccion;
37    }
38
39    public void setCedulaRep (String cedulaRep) {
40        this.cedulaRep = cedulaRep;
41    }
42
43    public void setNombreRep (String nombreRep) {
44        this.nombreRep = nombreRep;
45    }
46
47    public void setApellidoRep (String apellidoRep) {
48        this.apellidoRep = apellidoRep;
49    }
50
51    public void setParentesco (String parentesco) {
52        this.parentesco = parentesco;
53    }
54
55    public void setTelefono (String telefono) {
56        this.telefono = telefono;
57    }
58
59    public void setDireccion (String direccion) {
60        this.direccion = direccion;
61    }
62
63 }
64
```

Figura 13. Archivo Generado.

Si se realiza algún cambio en el modelo o en el archivo .mtl únicamente se hace clic en Run Launch Acceleo y se corregirá el archivo automáticamente, no hay necesidad de volver a configurar Acceleo.

Creación del módulo Representante en NetBeans

Para crear el módulo de Representante primero se debe crear un proyecto en Eclipse, en donde se estructurará el proyecto en tres paquetes (Fig. 14):

- Paquete de conexión a la Base de Datos
- Paquete con el Get y Set
- Paquete en donde diseñaremos la ventana y su funcionalidad

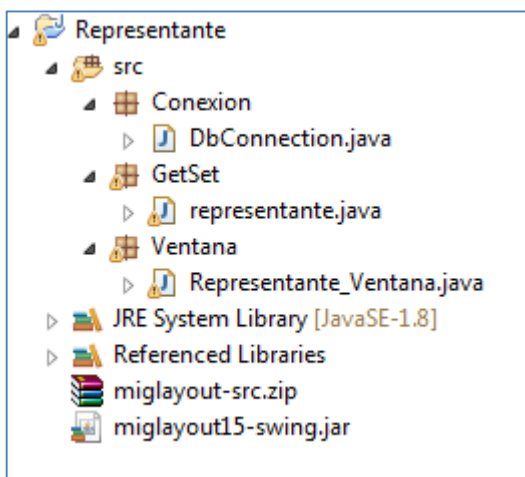


Figura 14. Organización de Paquetes del proyecto Eclipse.

En el paquete de conexión a la Base de Datos se agrega una clase en donde se establece los parámetros de conexión con la base de datos de MySQL (Fig. 15)

```
1 package Conexion;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.SQLException;
6
7 import javax.swing.JOptionPane;
8
9
10 public class DbConexion {
11     /*Parámetros de conexión*/
12     static String login = "root";
13     static String password = "*****";
14     static String url = "jdbc:mysql://localhost/sfime";
15
16     private Connection conexion = null;
17
18     private static String driver="com.mysql.jdbc.Driver";
19
20     public DbConexion(){
21         try {
22             Class.forName(driver);
23             conexion=DriverManager.getConnection(url,login,password);
24         } catch (ClassNotFoundException | SQLException e){
25             JOptionPane.showMessageDialog(null, "No se logró establecer conexión a la Base de Datos");
26         }
27     }
28
29     public Connection getConnection(){
30         return conexion;
31     }
32 }
```

Figura 15. Código de conexión a la Base de Datos.

En el paquete GetSet se añade una clase con el código generado automáticamente por Acceleo (Fig. 16)

```
1 package GetSet;
2
3 public class representante {
4     private String cedulaRep;
5
6     private String nombreRep;
7
8     private String apellidoRep;
9
10    private String parentesco;
11
12    private String telefono;
13
14    private String direccion;
15
16    public void representante() {}
17    public String getCedulaRep() {
18        return this.cedulaRep;
19    }
20
21    public String getNombreRep() {
22        return this.nombreRep;
23    }
24
25    public String getApellidoRep() {
26        return this.apellidoRep;
27    }
28
29    public String getParentesco() {
30        return this.parentesco;
31    }
32
33    public String getTelefono() {
34        return this.telefono;
35    }
36
37    public String getDireccion() {
38        return this.direccion;
39    }
40
41    public void setCedulaRep (String cedulaRep) {
42        this.cedulaRep = cedulaRep;
43    }
44
45    public void setNombreRep (String nombreRep) {
46        this.nombreRep = nombreRep;
47    }
48
49    public void setApellidoRep (String apellidoRep) {
50        this.apellidoRep = apellidoRep;
51    }
52
53    public void setParentesco (String parentesco) {
54        this.parentesco = parentesco;
55    }
56
57    public void setTelefono (String telefono) {
58        this.telefono = telefono;
59    }
60
61    public void setDireccion (String direccion) {
62        this.direccion = direccion;
63    }
64
65 }
```

Figura 16. Código Get y Set generados por Acceleo.

En el paquete Ventana se diseña la ventana que visualizará el usuario (Fig. 17), conjuntamente con las operaciones que se realizarán, las cuales son insertar, buscar, modificar, limpiar.



The image shows a graphical user interface window titled "REPRESENTANTE". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. The main content area contains the following elements:

- Label: "REPRESENTANTE" (centered at the top)
- Form fields:
 - Cédula: [text input field]
 - Nombre: [text input field]
 - Apellido: [text input field]
 - Parentesco: [text input field]
 - Teléfono: [text input field]
 - Dirección: [text area input field]
- Buttons at the bottom:
 - Insertar
 - Buscar
 - Limpiar
 - Modificar

Figura 17. Diseño de la ventana

El botón insertar permitirá al usuario agregar nuevos representantes, el botón buscar permite al usuario buscar a un representante a través de su número de cédula, el botón limpiar permite encerrar los campos dejando en blanco a estos, y el botón modificar se activará únicamente después de realizar la búsqueda de un representante. En la figura 18 se puede observar un fragmento de la programación de los botones.

```

btnInsertar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        bd = new DbConnection();
        Connection con=bd.getConnection();
        Statement st;
        representante rep=new representante();

        rep.setCedulaRep(cedula.getText());
        rep.setNombreRep(nombre.getText());
        rep.setApellidoRep(apellido.getText());
        rep.setParentesco(parentesco.getText());
        rep.setTelefono(telefono.getText());
        rep.setDireccion(direccion.getText());

        String sql = "insert into `representante` (`cedulaRep`, `nombreRep`, `apellidoRep`, `parentesco`,
        +rep.getCedulaRep()+"', '"+rep.getNombreRep()+"', '"+rep.getApellidoRep()+"', '"+rep.getPa
        +rep.getTelefono()+"', '"+rep.getDireccion()+"'";

        try
        {
            st=con.createStatement();
            st.executeUpdate(sql);
            con.close();
            st.close();
            JOptionPane.showMessageDialog(null, "Se ha Insertado Exitosamente", "Información", JOptionPane..
            limpiar();
        }
        catch(SQLException e){
            JOptionPane.showMessageDialog(null, "No se logró establecer conexión a la Base de Datos", "Er
        }catch(Exception e){
            JOptionPane.showMessageDialog(null, "No se logró establecer conexión a la Base de Datos", "Er
        }
    }
}

```

Figura 18. Programación del botón Insertar.

A continuación se observa mediante las figuras (19 – 26) el funcionamiento del módulo y su conexión con la base de datos:

The image shows a Java Swing window titled "REPRESENTANTE". The window has a standard title bar with minimize, maximize, and close buttons. The main content area is light gray and contains the following elements:

- A title "REPRESENTANTE" centered at the top.
- Six text input fields arranged vertically, each preceded by a label: "Cédula:", "Nombre:", "Apellido:", "Parentesco:", "Teléfono:", and "Dirección:". The "Dirección:" field is significantly larger than the others.
- Four buttons at the bottom, arranged horizontally: "Insertar", "Buscar", "Limpiar", and "Modificar".

Figura 19. Ingreso al módulo/Ventana después de dar clic en Limpiar.

REPRESENTANTE

Cédula:

Nombre:

Apellido:

Parentesco:

Teléfono:

Dirección:

Figura 20. Ingreso de datos.

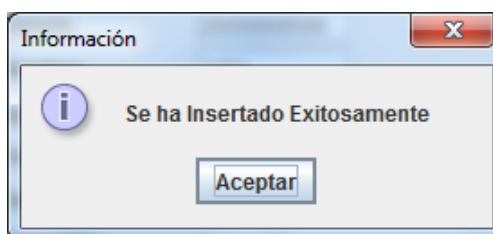


Figura 21. Mensaje después de Insertar.

```

C:\Program Files\MySQL\MySQL Server 5.1\bin>mysql.exe
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 54
Server version: 5.1.57-community MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use sfime;
Database changed
mysql> select * from representante
-> ;
+-----+-----+-----+-----+-----+-----+
| cedulaRep | nombreRep | apellidorep | parentesco | telefono | direccion |
+-----+-----+-----+-----+-----+-----+
| 2000069506 | Julia    | Salvador    | Ninguno    | 2347513  | Conocoto  |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> _
  
```

Figura 22. Datos insertados en tabla Representante.



REPRESENTANTE

Cédula:	2000069506
Nombre:	Julia
Apellido:	Salvador
Parentesco:	Ninguno
Teléfono:	2347513
Dirección:	Conocoto

Insertar Buscar Limpiar Modificar

Figura 23. Datos buscados por número de cédula y activación del botón Modificar.



REPRESENTANTE

Cédula:	2000069506
Nombre:	Julia
Apellido:	Salvador
Parentesco:	Ninguno
Teléfono:	2347513
Dirección:	Barrio San Miguel

Insertar Buscar Limpiar Modificar

Figura 24. Modificación de datos.

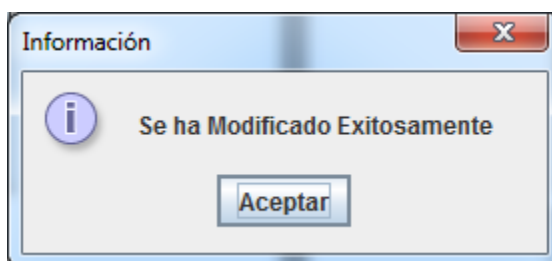


Figura 25. Mensaje de modificación de datos.

```

C:\Program Files\MySQL\MySQL Server 5.1\bin>mysql.exe
mysql> use sfime;
Database changed
mysql> select * from representante
-> ;
+-----+-----+-----+-----+-----+-----+
| cedulaRep | nombreRep | apellidorep | parentesco | telefono | direccion |
+-----+-----+-----+-----+-----+-----+
| 2000069506 | Julia    | Salvador    | Ninguno    | 2347513 | Conocoto  |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from representante;
+-----+-----+-----+-----+-----+-----+
| cedulaRep | nombreRep | apellidorep | parentesco | telefono | direccion |
+-----+-----+-----+-----+-----+-----+
| 2000069506 | Julia    | Salvador    | Ninguno    | 2347513 | Barrio San Miguel |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>

```

Figura 26. Datos modificados en tabla Representante.

3.2.2. NetBeans – Java

Para realizar el módulo Representante se creará un proyecto en NetBeans, pero el modelo UML se realiza en una herramienta CASE y éste no afecta al desarrollo del proyecto, el proyecto se encuentra organizado en tres paquetes (Fig. 27):

- Paquete para la conexión a la Base de Datos
- Paquete con los Get y Set de la tabla representante
- Paquete en para el diseño y funcionalidad de la ventana

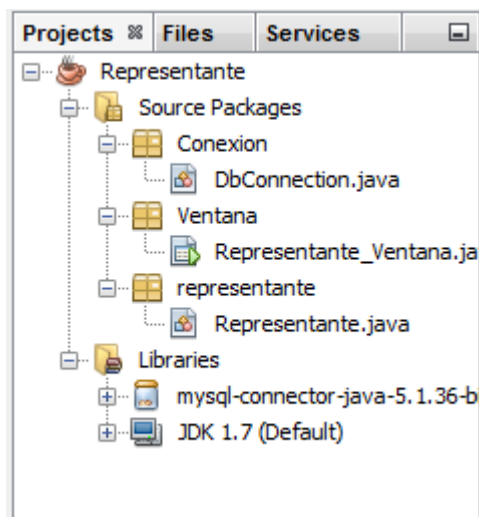


Figura 27. Organización de Paquetes del proyecto NetBeans.

En el paquete de conexión a la Base de Datos se agrega una clase en donde se establecen los parámetros de conexión con la base de datos de MySQL (Fig. 28)

```

14  * @author Juli
15  */
16  public class DbConnection {
17      /*Parámetros de conexión*/
18      static String login = "root";
19      static String password = "juli";
20      static String url = "jdbc:mysql://localhost:3306/sfime";
21
22      private Connection conexion = null;
23
24      //
25      private static String driver="com.mysql.jdbc.Driver";
26
27      public DbConnection(){
28          try {
29              Class.forName(driver);
30              conexion=DriverManager.getConnection(url,login,password);
31          } catch (ClassNotFoundException | SQLException e){
32              JOptionPane.showMessageDialog(null, "No se logró establecer conexión a la Base de Datos", "Error", JOptionPane.
33          }
34
35      public Connection getConnection(){
36          return conexion;
37      }
38  }
39

```

Figura 28. Clase de conexión a la Base de Datos

En el paquete representante se agregará una clase en la cual se realizará la declaración de los get y set de la tabla representante que se van a utilizar (Fig. 29).

```

1 package representante;
2
3
4
5
6
7
8
9
10
11 public class Representante {
12
13
14     private String cedulaRep;
15
16     private String nombreRep;
17
18     private String apellidoRep;
19
20     private String parentesco;
21
22     private String telefono;
23
24     private String direccion;
25
26     public void representar() {}
27     public String getCedulaRep() {
28         return this.cedulaRep;
29     }
30
31     public String getNombreRep() {
32         return this.nombreRep;
33     }
34
35     public String getApellidoRep() {
36         return this.apellidoRep;
37     }
38
39     public String getParentesco() {
40         return this.parentesco;
41     }
42
43     public String getTelefono() {
44         return this.telefono;
45     }
46
47     public String getDireccion() {
48         return this.direccion;
49     }
50
51     public void setCedulaRep (String cedulaRep) {
52         this.cedulaRep = cedulaRep;
53     }
54
55     public void setNombreRep (String nombreRep) {
56         this.nombreRep = nombreRep;
57     }
58
59     public void setApellidoRep (String apellidoRep) {
60         this.apellidoRep = apellidoRep;
61     }
62
63     public void setParentesco (String parentesco) {
64         this.parentesco = parentesco;
65     }
66
67     public void setTelefono (String telefono) {
68         this.telefono = telefono;
69     }
70
71     public void setDireccion (String direccion) {
72         this.direccion = direccion;
73     }
74
75
76
77

```

Figura 29. Clase Get y Set de la tabla representante.

En el paquete Ventana se diseñará la ventana que visualizará el usuario (Fig. 30), conjuntamente con las operaciones que se realizarán, las cuales son insertar, buscar, modificar, limpiar.



The image shows a graphical user interface window titled "REPRESENTANTE". It contains a form with the following fields and controls:

- Cédula:** A single-line text input field.
- Nombre:** A single-line text input field.
- Apellido:** A single-line text input field.
- Parentesco:** A single-line text input field.
- Teléfono:** A single-line text input field.
- Dirección:** A multi-line text area with a vertical scrollbar on the right and a horizontal scrollbar at the bottom.

At the bottom of the window, there are four buttons: "Insertar", "Buscar", "Limpiar", and "Modificar".

Figura 30. Diseño de la Ventana en NetBeans.

Al igual que la ventana realizada en Eclipse los botones cumplen las mismas funciones, el botón insertar permitirá al usuario agregar nuevos representantes, el botón buscar permite al usuario buscar a un representante a través de su número de cédula, el botón limpiar permite encerrar los campos dejando en blanco a estos, y el botón modificar se activará únicamente después de realizar la búsqueda de un representante. En la figura 31 se puede observar un fragmento de la programación de los botones.

```

225 private void btnBuscarActionPerformed(java.awt.event.ActionEvent evt) {
226     // TODO add your handling code here:
227     btnModificar.setEnabled(true);
228     bd = new DbConnection();
229     Connection con=bd.getConnection();
230     Statement st;
231     ResultSet rs;
232     Representante rep=new Representante();
233
234     rep.setCedulaRep(cedula.getText());
235     rep.setNombreRep(nombre.getText());
236     rep.setApellidoRep(apellido.getText());
237     rep.setParentesco(parentesco.getText());
238     rep.setTelefono(telefono.getText());
239     rep.setDireccion(direccion.getText());
240
241     String sql="select * from representante where `cedulaRep`='"+rep.getCedulaRep()+"'";
242     try
243     {
244         st=con.createStatement();
245         rs = st.executeQuery(sql);
246         while (rs.next()) {
247             nombre.setText(rs.getString(2));
248             apellido.setText(rs.getString(3));
249             parentesco.setText(rs.getString(4));
250             telefono.setText(rs.getString(5));

```

Figura 31. Fragmento de la codificación del botón Buscar.

A continuación se observará mediante las figuras (32 – 39) el funcionamiento del módulo y su conexión con la base de datos:

The image shows a Java Swing window titled "REPRESENTANTE". It features a light gray background and a title bar with standard window controls. The window contains six text input fields, each with a label to its left: "Cédula:", "Nombre:", "Apellido:", "Parentesco:", "Teléfono:", and "Dirección:". The "Dirección:" field is a text area with a vertical scrollbar. At the bottom of the window, there are four buttons: "Insertar", "Buscar", "Limpiar", and "Modificar".

Figura 32. Ingreso al módulo/Ventana después de dar clic en Limpiar – NetBeans.

REPRESENTANTE

Cédula:

Nombre:

Apellido:

Parentesco:

Teléfono:

Dirección:

Figura 33. Ingreso de datos Ventana NetBeans.

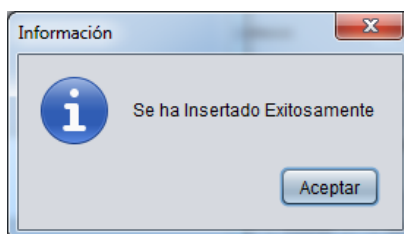


Figura 34. Mensaje de Insertado Exitoso Ventana NetBeans.

```

C:\Program Files\MySQL\MySQL Server 5.1\bin>mysql.exe
mysql> select * from representante;
+----+-----+-----+-----+-----+-----+
| cedulaRep | nombreRep | apellidorep | parentesco | telefono | direccion |
+----+-----+-----+-----+-----+-----+
| 2000069506 | Julia | Salvador | Ninguno | 2347513 | Barrio San Miguel |
+----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from representante;
+----+-----+-----+-----+-----+-----+
| cedulaRep | nombreRep | apellidorep | parentesco | telefono | direccion |
+----+-----+-----+-----+-----+-----+
| 1706529326 | Mónica | Mayorga | Ninguno | 052521506 | Galápagos |
| 2000069506 | Julia | Salvador | Ninguno | 2347513 | Conocoto |
+----+-----+-----+-----+-----+-----+
2 rows in set (0.03 sec)

mysql>
  
```

Figura 35. Datos insertados en tabla Representante mediante NetBeans.



A screenshot of a NetBeans window titled "REPRESENTANTE". The window contains a form with the following fields and values:

Field	Value
Cédula:	1706529326
Nombre:	Mónica
Apellido:	Mayorga
Parentesco:	Ninguno
Teléfono:	052521506
Dirección:	Galápagos

At the bottom of the window, there are four buttons: "Insertar", "Buscar", "Limpiar", and "Modificar". The "Modificar" button is highlighted with a blue border, indicating it is active.

Figura 36. Datos buscados por número de cédula y activación del botón Modificar Ventana NetBeans.



A screenshot of a NetBeans window titled "REPRESENTANTE". The window contains a form with the following fields and values:

Field	Value
Cédula:	1706529326
Nombre:	Mónica
Apellido:	Mayorga
Parentesco:	Ninguno
Teléfono:	052521506
Dirección:	San Cristóbal

At the bottom of the window, there are four buttons: "Insertar", "Buscar", "Limpiar", and "Modificar". The "Modificar" button is highlighted with a blue border, indicating it is active.

Figura 37. Modificación de datos Ventana NetBeans.

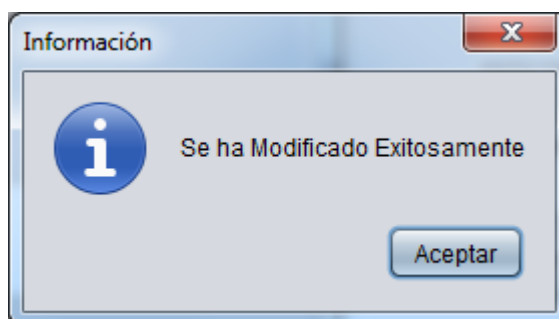


Figura 38. Mensaje de modificación de datos Ventana NetBeans.

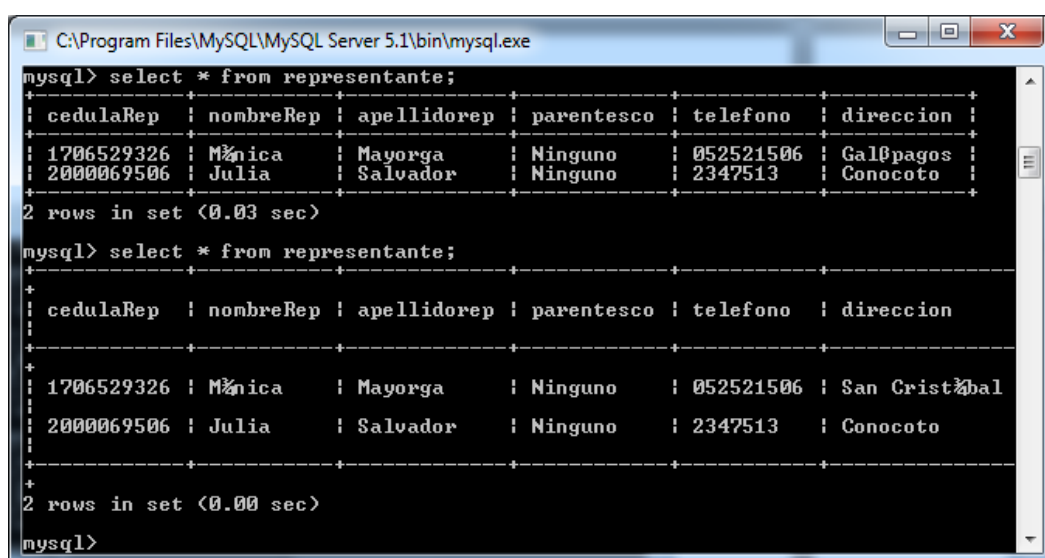


Figura 39. Datos modificados en tabla Representante.

3.3. Cuadro Comparativo

En la siguiente tabla se realizará un análisis entre la programación con la ayuda de Acceleo y la programación tradicional.

Tabla 1
Cuadro comparativo Acceleo / Java (NetBeans)

	Acceleo (Eclipse)	Java (NetBeans)
Tamaño del Archivo	456 KB	174 KB
Modelos UML	Se realiza el modelo en el proyecto	Es necesario usar una herramienta CASE
Líneas de código	34 Para crear el la plantilla con las reglas de generación	65 Para crear la clase con los get y set de la tabla
Tiempo de Programación	5 min Cuando hay más tablas el tiempo sigue siendo el mismo ya que utiliza el mismo código pero genera un archivo por cada tabla	10 min Cuando hay más tablas el tiempo aumenta dependiendo de la cantidad de tablas y atributos que exista
Errores	Los errores más comunes se dan al momento de escribir las reglas de generación ya que el texto se copia directamente al archivo generado	Los errores que pueden suceder son que no se programe un atributo o que se escriba de manera errónea un atributo.
Productividad	Mejora la productividad ya que reduce el tiempo de programación y se puede reusar el código del template con otro modelo, convirtiendo así en Acceleo en una herramienta eficiente ya que utiliza menos recursos	Es menos eficiente ya que generar los métodos get y set se vuelve un trabajo repetitivo y que tiene que ser programado atributo por atributo de cada clase.
Portabilidad	Al generar el código, podemos usar este en diferentes plataformas que utilicen el lenguaje de programación Java	Al ser un archivo Java se puede utilizar el código en otra plataforma que utilice Java.
Mantenimiento	Su mantenimiento es mejor ya que para corregir algún error de sintaxis o corrección	Su mantenimiento es más largo ya que si se desea una corrección en modelos o de

Documentación	en el modelo, se corrige en un único archivo y reemplazar el archivo automático.	sintaxis, se corrige por separado y para mantenimiento de modelo se debe usar la herramienta CASE necesaria y corregir en el archivo donde se encuentre la información relacionada
	Utiliza los modelos CIM, PIM y PSM	Se debe elegir una metodología para desarrollar el proyecto de una manera ordenada

CAPÍTULO 4

DESARROLLO DEL SISTEMA

4.1. Modelo Independiente de la Computación – CIM

El CIM es un modelado de negocios y requerimientos, que tiene como objetivo describir el negocio.

El modelado de negocio es el encargado de modelar el funcionamiento de una organización a través de sus procesos de negocio, para lo cual se utiliza los casos de uso y los diagramas de actividades.

Para los requerimientos se realiza el desarrollo del documento de Especificación de Requerimientos (Anexo A2).

A continuación se observa el Diagrama de Casos de Uso (figura 40) y los Requisitos Funcionales (tabla 2 – tabla 15).

4.1.1. Diagrama de Casos De Uso

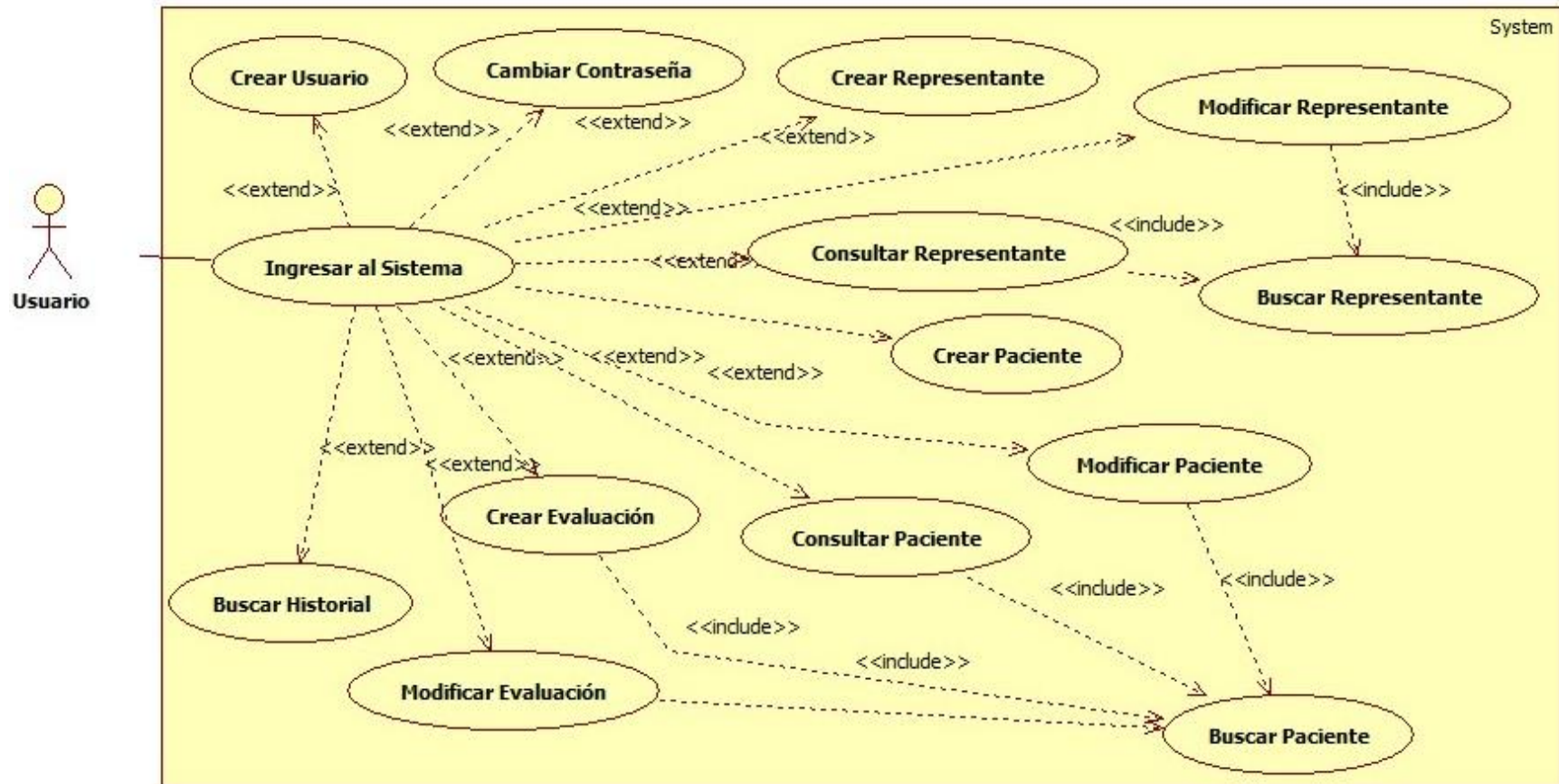


Figura 40. Diagrama de Casos de Uso.

4.1.2. Requisitos Funcionales

Tabla 2
CU01 Ingresar al Sistema

CU01	Ingresar al Sistema	
Actores	Usuario	
Descripción	El usuario ingresará su nombre de usuario que puede tener letras y/o números y la contraseña que tendrá una longitud mínima de 6 caracteres los cuales pueden ser números y/o letras, para poder logearse e ingresar al sistema.	
Precondición	El usuario debe conocer su usuario y contraseña.	
Secuencia Normal	Paso	Acción
	1	Ingresar nombre de Usuario.
	2	Ingresar la contraseña.
	3	Clic en Ingresar
Postcondición	Ingreso al sistema.	
Excepciones	Paso	Acción
	1	En el caso de que el usuario o la contraseña sean incorrectos, aparecerá un mensaje en donde dirá "Usuario o contraseña inválidos".
Rendimiento	El ingreso al Sistema es instantáneo, y el usuario tendrá acceso a los cuatro módulos del producto software.	
Frecuencia	Este caso de uso se realiza una vez cada se ingrese al sistema.	
Importancia	Importante	
Urgencia	No hay urgencia inmediata	
Comentarios	Proveer una interfaz sencilla e intuitiva al usuario	

Tabla 3
CU02 Crear Usuario

CU02	Crear Usuario										
Actores	Usuario										
Descripción	El usuario podrá crear a un nuevo usuario asignándole un nombre único y una contraseña.										
Precondición	Ingresar al Sistema, ingresar al módulo de Usuario, opción Crear Usuario.										
Secuencia Normal	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Escribir un nombre de Usuario</td> </tr> <tr> <td>2</td> <td>Comprobar si el usuario está disponible</td> </tr> <tr> <td>3</td> <td>Escribir una contraseña</td> </tr> <tr> <td>4</td> <td>Crear el nuevo Usuario</td> </tr> </tbody> </table>	Paso	Acción	1	Escribir un nombre de Usuario	2	Comprobar si el usuario está disponible	3	Escribir una contraseña	4	Crear el nuevo Usuario
Paso	Acción										
1	Escribir un nombre de Usuario										
2	Comprobar si el usuario está disponible										
3	Escribir una contraseña										
4	Crear el nuevo Usuario										
Postcondición	Mensaje de Creación exitosa										
Excepciones	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>En el caso de que el nombre de usuario no esté disponible el sistema emitirá un mensaje de error y el usuario deberá ingresar un nuevo nombre y deberá volver a comprobar si el nombre está disponible</td> </tr> <tr> <td>2</td> <td>En caso de que la contraseña tenga menos de seis (6) caracteres el sistema emitirá un mensaje de error y el usuario debe ingresar una nueva contraseña</td> </tr> </tbody> </table>	Paso	Acción	1	En el caso de que el nombre de usuario no esté disponible el sistema emitirá un mensaje de error y el usuario deberá ingresar un nuevo nombre y deberá volver a comprobar si el nombre está disponible	2	En caso de que la contraseña tenga menos de seis (6) caracteres el sistema emitirá un mensaje de error y el usuario debe ingresar una nueva contraseña				
Paso	Acción										
1	En el caso de que el nombre de usuario no esté disponible el sistema emitirá un mensaje de error y el usuario deberá ingresar un nuevo nombre y deberá volver a comprobar si el nombre está disponible										
2	En caso de que la contraseña tenga menos de seis (6) caracteres el sistema emitirá un mensaje de error y el usuario debe ingresar una nueva contraseña										
Rendimiento	El sistema deberá crear al nuevo usuario al instante.										
Frecuencia	Este caso de uso no es frecuente, debido a que pocas personas serán las usuarias del sistema.										
Importancia	Baja importancia.										
Urgencia	No hay urgencia inmediata.										
Comentarios	Interfaz intuitiva										

Tabla 4
CU03 Cambiar contraseña

CU03	Cambiar contraseña	
Actores	Usuario	
Descripción	El usuario puede cambiar su contraseña las veces que desea.	
Precondición	El usuario debe ingresar al sistema, al módulo de Usuario, opción Cambiar Contraseña y conocer su contraseña.	
Secuencia Normal	Paso	Acción
	1	Escribir la contraseña anterior
	2	Escribir la nueva contraseña
	3	Guardar los cambios
Postcondición	Mensaje exitoso	
Excepciones	Paso	Acción
	1	En el caso de que la contraseña anterior sea incorrecta, el sistema emitirá un mensaje de error.
	2	En el caso de que la nueva contraseña no cumpla con el mínimo de caracteres, se emitirá un mensaje de error.
Rendimiento	El sistema deberá ejecutar esta acción en tiempo instantáneo.	
Frecuencia	Este caso de uso se espera que no se lleve muy frecuente.	
Importancia	Importante	
Urgencia	Baja	
Comentarios	Intuitiva interfaz para el usuario	

Tabla 5
CU04 Crear Representante

CU04	Crear Representante	
Actores	Usuario	
Descripción	El usuario podrá crear a un nuevo representante, ingresando los datos que se solicita.	
Precondición	Ingresar al Sistema, e ingresar al módulo de Representante.	
Secuencia Normal	Paso	Acción
	1	Ingresar los datos solicitados en el formulario: <ul style="list-style-type: none"> • Cédula • Nombre • Apellido • Dirección • Teléfono • Parentesco
	2	Crear el nuevo representante
Postcondición	Mensaje exitoso	
Excepciones	Paso	Acción
	1	En el caso de que no se ingrese los datos obligatorios el sistema emitirá un mensaje de error.
	2	En el caso de que la cédula del representante ya exista y se desee crear el representante el sistema emitirá un mensaje de error.
Rendimiento	El sistema deberá ejecutar esta acción en tiempo instantáneo.	
Frecuencia	Este caso de uso es de frecuencia media.	
Importancia	Importancia	
Urgencia	Urgencia media	
Comentarios	Intuitiva interfaz para el usuario	

Tabla 6
CU05 Buscar Representante

CU05	Buscar Representante	
Actores	Usuario	
Descripción	El usuario podrá buscar a un representante, ingresando la cédula del mismo.	
Precondición	Ingresar al Sistema, e ingresar al módulo de Representante.	
Secuencia Normal	Paso	Acción
	1	Ingresar el número de cédula del representante a ser buscado.
	2	Buscar el representante
Postcondición	La información solicitada	
Excepciones	Paso	Acción
	1	En el caso de que la cédula no exista, el sistema emitirá un mensaje de error.
Rendimiento	El sistema deberá publicar la información del representante de una manera inmediata.	
Frecuencia	Este caso de uso es de frecuencia media.	
Importancia	Importancia	
Urgencia	Urgencia media	
Comentarios	Intuitiva interfaz para el usuario	

Tabla 7
CU06 Modificar Representante

CU06	Modificar Representante	
Actores	Usuario	
Descripción	El usuario podrá buscar a un representante, ingresando la cédula del mismo.	
Precondición	Ingresar al Sistema, ingresar al módulo de Representante y buscar representante.	
Secuencia Normal	Paso	Acción
	1	En el formulario cambiar la información a ser modificada
	2	Guardar los cambios
Postcondición	Mensaje exitoso	
Excepciones	Paso	Acción
	1	El número de cédula no puede ser cambiado.
	2	Los campos obligatorios no pueden quedar en blanco.
Rendimiento	El sistema deberá guardar los cambios inmediatamente.	
Frecuencia	Este caso de uso es de frecuencia media.	
Importancia	Importancia	
Urgencia	Urgencia media	
Comentarios	Intuitiva interfaz para el usuario	

Tabla 8
CU07 Consultar Representante

CU06	Consultar Representante	
Actores	Usuario	
Descripción	El usuario podrá buscar a un representante, ingresando la cédula del mismo.	
Precondición	Ingresar al Sistema, ingresar al módulo de Representante y buscar representante.	
Secuencia Normal	Paso 1	Acción En el formulario se mostrarán los datos del representante consultado
Postcondición	Información solicitada	
Excepciones	Paso 1	Acción En caso de que el representante no exista, el formulario quedará en blanco
Rendimiento	El sistema deberá mostrar lo consultado.	
Frecuencia	Este caso de uso es de frecuencia media.	
Importancia	Importancia	
Urgencia	Urgencia media	
Comentarios	Intuitiva interfaz para el usuario	

Tabla 9
CU08 Crear Paciente

CU07	Crear Paciente						
Actores	Usuario						
Descripción	El usuario podrá crear a un nuevo paciente, ingresando los datos que se solicita.						
Precondición	Ingresar al Sistema, e ingresar al módulo de Paciente.						
Secuencia Normal	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Ingresar los datos solicitados en el formulario: <ul style="list-style-type: none"> • Cédula • Nombre • Apellido • Representante • Fecha de Nacimiento • Discapacidad </td> </tr> <tr> <td>2</td> <td>Crear el nuevo paciente</td> </tr> </tbody> </table>	Paso	Acción	1	Ingresar los datos solicitados en el formulario: <ul style="list-style-type: none"> • Cédula • Nombre • Apellido • Representante • Fecha de Nacimiento • Discapacidad 	2	Crear el nuevo paciente
Paso	Acción						
1	Ingresar los datos solicitados en el formulario: <ul style="list-style-type: none"> • Cédula • Nombre • Apellido • Representante • Fecha de Nacimiento • Discapacidad 						
2	Crear el nuevo paciente						
Postcondición	Mensaje exitoso						
Excepciones	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>En el caso de que no se ingrese los datos obligatorios el sistema emitirá un mensaje de error.</td> </tr> <tr> <td>2</td> <td>En el caso de que la cédula del paciente ya exista y se desee crear el representante el sistema emitirá un mensaje de error.</td> </tr> </tbody> </table>	Paso	Acción	1	En el caso de que no se ingrese los datos obligatorios el sistema emitirá un mensaje de error.	2	En el caso de que la cédula del paciente ya exista y se desee crear el representante el sistema emitirá un mensaje de error.
Paso	Acción						
1	En el caso de que no se ingrese los datos obligatorios el sistema emitirá un mensaje de error.						
2	En el caso de que la cédula del paciente ya exista y se desee crear el representante el sistema emitirá un mensaje de error.						
Rendimiento	El sistema deberá ejecutar esta acción en tiempo instantáneo.						
Frecuencia	Este caso de uso es de frecuencia media.						
Importancia	Importancia						
Urgencia	Urgencia media						
Comentarios	Intuitiva interfaz para el usuario						

Tabla 10
CU09 Buscar Paciente

CU08	Buscar Paciente	
Actores	Usuario	
Descripción	El usuario podrá buscar a un paciente, ingresando la cédula del mismo.	
Precondición	Ingresar al Sistema, e ingresar al módulo de Paciente.	
Secuencia Normal	Paso	Acción
	1	Ingresar el número de cédula del paciente a ser buscado.
	2	Buscar el paciente
Postcondición	La información solicitada	
Excepciones	Paso	Acción
	1	En el caso de que la cédula no exista, el sistema emitirá un mensaje de error.
Rendimiento	El sistema deberá publicar la información del paciente de una manera inmediata.	
Frecuencia	Este caso de uso es de frecuencia media.	
Importancia	Importancia	
Urgencia	Urgencia media	
Comentarios	Intuitiva interfaz para el usuario	

Tabla 11
CU10 Modificar Paciente

CU09	Modificar Paciente	
Actores	Usuario	
Descripción	El usuario podrá buscar a un paciente, ingresando la cédula del mismo.	
Precondición	Ingresar al Sistema, ingresar al módulo de Paciente y buscar paciente.	
Secuencia Normal	Paso	Acción
	1	En el formulario cambiar los datos que se requiera
	2	Guardar los cambios
Postcondición	Mensaje exitoso	
Excepciones	Paso	Acción
	1	El número de cédula no puede ser cambiado.
	2	Los campos obligatorios no pueden quedar en blanco.
Rendimiento	El sistema deberá guardar los cambios inmediatamente.	
Frecuencia	Este caso de uso es de frecuencia media.	
Importancia	Importancia	
Urgencia	Urgencia media	
Comentarios	Intuitiva interfaz para el usuario	

Tabla 12
CU11 Consultar Paciente

CU09	Consultar Paciente	
Actores	Usuario	
Descripción	El usuario podrá buscar a un paciente, ingresando la cédula del mismo.	
Precondición	Ingresar al Sistema, ingresar al módulo de Paciente y buscar paciente.	
Secuencia Normal	Paso 1	Acción En el formulario aparecerán los datos del paciente consultado
Postcondición	Información solicitada	
Excepciones	Paso 1	Acción En el caso de no existir el paciente el formulario quedará en banco.
Rendimiento	El sistema deberá mostrar los datos inmediatamente.	
Frecuencia	Este caso de uso es de frecuencia media.	
Importancia	Importancia	
Urgencia	Urgencia media	
Comentarios	Intuitiva interfaz para el usuario	

Tabla 13
CU12 Crear Evaluación

CU10	Crear Evaluación	
Actores	Usuario	
Descripción	El usuario podrá crear a una nueva evaluación, ingresando los datos que se solicita.	
Precondición	Ingresar al Sistema, ingresar al módulo de Ficha Médica y buscar paciente.	
Secuencia Normal	Paso	Acción
	1	Ingresar los datos solicitados en el formulario <ul style="list-style-type: none"> • Fecha • Médico • Evaluación • Tratamiento • Observaciones
	2	Crear la nueva evaluación
Postcondición	Mensaje exitoso	
Excepciones	Paso	Acción
	1	En el caso de que no se ingrese los datos obligatorios el sistema emitirá un mensaje de error.
Rendimiento	El sistema deberá ejecutar esta acción en tiempo instantáneo.	
Frecuencia	Este caso de uso es de frecuencia alta.	
Importancia	Importancia alta	
Urgencia	Urgencia alta	
Comentarios	Intuitiva interfaz para el usuario	

Tabla 14
CU13 Buscar Historial

CU11	Buscar Historial	
Actores	Usuario	
Descripción	El usuario podrá buscar el historial de un paciente, ingresando la cédula del mismo.	
Precondición	Ingresar al Sistema, e ingresar al módulo de Ficha Médica.	
Secuencia Normal	Paso	Acción
	1	Ingresar el número de cédula del paciente a ser buscado.
	2	Buscar el historial
Postcondición	La información solicitada	
Excepciones	Paso	Acción
	1	En el caso de que la cédula no exista, el sistema emitirá un mensaje de error.
Rendimiento	El sistema deberá publicar la información de la ficha médica del paciente de una manera inmediata.	
Frecuencia	Este caso de uso es de frecuencia alta.	
Importancia	Importancia alta	
Urgencia	Urgencia alta	
Comentarios	Intuitiva interfaz para el usuario	

Tabla 15
CU14 Modificar Evaluación

CU11	Modificar Evaluación	
Actores	Usuario	
Descripción	El usuario podrá modificar la(s) evaluación(es) de un paciente, ingresando la cédula del mismo.	
Precondición	Ingresar al Sistema, e ingresar al módulo de Ficha Médica.	
Secuencia Normal	Paso	Acción
	1	Ingresar el número de cédula del paciente a ser buscado.
	2	Buscar el historial
	3	Modificar la(s) Evaluación(es)
	4	Guardar los cambios
Postcondición	La información solicitada	
Excepciones	Paso	Acción
	1	En el caso de que la cédula no exista, el sistema emitirá un mensaje de error.
Rendimiento	El sistema deberá publicar la información de la ficha médica del paciente de una manera inmediata.	
Frecuencia	Este caso de uso es de frecuencia alta.	
Importancia	Importancia alta	
Urgencia	Urgencia alta	
Comentarios	Intuitiva interfaz para el usuario	

4.2. MODELO INDEPENDIENTE DE LA PLATAFORMA – PIM

El modelo PIM describe la arquitectura del sistema, en donde se representa la estructura, funcionalidad y restricciones del sistema, sin importar la plataforma que se va a utilizar.

Para representar el PIM se utiliza el Diagrama de Clases Figura 53.

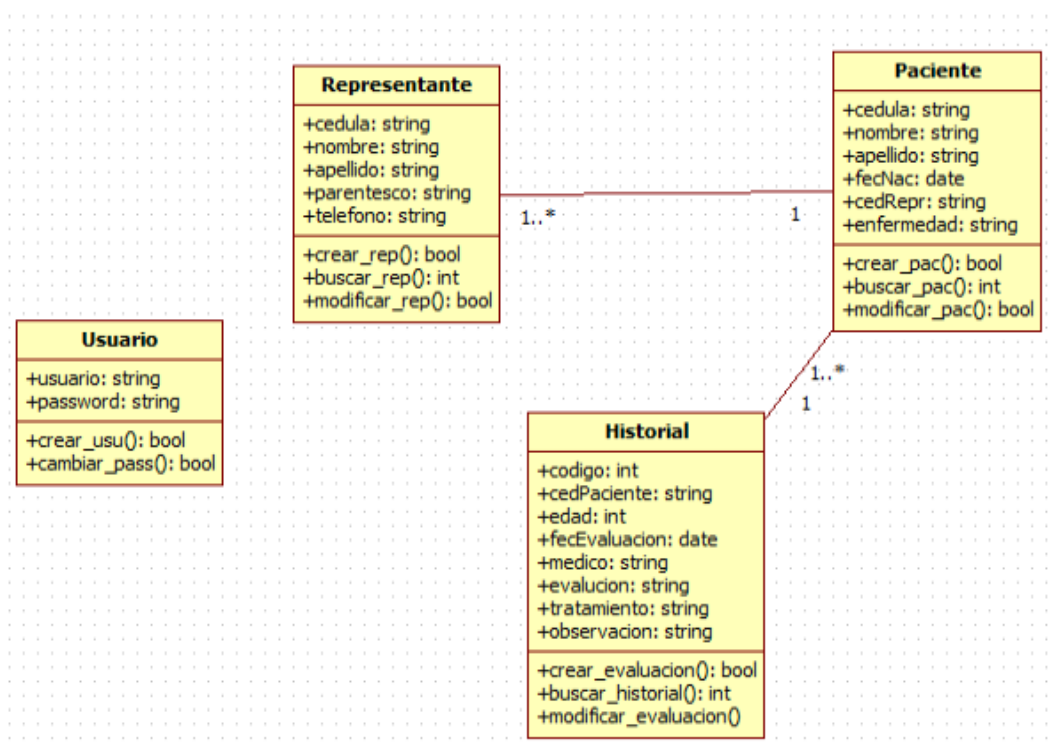


Figura 41. Diagrama de Clases

4.3. MODELO ESPECÍFICO DE PLATAFORMA – PSM

El PSM describe la implementación, con detalles específicos de la plataforma, esto es generado a partir del PIM, pero en diferente nivel de abstracción. Se ha generado los siguientes modelos:

- Modelo Entidad - Relación, el cual describe el esquema de la base de datos (Figura 42).

- Modelo UML Acceleo, es el modelo realizado con Acceleo para la generación de código (Figura 43).

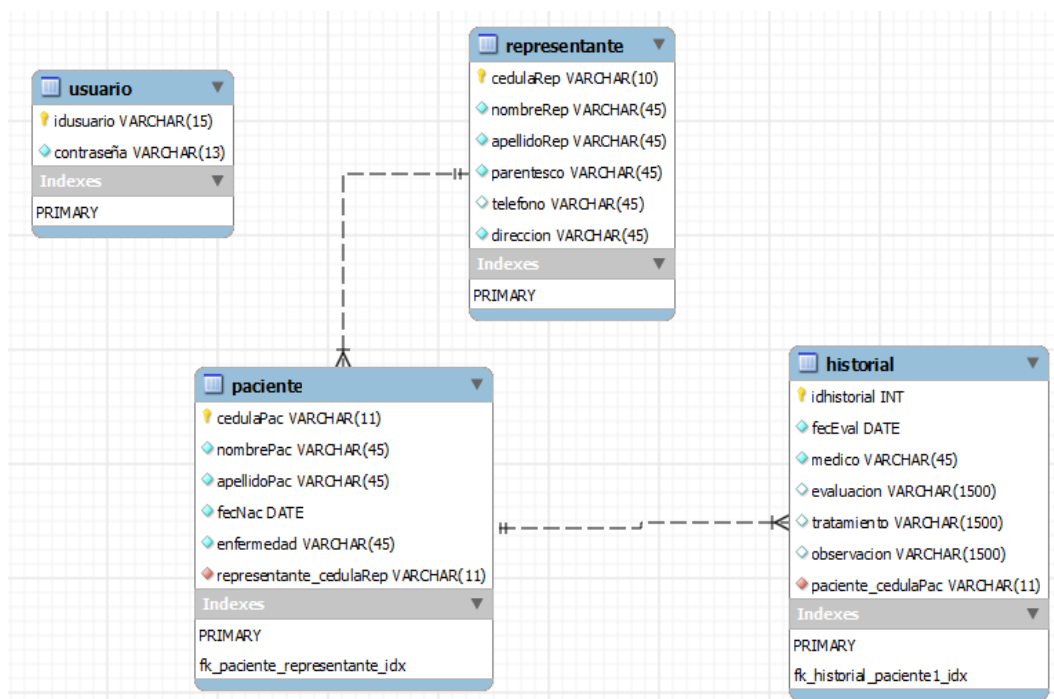


Figura 42. Diagrama Entidad – Relación

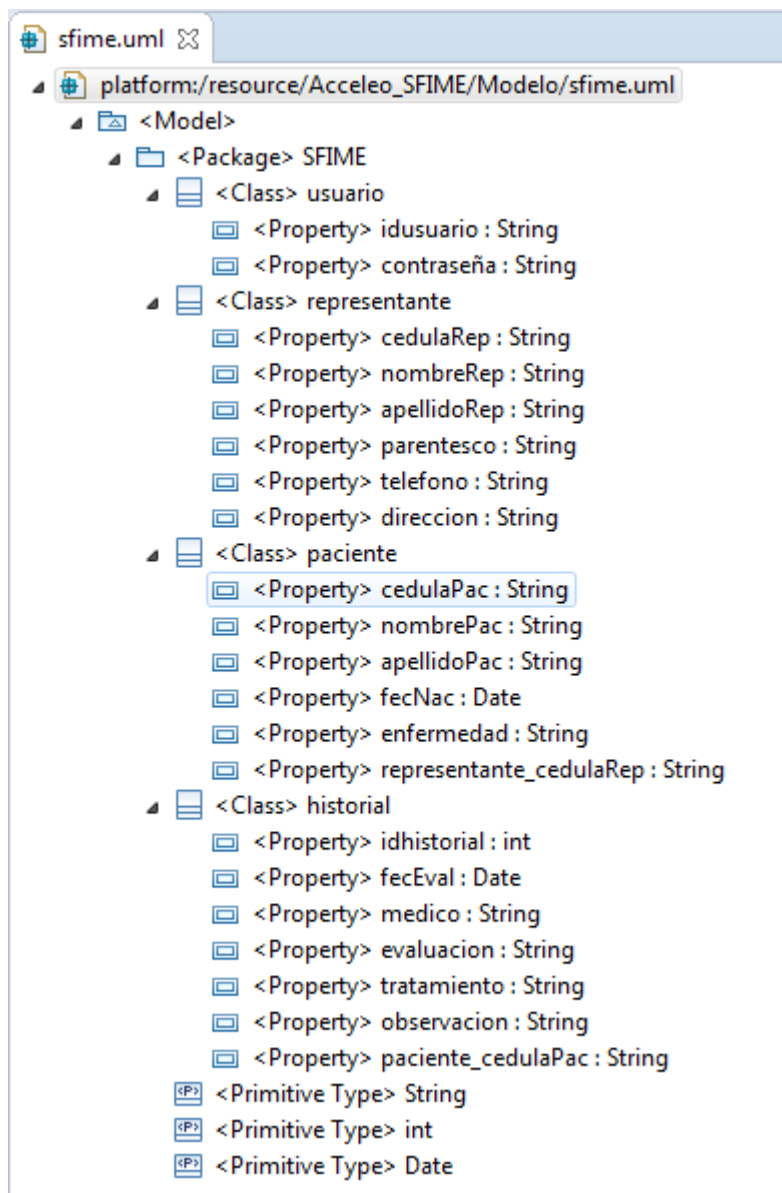


Figura 43. Modelo UML – Acceleo.

4.4. IMPLEMENTACIÓN DEL SISTEMA

Para la realización del presente proyecto se decidió utilizar una programación mixta, donde los Get y Set del proyecto son autogenerados con la ayuda de Acceleo y el resto de la programación es Programación Tradicional.

4.5. PRUEBAS DEL SISTEMA

Para verificar que los procesos cumplan con los requisitos se ha decidido realizar dos tipos de pruebas las cuales son: prueba de unidad y prueba de integración; las cuales se han ido realizando a lo largo del desarrollo, llegando a la conclusión de que el Sistema funciona correctamente.

Prueba de Unidad:

Ésta prueba evalúa a las clases de un programa, con el objetivo de probar el sistema en pequeñas partes.

El resultado es: la comunicación entre la base de datos y el sistema es correcta y funciona con todos los módulos, sin evidenciar ruptura de comunicación en ningún caso.

Prueba de Integración:

En ésta prueba se verifica que la funcionalidad sea correcta cumpliendo con los requisitos establecidos (Tabla 16). La prueba de Integración se realiza por medio de la prueba de caja negra. Los resultados de la prueba serán reflejados en la tabla 17.

Tabla 16
Prueba de Casos de Uso

Nombre	Caso de Uso	Resultado
Ingresar al Sistema	CU01	Exitoso
Crear Usuario	CU02	Exitoso
Cambiar Contraseña	CU03	Exitoso
Crear Representante	CU04	Exitoso
Buscar Representante	CU05	Exitoso
Modificar Representante	CU06	Exitoso
Consultar Representante	CU07	Exitoso

CONTINÚA →

Crear Paciente	CU08	Exitoso
Buscar Paciente	CU09	Exitoso
Modificar Paciente	CU10	Exitoso
Consultar Paciente	CU11	Exitoso
Crear Evaluación	CU12	Exitoso
Buscar Historial	CU13	Exitoso
Modificar Evaluación	CU14	Exitoso

Tabla 17
Prueba de Caja Negra

Caja Negra	Falla	Corrección
Prueba de Entrada de Datos	No se presentaron	_____
Pruebas de consulta (Listado, actualización y eliminación de registros)	No se presentaron	_____
Prueba de ingreso de cuenta del sistema	No se presentaron	_____
Pruebas de salidas de información	No se presentaron	_____

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Se estudió la herramienta Acceleo mediante el desarrollo del módulo “Representante” del “Sistema de Gestión de Fichas Médicas” y se comprobó los siguientes beneficios:

- En cuanto a la productividad, el tiempo de programación se reduce debido a que la herramienta ayuda a minimizar la programación repetitiva de los métodos “Get” y “Set” de los atributos de las clases. En el caso práctico se logró ahorrar un 8% de la programación.
- En cuanto a la portabilidad el código generado lo podemos utilizar en diferentes IDEs en los que se esté utilizando JAVA, ya que el código fue generado en dicho lenguaje.
- En cuanto al mantenimiento es más eficiente ya que permite realizar las correcciones necesarias en el modelo y volver a generar el código, sin necesidad de programar otras líneas de código.

Con la investigación acerca del tema de MDA se ha evidenciado que el diseño se encuentra separado de la arquitectura y la tecnología.

MDA está basado en tres modelos para el desarrollo: CIM – Modelo Independiente de la Computación, PIM – Modelo Independiente de la Plataforma y PSM – Modelo Específico de Plataforma, los cuales son los encargados de dirigir todo el proceso de desarrollo.

El CIM es desarrollado por el Analista de Negocios, en éste se representar el modelo de negocios y los requerimientos, siendo la base para el resto de modelos.

El PIM es desarrollado por el Arquitecto de Software, en éste modelo describe la solución, se representa un análisis y diseño independiente de la tecnología y la plataforma.

El PSM es realizado por el Desarrollador o Programador, éste modelo contiene los detalles de la plataforma.

Mediante el desarrollo de la aplicación práctica se constató que Acceleo es una herramienta que brinda los siguientes beneficios: es fácil de utilizar, cuenta con la característica de Eclipse de colorear la sintaxis y su trazabilidad permite encontrar los elementos del modelo.

El Sistema de Gestión de Fichas Médicas fue desarrollado con Acceleo – Eclipse y NetBeans; con Acceleo se ahorra tiempo y líneas de código ya que se escribe únicamente las líneas de código de la plantilla, la documentación son los diagramas realizados en los modelos MDA, aunque el tamaño del archivo será mayor ya que dentro del proyecto se encuentran las clases con sus atributos y tipos.

5.2. Recomendaciones

Se recomienda al departamento de Ciencias de la Computación que incorpore el estudio de MDA y sus herramientas en la malla curricular.

Se recomienda para un futuro tema de estudio el análisis entre herramientas MDA para la evaluación de la mejor herramienta.

Se recomienda en un futuro realizar una conferencia de MDA para difundir el tema con la ayuda de expertos.

Se recomienda incentivar a los estudiantes para que éstos investiguen acerca de las tendencias de programación.

Se recomienda realizar trabajos con diferentes tendencias de programación y no utilizar únicamente la programación tradicional.

Bibliografía

- Alan Hartman, D. K. (2005). *Model Driven Architecture - Foundations and Applications*. Alemania: Springer.
- Asociación Corazones Unidos . (29 de Marzo de 2010). Estatutos de la Asociación Corzones Unidos. *Capítulo I: Nombre y Dirección*. Pto. Baquerizo Moreno, Galápagos, Ecuador.
- Begaudeau, S. (08 de Junio de 2012). *WikiAcceleo*. Recuperado el 15 de Agosto de 2015, de Eclipse: <http://wiki.eclipse.org/Acceleo>
- Cañadas, J., Túnez, S., & Palma, J. (2004). *Workshop: Universidad Politécnica de Valencia*. Recuperado el 6 de Abril de 2014, de Universidad Politécnica de Valencia: <http://users.dsic.upv.es/workshops/dsdm04/files/04-Canadas.pdf>
- Carrera Sepúlveda, S., Varas Contreras, M., & Urrutia Sepúlveda, A. (2010). Transformación de esquemas multidimensionales difusos desde el nivel conceptual al nivel lógico. *INGENIARE, Revista Chilena de Ingeniería*, 166.
- Cerezo, A., & Centeno, J. (27 de Junio de 2008). *SALAS: Un Sistema de Reserva de Aulas en Ruby on Rails*. Recuperado el 9 de Diciembre de 2012, de Universidad Rey Juan Carlos - España: <http://upcommons.upc.edu/pfc/bitstream/2099.1/5542/2/Memoria.pdf>
- Colaborador guest83f0d26 de Slide Share. (27 de Noviembre de 2009). *MDA: Slide Share*. Recuperado el 12 de Diciembre de 2014, de Slide Share: <http://es.slideshare.net/guest83f0d26/mda-2596889>
- Colaboradores de Object Management Group. (19 de Marzo de 2015). *MDA Specifications: OMG*. Recuperado el 26 de Junio de 2015, de OMG: <http://www.omg.org/mda/specs.htm>
- Colaboradores OMG. (7 de Agosto de 2015). *OMG's MetaObject Facility: OMG*. Recuperado el 18 de Octubre de 2015, de OMG: <http://www.omg.org/mof/>
- Debrauwer, L., & Van delLeyde, F. (2005). *UML 2: iniciación, ejemplos y ejercicios corregidos*. Barcelona: ENI.
- Etienne, A., Mohamed Mahdi, B., & Choppy, C. (2014). *Computer Science: arXiv.org*. Recuperado el 24 de Octubre de 2015, de arXiv.org: <http://arxiv.org/pdf/1405.1112.pdf>

- Fernandez, O. (2010). *The Rails 3 Way* (2da ed.). Boston, USA: Addison-Wesley Professional.
- García Molina, J., Rodríguez, J., Menárguez, M., Ortín, M. J., & Sánchez, J. (2004). *Workshops: Universidad Politécnica de Valencia*. Recuperado el 6 de Abril de 2014, de Universidad Politécnica de Valencia: <http://users.dsic.upv.es/workshops/dsdm04/files/09-Garcia.pdf>
- Jose, N., & Alfonso, E. (2012). *Planificación - Arquitectura de la Información en Entornos Web*. Recuperado el 5 de Diciembre de 2012, de Universidad Tecnológica Nacional de Argentina: http://www.frm.utn.edu.ar/archivos/oferta%20educativa/ing_sistemas/programas/Planificacion_Arqu_de_informacion_2012.pdf
- López, D., González, M., López, M., & Iduñate, E. (2006). *Proceso de Desarrollo de Software Mediante Herramientas MDA*. Recuperado el 10 de Octubre de 2013, de Revista Iberoamericana de Sistemas, Cibernética e Informática: [http://www.iiisci.org/journal/CV\\$/risici/pdfs/C476A1.pdf](http://www.iiisci.org/journal/CV$/risici/pdfs/C476A1.pdf)
- Marco de Referencia para la Evaluación de Herramientas Basadas en MDA*. (11 de Marzo de 2007). Recuperado el 11 de Octubre de 2013, de Servidor Web Postgrado en Computación UCV: http://kuainasi.ciens.ucv.ve/ideas07/documentos/articulos_ideas/articulo65.pdf
- Mellor, S. (2004). *MDA Distilled: Principles of Model-driven Architecture*. Boston: Addison-Wesley Professional.
- MySQL. (2013). *About MySQL*. Recuperado el 10 de Noviembre de 2013, de MySQL: <http://www.mysql.com/>
- Oracle Corporation. (2015). *Bienvenido a NetBeans y www.netbeans.org*. Recuperado el 17 de Mayo de 2014, de NetBeans: https://netbeans.org/index_es.html
- Pérez, J. M., & Ruiz, F. (Marzo de 2007). *Departamento de Tecnologías y Sistemas de la Información*. Recuperado el 10 de Noviembre de 2013, de Universidad de Castilla-La Mancha (UCLM): <https://www.uclm.net/dep/tsi/pdf/UCLM-TSI-002.pdf>
- Pressman, R. (2010). *Ingeniería de Software Un enfoque práctico*. McGraw Hill.

- Quintero, J. B., & Anaya, R. (Diciembre de 2007). *MDA Y EL PAPEL DE LOS MODELOS EN EL PROCESO DE DESARROLLO DE SOFTWARE*. Recuperado el 12 de Octubre de 2013, de Revista EIA: <http://revista.eia.edu.co/articulos8/Art.10.pdf>
- Román Alcaraz, N. (13 de Noviembre de 2012). *Facultad de Ciencia y Tecnología: Universidad Católica "Nuestra Señora de la Asunción"*. Recuperado el 15 de Abril de 2014, de Universidad Católica "Nuestra Señora de la Asunción": <http://www.cyt.uc.edu.py/jit-cita/2012/files/charlas/charla8.pdf>
- Ruiz Rube, I. (05 de Diciembre de 2013). *P8 - Transformaciones de modelo a texto con Acceleo*. Recuperado el 15 de Agosto de 2015, de Universidad de Cádiz: https://ocw.uca.es/pluginfile.php/2501/mod_resource/content/0/P8%20-%20Transformaciones%20de%20modelo%20a%20texto%20con%20Acceleo.pdf
- Ruiz Rube, I. (05 de Diciembre de 2013). *Procesadores de Lenguaje II: Universidad de Cádiz*. Recuperado el 25 de Octubre de 2015, de Universidad de Cádiz: https://ocw.uca.es/pluginfile.php/2501/mod_resource/content/0/P8%20-%20Transformaciones%20de%20modelo%20a%20texto%20con%20Acceleo.pdf
- StarUML. (2013). *About*. Recuperado el 5 de Diciembre de 2013, de StarUML: <http://staruml.sourceforge.net/en/about.php>
- Stephens, M., & Rosenberg, D. (2010). *Design Driven Testing: Test Smarter, Not Harder* (1era ed.). New York, USA: Apress.
- The Eclipse Foundation. (2015). *Acceleo*. Recuperado el 15 de Agosto de 2015, de Eclipse: <http://www.eclipse.org/acceleo/>
- Valim, J. (2011). *Crafting Rails Applications: Expert Practices for Everyday Rails Development* (1era ed.). Pragmatic Bookshelf.
- Weitzenfeld, A. (2005). *Ingeniería de Software Orientada a Objetos con UML, Java e Internet*. Mexico: Thomson.
- Wynne, M., & Hellesoy, A. (2012). *The Cucumber Book: Behaviour-Driven Development for Testers and Developers*. USA: Pragmatic Bookshelf.