

UNIVERSIDAD TÉCNICA DE COTOPAXI



UNIDAD ACADÉMICA DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

CARRERA DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS COMPUTACIONALES

TESIS DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIEROS EN INFORMÁTICA Y SISTEMAS COMPUTACIONALES

TEMA:

“ESTUDIO COMPARATIVO ENTRE LAS HERRAMIENTAS LIBRES DE ANALISIS Y DESARROLLO STARUML Y FUJABA PARA ESTABLECER LA USABILIDAD Y LA APLICABILIDAD DE LA INGENIERIA INVERSA DENTRO DEL PROCESO DE DESARROLLO DE SOFTWARE EN LA EMPRESA MERCADO LATACUNGA”.

POSTULANTES: CEPEDA VACA LUIS ORLANDO

MEDINA CHILUISA FABIAN EDUARDO

DIRECTOR: ING. FRANKLIN MONTALUISA

LATACUNGA, OCTUBRE DEL 2013

DECLARACION DE AUTORIA

Latacunga, 16 de Octubre del 2013

Declaramos que somos los únicos autores de este trabajo de Diploma y autorizamos a la empresa “Mercado Latacunga”, de la ciudad de Latacunga hacer uso del mismo con la finalidad que estimen conveniente.

.....

Luis Orlando Cepeda.

CI: 0502346828

.....

Fabián Eduardo Medina.

CI: 0502612443

CERTIFICACION

Debo certificar que cumpliendo con lo estipulado en el Capítulo V, Art. 12, literal f del reglamento de la Universidad Técnica de Cotopaxi, que el tema de tesis titulado **ESTUDIO COMPARATIVO ENTRE LAS HERRAMIENTAS LIBRES DE ANALISIS Y DESARROLLO STARUML Y FUJABA PARA ESTABLECER LA USABILIDAD Y LA APLICABILIDAD DE LA INGENIERIA INVERSA DENTRO DEL PROCESO DE DESARROLLO DE SOFTWARE EN LA EMPRESA MERCADO LATACUNGA**, propuesto por los Egresados Cepeda Vaca Luis Orlando y Medina Chiluisa Fabián Eduardo, ha concluido el presente trabajo de Investigación de acuerdo a los planteamientos formulados en el plan de tesis, siendo ejecutado y revisado meticulosamente.

Atentamente,

.....
Ing. Franklin Montaluisa.
DIRECTOR DE TESIS



UNIVERSIDAD TÉCNICA DE COTOPAXI
UNIDAD ACADÉMICA DE CIENCIAS ADMINISTRATIVAS Y
HUMANÍSTICAS

LATACUNGA – ECUADOR

AVAL DE TRADUCCIÓN

En calidad de Docente del Centro Cultural de Idiomas de la Universidad Técnica De Cotopaxi, yo Lic. Mishelle Velástegui con la C.I. 050187099-2, CERTIFICO que he realizado la respectiva revisión del Abstract; con el tema: “ESTUDIO COMPARATIVO ENTRE LAS HERRAMIENTAS LIBRES DE ANALISIS Y DESARROLLO STARUML Y FUJABA PARA ESTABLECER LA USABILIDAD Y LA APLICABILIDAD DE LA INGENIERIA INVERSA DENTRO DEL PROCESO DE DESARROLLO DE SOFTWARE EN LA EMPRESA MERCADO LATACUNGA” cuyos autores son: Medina Chiluisa Fabián Eduardo y Cepeda Vaca Luis Orlando y director de tesis Ing. Franklin Montaluisa.

Latacunga, Junio del 2013

Docente:

Lic. Mishelle Velástegui
C.I. 050187099-2



Latacunga, 10 de octubre del 2013

CERTIFICADO

A Quien Interese.

Por medio del presente certifico que CEPEDA VACA LUIS ORLANDO con cédula de ciudadanía número 0502346828, y MEDINA CHILUISA FABIAN EDUARDO con cédula de ciudadanía número 0502612443, realizaron el estudio e implementación de su proyecto de tesis en la empresa **MERCADO LATACUNGA**, con el tema **“ESTUDIO COMPARATIVO ENTRE LAS HERRAMIENTAS LIBRES DE ANALISIS Y DESARROLLO STARUML Y FUJABA PARA ESTABLECER LA USABILIDAD Y LA APLICABILIDAD DE LA INGENIERIA INVERSA DENTRO DEL PROCESO DE DESARROLLO DE SOFTWARE”**.

Los señores antes mencionados pueden hacer uso de este certificado para los fines que crean pertinentes.

Es todo en cuanto puedo certificar.



contacto@mercadolatacunga.com

Av. Amazonas 5-37 Guayaquil Edif. C&C Corrales - Castellano 2do piso oficina # 102
junto a **UNIBANCO**
(03) 2802-651

v

AGRADECIMIENTO

Agradecemos a la Universidad Técnica de Cotopaxi y a todo su plantel docente de la Carrera Ingeniería en Informática y Sistemas Computacionales por brindarnos los medios y conocimientos necesarios para acceder a una formación académica que nos permitirá desempeñarnos profesionalmente en el futuro próximo.

En especial queremos agradecerle todo el apoyo recibido y buena predisposición totalmente desinteresada, por nuestro profesores, quienes nos guiaron durante todo este proceso ofreciéndonos sus conocimientos y horas de su tiempo.

Finalmente un agradecimiento especial al Ingeniero Franklin Montaluisa, quien nos guió durante todo este proceso ofreciéndonos sus conocimientos y horas de su tiempo.

Luis y Fabián

DEDICATORIA

Dedico este proyecto y toda mi carrera universitaria a Dios por ser quien ha estado a mi lado en todo momento dándome las fuerzas necesarias para continuar luchando día tras día y seguir adelante rompiendo todas las barreras que se me presenten. Les agradezco a mis padres, a mi abuelito, a mis hermanos, sobrinos, tíos y primos ya que gracias a ellos soy quien soy hoy en día, quienes fueron los que me dieron ese cariño y calor humano necesario, son los que han velado por mi salud, mis estudios, mi educación alimentación entre otros, son a ellos a quien les debo todo, horas de consejos , de regaños, de reprimendas de tristezas y de alegrías de las cuales estoy muy seguro que las han hecho con todo el amor del mundo para formarme como un ser integral y de las cuales me siento extremadamente orgulloso.

Le agradezco a mi amada esposa Silvana y a mis pequeñas y adoradas hijas Angie Monserrath y Doménica| Luciana, las cuales han estado a mi lado, han compartido todos mis años de esfuerzo siempre alerta ante cualquier problema que se me puedan presentar, dándome el apoyo necesario para poder culminar mi carrera.

Luis Orlando Cepeda Vaca

DEDICATORIA

La concepción de este proyecto está dedicado al creador de todas las cosas, el que me ha dado fortaleza para continuar cuando a punto de caer he estado; por ello, con toda la humildad que de mi corazón puede emanar, dedico primeramente mi trabajo a Dios.

De igual forma, dedico esta tesis a mi madre pilar fundamental en mi vida, que ha sabido formarme con buenos sentimientos, hábitos y valores, lo cual me ha ayudado a salir adelante en los momentos más difíciles.

A mis hermanos que siempre han estado junto a mí y brindándome su apoyo, muchas veces poniéndose en el papel de padre.

Al hombre que me dio la vida, el cual a pesar de nuestra diferencias me apoya.

A mi familia en general, porque me han brindado su apoyo incondicional y por compartir conmigo buenos y malos momentos.

Fabián Eduardo Medina Chiluisa

INDICE

Portada	i
Declaración de Autenticidad	ii
Certificación Director Tesis	iii
Certificación Summary	iv
Certificación Implantación	v
Agradecimiento	vi
Dedicatoria	vii
Índice	ix
Resumen	xv
Summary	xvi
Introducción	1
CAPITULO I	
Fundamentación Teórica	3
1.1. Herramientas de Software Libre	3
1.1.1. Software Libre	3
1.1.2. Tipos de Software Libre	4
1.1.2.1. Software de dominio público.	4
1.1.2.2. Software protegido con copyleft.	5
1.1.2.3. Software no protegido con copyleft.	5
1.1.2.4. Software cubierto con la GPL.	6
1.1.2.5. Software privativo.	7
1.1.2.6. Freeware.	8
1.1.2.7. Shareware	8
1.1.3. Staruml	9
1.1.4. Uml	10
1.1.5. Fujaba	11
1.2. Ingeniería Inversa en el Proceso de Desarrollo	12
1.2.1. Ingeniería Inversa	12

1.2.2. Tipos de Ingeniería Inversa	14
1.2.2.1. Ingeniería inversa de datos	14
1.2.2.2. Ingeniería inversa de lógica o de proceso	14
1.2.2.3. Ingeniería inversa de interfaces de usuario	15
1.2.3. Herramientas para la Ingeniería	15
1.2.3.1. Los Depuradores	15
1.2.3.2. Las Herramientas de Inyección de Fallos	16
1.2.3.3. Los Desensambladores	17
1.2.3.4. Los compiladores Inversos o Decompiladores	17
1.2.3.5. Las Herramientas CASE	18
1.2.3.5.1. Componentes de una Herramienta Case.	19
1.2.3.5.2. Estructura general de una Herramienta Case.	20
1.2.3.5.3. Características de las Herramientas Case.	21

CAPITULO II:

2. Descripción, Análisis e Interpretación de Resultados	23
2.1 Entorno de la Empresa Mercado Latacunga	23
2.1.1 Antecedentes Históricos	23
2.1.2 Funciones	24
2.1.3. Objetivos de la empresa Mercado Latacunga	25
2.1.4. Misión de la empresa Mercado Latacunga.	25
2.1.5. Visión de la empresa Mercado Latacunga.	25
2.1.6. Estructura Organizacional	26
2.1.7. Actividades que se desempeña dentro de la empresa Mercado Latacunga.	27
2.1.7.1. Departamento de Gerencia.	27
2.1.7.2. Departamento de Sub Gerencia.	27
2.1.7.3. Departamento de Contabilidad.	27
2.1.7 .4. Departamento de Proyectos.	28
2.1.7 .5. Departamento de Diseño Gráfico.	29
2.1.7 .6. Departamento de Calidad.	30
2.2 Tipo de Investigación.	31
2.2.1. Investigación de campo.	31

2.2.2. Investigación Bibliográfica.	31
2.2.3. Tipo de investigación experimental.	32
2.3 Análisis de los Resultados de la Encuesta	33

CAPITULO III:

Estudio comparativo entre las herramientas libres de análisis y desarrollo staruml y fujaba para establecer la usabilidad y la aplicabilidad de la ingeniería inversa dentro del proceso de desarrollo de software en la Empresa Mercado Latacunga	42
--	----

3.1. Propuesta	42
3.2 Objetivos	42
3.3 Justificación de la Propuesta	43
3.4 Factibilidad de Aplicar la Propuesta	45
3.5 Desarrollo de la Propuesta	45
3.5.1 Descripción de las herramientas de análisis	45
3.5.1.1. Open Source	45
3.5.1.2. Uml	46
3.5.1.3. starUML	48
3.5.1.4. Java.	48
3.5.1.5. Fujaba.	48
3.5.1.6. Power Designer.	49
3.5.2 Análisis Comparativo entre las herramientas starUML y Fujaba.	49
3.5.2.1 Recopilación - clasificación de herramientas de modelado de procesos.	49
3.5.2.2. Asignación de pesos	49
3.5.2.3. Estudio Comparativo.	51
3.5.2.4. Recopilación - clasificación de herramientas modelado.	51
3.5.2.5. Herramientas Descripción del modelado de proceso.	51
3.5.2.6. Evaluación de las herramientas.	51
3.5.2.7. Tipo de Software	52
3.5.2.8. Plataforma.	52
3.5.2.9. Generación de código.	53

3.5.2.10. Soporte de ingeniería inversa.	53
3.5.2.11. Soporte a UML.	53
3.5.2.12. Estabilidad de la herramienta.	53
3.5.2.13 Diagramas UML	53
3.5.2.14. Selección de la herramienta	54
3.5.3. Caso Práctico	55
3.5.3. Análisis Caso Práctico	57
3.6. Actores y Responsabilidades	57
3.7. Verificación de Hipótesis	58
3.8. Verificación de Objetivos	61
3.9. Conclusiones y Recomendaciones	61
3.9.1. Conclusiones	61
3.9.2. Recomendaciones	62
CAPITULO IV	
Conclusiones y Recomendaciones	
4.1 Conclusiones	63
4.2 Recomendaciones	63
Bibliografía	64
Glosario de Términos	68
Anexos	72

ÍNDICE DE GRÁFICOS

Grafico N° 2.1. Organigrama de la empresa Mercado Latacunga	26
Gráfico N° 2.2. Valoración de la utilización del software libre.	34
Gráfico N° 2.3. Beneficios del software libre.	35
Gráfico N° 2.4. Herramientas libres para el desarrollo de software.	36
Gráfico N° 2.5. Herramientas de análisis para el desarrollo de software.	37
Gráfico N° 2.6. Metodología(s) para el desarrollo de software.	38
Gráfico N° 2.7. Conoce herramientas libres de análisis.	39
Gráfico N° 2.8. Herramientas de análisis StarUML y Fujaba.	40
Gráfico N° 2.9. Usted implementaría en el desarrollo de software.	41
Grafico N° 3.1. StarUML	56
Grafico N° 3.2. Fujaba	56

ÍNDICE DE TABLAS

Tabla N° 2.1. Valoración De La Utilización Del Software Libre.	34
Tabla N° 2.2. Beneficios del software libre.	35
Tabla N° 2.3. Herramientas libres para el desarrollo de software.	36
Tabla N° 2.4. Herramientas de análisis para el desarrollo de software.	37
Tabla N° 2.5. Metodología(s) para el desarrollo de software.	38
Tabla N° 2.6. Conoce herramientas libres de análisis.	39
Tabla N° 2.7. Herramientas de análisis StarUML y Fujaba.	40
Tabla N° 2.8. Usted implementaría en el desarrollo de software.	41
Tabla N° 3.1. Diagramas	47
Tabla N° 3.2. Asignación de Pesos	51
Tabla N° 3.3. Matriz Comparativa de las Herramientas	52
Tabla N° 3.4. Matriz Comparativa de las Herramientas con pesos	55
Tabla N° 3.5. Tabulación de Encuestas Empleadas	58
Tabla N° 3.6. Frecuencias Observables	59
Tabla N° 3.7. Frecuencias Esperadas	59
Tabla N° 3.8. Cálculo del $\chi^2 = \frac{(f_o - f_e)^2}{f_e}$	60

TEMA: “ESTUDIO COMPARATIVO ENTRE LAS HERRAMIENTAS LIBRES DE ANALISIS Y DESARROLLO STARUML Y FUJABA PARA ESTABLECER LA USABILIDAD Y LA APLICABILIDAD DE LA INGENIERIA INVERSA DENTRO DEL PROCESO DE DESARROLLO DE SOFTWARE EN LA EMPRESA MERCADO LATACUNGA”

Autores: Cepeda Vaca Luis Orlando
Medina Chiluisa Fabián Eduardo

RESUMEN

Fue desarrollado con tecnología OpenSource, es decir, que es de acceso gratuito en Internet. Todas las herramientas para el desarrollo han sido descargadas de Internet, así tenemos starUML, Fujaba. Las demás herramientas utilizadas trabajan bajo licencias adquiridas. El presente artículo, muestra un estudio comparativo de las herramientas empleadas para el diseño de procesos, teniendo en cuenta sus características y ventajas principales, que ayudarán a establecer cuál de éstas tiene un mejor desempeño y brinda mayores prestaciones, al momento de utilizarlas. Este trabajo hace parte de investigación "el estudio comparativo de herramientas de análisis en el desarrollo de software", que aportará el medio de selección apropiado de acuerdo a las características relevantes que determinan el estudio, determinando la factibilidad de su uso de acuerdo a las ventajas que aportarían cada uno de ellas. Se presentan aspectos generales del trabajo enfocado bajo ciertas normas de ingeniería que se deben seguir; luego se plantean algunas funciones de los grupos que hacen parte del equipo de trabajo; posteriormente se comentan algunos casos prácticos en los cuales se han utilizado técnicas y métodos de ingeniería de software libre. Y finalmente se dan algunas conclusiones.

THEME: ‘COMPARATIVE STUDY BETWEEN OF FREE TOOLS OF ANALYSIS AND STARUML DEVELOPMENT AND FUJABA FOR ESTABLISH THE USABILITY AND APPLICABILITY OF THE REVERSE ENGINEERING IN THE SOFTWARE DEVELOPMENT PROCESS IN LATACUNGA MARKET-COMPANY’

Authors: Cepeda Vaca Luis Orlando
Medina Chiluisa Fabián Eduardo

SUMMARY

Was developed with Open Source technology, namely that is free on the Internet. All development tools have been downloaded from the Internet, so we have starUML, Fujaba. The other tools used works under licenses purchased. This article sample a comparative study of the used tools for the process design, given their characteristics and the principal advantages, that helped to establish which have the best performance and provides the best services at the time of use. This work is part of investigation "Comparative study of analysis tools in software development", which will provide the appropriate selection medium according to the relevant features which determine the study, determining the feasibility of their use according to the benefits posed by each of them. We present general aspects of the work focused on certain engineering standards to be followed, then raises some functions of the groups that are part of the work team. Later discusses some practical cases in which techniques have been used and software engineering methods free. And finally, some conclusions.

INTRODUCCION

Es necesario conocer las necesidades, demandas que tienen los clientes para el desarrollo de software, por lo tanto, los estudios de requerimientos del software, constituyen la herramienta idónea para conseguir ese fin. Además, permiten evaluar la calidad, para establecer procesos de mejoras e innovación durante el desarrollo en aquellos que presentan debilidades, así como fortalecer los que están en funcionamiento. Por lo anterior, esta investigación centra su atención en la ejecución de un estudio de comparación entre las herramientas libres (StarUML y Fujaba) y las herramientas con licencia (Rational Rose, etc.), a partir de ello, presentar un modelo practico para su realización y aplicación al momento de desarrollar un software para cualquier empresa, que permita evaluar las ventajas que estos prestan. Del mismo modo, su calidad, con el propósito de plantear mejoras necesarias para que logren la óptima satisfacción de sus clientes.

Debido a que los usuarios son la razón de ser, es indispensable que los encargados del desarrollo conozcan a sus clientes para que puedan satisfacer sus demandas informativas y proporcionar servicios de calidad. Por lo cual, el programador tiene la responsabilidad de afrontar y asumir los nuevos retos que exige la sociedad de la información y la diversidad de clientes que acuden, que cada vez son más exigentes y requieren un software de calidad. Este documento muestra un nuevo enfoque, que permite observar de manera diferente la forma de desarrollar software, y, por ende, traza algunas pautas para que este trabajo de carácter colectivo se realice de manera eficiente. La distribución del escrito está dada de forma tal que el lector tiene una guía práctica y secuencial para adentrarse en conceptos básicos de las herramientas utilizadas.

En primera instancia se presentan aspectos generales del trabajo enfocado bajo ciertas normas de ingeniería que se deben seguir; luego se plantean algunas funciones de los grupos que hacen parte del equipo de trabajo; posteriormente se comentan algunos casos prácticos en los cuales se han utilizados técnicas y métodos de ingeniería de software libre. Y finalmente se dan algunas

conclusiones. Algunos de los aportes aquí presentados son parte de la experiencia, en el desarrollo con ingeniería de software y de proyectos de software libre que se desarrollan dentro de la empresa “MERCADO LATACUNGA”.

El CAPÍTULO I denominado "FUNDAMENTACIÓN TEÓRICA", presenta un enfoque sobre las nociones fundamentales de las herramientas libres, historia, tecnologías de desarrollo, lenguajes de programación con orientación web.

El CAPÍTULO II denominado "DESCRIPCIÓN, ANÁLISIS E INTERPRETACIÓN DE RESULTADOS", presenta una descripción de la finalidad del estudio, la localidad y personas comprometidas, y sus generalidades. El análisis descriptivo se centra en cómo, dónde y quién recolectó la información, lo cual implica revisar la información, identificar vínculos, patrones y temas comunes, ordenar los hechos y presentarlos como son.

El CAPÍTULO III denominado “ESTUDIO COMPARATIVO ENTRE LAS HERRAMIENTAS LIBRES DE ANALISIS Y DESARROLLO STARUML Y FUJABA PARA ESTABLECER LA USABILIDAD Y LA APLICABILIDAD DE LA INGENIERIA INVERSA DENTRO DEL PROCESO DE DESARROLLO DE SOFTWARE EN LA EMPRESA MERCADO LATACUNGA”, describe una comparación entre las Fujaba y starUML que permitirá determinar un estudio comparativo de adaptación para el desarrollo del software.

El CAPÍTULO IV presenta las conclusiones y recomendaciones en donde se encuentra descrito hasta qué punto fue factible la realización del estudio comparativo y algunas recomendaciones para que en la posteridad se dé el tratamiento de las herramientas libres en los centros de educación y en especial dentro de esta alma mater.

CAPITULO I

FUNDAMENTACIÓN TEÓRICA

1.1. HERRAMIENTAS DE SOFTWARE LIBRE

1.1.1. Software Libre

De acuerdo con la página web <http://www.gnu.org/philosophy/free-sw.es.html> nos dice que **SOFTWARE LIBRE**: “Significa que el software respeta la libertad de los usuarios y la comunidad. En términos generales, **los usuarios tienen la libertad de copiar, distribuir, estudiar, modificar y mejorar el software**. Con estas libertades, los usuarios (tanto individualmente como en forma colectiva) controlan el programa y lo que hace.”. (CONSULTADO EL 15-07-2013).

De acuerdo con la página web <http://raifranco.blogspot.com/2012/05/definicion-de-software-libre.html> nos dice que **SOFTWARE LIBRE**: “Es la denominación del software que respeta la libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, modificado, y redistribuido libremente.”. (CONSULTADO EL 17-07-2013).

De lo mostrado anteriormente se puede decir que software libre es aquel que una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

1.1.2. Tipos de Software Libre

1.1.2.1. Software de dominio público.

De acuerdo con la página web http://www.reducativa.com/webquests/licencias/dominio_publico.html nos dice que **SOFTWARE DE DOMINIO PÚBLICO**: “Es aquél por el que no es necesario solicitar ninguna licencia y cuyos derechos de explotación son para toda la humanidad, porque pertenece a todos por igual. Cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría original. Este software sería aquél cuyo autor lo dona a la humanidad o cuyos derechos de autor han expirado. El software de dominio público (software que no está sujeto al copyright de nadie) es software libre; pero cualquiera puede hacer una versión modificada propietaria a partir de él. En ese mismo sentido, muchos programas libres están sujetos a copyright pero se distribuyen mediante sencillas licencias permisivas que admiten las versiones modificadas propietarias.” (CONSULTADO EL 17-07-2013).

De acuerdo con la página web <http://www.cavsi.com/preguntasrespuestas/que-es-software-de-dominio-publico/> nos manifiesta que **SOFTWARE DE DOMINIO PÚBLICO**: “El software de dominio público es un software libre que no tiene un propietario, por ende no existen derechos de autor, licencias o restricciones de distribución. Por este concepto, el software de dominio público se diferencia de un freeware, el cual conserva los derechos de autor. El software de dominio público puede ser libremente utilizado, copiado o alterado, porque legalmente nadie es dueño de los derechos de restringir su uso.” (CONSULTADO EL 17-07-2013).

De lo presentado anteriormente se puede decir que software de dominio público es el software que no está protegido por las leyes de derechos de autor y puede ser copiado por cualquier persona que lo requiera sin costo alguno y ser modificadas como si fueran propiedad de este último.

1.1.2.2. Software protegido con copyleft.

De acuerdo con la página web <http://www.deciencias.net/simulaciones/paginas/libre.htm>, nos dice que **SOFTWARE PROTEGIDO CON COPYLEFT**: “El software protegido con copyleft es software libre cuyos términos de distribución no permiten a los redistribuidores agregar ninguna restricción adicional cuando éstos redistribuyen o modifican el software. Esto significa que cada copia del software, aun si ha sido modificado, debe ser software libre. Copyleft es un concepto general; para proteger actualmente un programa con copyleft, necesita usar un conjunto específico de términos de distribución. Hay muchas maneras posibles de escribir términos copyleft de distribución.” (CONSULTADO EL 17-07-2013).

De acuerdo con la página web http://e-educativa.catedu.es/44003247/aula/archivos/repositorio//0/236/Tipos_de_software.pdf, nos indica que **SOFTWARE PROTEGIDO CON COPYLEFT** : “El software protegido con copyleft es software libre cuyos términos de distribución aseguran que todas las copias de todas las versiones son software libre. Esto significa, por ejemplo, que las licencias copyleft no permiten a terceros agregar ningún requisito adicional (a excepción de un conjunto limitado de requisitos para aumentar su protección) y exige que el código fuente sea público.” (CONSULTADO EL 17-07-2013).

De lo enseñado anteriormente se puede decir que software protegido con copyleft es aquel que no puede ser modificado y luego que el programador quiera venderlo. Su copyleft asegura que sus modificaciones se distribuyan gratis también y con su código fuente.

1.1.2.3. Software no protegido con copyleft.

De acuerdo con la página web http://e-educativa.catedu.es/44003247/aula/archivos/repositorio//0/236/Tipos_de_software.pdf, nos señala que **SOFTWARE LIBRE NO PROTEGIDO CON COPYLEFT**: “El software libre

no protegido con copyleft, incluye la autorización del autor para redistribuir y modificar el software, así como el permiso para añadirle restricciones adicionales. El que un programa sea libre pero no esté protegido con copyleft, implica que algunas copias o versiones modificadas del mismo pueden no ser completamente libres. Una compañía de software podría compilar el programa, con o sin modificaciones, y distribuir el archivo ejecutable como un producto de software privativo.” (CONSULTADO EL 17-07-2013).

De acuerdo con la página web <http://www.deciencias.net/simulaciones/paginas/libre.htm>, nos señala que **SOFTWARE LIBRE NO PROTEGIDO CON COPYLEFT**: “El software libre no protegido con copyleft viene desde el autor con autorización para redistribuir y modificar así como para añadirle restricciones adicionales. Si un programa es libre pero no protegido con copyleft, entonces algunas copias o versiones modificadas pueden no ser libres completamente. Una compañía de software puede compilar el programa, con o sin modificaciones, y distribuir el archivo ejecutable como un producto propietario de software.” (CONSULTADO EL 17-07-2013).

De lo citado anteriormente se puede decir que software no protegido con copyleft es aquel software que viene con permiso de redistribución y modificación, como así también con el permiso de agregarle restricciones.

1.1.2.4. Software cubierto con la GPL.

De acuerdo con la página web http://scielo.sld.cu/scielo.php?pid=S1024-94352005000600009&script=sci_arttext, nos señala que **SOFTWARE CUBIERTO CON GPL**: “es un conjunto específico de términos de distribución para proteger con copyleft a un programa. El Proyecto GNU la utiliza como los términos de distribución para la mayoría del software GNU.” (CONSULTADO EL 17-07-2013).

De lo mencionado anteriormente se puede decir que software cubierto con GPL es aquel software que viene con una licencia la misma que es la que permite utilizar, modificar el código o el programa.

1.1.2.5. *Software privativo.*

De acuerdo con la página web <http://www.gnu.org/philosophy/categories.es.html>, nos manifiesta que: “El software privativo es otro nombre para designar el software que no es libre. En el pasado habíamos subdividido el software que no es libre en «software semilibre», que podía ser modificado y redistribuido sin fines comerciales, y «software privativo», que no podía ser modificado ni redistribuido. Pero hemos abandonado esta distinción y ahora utilizamos el término «software privativo» como sinónimo de software que no es libre.” (CONSULTADO EL 17-07-2013).

De acuerdo con la página web <http://jalvarezm.wordpress.com/2006/09/30/el-software-privativo/>, nos manifiesta que: “Se conoce como software privativo a los programas informáticos amparados bajo licencias que reservan algunos o todos los derechos de uso, copia, modificación y distribución para el fabricante, quien previo pago de una regalía concede el uso de una copia ejecutable del programa al titular de la licencia. El usuario no es dueño del software que está funcionando en su computador, el propietario sigue siendo el fabricante y no faculta al usuario a realizar ninguna modificación en él, ni a tampoco estudiarlo por ninguna vía para determinar cómo realiza sus funciones. Asimismo, el usuario tiene prohibida la divulgación por cualquier medio de fallos que descubriese en el programa. E incluso, algunas licencias impiden al comprador la transferencia de la titularidad a un tercero.” (CONSULTADO EL 17-07-2013).

De lo señalado anteriormente se puede decir que el software privativo es cualquier programa informático en el que el usuario tiene limitaciones para usarlo, modificarlo o redistribuirlo.

1.1.2.6. Freeware.

De acuerdo con la página web <http://www.taringa.net/posts/info/9270788/Definiciones-Shareware-Freeware-Demo-y-Licencia.html>, manifiesta que **FREEWARE:** “El término freeware define un tipo de software privativo que se distribuye sin costo, disponible para su uso y por tiempo ilimitado siendo una variante gratuita del shareware, en el que la meta es lograr que un usuario pruebe el producto durante un tiempo ("trial" limitado, y si le satisface, pague por él, habilitando toda su funcionalidad. A veces se incluye el código fuente pero no es lo usual.” (CONSULTADO EL 17-07-2013).

De acuerdo con la página web <http://www.ecured.cu/index.php/Freeware>, manifiesta que **FREEWARE:** “Es el software que está disponible para su uso sin costo alguno o algún costo nominal voluntario, este está protegido por su desarrollador, que retiene todos los derechos para controlar su distribución[1][2]. Generalmente el freeware incluye una licencia de uso, que permite su redistribución pero con algunas restricciones, entre estas están la prohibición de modificación de la aplicación, prohibición de venta o la restricción de uso para usuarios con fines comerciales o el requerimiento de un pago para permitirlo.” (CONSULTADO EL 17-07-2013).

De lo citado anteriormente se puede decir que Freeware es aquel que suele incluir una licencia de uso, que permite redistribuir pero con algunas restricciones por ejemplo, no permite modificarlo, ni venderlo.

1.1.2.7. Shareware

De acuerdo con la página web <http://www.aulaclie.es/articulos/licencias.html> manifiesta que **SHAREWARE:** “Es otra modalidad de comercialización todavía más extendida, el programa se distribuye con limitaciones, bien como versión de demostración o evaluación, con funciones o características limitadas o con un uso restringido a un límite de tiempo establecido (por ejemplo 30 días). Así, se le da al

usuario la oportunidad de probar el producto antes de comprarlo y, más tarde, adquirir la versión completa del programa.” (CONSULTADO EL 18-07-2013).

De acuerdo con la página web <http://alvarolivares.wordpress.com/2009/02/25/software-libre-concepto-y-mercado-freeware-y-shareware/> manifiesta que **SHAREWARE**: “Es una modalidad de distribución de software, para que el mismo pueda ser evaluado de forma gratuita, dicha copia de evaluación puede no ser completa o bien caducar después de un período de uso. Si el usuario desea utilizar de forma permanente o completa dicho software, debe comprar su licencia. Con posterioridad a la compra el desarrollador envía al usuario una copia completa del programa o la clave necesaria para superar las limitaciones señaladas.” (CONSULTADO EL 18-07-2013).

De lo mencionado anteriormente se puede decir que Shareware es un tipo de software que es distribuido gratuitamente exclusivamente para ser probado, pero posee restricciones en su funcionalidad o disponibilidad, por lo general son limitados a 30 días de uso, pero también algunos desactivan opciones como "Guardar", o tienen limitado el número de veces que pueden ejecutarse, En caso de que al usuario le convenza el software podrá comprarlo

1.1.3. Staruml

De acuerdo con el autor Cessario Jorge Andrés (2009) manifiesta que **STARMUL**: “StarUML es un lenguaje de modelado para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso usar”. (CONSULTADO EL 18-07-2013).

De acuerdo con la página web <http://black-byte.com/review/staruml/> menciona que **StarUML**: “Es una herramienta para el modelamiento de software basado en los estándares UML (Unified Modeling Language) y MDA (Model Driven Architecture), que en un principio era un producto comercial y que hace cerca de un año paso de ser un proyecto comercial (anteriormente llamado plastic) a uno de licencia abierta GNU/GPL.” (CONSULTADO EL 18-07-2013).

De lo expuesto anteriormente se puede decir que StarUML es una herramienta para el modelamiento de software basado en los lenguajes UML (Unified Modeling Language) y MDA (Model Driven Architecture), y que nos permite describir métodos o procesos dentro del desarrollo del software.

1.1.4. Uml

De acuerdo con la página web http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado, manifiesta que **UML**: “UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas”.

Según Corradini Horacio (2008) menciona **UML: “Lenguaje Unificado de Modelado** (LUM) o (**UML**, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del

sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables”.

De lo mostrado anteriormente se puede decir que el UML es un popular lenguaje de modelado de sistemas de software. Se trata de un lenguaje gráfico para construir, documentar, visualizar y especificar un sistema de software. Entre otras palabras, UML se utiliza para definir un sistema de software; además posee la riqueza suficiente como para crear un modelo del sistema, pudiendo modelar los procesos de negocios, funciones, esquemas de bases de datos, expresiones de lenguajes de programación, etc. Para ello utiliza varios tipos diferentes de diagramas

1.1.5. Fujaba

Según la página web <http://translate.google.com.ec/translate?hl=es&sl=en&u=http://www.fujaba.de/&prev=/search%3Fq%3Dque%2Bes%2Bfujaba%26biw%3D1230%26bih%3D567> menciona que **FUJABA**: “Es una herramienta CASE libre ofreciendo a los desarrolladores con la ayuda para la ingeniería de software basado en modelos y reingeniería. Los proyectos Fujaba su objetivo es la elaboración y ampliación de la suite de herramientas Fujaba y ofreciendo así una platform extensible para los investigadores de ingeniería de software.” (CONSULTADO EL 18-07-2013).

Según la página web <http://translate.google.com.ec/translate?hl=es&sl=en&u=http://java-source.net/open-source/uml-modeling/fujaba&prev=/search%3Fq%3Dfujaba%26biw%3D1230%26bih%3D567> menciona que **FUJABA**: “Es el acrónimo de 'From UML para Java And Back Again' y en un principio tiene como objetivo proporcionar un entorno para la ingeniería de ida y vuelta con UML como lenguaje de programación visual.” (CONSULTADO EL 18-07-2013).

De lo presentado anteriormente se puede ver que Fujaba es una herramienta CASE extensible mediante plugins, y pensada para realizar ingeniería de ida y vuelta empleando UML y Java.

1.2. INGENIERÍA INVERSA EN EL PROCESO DE DESARROLLO

1.2.1. Ingeniería Inversa

De acuerdo con Reyes Rodríguez, Antonio Manuel (2008). Dice que **INGENIERIA INVERSA**: “El objetivo de la ingeniería inversa es obtener información a partir de un producto accesible al público, con el fin de determinar de qué está hecho, qué lo hace funcionar y cómo fue fabricado. Los productos más comunes que son sometidos a la ingeniería inversa son los programas de computadoras y los componentes electrónicos, pero básicamente casi cualquier proceso puede ser sometido a un análisis de Ingeniería Inversa.”

Según Montaña Cruz (2009). **INGENIERIA INVERSA**: “La ingeniería inversa es un método de resolución. Aplicar ingeniería inversa a algo supone profundizar en el estudio de su funcionamiento, hasta el punto de que podemos llegar a entender, modificar, y mejorar dicho modo de funcionamiento.

Pero este término no sólo se aplica al software de protección. Así pues se considera ingeniería inversa también al estudio de todo tipo de elementos, por ejemplo equipos electrónicos, microcontroladores, etc., siempre y cuando el resultado de dicho estudio repercuta en el entendimiento de su funcionamiento.

Se denomina ingeniería inversa del software a la actividad que se ocupa de descubrir cómo funciona un programa, función o característica de cuyo código fuente no se dispone, hasta el punto de poder modificar ese código o generar código propio que cumpla las mismas funciones. La gran mayoría del software de pago incluye en su licencia una prohibición expresa de aplicar ingeniería inversa a

su código, con el intento de evitar que se pueda modificar su código y que así los usuarios tengan que pagar si quieren usarlo.”

Según García-Badell Lapetra, José Javier (2009) **INGENIERIA INVERSA**: “Se denomina ingeniería inversa del software a la actividad que se ocupa de descubrir cómo funciona un programa, función o característica de cuyo código fuente no se dispone, hasta el punto de poder modificar ese código”.

La idea general de emplear la Ingeniería Inversa es:

- Conocer a fondo la aplicación y generar un mejor código.
- Migrar la aplicación a un nuevo sistema operativo.
- Hacer o completar la documentación.

Verificar que el código en estudio cumple las especificaciones de diseño estándar. La información extraída de la lectura del código fuente son las especificaciones de diseño: Modelos de flujo de control, diagramas de diseño, documentos de especificación de diseño, pudiendo tomarse estas especificaciones como nuevo punto de partida para aplicar ingeniería inversa y obtener información a mayor nivel de abstracción.

Los lenguajes de programación orientados a objetos no contienen ni la sintaxis ni la semántica necesaria para expresar directamente las asociaciones. Por tanto, hay que implementar las asociaciones UML mediante una adecuada combinación de clases, atributos y métodos. Este artículo presenta algunos principios para la implementación de asociaciones binarias UML en Java, prestando especial atención a la multiplicidad, la navegabilidad y visibilidad. Nuestro análisis ha encontrado algunas paradojas en la especificación de la visibilidad para las asociaciones bidireccionales. Hemos utilizado estos principios para escribir una serie de plantillas de código que utilizamos en combinación con una herramienta para generar el código de las asociaciones, que son leídas de un modelo almacenado en formato XMI.

Este método es denominado ingeniería inversa porque avanza en dirección opuesta a las tareas habituales de ingeniería, que consisten en utilizar datos técnicos para elaborar un producto determinado. En general si el producto u otro material que fue sometido a la ingeniería inversa fue obtenido en forma apropiada, entonces el proceso es legítimo y legal. De la misma forma, pueden fabricarse y distribuirse, legalmente, los productos genéricos creados a partir de la información obtenida de la ingeniería inversa, como es el caso de algunos proyectos de Software libre ampliamente conocidos.”

De lo presentado anteriormente se puede decir que la ingeniería inversa se define como el proceso de construir especificaciones de un mayor nivel de abstracción partiendo del código fuente de un sistema software o cualquier otro producto

1.2.2. TIPOS DE INGENIERIA INVERSA

1.2.2.1. Ingeniería inversa de datos

De acuerdo a página web <http://cnx.org/content/m17432/latest/> manifiesta que **Ingeniería inversa de datos:** “Se aplica sobre algún código de bases datos (aplicación, código SQL, etc) para obtener los modelos relacionales o sobre el modelo relacional para obtener el diagrama entidad-relación.” (CONSULTADO EL 18-07-2013).

De lo citado anteriormente se puede decir que la ingeniería inversa de datos es el que permite la obtención de una representación conceptual de un esquema de base de datos a partir de su codificación.

1.2.2.2. Ingeniería inversa de lógica o de proceso

De acuerdo a página web <http://cnx.org/content/m17432/latest/> manifiesta que **Ingeniería inversa de lógica o de proceso:** “Cuando la ingeniería inversa se aplica sobre código de un programa para averiguar su lógica o sobre cualquier

documento de diseño para obtener documentos de análisis o de requisitos.” (CONSULTADO EL 18-07-2013).

De lo señalado anteriormente se puede decir que la ingeniería inversa de lógica o de procesos es la que nos permite averiguar su lógica o sobre cualquier documento de diseño para obtener documentos de análisis o de requisitos.

1.2.2.3. Ingeniería inversa de interfaces de usuario

De acuerdo a página web <http://cnx.org/content/m17432/latest/> manifiesta que **TIPOS DE INGENIERÍA INVERSA:** “Se aplica con objeto de mantener la lógica interna del programa para obtener los modelos y especificaciones que sirvieron de base para la construcción de la misma, con objeto de tomarlas como punto de partida en procesos de ingeniería directa que permitan modificar dicha interfaz.” (CONSULTADO EL 18-07-2013).

De lo referido anteriormente se puede decir que la ingeniería inversa de interfaces de usuario nos permite obtener la relación entre los distintos componentes de la interfaz de usuario, la cual nos permitirá identificar los diferentes tipos de modelos que actúan en el modelo de interfaz.

1.2.3. Herramientas para la Ingeniería

1.2.3.1. Los Depuradores

De acuerdo a página web <http://cnx.org/content/m17432/latest/> manifiesta que **Los Depuradores:** “Un depurador es un programa que se utiliza para controlar otros programas. Permite avanzar paso a paso por el código, rastrear fallos, establecer puntos de control y observar las variables y el estado de la memoria en un momento dado del programa que se esté depurando. Los depuradores son muy valiosos a la hora de determinar el flujo lógico del programa.” (CONSULTADO EL 18-07-2013).

De acuerdo a página web <http://learnassembler.com/gdbesp.html> manifiesta que **Los Depuradores:** “Un depurador es una aplicación que permite correr otros programas, permitiendo al usuario ejercer cierto control sobre los mismos a medida que los estos se ejecutan, y examinar el estado del sistema (variables, registros, banderas, etc.) en el momento en que se presente algún problema. El propósito final de un depurador consiste en permitir al usuario observar y comprender lo que ocurre "dentro" de un programa mientras el mismo es ejecutado.” (CONSULTADO EL 18-07-2013).

De lo antedicho anteriormente se puede decir que los depuradores son programas que se usan para probar y eliminar los errores de otros programas.

1.2.3.2. Las Herramientas de Inyección de Fallos

De acuerdo a página web <http://cnx.org/content/m17432/latest/> manifiesta que **Las Herramientas de Inyección de Fallos:** “Las herramientas que pueden proporcionar entradas malformadas con formato inadecuado a procesos del software objetivo para provocar errores son una clase de herramientas de inserción de fallos. Los errores del programa pueden ser analizados para determinar si los errores existen en el software objetivo.” (CONSULTADO EL 21-07-2013).

De acuerdo a página web http://centrodeartigos.com/articulos-educativos/article_11736.html, manifiesta que **Las Herramientas de Inyección de Fallos:** “es una técnica para la mejora de la cobertura de una prueba mediante la introducción de faltas para probar las rutas de código, en particular, las rutas de código de gestión de errores, que de otro modo raramente seguirse.” (CONSULTADO EL 21-07-2013).

De lo mostrado anteriormente se puede decir que las herramientas de inyección de fallos es un método que consiste en modelar el comportamiento del sistema a

testar usando un lenguaje de alto nivel, y al momento de ser ejecutado emitirá un reporte de los fallos que tenga.

1.2.3.3. Los Desensambladores

De acuerdo a página web <http://cnx.org/content/m17432/latest/> manifiesta que **Los Desensambladores:** “Se trata de una herramienta que convierte código máquina en lenguaje ensamblador. El lenguaje ensamblador es una forma legible para los humanos del código máquina. Los desensambladores revelan que instrucciones máquinas son usadas en el código. El código máquina normalmente es específico para una arquitectura dada del hardware. De forma que los desensambladores son escritor expresamente para la arquitectura del hardware del software a desensamblar.” (CONSULTADO EL 21-07-2013).

De acuerdo a página web <http://www.viruslist.com/sp/hackers/glossary?glossid=153598730> manifiesta que **Los Desensambladores:** “Son herramientas utilizadas para transformar el código binario en lenguaje armado.” (CONSULTADO EL 21-07-2013).

De lo manifiesto anteriormente se puede decir que los desensambladores es un programa de computadora que traduce el lenguaje de máquina a lenguaje ensamblador,

1.2.3.4. Los compiladores Inversos o Decompiladores

De acuerdo a página web <http://cnx.org/content/m17432/latest/> manifiesta que **Los compiladores Inversos o Decompiladores:** “es una herramienta que transforma código en ensamblador o código máquina en código fuente en lenguaje de alto nivel. También existen decompiladores que transforman lenguaje intermedio en código fuente en lenguaje de alto nivel. Estas herramientas son sumamente útiles para determinar la lógica a nivel superior como bucles o declaraciones if-then de los programas que son decompilados. Los

decompiladores son parecidos a los desensambladores pero llevan el proceso un importante paso más allá.” (CONSULTADO EL 22-07-2013).

De acuerdo a página web <http://www.buenastareas.com/ensayos/Decompiladores/172515.html> muestra que **Los compiladores Inversos o Decompiladores:** “Realizan la tarea inversa a los compiladores, es decir son un caso particular de los traductores en los cuales el programa fuente es un lenguaje de bajo nivel y el lenguaje objeto es un lenguaje de nivel superior. Esta tarea es difícil, sobre todo si se desea que el lenguaje objeto tenga una cierta lógica, a no ser que se conozca la forma en que se obtuvo el programa de bajo nivel (por ejemplo si se generó con un compilador determinado, y se tiene el esquema de generación de código de dicho compilador). (CONSULTADO EL 22-07-2013).

De lo mostrado anteriormente se puede decir que los compiladores inversos o decompiladores es un programa que realiza la operación inversa de un compilador: obtener el código fuente de un programa compilado.

1.2.3.5. Las Herramientas CASE

De acuerdo a página web <http://cnx.org/content/m17432/latest/> **Las Herramientas CASE:** “Las herramientas de ingeniería de sistemas asistida por ordenador (Computer-Aided Systems Engineering – CASE) aplican la tecnología informática a las actividades, las técnicas y las metodologías propias de desarrollo de sistemas para automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas. En el caso de la ingeniería inversa generalmente este tipo de herramientas suelen englobar una o más de las anteriores junto con otras que mejoran el rendimiento y la eficiencia.” (CONSULTADO EL 22-07-2013).

De acuerdo con Kendall, Keneth y Kendall, Julie (2009) **Las Herramientas CASE:** “La ingeniería de sistemas asistida por ordenador es la aplicación de tecnología informática a las actividades, las técnicas y las metodologías propias de desarrollo, su objetivo es acelerar el proceso para el que han sido diseñadas, en el

caso de CASE para automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas.

Cuando se hace la planificación de la base de datos, la primera etapa del ciclo de vida de las aplicaciones de bases de datos, también se puede escoger una herramienta CASE (Computer-Aided Software Engineering) que permita llevar a cabo el resto de tareas del modo más eficiente y efectivo posible. Una herramienta CASE suele incluir:

Un diccionario de datos para almacenar información sobre los datos de la aplicación de bases de datos.

Herramientas de diseño para dar apoyo al análisis de datos.

Herramientas que permitan desarrollar el modelo de datos corporativo, así como los esquemas conceptual y lógico.”

De lo manifestado anteriormente se puede manifestar que Las herramientas CASE permite mejorar la productividad en el desarrollo de una aplicación de bases de datos a través de los diferentes modelos que tiene esta herramienta para lograr apoyar las fases de ciclo de vida del desarrollo de los sistemas

1.2.3.5.1. Componentes de una Herramienta Case.

De acuerdo a página web <http://herramientascase.wordpress.com/integracion-estructura-componentes/> manifiesta que componentes de una herramienta CASE: “De una forma esquemática podemos decir que una herramienta CASE se compone de los siguientes elementos:

Repositorio (diccionario): donde se almacenan los elementos definidos o creados por la herramienta, y cuya gestión se realiza mediante el apoyo de un Sistema de Gestión de Base de Datos (SGBD) o de un sistema de gestión de ficheros.

Meta modelo (no siempre visible): que constituye el marco para la definición de las técnicas y metodologías soportadas por la herramienta.

Carga o descarga de datos: son facilidades que permiten cargar el repertorio de la herramienta CASE con datos provenientes de otros sistemas, o bien generar a partir de la propia herramienta esquemas de base de datos, programas, etc. que pueden, a su vez, alimentar otros sistemas. Este elemento proporciona así un medio de comunicación con otras herramientas.

Comprobación de errores: facilidades que permiten llevar a cabo un análisis de la exactitud, integridad y consistencia de los esquemas generados por la herramienta.

Interfaz de usuario: que constará de editores de texto y herramientas de diseño gráfico que permitan, mediante la utilización de un sistema de ventanas, iconos y menús, con la ayuda del ratón, definir los diagramas, matrices, etc. que incluyen las distintas metodologías.” (CONSULTADO EL 22-07-2013).

1.2.3.5.2. Estructura general de una Herramienta Case.

Según la página web <http://paulchasiluisa.galeon.com/> manifiesta que la **Estructura general de una Herramienta Case**: “Se basa en la siguiente terminología:

- CASE de alto nivel son aquellas herramientas que automatizan o apoyan las fases finales o superiores del ciclo de vida del desarrollo de sistemas como la planificación de sistemas, el análisis de sistemas y el diseño de sistemas.
- CASE de bajo nivel son aquellas herramientas que automatizan o apoyan las fases finales o inferiores del ciclo de vida como el diseño detallado de sistemas, la implantación de sistemas y el soporte de sistemas.
- CASE cruzado de ciclo de vida se aplica a aquellas herramientas que apoyan actividades que tienen lugar a lo largo de todo el ciclo de vida, se incluyen actividades como la gestión de proyectos y la estimación”. (CONSULTADO EL 22-07-2013).

A través de esta cita se puede determinar que con la ayuda de estas herramientas ya se dan paso algunas etapas del ciclo de vida del sistema, tales como el desarrollo e implantación las mismas que permiten el desarrollo del software.

1.2.3.5.3. Características de las Herramientas Case.

Según el criterio de varios autores sostiene que las **Características de las Herramientas Case** son:

- El primer aspecto a destacar es la aparición comercial de los Editores dirigidos por la Sintaxis, también llamados Editores Estructurales o Editores Sensibles, es decir, los sistemas de edición de programas que conocen las Sintaxis del lenguaje que editan y contienen esqueletos o plantillas de sus sentencias. Normalmente lo que se vende es un generador del editor a partir de la gramática BNF del lenguaje, junto con las gramáticas de los lenguajes más comunes (Fortran, Cobol, Pascal, C, etc.): ello permite al instante de editores para estos lenguajes de mayor uso y, además, generarnos un editor para cualquier otro lenguaje (de comandos, de query, etc.).
- El segundo aspecto es la creciente identificación entre lenguaje de programación, y entorno de desarrollo asociado. En el momento actual es muy frecuente (y cada vez lo es mas) que, bajo el nombre de un lenguaje, nos ofrezcan un conjunto integrado e interactivo de herramientas de desarrollo, incluyendo editor (dirigido por la sintaxis en algunos casos), depurador ("debugger") de alto nivel, y compilador. Es el caso, por ejemplo, del Quick C para Pc`s o del Light speed Pascal para Macintosh. Si hojeamos los catálogos de Software disponible por terceros para estaciones de trabajo tipo Micro Vax o sun, encontraremos, bajo el epígrafe "Lenguajes" o "compiladores" muchos entornos de este tipo que, siendo herramientas CASE, no se acogen a estás etiqueta, que, como

veremos en el párrafo siguiente, se suele asociar a los entornos de ayuda a diversos métodos de análisis y diseño.

- El tercer segmento o tipo de herramientas que ha irrumpido en el mercado, es el de los entornos diseñados como soporte de un método. Es este aspecto al que, para muchos, define lo que es una herramienta CASE, tal es la abundancia de este tipo de productos de este tipo que han aparecido recientemente, cosa, por otra parte, natural, si estamos de acuerdo con la afirmación de que la única forma de implantar el uso de un método es si existe una herramienta asociada. Una estrategia comercial mínima no se arriesgaría con métodos de poca difusión, así que se han centrado en los métodos de análisis y diseño de sistemas más populares (en los de USA), o de aprendizaje más sencillo.”

CAPITULO II

DESCRIPCION, ANALISIS E INTERPRETACION DE RESULTADOS

2.1. ENTORNO DE LA EMPRESA MERCADO LATACUNGA.

2.1.1 Antecedentes históricos.

La empresa se fundó en Abril, 2009 por iniciativa del Ing. José Napoleón Páez, como persona física, en la que inició sus operaciones el día 1 de junio de 2009 con el nombre de Mercado Latacunga el objeto de efectuar el comercio electrónico donde los usuarios compran y venden productos a través de Internet.

En Mercado Latacunga se promocionan sus productos a pequeñas y medianas empresas, productores, fabricantes, importadores, emprendedores, minoristas, mayoristas, individuos particulares, concesionarios, etc. Para vender en Mercado Latacunga el usuario deberá completar un formulario de venta con todos los datos detalles y fotos del producto que se ofrece.

Luego, ambos pueden calificarse para contarle al resto de la comunidad de usuarios cuál fue su experiencia en cuanto a su contraparte, el producto y la transacción.

Costos del servicio: existen diferentes servicios y costos para utilizar Mercado Latacunga como plataforma de comercialización de productos. Se pueden realizar desde publicaciones gratuitas hasta publicaciones con gran nivel de exposición,

incluso en la página principal del sitio. Todos los costos están detallados en las páginas de ayuda del sitio.

Atención al Cliente: en cada uno de los casos en los que opera Mercado Latacunga los usuarios cuentan con secciones de Ayuda desde donde pueden informarse en detalle acerca del funcionamiento de la plataforma (lo que es altamente recomendable realizar antes de comenzar a utilizar los servicios), pueden contactarse con el equipo de atención al cliente para realizar consultas.

Desarrollo de software a la medida. Ofrecemos a nuestros clientes un proceso completo de desarrollo de software, que contempla desde la investigación y el análisis de las necesidades reales del cliente, hasta el diseño, desarrollo e implementación de una solución informática que brinde a sus usuarios un incremento en la productividad y facilidad en el manejo de la información.

Perfil de nuestros profesionales

Áreas de desempeño

Mercado Latacunga está conformada por profesionales en el área de la tecnología de información. Actualmente cuenta con personal capacitado en las áreas de:

- *Ingeniería de Software:*
Análisis de problemas y diseño de soluciones:
Diseño avanzado con UML
Modelado de bases de datos
- *Gestión de proyectos de software:*
Aseguramiento de la calidad del software
Control de cambios en el software
Control de riesgos en proyectos de software

2.1.2 Funciones.

Programar, organizar, dirigir y coordinar las actividades concernientes, para lo cual busca acercamientos con los clientes, empresas de toda la provincia.

2.1.3. Objetivos de la empresa Mercado Latacunga.

Somos una de empresa publicidad y desarrollo de software a medida que ofrece a sus clientes desarrollos web corporativos utilizando las más modernas tecnologías existentes en el mercado. Como empresa de desarrollo Web contamos con el equipo técnico necesario para implementar de manera eficaz la lógica empresarial. Nuestro equipo de desarrollo Web y desarrollo de software a medida implementa soluciones completas orientadas a resultados y junto al cliente planteamos y definimos los objetivos a lograr. Aportamos un importante valor agregado a nuestros clientes con la implementación de las técnicas más modernas de desarrollo web. Promocionar los servicios de los diferentes locales comerciales de la provincia.

Realizar desarrollo de software cliente-servidor (client-server) con herramientas de última generación, que nos permiten generar aplicaciones ágiles, escalables y modulares.

2.1.4. Misión de la empresa Mercado Latacunga.

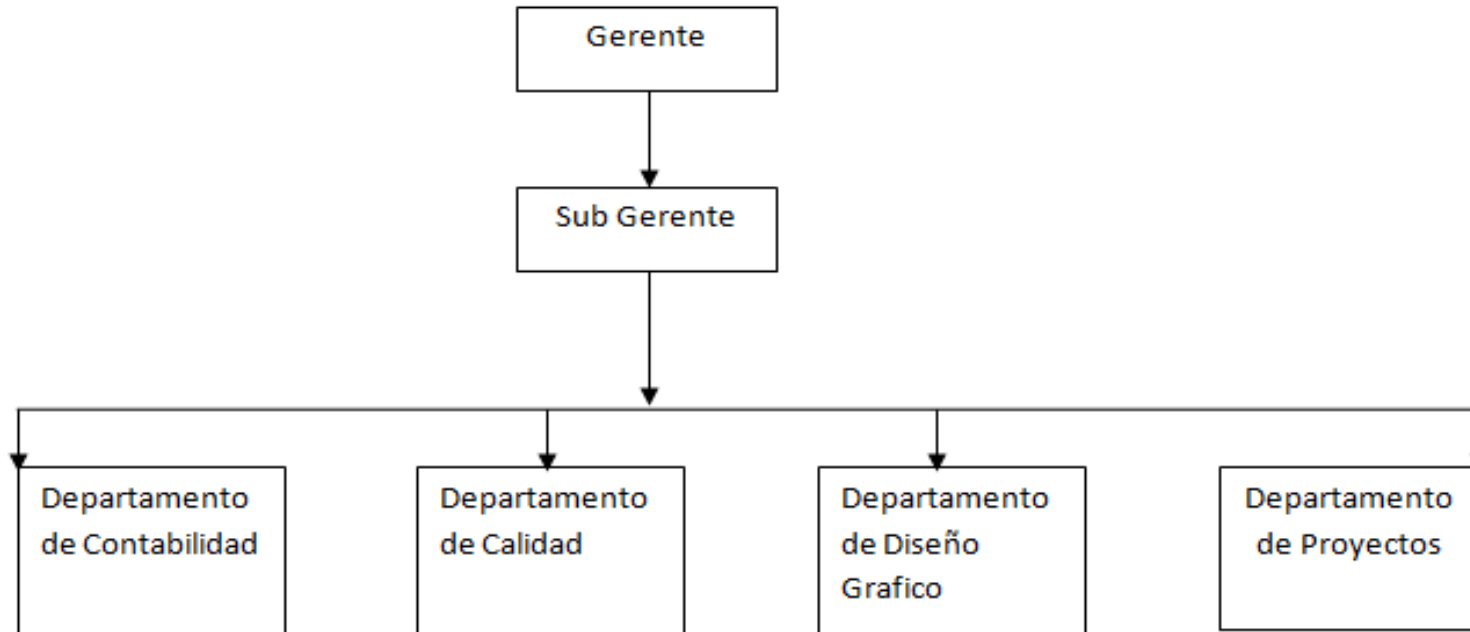
Brindar diferentes servicios informáticos, publicidad en internet a nivel nacional, entregando soluciones de alto nivel de calidad con un equipo humano altamente calificado utilizando tecnología de última generación.

2.1.5. Visión de la empresa Mercado Latacunga.

Llegar al mercado provincial, nacional a través de nuestra experiencia en el desarrollo de soluciones efectivas y de calidad.

2.1.6. Estructura organizacional.

GRAFICO N° 2.1. ORGANIGRAMA DE LA EMPRESA MERCADO LATACUNGA



Realizado por: Fabián Medina, Orlando Cepeda

2.1.7. Actividades que se desempeña dentro de la empresa Mercado Latacunga.

2.1.7 .1. Departamento de Gerencia.

El Departamento de Gerencia desarrolla las funciones de gestión de los recursos humanos, incluida la formación interna del personal, recursos económico-financieros, informáticos, logísticos y materiales, con el fin de prestar el apoyo necesario a los órganos y unidades de la Agencia para el cumplimiento de sus funciones. Asimismo es el responsable de la recopilación y difusión de conocimiento y documentación de las materias competencia de la Agencia, a través del centro de documentación e información de la Agencia. Es la oficina de contabilidad de la Agencia a todos los efectos legales, y el órgano encargado de establecer el sistema de contabilidad de gestión que permita efectuar el seguimiento del cumplimiento de los compromisos asumidos en el Contrato de gestión.

2.1.7.2. Departamento de Sub Gerencia.

Las funciones de la sub gerencia es participar en la planificación y supervisión del Desarrollo de los proyectos y sus respectivas prácticas. Coordinar y facilitar la ejecución de los servicios administrativos de apoyo a la gestión de proyectos de la empresa, priorizar las acciones a implementar, optimizando la sinergia de sus unidades de trabajo, evaluando la gestión y colocando las bases para la mejora continua, en el marco de un servicio de calidad y transparencia.

2.1.7.3. Departamento de Contabilidad.

La sección de Contabilidad General lleva, para los efectos de control, un registro sumario de la información obtenida y empleada por todas las demás secciones. Constituye, por tanto, la sección central de control a la que afluye ordenadamente

el resumen de las cifras registradas al detalle en las demás secciones. Esta sección tiene a su cargo los siguientes particulares:

El Mayor General, que contiene las cuentas de control para cada sección de la empresa.

El Diario General como medio de pase para hacer los asientos en las cuentas de Mayor General.

La preparación de comprobantes relativos a todos los cargos. Estos cargos pueden originarse en las transacciones de las distintas secciones así como en la misma sección de contabilidad general.

La preparación de los informes financieros que se establecen periódicamente.

La preparación de informes específicos que requieran la gerencia general o gerencia financiera.

La preparación de informes específicos que requieren las entidades gubernamentales. (SRI, SUPERINTENDENCIA DE COMPAÑIAS, MUNICIPIO. ETC.).

2.1.7.4. Departamento de Proyectos.

El departamento de proyectos entre las funciones principales, que debe cumplir son:

Elaborar y dar apoyo técnico a todas las instancias de la empresa en proyectos basados en las necesidades de sus clientes, a fin de conseguir financiamiento externo adicional al correspondiente por ley.

Coadyuvar a la formación de los empleados y en la elaboración de proyectos, basados en formatos específicos.

Formación de grupos interdisciplinarios para la elaboración de proyectos, en áreas específicas de acuerdo a la naturaleza de cada proyecto.

Coordinación, general con las autoridades de la empresa, departamentales y de otras dependencias, relacionadas con las acciones correspondientes a la generación de proyectos.

Crear un banco de proyectos de la empresa que solucionen necesidades, que no pueden lograrse por la vía regular, es decir requieren otro tipo de financiamiento.

2.1.7.5. Departamento de Diseño Gráfico.

Depende en forma directa de la Dirección General, en que se inserte el Departamento a su cargo, y excepcionalmente de la estructura de conducción.

Organiza las tareas a nivel de Departamento y ejerce control directo sobre el/los Jefe/s de División. Ejecuta tareas de carácter profesional, de nivel superior. Brinda asesoramiento a los niveles de jerarquía superior. Colabora en la preparación y control de programas y proyectos. Tiene a su cargo la supervisión y control del desempeño del personal del tramo intermedio.

Sus funciones son:

- Trabajar en la interpretación, ordenamiento y presentación de los mensajes visuales, que forma parte de la planificación y estructuración de las comunicaciones, es responsable del diseño Web del sitio oficial de la empresa.
- Difundir las actividades a través de materiales gráficos e impresos así como una identidad institucional a través del diseño.
- Coordinar las publicaciones impresas y electrónicas de la Dirección.

- Brindar una visión integradora del proceso desde el concepto y la creación hasta las herramientas tecnológicas necesarias para su materialización.
- Captar la necesidad de las distintas áreas de la empresa, para diseñar estrategias de comunicación visual que den respuestas a las demandas planteadas.
- Colaborar con las diversas dependencias internas de la empresa para la confección de material gráfico y de comunicación.
- Intervenir en el diseño de campañas de difusión y de promoción de la empresa.
- Orientar y evaluar las acciones conducentes para potenciar el necesario perfeccionamiento de los agentes del área.

2.1.7.6. Departamento de Calidad.

El Departamento de Calidad de los Servicios es el encargado de desarrollar las funciones de elaboración del Informe al gerente sobre el seguimiento de las actuaciones comprendidas en el plan de gestión de calidad de los proyectos de desarrollo de software para promover y desarrollar actividades de análisis, y asesoramiento tendentes a la mejora de la calidad en la gestión pública y privada, por el que se establece el marco general para la mejora de la calidad en la Administración General del Estado, en lo referente al plan de gestión de calidad del servicio.

2.2 TIPO DE INVESTIGACIÓN.

2.2.1. Investigación de campo.

La investigación de campo se presenta mediante la manipulación de una variable externa no comprobada, en condiciones rigurosamente controladas, con el fin de describir de qué modo o porque causas se produce una situación o acontecimiento particular.

Podríamos definirla diciendo que es el proceso que, utilizando el método científico, permite obtener nuevos conocimientos en el campo de la realidad social. (Investigación pura), o bien estudiar una situación para diagnosticar necesidades y problemas a efectos de aplicar los conocimientos con fines prácticos (investigación aplicada).

Este tipo de investigación es también conocida como investigación in situ ya que se realiza en el propio sitio donde se encuentra el objeto de estudio. Ello permite el conocimiento más a fondo del investigador, puede manejar los datos con más seguridad y podrá soportarse en diseños exploratorios, descriptivos y experimentales, creando una situación de control en la cual manipula sobre una o más variables dependientes (efectos).

2.2.2. Investigación Bibliográfica

La investigación bibliográfica es aquella etapa de la investigación científica donde se explora qué se ha escrito en la comunidad científica sobre un determinado tema o problema. ¿Qué hay que consultar, y cómo hacerlo?

La bibliografía está constituida por un método científico y por el objetivo de la ciencia, erigiéndose entre el conocimiento vulgar y el científico.

Utiliza un método histórico que consta de apoyo teórico y metodología científica

2.2.3. Tipo de investigación experimental

La investigación experimental está integrada por un conjunto de actividades metódicas y técnicas que se realizan para recabar la información y datos necesarios sobre el tema a investigar y el problema a resolver, se presenta mediante la manipulación de una variable experimental no comprobada, en condiciones rigurosamente controladas, con el fin de describir de qué modo o por qué causa se produce una situación o acontecimiento particular.

Su diferencia con los otros tipos de investigación es que el objetivo de estudio y su tratamiento dependen completamente del investigador, de las decisiones que tome para manejar su experimento.

El experimento es una situación provocada por el investigador para introducir determinadas variables de estudio manipuladas por él para controlar el aumento o disminución de esas variables y su efecto en las conductas observadas.

Esto da lugar al desarrollo de investigaciones conocidas como cuantitativas, las cuales se apoyan en las pruebas estadísticas tradicionales. Pero especialmente en el ámbito de las ciencias sociales se observan fenómenos complejos y que no pueden ser alcanzados ser observados a menos que se realicen esfuerzos con alto grado de subjetividad y orientados hacia las cualidades más que a la cantidad. Así se originan diversas metodologías para la recolección y análisis de datos (no necesariamente numéricos) con los cuales se realiza la investigación conocida con el nombre de Cualitativa.

2.3 ANÁLISIS DE LOS RESULTADOS DE LA ENCUESTA APLICADA AL PERSONAL DE LA EMPRESA MERCADO LATACUNGA.

La investigación propuesta se realizará a los administradores y al personal de los diferentes departamentos de la empresa, estarán enfocadas a 6 Administradores de la empresa y 44 empleados en su totalidad correspondiente a cada departamento de la empresa Mercado Latacunga, obteniendo así un total de 50 personas involucradas para la recolección de datos.

En base a las preguntas planteadas se ha podido rescatar los siguientes resultados que se constituyen en fuente confiable para poder implementar cambios en el método de presentación del estudio comparativo entre las herramientas libres de análisis y desarrollo starUML y Fujaba para establecer la usabilidad y la aplicabilidad de la ingeniería inversa dentro del proceso de desarrollo de software.

1.- ¿Ha utilizado usted el software libre?

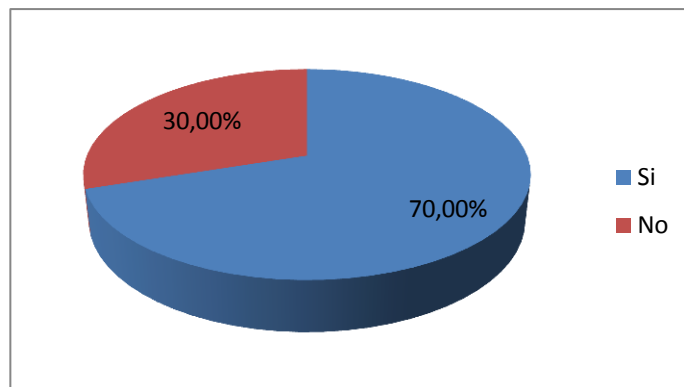
TABLA N° 2.1.- VALORACIÓN DE LA UTILIZACION DEL SOFTWARE LIBRE.

Alternativa	Valor	Porcentaje
Si	35	70,00%
No	15	30,00%
Total	50	100%

Fuente: Encuestados

Realizado por: Fabián Medina, Orlando Cepeda

GRÁFICO N°2.2.- VALORACIÓN DE LA UTILIZACION DEL SOFTWARE LIBRE.



Fuente: Encuesta

Realizado por: Fabián Medina, Orlando Cepeda

Después de haber obtenido los datos de la población, podemos darnos cuenta que un 70% de las personas encuestadas consideran que SI han utilizado el software libre, mientras que un 30% de los encuestados manifestaron que NO lo han utilizado.

2.- ¿Conoce usted los beneficios del software libre?

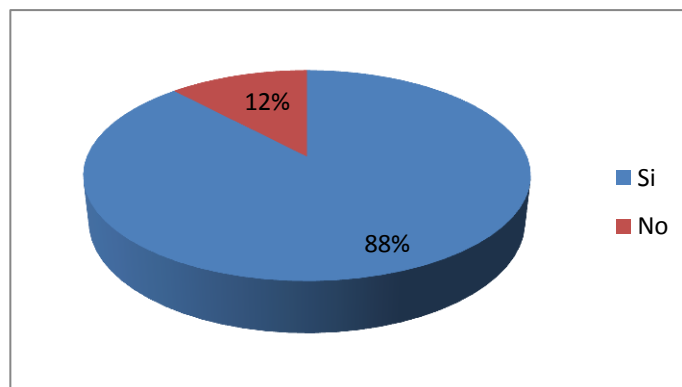
TABLA N° 2.2. BENEFICIOS DEL SOFTWARE LIBRE.

Alternativa	Valor	Porcentaje
Si	44	88,00,%
No	6	12,00%
Total	50	100%

Fuente: Encuesta

Realizado por: Fabián Medina, Orlando Cepeda

GRÁFICO N°2.3. CONOCEN LOS BENEFICIOS DEL SOFTWARE LIBRE.



Fuente: Encuesta

Realizado por: Fabián Medina, Orlando Cepeda

De las personas encuestadas opinan un 88% dicen SI conocer los veneficios que presta el software libre para el desarrollo de sistemas, mientras que un 12% No dice conocer los beneficios de esta.

3. ¿Utiliza usted herramientas libres para el desarrollo de software en la empresa?

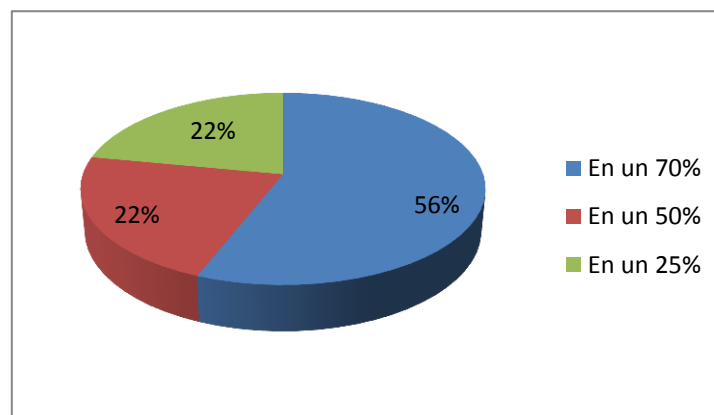
TABLA N°2.3.- HERRAMIENTAS LIBRES PARA EL DESARROLLO DE SOFTWARE.

En un.	Valor	Porcentaje
70%	28	56,00%
50%	11	22,00%
25%	11	22,00%
Total	50	100%

Fuente: Encuesta

Realizado por: Fabián Medina, Orlando Cepeda

GRÁFICO N°2.4. USO DE HERRAMIENTAS LIBRES PARA EL DESARROLLO DE SOFTWARE.



Fuente: Encuesta

Realizado por: Fabián Medina, Orlando Cepeda

En la pregunta aplicada, es evidente que un grupo de 28 dice que se utiliza el software libre en un 70%, mientras que el otro grupo de 11 encuestados considera que se utiliza en un 50% y otro grupo de 11 encuestados mantienen que se utiliza en un 25% el software libre para el desarrollo de software en la empresa. Hay que aclarar que el porcentaje de 25% está enfocado al desarrollo de software con herramientas de licencia o propietarias.

4. ¿Maneja usted herramientas de análisis para el desarrollo de software en su empresa?

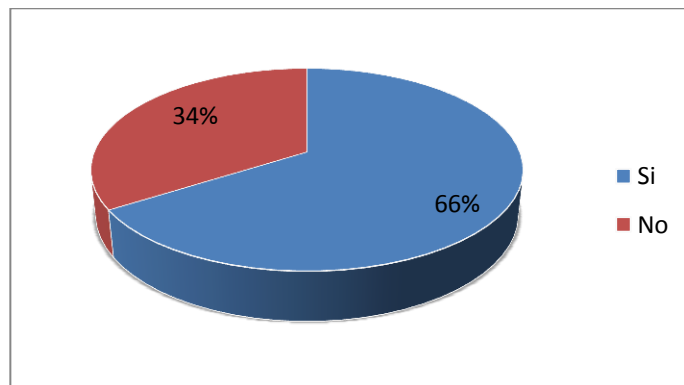
TABLA N°2.4. HERRAMIENTAS DE ANÁLISIS PARA EL DESARROLLO DE SOFTWARE.

Alternativa	Valor	Porcentaje
Si	33	66,00%
No	17	34,00%
Total	50	100%

Fuente: Encuesta

Realizado por: Fabián Medina, Orlando Cepeda

GRÁFICO N°2.5. USO DE HERRAMIENTAS DE ANÁLISIS PARA EL DESARROLLO DE SOFTWARE.



Fuente: Encuesta

Realizado por: Fabián Medina, Orlando Cepeda

Después de haber obtenido los datos de la encuesta realizada, podemos evidenciar que el 66% de los involucrados manejan herramientas de análisis para el desarrollo de software, mientras que un 34% manifiesta no manejan este tipo de herramientas.

5. ¿Qué metodología (as) utiliza usted para el desarrollo de software?

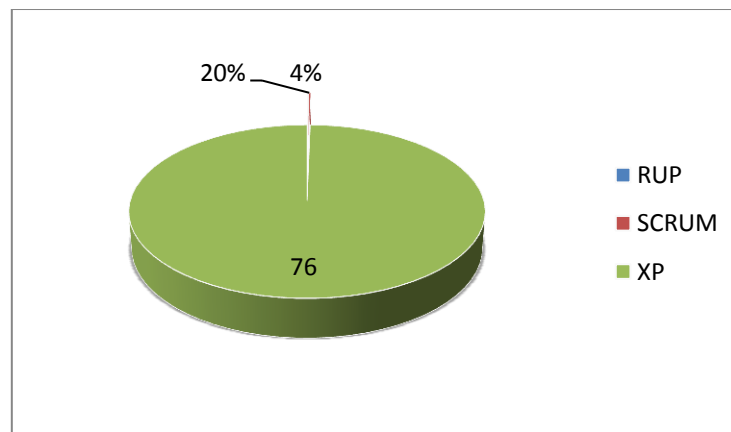
TABLA N°2.5. METODOLOGIA(S) PARA EL DESARROLLO DE SOFTWARE.

Alternativa	Valor	Porcentaje
RUP	2	4,00%
SCRUM	10	20,00%
XP	38	76,00%
Total	50	100%

Fuente: Encuesta

Realizado por: Fabián Medina, Orlando Cepeda

GRÁFICO N° 2.6. METODOLOGIAS UTILIZADAS PARA EL DESARROLLO DE SOFTWARE.



Fuente: Encuesta

Realizado por: Fabián Medina, Orlando Cepeda

Al respecto, la mayoría de encuestados esto es el 76%, manifiestan que la metodología que ellos utilizan es la metodología XP, mientras que un 20% utilizan a SCRUM como metodología, y un 4% utilizan la metodología RUP para proyectos a largo plazo.

6. ¿Conoce usted herramientas libres de análisis a más de las ya conocidas como Rational Rose, Power Designer?

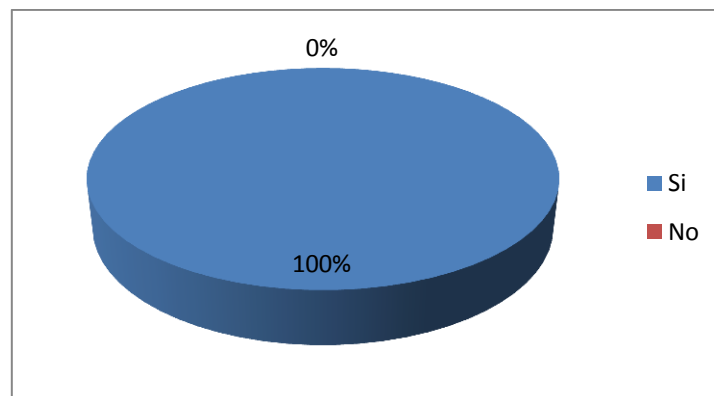
TABLA N°2.6. CONOCE HERRAMIENTAS LIBRES DE ANALISIS.

Alternativa	Valor	Porcentaje
Si	50	100,00%
No	0	0,00%
Total	50	100%

Fuente: Encuesta

Realizado por: Fabián Medina, Orlando Cepeda

GRÁFICO N°2.7. CONOCE HERRAMIENTAS LIBRES DE ANALISIS.



Fuente: Encuesta

Realizado por: Fabián Medina, Orlando Cepeda

De los resultados obtenidos sobre esta interrogante se pudo evidenciar que un contundente 100% de los encuestados consideran que SI conocen herramientas libres de análisis para el desarrollo de software y con un 0% correspondiente al NO.

7. ¿Ha escuchado usted sobre las herramientas de análisis starUML y Fujaba?

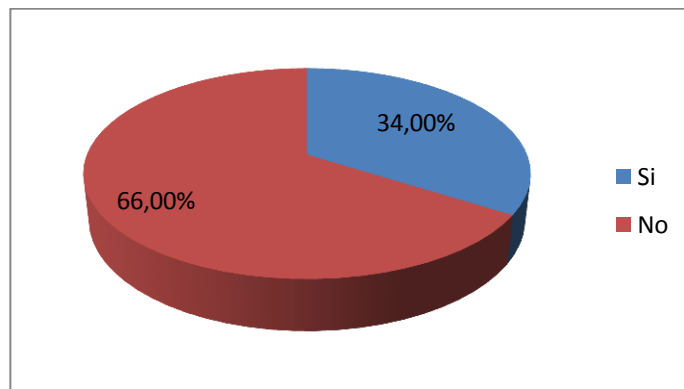
TABLA N°2.7. HERRAMIENTAS DE ANÁLISIS STARUML Y FUJABA.

Alternativa	Valor	Porcentaje
Si	17	34,00%
No	33	66,00%
Total	50	100%

Fuente: Encuesta

Realizado por: Fabián Medina, Orlando Cepeda

GRÁFICO N°2.8. HERRAMIENTAS DE ANÁLISIS STARUML Y FUJABA.



Fuente: Encuesta

Realizado por: Fabián Medina, Orlando Cepeda

Después de haber obtenido los datos de la encuesta realizada, podemos evidenciar que el 66% de los involucrados manifiestan que NO han escuchado de las herramientas de análisis starUML y Fujaba. Mientras que un 34% SI ha escuchado de las herramientas.

8. ¿Al conocer las múltiples ventajas de las herramientas libres de análisis, que las tradicionales, usted implementaría en el desarrollo de software para su empresa?

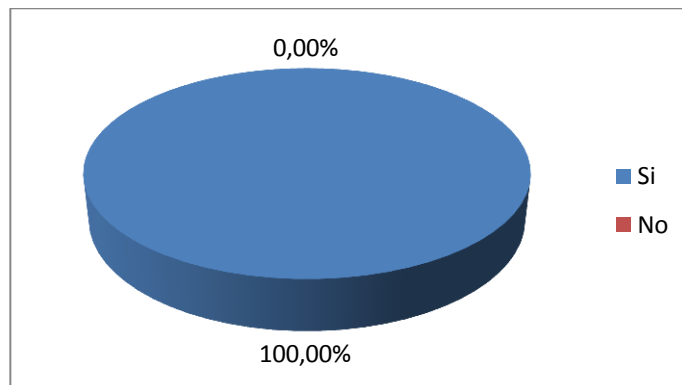
TABLA N°2.8. USTED IMPLEMENTARÍA EN EL DESARROLLO DE SOFTWARE.

Alternativa	Valor	Porcentaje
Si	50	100,00%
No	0	00,00%
Total	50	100%

Fuente: Encuesta

Realizado por: Fabián Medina, Orlando Cepeda

GRÁFICO N°2.9. USTED IMPLEMENTARÍA EN EL DESARROLLO DE SOFTWARE.



Fuente: Encuesta

Realizado por: Fabián Medina, Orlando Cepeda

En la pregunta realizada los encuestados manifiestan que al conocer de las múltiples ventajas de las herramientas libres de análisis SI lo implementarían para el desarrollo de software de la empresa y con un 0% en contra.

CAPITULO III:

ESTUDIO COMPARATIVO ENTRE LAS HERRAMIENTAS LIBRES DE ANALISIS Y DESARROLLO STARUML Y FUJABA PARA ESTABLECER LA USABILIDAD Y LA APLICABILIDAD DE LA INGENIERIA INVERSA DENTRO DEL PROCESO DE DESARROLLO DE SOFTWARE EN LA EMPRESA MERCADO LATACUNGA

3.1. PROPUESTA

El presente trabajo de investigación, muestra un estudio comparativo de las herramientas empleadas para el diseño de procesos en el desarrollo de análisis de software, teniendo en cuenta sus características y ventajas principales, que ayudarán a establecer cuál de éstas tiene un mejor desempeño y brinda mayores prestaciones, al momento de utilizarlas.

Para esto se ha desarrollado un caso práctico utilizando las dos herramientas y realizando el mismo ejercicio, que aportará el medio de selección apropiado de acuerdo a las características relevantes que determinan el estudio.

3.2 OBJETIVOS

General:

Realizar un estudio comparativo entre las herramientas libres de análisis y desarrollo, el mismo que se desarrollara con la ayuda de los programas

StarUML y Fujaba para establecer la usabilidad y la aplicabilidad de la ingeniería inversa dentro del proceso de desarrollo de software.

Específicos:

- Realizar un estudio comparativo entre las herramientas libres (starUML, Fujaba) y propietarias como (Rational rose, Power Designer).
- Emitir un criterio favorable para las herramientas libres StarUML y Fujaba en la reducción en el costo de desarrollo de los sistemas informáticos.

3.3 JUSTIFICACIÓN DE LA PROPUESTA

Teniendo en cuenta que la mayoría de aplicaciones modernas están siendo desarrolladas bajo el modelo orientado a objetos los mismos que se basa en conceptos como clases, métodos, atributos, herencia, composición, polimorfismo, etc. Es por eso que para nuestro estudio hemos utilizado las herramientas CASE Orientada a Objetos ya que proveen soporte para el modelado y metodologías bajo el paradigma OO y muchas generan el código base de las aplicaciones OO. Las últimas versiones de las herramientas CASE OO ya generan para el lenguaje como Java. Y muchas otras CASE OO ya están soportando bases de datos relacionales mediante la realización de un diseño y modelado incluyendo la generación de esquemas o bien hasta la ingeniería inversa de las tablas y otros elementos.

Se ha considerado que como la mayoría de las herramientas CASE hacen uso del UML y sirven para crear diagramas que representan el modelo de los objetos usando la notación del UML.

Para el estudio comparativo se cuenta con suficiente referencia humana, bibliográfica, entre otros, el mismo que facilitará el desarrollo del proyecto.

Por medio del estudio comparativo se espera mejorar los procesos aumentando la productividad en el desarrollo de software reduciendo el coste de las mismas.

En las últimas décadas se ha trabajado en el área de desarrollo de sistemas para encontrar técnicas que permitan incrementar la productividad y el control de calidad en cualquier proceso de elaboración de software, y hoy en día la tecnología CASE (Computer Aided Software Engineering ó Ingeniería de Software Asistida por Computadora) reemplaza al papel y al lápiz por el ordenador para transformar la actividad de desarrollar software en un proceso automatizado.

Es por esta razón que fomentando la informatización de la informática es decir la automatización del desarrollo del software, se contribuirá así a elevar la productividad y la calidad de en el desarrollo de los sistemas de información de forma análoga a lo que suponen las técnicas CAD/CAM en el área de fabricación en la empresa “MERCADO LATACUNGA”.

Sin lugar a dudas las herramientas CASE como StarUML y Fujaba han venido a revolucionar la forma de automatizar los aspectos clave en el desarrollo de los sistemas de información, debido a la gran plataforma de seguridad que ofrecen a los sistemas que las usan y es que éstas, brindan toda una gama de componentes que incluyen todas o la mayoría de los requisitos necesarios para el desarrollo de los sistemas, han sido creadas con una gran exactitud en torno a las necesidades de los desarrolladores de sistemas para la automatización de procesos incluyendo el análisis, diseño e implantación.

El estudio comparativo entre estas herramientas libres de análisis y desarrollo StarUML y Fujaba permitirá establecer a la empresa “Mercado Latacunga” la usabilidad y la aplicabilidad en el proceso de desarrollo de software permitiendo sacar al mercado un producto de calidad por lo cual la investigación realizada está enfocada al uso de las herramientas libres permitiendo mediante la comparación

analizar la rentabilidad de estas herramientas con respecto a las pagadas o con licencia.

3.4 FACTIBILIDAD DE APLICAR LA PROPUESTA

Este tema de investigación, presenta un estudio comparativo de algunas de las herramientas empleadas para el modelado de procesos, que se logró, luego de una revisión teórica, y la aplicación de las mismas de forma práctica, teniendo en cuenta sus características principales, funcionalidad y estandarización.

La comparación se realizó debido a la importancia que representa la adecuada selección de las herramientas starUML y Fujaba, para obtener el éxito en cualquier tipo de proyecto de desarrollo de software, y al aporte que representa dentro de cada uno de los proyectos de investigación para desarrollo de software.

A demás se pudo evidenciar que el uso de las herramientas libres estudiadas tienen igual o mayor productividad que las herramientas que no son libres, y con esto, se pudo evidenciar que con la utilización de las mismas benefició en la cuestión de costo dentro del proceso de desarrollo de software.

3.5 DESARROLLO DE LA PROPUESTA

Para la aplicación de la propuesta se utilizaron algunas herramientas que se detallan a continuación.

3.5.1 Descripción de las herramientas de análisis

3.5.1.1. Open Source.

Los programas que se distribuyen bajo la licencia Open Source(código abierto) son aquellos que ponen a disposición de la gente el código fuente del programa para que este pueda ser estudiado, modificado y mejorado.

Un programa Open Source posee las siguientes características y ventajas:

Flexibilidad: al tener disponible su código fuente, los desarrolladores pueden aprender y modificar los programas a su gusto para realizar tareas específicas. Además, se genera un flujo constante de ideas que mejora la calidad de los programas.

Fiabilidad y seguridad: al contar con unos cuantos programadores mirando el mismo trabajo simultáneamente, los errores se detectan y corrigen con anterioridad, por lo que el producto resultante es más confiable y efectivo que el comercial.

Rapidez de desarrollo: las actualizaciones y ajustes se llevan a cabo por medio de una comunicación constante vía internet. Debido a la gran cantidad de herramientas y librerías disponibles, se requieren menores tiempos de desarrollo.

Relación con el usuario: el programador puede definir mejor las necesidades reales de su cliente, y en consecuencia crear un producto específico para él.

Libre: es de libre distribución, las personas pueden regalarlo, venderlo o prestarlo.

Combate de manera muy efectiva la piratería de software.

3.5.1.2. Uml

Debido a que muchos analistas de sistemas no utilizan un método ordenado y sistemático para el análisis y diseño de sus proyectos, la elaboración de este documento se justifica ante la necesidad de conocer esta popular y globalmente aceptada herramienta conocida como UML, que permite desarrollar sistemas metódicamente y paso a paso, documentando gráficamente mediante diagramas lo

que se podría conocer como "*el plano del sistema*", si se hiciera la analogía con los planos a la hora de construir un edificio. A continuación se presentan los diagramas utilizados en UML para el modelado de sistemas:

TABLA N° 3.1. DIAGRAMAS

Diagrama de caso de uso	Determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar.
Diagrama de estados	Son una técnica conocida para explicar el comportamiento de un sistema. Detallan todos los estados posibles en los que puede entrar un objeto particular y la manera en que cambia el estado del objeto, como resultado de los sucesos que llegan a él.
Diagrama de secuencias	Muestra la interacción gráfica de un conjunto de objetos en una aplicación a través del tiempo.
Diagrama de colaboraciones	Es una forma de representar interacción entre objetos
Diagrama de actividades	Se emplea para presentar un panorama simplificada de lo que ocurre durante una operación o proceso, como un flujo de trabajo a través de una serie de actividades.
Diagrama de componentes	Como su nombre lo señala, un esquema o diagrama que muestra las interacciones y relaciones de los componentes de un modelo de manera gráfica.
Diagrama de distribución	Es una forma gráfica de los pasos que se siguen en toda una secuencia de actividades, dentro de un proceso o un procedimiento, reconociéndolos mediante símbolos de acuerdo con su naturaleza.
Diagrama de clases	Es un tipo de diagrama fijo que describe la estructura de un sistema manifestando sus clases, atributos y las relaciones entre ellos.
Diagrama de objetos	Estos diagramas en el proceso de análisis y diseño de sistemas son aquellos que muestra un conjunto de objetos y sus relaciones en un momento concreto

Realizado por: Fabián Medina, Orlando Cepeda

3.5.1.3. StarUML

Al ser StarUML parte de nuestro estudio comparativo se pudo comprobar que es un software muy amigable, para trabajar tanto en UML, como en MDA, de forma rápida y flexible. También se puede decir que es una excelente alternativa a otra herramienta comercial como lo es Rational Rose, sin que esta deje de ser compatible, ya que importa perfectamente los archivos de Rational Rose.

3.5.1.4. Java

Es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

3.5.1.5. Fujaba

Fujaba es un Open Source UML CASE proyecto herramienta comenzó en el grupo de ingeniería de software de la Universidad de Paderborn en 1997. En 2002, Fujaba ha sido rediseñado y se convirtió en el Fujaba Tool Suite, con los desarrolladores de plug-in de la arquitectura, lo que permite añadir funcionalidad fácilmente manteniendo un control total sobre sus contribuciones.

Mediante el uso de diagramas UML y patrones de diseño de software permite también la generación de código Java a partir de los diagramas y la aplicación de ingeniería inversa al código para crear los diagramas.

Fujaba tenía como objetivo de apoyar el software de avance y retroceso de la ingeniería. **Anexo 2**

3.5.1.6. Power Designer

Es la herramienta para el análisis, diseño inteligente y construcción sólida de una base de datos y un desarrollo orientado a modelos de datos a nivel físico y conceptual, que da a los desarrolladores Cliente/Servidor la más firme base para aplicaciones de alto rendimiento. Ofrece un acercamiento de diseño para optimizar las estructuras de las bases de datos.

3.5.2 Análisis Comparativo entre las herramientas starUML y Fujaba

La Planificación de un Proyecto de Software hoy en día es una de las áreas clave dentro del proceso de desarrollo de software. Por lo tanto, una tarea de tal relevancia requiere especial atención. Para esto se ha propuesto un análisis de las dos herramientas CASE para el desarrollo de software ya que la utilización de estas herramientas son una actividad clave para el éxito de un proyecto de software.

3.5.2.1 Recopilación - clasificación de herramientas de modelado de procesos

Para la clasificación de las herramientas de modelado se tomaron en cuenta sus ventajas, potencialidades que ofrecen cada una de ellas para el proceso de diseño de software, siendo esta actividad de gran importancia. **Anexo 1**

3.5.2.2. Asignación de pesos.

De acuerdo a las características e importancia se procedió asignar pesos para realizar el estudio comparativo el mismo que se tomaron en cuenta los siguientes parámetros.

Para poder asignar pesos o valores a cada herramienta hemos realizado una ponderación de un valor relativo a los distintos niveles de logro de las metas, para que con esto se pueda tener un valor certero que me permita calificar su desempeño.

Para el grupo investigador se le vio conveniente asignar un valor de 25% para aquel que cumpla con un desempeño sobresaliente, es decir aquellos que superan las expectativas de la meta.

A el desempeño satisfactorio representa el valor que implica el cumplimiento esperado de la meta y se le asignado un peso ponderado de 20%.

Para el desempeño aceptable representa el valor del 15% de aprobatorio que nos indica que el cumplimiento está por debajo de lo esperado en la meta, pero aun así su desempeño sigue siendo todavía aceptable.

El desempeño no aceptable representa el valor de un 10% que implica un cumplimiento mínimo.

En la suma de todos los desempeños nos dará un resultado del 100% **Tabla N° 3.2.**

Parámetros para el estudio comparativo

- ✓ Software libre
- ✓ Multiplataforma
- ✓ Uml
- ✓ Ingeniería inversa
- ✓ Genera código
- ✓ Estabilidad herramienta
- ✓ Soporte

Acorde a las características seleccionadas, se asigna un valor de importancia a cada una de ellas, a fin de facilitar la elección de la herramienta a utilizar. Las tablas con las consideraciones para la evaluación de las herramientas se muestran en el siguiente **Anexo 1.**

3.5.2.3. Estudio Comparativo.

Para realizar el estudio comparativo, se llevaron a cabo las actividades planteadas en las cuales permitieron la obtención de los resultados que se muestran a continuación:

3.5.2.4. Recopilación - clasificación de herramientas modelado.

De acuerdo con la investigación realizada, se encontraron algunas de las herramientas de modelado de procesos disponibles en el mercado, clasificadas acorde al tipo de software en el que se encuentran enmarcadas en base a una matriz de análisis. **Tabla N° 3.3**

3.5.2.5. Herramientas Descripción del modelado de proceso.

Acorde al listado recopilado, se realizó una breve descripción de cada una de las herramientas, con el fin de tener un panorama amplio de algunas de sus características y utilidades **Anexo 1 y Anexo 2.**

3.5.2.6. Evaluación de las herramientas.

Para realizar la evaluación de las herramientas se elaboró un cuadro comparativo, que permitió indicar el peso correspondiente según la característica o características presentes en cada caso. Una vez asignados los valores, se totalizaron los resultados.

TABLA N° 3.2. ASIGNACIÓN DE PESOS

PARAMETROS	PESO
Software libre	10
Multiplataforma	10
UML	20
Ingeniería Inversa	25
Generador de Código	15

Estabilidad	10
Soporte	10
	TOTAL 100%

Realizado por: Fabián Medina, Orlando Cepeda

TABLA N° 3.3. MATRIZ COMPARATIVA DE LAS HERRAMIENTAS

Herramientas	Software libre	Multiplataforma	Uml	Ingeniería inversa	Genera código	Estabilidad herramienta	Soporte
FUJABA	X	X	X	X		X	
STARUML	X	X	X	X	X	X	X
POWER DESIGNER				X		X	X
RATIONAL ROSE			X	X		X	X

Realizado por: Fabián Medina, Orlando Cepeda

3.5.2.7 Tipo de Software.

Hace referencia al tipo de software de acuerdo a su licenciamiento ya sean estas libres o pagadas para su respectiva comparación.

A cada característica elegida, se le asignaron pesos los cuales nos permitirán tener una mejor perspectiva al momento de la comparación.

3.5.2.8. Plataforma.

Teniendo en cuenta que la empresa Mercado Latacunga maneja diferentes plataformas para el desarrollo y puesta en marcha de sus productos, se considera

la posibilidad de que la propuesta desarrollada pueda ser empleada en múltiples plataformas.

3.5.2.9. Generación de código.

Durante el proceso mediante un caso práctico se realizó un pequeño diagrama a partir del cual se vio si la herramienta tenía en una de sus características el generar código a partir de un diagrama.

3.5.2.10. Soporte de ingeniería inversa.

De igual manera para este punto se tomó en cuenta el caso práctico el cual nos va a demostrar la utilidad y la ventaja que tiene la herramienta al momento de realizar una ingeniería inversa a través de un diagrama.

3.5.2.11. Soporte a UML.

Teniendo en cuenta que UML, es un lenguaje estándar para el modelado de diagramas, respaldado por el OMG, se requiere que la herramienta seleccionada tenga soporte a este tipo de diagramas para lo cual nos basamos en la investigación realizada mediante fuentes bibliográficas y web que nos ayudaron para la selección de las herramientas que se utilizaran para el estudio comparativo.

3.5.2.12. Estabilidad de la herramienta.

Para esto se necesita que la herramienta sea estable para no tener ningún inconveniente al momento de la realización de diseño para el desarrollo del software.

3.5.2.13 Diagramas UML.

En cuanto al soporte que brindan en la generación de diagramas de casos de uso, clases, estados, actividad, colaboración, secuencia, componentes y despliegue, son

de gran importancia para su elección.

3.5.2.14. Selección de la herramienta.

De acuerdo al cuadro de evaluación generado anteriormente, se eligieron a cuatro de las herramientas, starUML, Fujaba PowerDesigner, Rational Rose.

Una vez seleccionadas las herramientas, se procedió a analizar cada una con el fin de determinar cuál de éstas brinda mejores prestaciones para el desarrollo de los proyectos dentro de la empresa mediante una matriz comparativa, tomando también en cuenta la generación de diagramas de casos de uso.

Mediante la matriz de resultados (**Tabla 3.5**) se vio que la herramienta sobresaliente fue StarUML la misma que mediante un análisis y el caso práctico se pudo observar las potencialidades que ofrece esta herramienta.

StarUML. La primera y más esencial es la inclusión de todos los diagramas UML:

- Diagrama de casos de uso.
- Diagrama de clase.
- Diagrama de secuencia
- Diagrama de colaboración.
- Diagrama de estados
- Diagrama de actividad.
- Diagrama de componentes
- Diagrama de despliegue.
- Diagrama de composición estructural.

Usabilidad: el programa es muy utilizado para los desarrollos de proyectos de software.

Generar documentación sobre los diagramas creados en Word, Excel y PowerPoint.

Generar código C#, Java y C++ a partir de los diagramas y viceversa (ingeniería inversa).

**TABLA N° 3.4. MATRIZ COMPARATIVA DE LAS HERRAMIENTAS
CON PESOS**

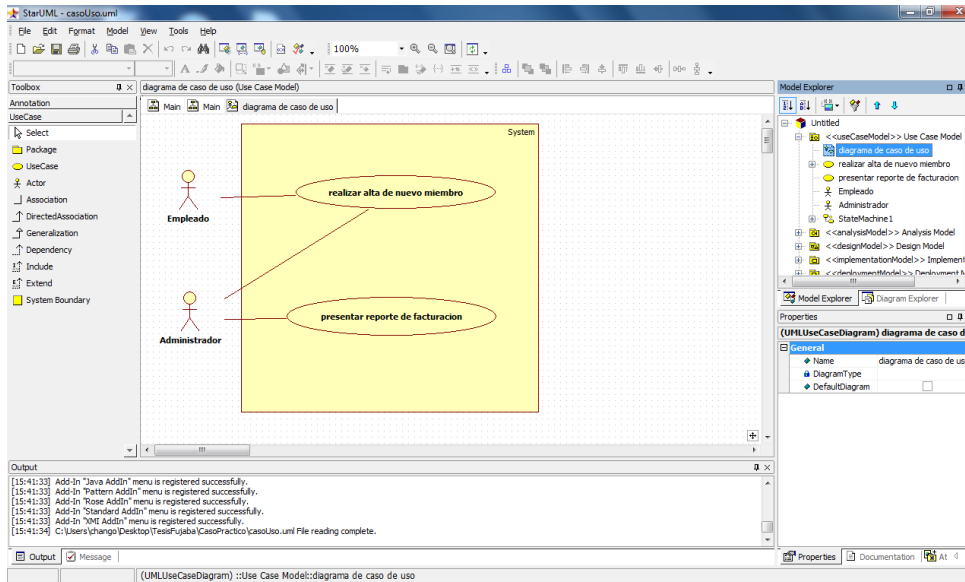
Herramientas	Software libre	Multiplataforma	Uml	Ingeniería inversa	Genera código	Estabilidad herramienta	SopORTE	Total
FUJABA	10%	10%	20%	25%		10%		75%
STARUML	10%	10%	20%	25%	15%	10%	10%	100%
POWER DESIGNER				25%		10%	10%	45%
RATIONAL ROSE			20%	25%		10%	10%	70%

Realizado por: Fabián Medina, Orlando Cepeda

3.5.3. Caso Práctico.

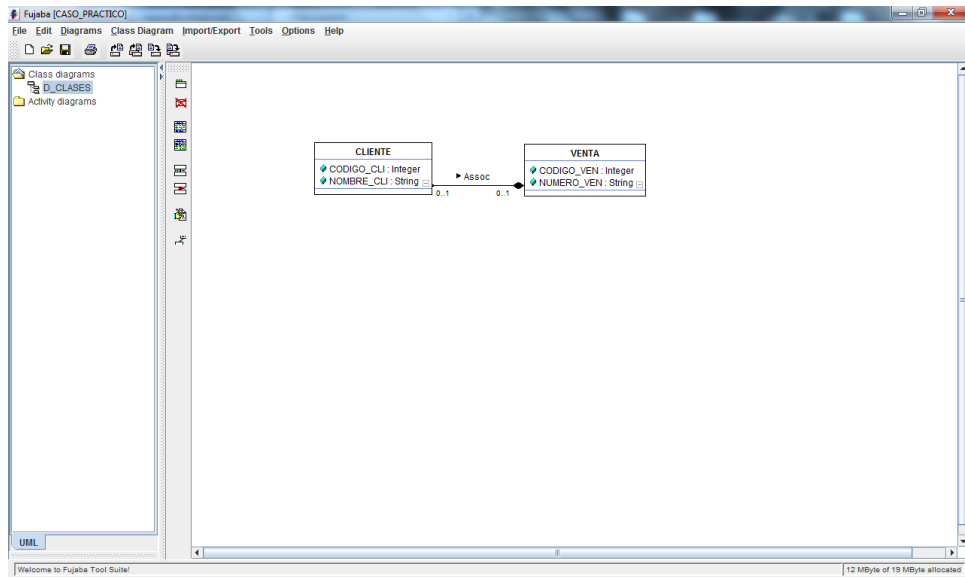
Para la presente investigación se realizó un caso práctico de las dos herramientas que obtuvieron el mejor porcentaje dentro de nuestro análisis que se realizó mediante la evaluación a través de una matriz dando los siguientes resultados **Anexo3 y Anexo 4:**

GRAFICO N° 3.1. STARUML



Realizado por: Fabián Medina, Orlando Cepeda

GRAFICO N° 3.2. FUJABA



Realizado por: Fabián Medina, Orlando Cepeda

3.5.4. Análisis del Caso Práctico

Una vez que se realizó el diseño de una aplicación en las dos herramientas seleccionadas se pudo observar efectivamente que la herramienta con mayor bondad es StarUML ya que nos permitió mayor usabilidad y mejor proceso de

análisis de los diferentes diagramas UML exportación de código e ingeniería inversa, obteniendo una estructura “base” tanto en el modelo como en código.

3.6. ACTORES Y RESPONSABILIDADES

Asesor (Manager)

Ing. Franklin Montaluisa.

El líder del equipo - toma las decisiones importantes

Principal responsable del proceso

Rastreador (Tracker)

CEPEDA VACA LUIS ORLANDO

MEDINA CHILUISA FABIAN EDUARDO

MetricMan

Observa sin molestar

Conserva datos históricos

Probador (Tester)

CEPEDA VACA LUIS ORLANDO

MEDINA CHILUISA FABIAN EDUARDO

Ayuda al cliente con las pruebas funcionales

Se asegura de que los test funcionales se ejecutan

3.7. VERIFICACIÓN DE HIPÓTESIS

HIPÓTESIS:

“Las herramientas libres de análisis y desarrollo StarUML y FUJABA permitirá mejorar la usabilidad y la aplicabilidad de la ingeniería inversa dentro del proceso de desarrollo de software.”

HIPÓTESIS NULA Ho:

“Las herramientas libres de análisis y desarrollo StarUML y FUJABA no permitirá mejorar la usabilidad y la aplicabilidad de la ingeniería inversa dentro del proceso de desarrollo de software.”

HIPÓTESIS ALTERNATIVA H1:

“Las herramientas libres de análisis y desarrollo StarUML y FUJABA si permitirá mejorar la usabilidad y aplicabilidad de la ingeniería inversa dentro del proceso de desarrollo de software.”

TABLA N° 3.5. TABULACIÓN DE ENCUESTAS EMPLEADAS

N:	SI	NO	TOTAL
1	35	15	50
2	44	6	50
3	28	22	50
4	33	17	50
5	12	38	50
6	50	0	50
7	17	33	50
8	50	0	50

Realizado por: Fabián Medina, Orlando Cepeda

RESUMEN DE FRECUENCIAS OBSERVABLES (fo)

TABLA N° 3.6. FRECUENCIAS OBSERVABLES

XI	SI	NO	TOTAL
1	35	15	50
2	44	6	50
3	28	22	50
4	33	17	50
5	12	38	50
6	50	0	50
7	17	33	50
8	50	0	50
	219	131	400

Realizado por: Fabián Medina, Orlando Cepeda

RESUMEN DE FRECUENCIAS ESPERADAS

TABLA N° 3.7. FRECUENCIAS ESPERADAS

N:	SI	NO
1	27,37	36,02
2	27,37	36,02
3	27,37	36,02
4	27,37	36,02
5	27,37	36,02
6	27,37	36,02
7	27,37	36,02
8	27,37	36,02

Realizado por: Fabián Medina, Orlando Cepeda

$$fc = \frac{tf * tc}{tg}$$

$$fc = \frac{50 * 219}{400} = 27,37$$

$$fc = \frac{110 * 131}{400} = 36,02$$

CÁLCULO DEL χ^2

TABLA N° 3.8. CÁLCULO DEL $\chi^2 = \frac{(fo-fe)^2}{fe}$

N:	fo	fe	fo-fe	(fo-fe)²	$\chi^2 = \frac{(fo-fe)^2}{fe}$
1	35	27,37	7,63	58,21	2,12
2	44	27,37	16,63	276,55	10,10
3	28	27,37	0,63	0,39	0,01
4	33	27,37	5,63	31,69	1,15
5	12	27,37	-15,37	236,23	8,6
6	50	27,37	22,63	512,11	18,71
7	17	27,37	-10,37	107,53	3,92
8	50	27,37	22,63	512,11	18,71
9	15	36,02	-50,02	2502,00	69,46
10	6	36,02	-30,02	901,20	25,01
11	22	36,02	-14,02	196,56	5,45
12	17	36,02	-19,02	361,76	10,04
13	38	36,02	1,98	3,92	0,10
14	0	36,02	-36,02	1297,44	36,01
15	33	36,02	-3,02	9,12	0,25
16	0	36,02	-36,02	1297,44	36,01
TOTAL					245,65

Realizado por: Fabián Medina, Orlando Cepeda

$$gl = (nf - L) * (n - c1)$$

fo= Frecuencia observada

$$gl = (8 - 1) * (2 - 1)$$

fe= Frecuencia esperada

$$gl = 7 * 1 = 7$$

Tf= T. fila

Tc= T. columna

TG= T. general

gl= grados de libertad

$$(xt^2) = R = 14,1$$

$$xt^2 < xl^2$$

$$\chi^2 = 245,65$$

$$14,1 < 245,65$$

Una vez determinado el xt^2 y el xc^2 se establece que xt^2 (14,1) es menor que el $(XL)^2$ (245,65); por lo tanto se realiza la hipótesis nula (H_0) y se acepta la hipótesis alternativa (H_1) que dice:

“Las herramientas libres de análisis y desarrollo StarUML y FUJABA sí permitirá mejorar la usabilidad y la aplicabilidad de la ingeniería inversa dentro del proceso de desarrollo de software.”

3.8 VERIFICACIÓN DE OBJETIVOS

Al realizar la comparación de las herramientas de modelado libres y propietarias podemos decir que se cumplió todos los objetivos inicialmente planteados:

Analizar los fundamentos teóricos y conceptuales de las herramientas libres starUML y Fujaba.

Realizar un estudio comparativo entre las herramientas libres (starUML, Fujaba) y propietarias como (Rational Rose, Power Designer).

Mejorar la calidad del software con herramientas libres, al mismo nivel que se lo haría utilizando las herramientas propietarias. Reducir el costo de desarrollo de los sistemas informáticos.

3.9 CONCLUSIONES Y RECOMENDACIONES

3.9.1 Conclusiones

Para el desarrollo del estudio comparativo, fue necesario realizar una búsqueda a profundidad, sobre cada una de las herramientas seleccionadas con el fin de abstraer las características principales de las mismas, y así elegir los parámetros adecuados para llevar a cabo la comparación deseada.

En la búsqueda de las herramientas se pudo corroborar que existe un amplio número de opciones de tipo libre, que brindan funcionalidades importantes, para lograr llevar a cabo proyectos de desarrollo de software y a bajo costo.

Para realizar la comparación, se consideraron ocho características principales, a cada una de las cuales se les asignó un peso de acuerdo a la relevancia que representaban para el desarrollo de los proyectos. Los parámetros que obtuvieron un mayor valor, fueron los correspondientes a “soporte a UML” facilitando la utilización de otras herramientas y garantizando su correcto funcionamiento.

3.9.2 RECOMENDACIONES

En base al estudio realizado se recomienda las herramientas libres de análisis para el desarrollo de software.

Las herramientas estudiadas son de gran ayuda para las empresas ya que tienen cero costos.

StarUML es una muy buena herramienta de análisis que está al alcance de todas las empresas realizando las mismas aplicaciones de las propietarias.

CAPITULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- Gracias a este trabajo se pudo concluir que las herramientas libres para el desarrollo de software tienen la misma capacidad que las herramientas que tienen licencia o denominadas software propietario.
- Se puede concluir también que el uso de estas herramientas libres ayudan para que el costo de elaboración de software sea bajos.
- Con la ayuda de este trabajo investigativo, se pudo aplicar todos los conocimientos adquiridos durante el proceso de aprendizaje.

4.2. Recomendaciones

- El grupo de investigación recomienda el uso de las herramientas libres UML como Staruml para el proceso de desarrollo de software y en el proceso de la ingeniería inversa.
- Se recomienda que dentro de la malla de la carrera de Ingeniería en Informática y Sistemas Computacionales se dé el tratamiento o estudio de herramientas libres dentro del desarrollo de software, brindando a los estudiantes nuevas alternativas.

BIBLIOGRAFÍA

CITADA

- <http://www.gnu.org/philosophy/free-sw.es.html> (CONSULTADO EL 15-07-2013).
- <http://raifranco.blogspot.com/2012/05/definicion-de-software-libre.html> (CONSULTADO EL 17-07-2013).
- http://www.reducativa.com/webquests/licencias/dominio_publico.html (CONSULTADO EL 17-07-2013).
- <http://www.cavsi.com/preguntasrespuestas/que-es-software-de-dominio-publico/> (CONSULTADO EL 17-07-2013).
- <http://www.deciencias.net/simulaciones/paginas/libre.htm> (CONSULTADO EL 17-07-2013).
- http://e-educativa.catedu.es/44003247/aula/archivos/repositorio//0/236/Tipos_de_software.pdf (CONSULTADO EL 17-07-2013).
- http://e-educativa.catedu.es/44003247/aula/archivos/repositorio//0/236/Tipos_de_software.pdf (CONSULTADO EL 17-07-2013).
- <http://www.deciencias.net/simulaciones/paginas/libre.htm> (CONSULTADO EL 17-07-2013).
- http://scielo.sld.cu/scielo.php?pid=S1024-94352005000600009&script=sci_arttext, (CONSULTADO EL 17-07-2013).
- <http://www.gnu.org/philosophy/categories.es.html> (CONSULTADO EL 17-07-2013).
- <http://jalvarezm.wordpress.com/2006/09/30/el-software-privativo/>, (CONSULTADO EL 17-07-2013).
- <http://www.taringa.net/posts/info/9270788/Definiciones-Shareware-Freeware-Demo-y-Licencia.html>, (CONSULTADO EL 17-07-2013).
- <http://www.ecured.cu/index.php/Freeware>, (CONSULTADO EL 17-07-2013).

- <http://www.aulaclie.es/articulos/licencias.html> (CONSULTADO EL 18-07-2013).
- <http://alvarolivares.wordpress.com/2009/02/25/software-libre-concepto-y-mercado-freeware-y-shareware/> (CONSULTADO EL 18-07-2013).
- <http://black-byte.com/review/staruml/> (CONSULTADO EL 18-07-2013).
- http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado,
- <http://translate.google.com.ec/translate?hl=es&sl=en&u=http://www.fujaba.de/&prev=/search%3Fq%3Dque%2Bes%2Bfujaba%26biw%3D1230%26bih%3D567> (CONSULTADO EL 18-07-2013).
- <http://translate.google.com.ec/translate?hl=es&sl=en&u=http://java-source.net/open-source/uml-modeling/fujaba&prev=/search%3Fq%3Dfujaba%26biw%3D1230%26bih%3D567> (CONSULTADO EL 18-07-2013).
- <http://cnx.org/content/m17432/latest/> (CONSULTADO EL 18-07-2013).
- <http://learnassembler.com/gdbesp.html> (CONSULTADO EL 18-07-2013).
- http://centrodeartigos.com/articulos-educativos/article_11736.html, (CONSULTADO EL 21-07-2013).
- <http://cnx.org/content/m17432/latest/> (CONSULTADO EL 21-07-2013).
- <http://www.viruslist.com/sp/hackers/glossary?glossid=153598730> (CONSULTADO EL 21-07-2013).
- <http://cnx.org/content/m17432/latest/> (CONSULTADO EL 22-07-2013).
- <http://www.buenastareas.com/ensayos/Decompiladores/172515.html> (CONSULTADO EL 22-07-2013).
- <http://cnx.org/content/m17432/latest/> (CONSULTADO EL 22-07-2013).
- <http://herramientascase.wordpress.com/integracion-estructura-componentes/> (CONSULTADO EL 22-07-2013).
- <http://paulchasiluisa.galeon.com/> manifiesta que la Estructura general de una Herramienta Case: (CONSULTADO EL 22-07-2013).

BÁSICA

- Almazan, Fabio. (1996). *Desarrollo de software*. México: Editorial Trillas. 5ta edición. 302 p.
- Cessario, Jorge Andrés. (2002). *Sitios web sin problemas*. Madrid: Editorial McGraw Hill, 1ra edición, 22 - 25p.
- Corradini, Horacio. (2008). *Diseño de páginas web interactivas*. México: Editorial Alfaomega, 3ra edición, 16,17 p.
- García, Badell. y Lapetra, José Javier. (2009). *herramientas case*. España: Editorial Mera, 5ta edición, 125p
- Jacobson, Booch. (2001). *Lenguaje Unificado de Modelado*. Madrid: Editorial Addison Wesley, 1ra edición, 76,77p.
- Kendall, Albert. (2008). *Aplicación de tecnología Informática*. Italia: Editorial Lexus, 3ra Edición, 19 p.
- Montaña, Cruz. (2002). *Crear Servicios Web*, Perú: Editorial Mondadori, 3ra Edición, 202 p.
- Pérez, García. (2009.) *Modelos de componentes Rational Rose*. Colombia: Editorial Pearson Educación, 1ra Edición, 125-130 p.
- Reyes, Rodríguez. (2008). *Programación html*, España: Editorial Maza, 2da Edición, 50p

VIRTUAL

- <http://www.humbertocervantes.net/cursos/ayd/practicass/practica2/practica.html>
- <http://www.forosdelweb.com/f50/herramientas-case-herramientas-uml-798448/>
- <http://www.scribd.com/doc/25374125/Estudio-de-Herramientas-CASE-de-Soporte-a-UML-y-UML2>
- <http://www.softqanetwork.com/2006/05/staruml-herramienta-gratuita-de-uml/>

- <http://www.kriptopolis.org/introduccion-a-la-ingenieria-inversa-del-software>
- <http://fbio.uh.cu/bioinfo/glosario.html>
- <http://www.lcc.uma.es/~eat/services/html-js/manual1.html>.
- <http://www.librosweb.es/ajax/capitulo1.html><http://www.arqhys.com/conte>
nidos/documentacion y proyectos.html.
- [http://wiki.buenosaireslibre.org/Glosario Servidor](http://wiki.buenosaireslibre.org/Glosario%20Servidor).
- http://viaservercenter.com/index.php?option=com_content&task=view...&id=31&Itemid=46.
- <http://upsg01.foroactivo.com/tema-2-programacion-i-paralelo-20-f6/tema-2-uml-t166.htm>
- http://www.une.edu.ve/postgrado/intranet/investigacion_virtual/manual_para_la_elaboracion_tesis_de_grado.htm
- <http://www.ub.es/geocrit/b3w235.htm#Metodología%20para%20llevar%20a%20cabo%20investigaciones>
- http://eco.mdp.edu.ar/cendocu/opac/tesis/byd/duran_em.pdf
- <http://www.scribd.com/doc/260971/Ejemplos-de-ficha-bibliografica>
- http://translate.google.com.ec/translate?hl=es&sl=en&u=http://www.hpi.uni-potsdam.de/giese/projects/fujaba.html%3FL%3D1%26cHash%3D7e31f9324645305286dc6c9c443d7730&ei=dBc3Tea_PIXpgQfyi_21Aw&sa=X&oi=translate&ct=result&resnum=9&ved=0CFIQ7gEwCA&prev=/search%3Fq%3Ddiagramas%2Bque%2Butiliza%2Bfujaba%26hl%3Des%26biw%3D1003%26bih%3D562%26prmd%3Divns
- <http://www.scribd.com/doc/25374125/Estudio-de-Herramientas-CASE-de-Soporte-a-UML-y-UML2>
- <http://case-tools.org/tools/fujaba.html><http://redalyc.uaemex.mx/pdf/215/21513706.pdf>
- <http://www.scribd.com/doc/25374125/Estudio-de-Herramientas-CASE-de-Soporte-a-UML-y-UML2>

GLOSARIO DE TÉRMINOS

A:

Arquitectura de la Información (Ai): La Arquitectura de la Información (AI) es la disciplina y arte encargada del estudio, análisis, organización, disposición y estructuración de la información en espacios de información, y de la selección y presentación de los datos en los sistemas de información interactivos y no interactivos.

Código Abierto

(*Open-Source*) Relativo al software para el cual el código fuente está disponible en forma gratuita.

Código Fuente

Un conjunto de líneas que conforman un bloque de texto, escrito según las reglas sintácticas de algún lenguaje de programación destinado a ser legible por humanos.

Comparación

Se denomina comparación a la especificación de la situación o posición de una magnitud, cualidad o proceso, dentro de una escala a partir de un determinado punto de referencia. Dependiendo de la situación o posición del elemento respecto del punto de comparación, se establecen tres grados: *superioridad*, *inferioridad* e *igualdad*.

E:

Elementos

Los elementos que pueden aparecer en un Diagrama de Casos de Uso son: actores, casos de uso y relaciones entre casos de uso.

Estructura

Los mensajes se dibujan cronológicamente desde la parte superior del diagrama a la parte inferior; la distribución horizontal de los objetos es arbitraria. Durante el análisis inicial, el modelador típicamente coloca el nombre 'business' de un mensaje en la línea del mensaje. Más tarde, durante el diseño, el nombre 'business' es reemplazado con el nombre del método que está siendo llamado por un objeto en el otro. El método llamado, o invocado, pertenece a la definición de la clase instanciada por el objeto en la recepción final del mensaje.

Estudio Comparativo.

El diseño de la investigación comparativa es simple. Estudiamos ejemplares que pertenecen al mismo grupo pero que difieren en algunos aspectos. Estas diferencias llegan a ser el foco de la exanimación. La meta es descubrir por qué los casos son diferentes: para revelar la estructura subyacente general que genera o permite tal variación.

F:

Factibilidad

Se refiere a la disponibilidad de los recursos necesarios para llevar a cabo los objetivos o metas señalados. Generalmente la factibilidad se determina sobre un proyecto.

G:

Gnu

La licencia pública general (Gnu)

Gestión del Conocimiento

La Gestión del conocimiento (del inglés *Knowledge Management*) es un concepto aplicado en las organizaciones, que pretende transferir el conocimiento y

experiencia existente entre sus miembros, de modo que pueda ser utilizado como un recurso disponible para otros en la organización.

H:

Herencia

Se define como la reutilización de un objeto padre ya definido para poder extender la funcionalidad en un objeto hijo. Los objetos hijos heredan todas las operaciones y/o propiedades de un objeto padre.

O:

Operaciones

Comúnmente llamados métodos, son aquellas actividades o verbos que se pueden realizar con/para este objeto, como por ejemplo abrir, cerrar, buscar, cancelar, acreditar, cargar. De la misma manera que el nombre de un atributo, el nombre de una operación se escribe con minúsculas si consta de una sola palabra. Si el nombre contiene más de una palabra, cada palabra será unida a la anterior y comenzará con una letra mayúscula, a excepción de la primera palabra que comenzará en minúscula. Por ejemplo: abrirPuerta, cerrarPuerta, buscarPuerta, etc.

P:

Propiedades

También llamados **atributos** o características, son valores que corresponden a un objeto, como color, material, cantidad, ubicación.

T:

Transiciones

Las transiciones reflejan el paso de un estado a otro, bien sea de actividad o de

acción. Esta transición se produce como resultado de la finalización del estado del que parte el arco dirigido que marca la transición

ANEXOS

ANEXO 1

TIPO DE SOFTWARE	VENTAJAS
StarUML	<ul style="list-style-type: none"> ✓ Se maneja con facilidad ✓ Es gratuito, ósea, no necesita crack, ni serial, ni actualización. ✓ Es compatible con cualquier versión de Windows, No pesa mucho, solo pesa 27.5 Mb. ✓ Funciona desde una P IV, y es compatible a todas las computadoras.
Fujaba	<ul style="list-style-type: none"> ✓ Libre para su respectiva descarga. ✓ Su facilidad de uso le hace una de las herramientas más amigables para el analista de sistemas. ✓ Facilidad de crear diagramas a partir de código Java. ✓ Es capaz de hacer ingeniería inversa y crear los diagramas a partir del código Java

Realizado por: Fabián Medina, Orlando Cepeda

ANEXO 2

TIPO DE SOFTWARE	CARACTERÍSTICAS
starUML	<ul style="list-style-type: none"> ✓ Definir elementos propios para los diagramas, que no necesariamente pertenezcan al estándar de UML, ✓ La capacidad de generar código a partir de los diagramas y viceversa, actualmente funcionando para los lenguajes C++, C# y Java. ✓ Generar documentación en formatos Word, Excel y PowerPoint sobre los diagramas. ✓ Patrones GoF (Gang of Four), EJB (Enterprise JavaBeans) y personalizados. ✓ Plantillas de proyectos. ✓ Posibilidad de crear plugins para el programa.
Fujaba	<ul style="list-style-type: none"> ✓ Potente, fácil de usar, pero formales, visuales, sistema de software orientado a objetos lenguaje de especificación (diagramas de clases UML y diagramas de actividad especializada, llamada Historia de diagramas basado en las transformaciones gráfico). ✓ Generación de código Java basada en la especificación formal de la

estructura de un sistema y el comportamiento que se traduce en un prototipo de sistema de archivo ejecutable.

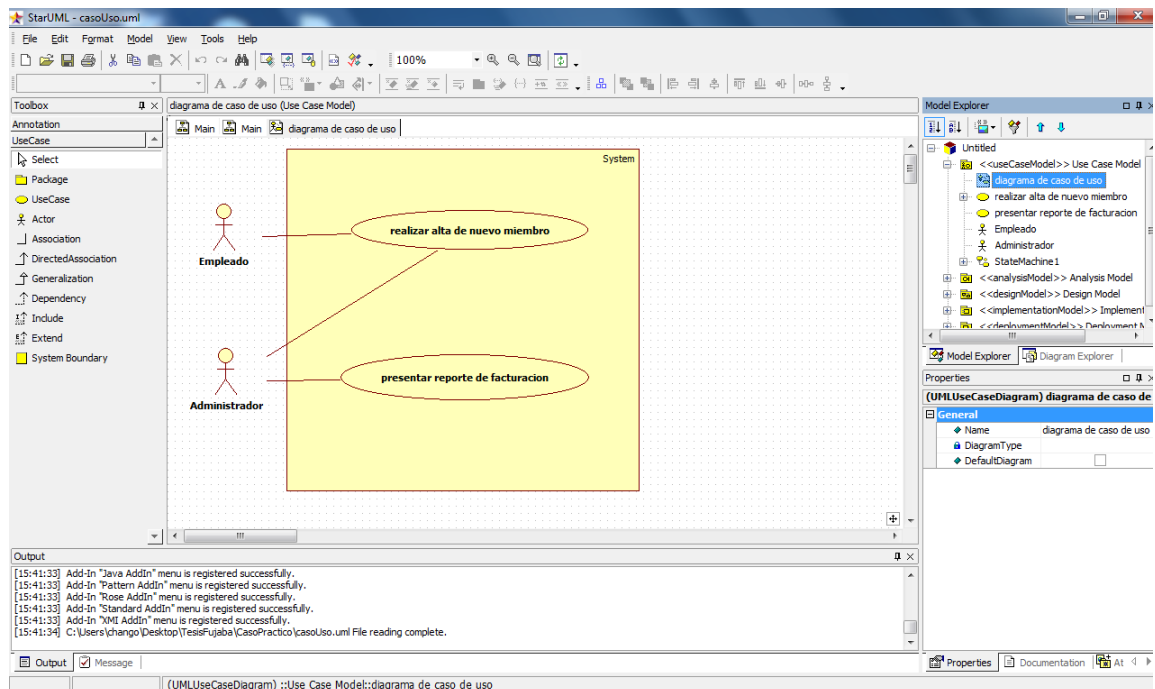
- ✓ Un caso extensible marco herramienta para los investigadores, que pueden desarrollar sus propios Fujaba plug-ins.
- ✓ Numerosos Fujaba plug-ins de la prestación de apoyo por ejemplo para la ingeniería inversa del código fuente mediante la creación de diagramas de clases UML, la detección de patrones de diseño, los idiomias, las pautas.

Realizado por: Fabián Medina, Orlando Cepeda

ANEXO 3

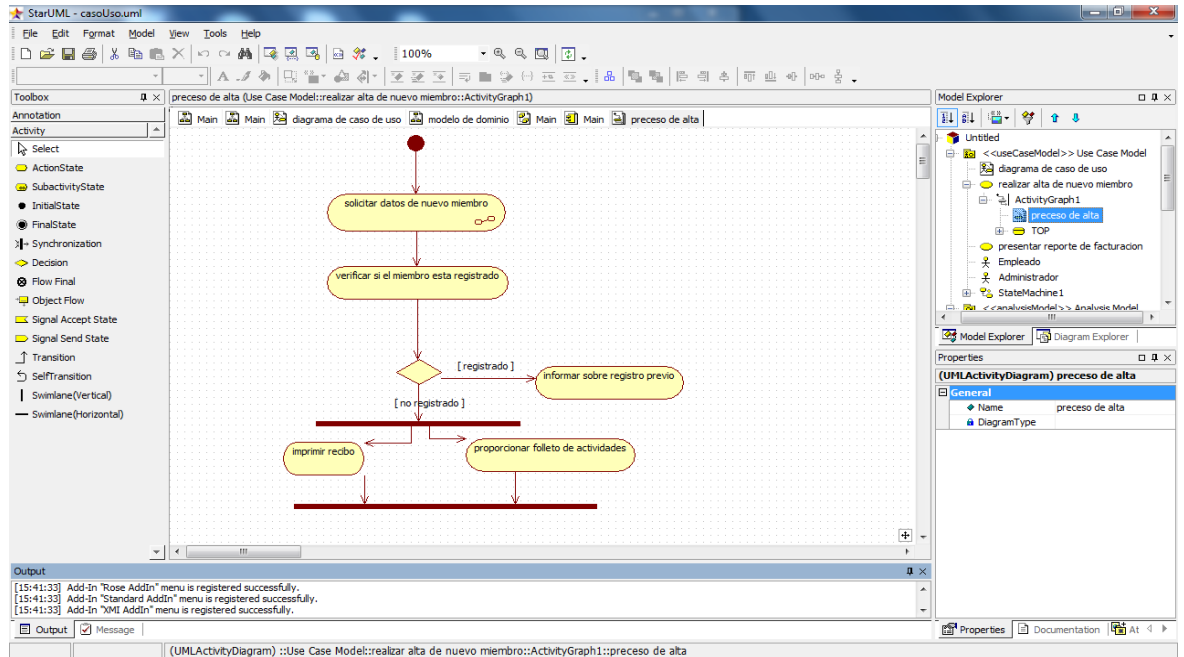
CASO PRÁCTICO:

STARUML



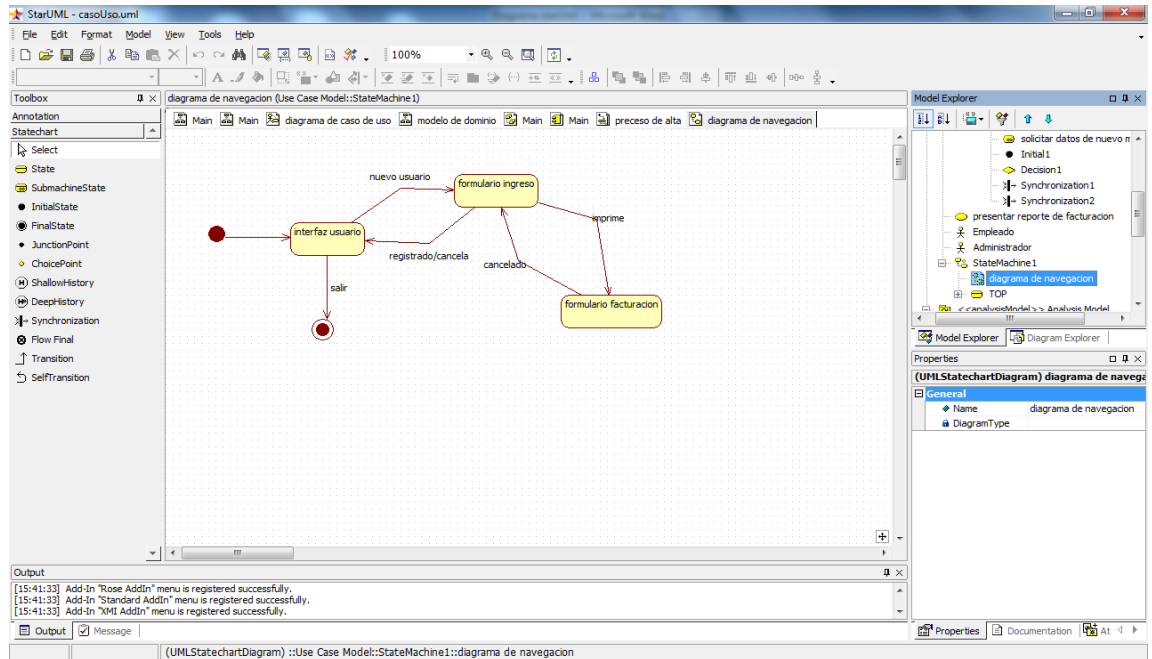
Realizado por: Fabián Medina, Orlando Cepeda

Diagrama de Actividad.

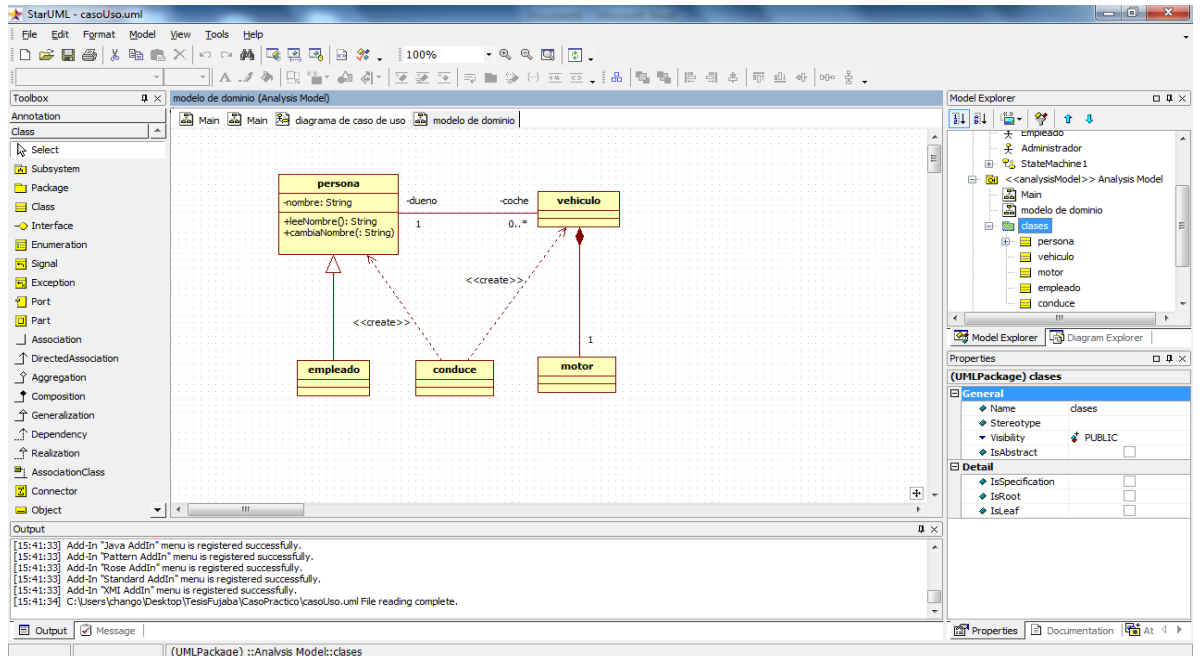


Realizado por: Fabián Medina, Orlando Cepeda

Diagrama de Navegación.

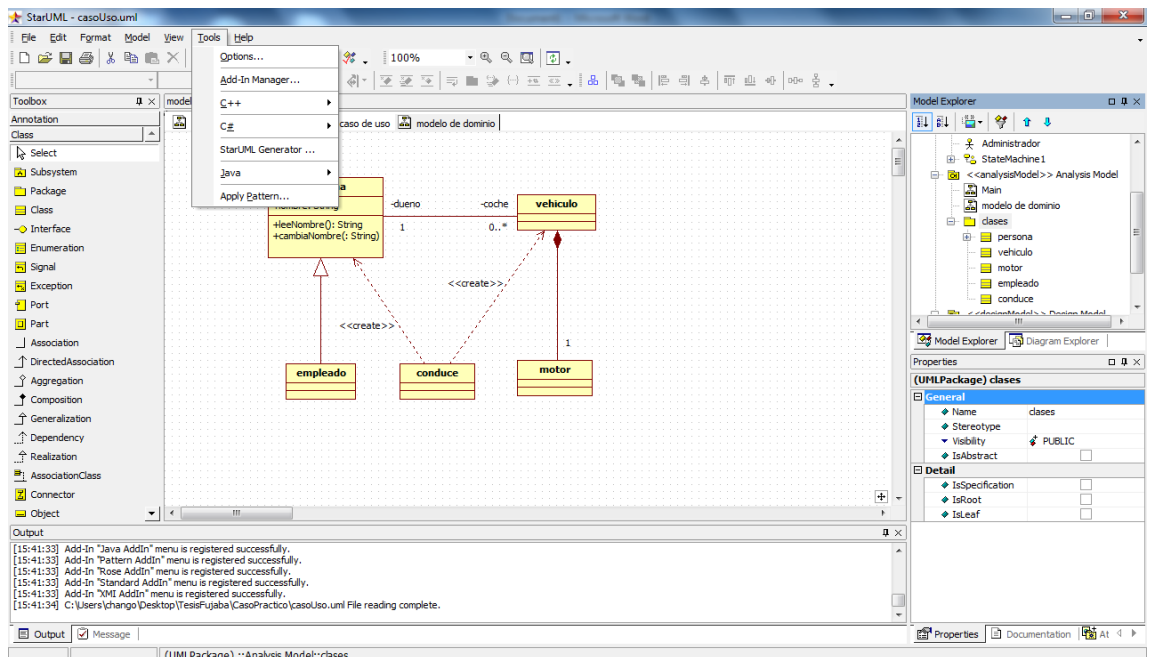


Realizado por: Fabián Medina, Orlando Cepeda



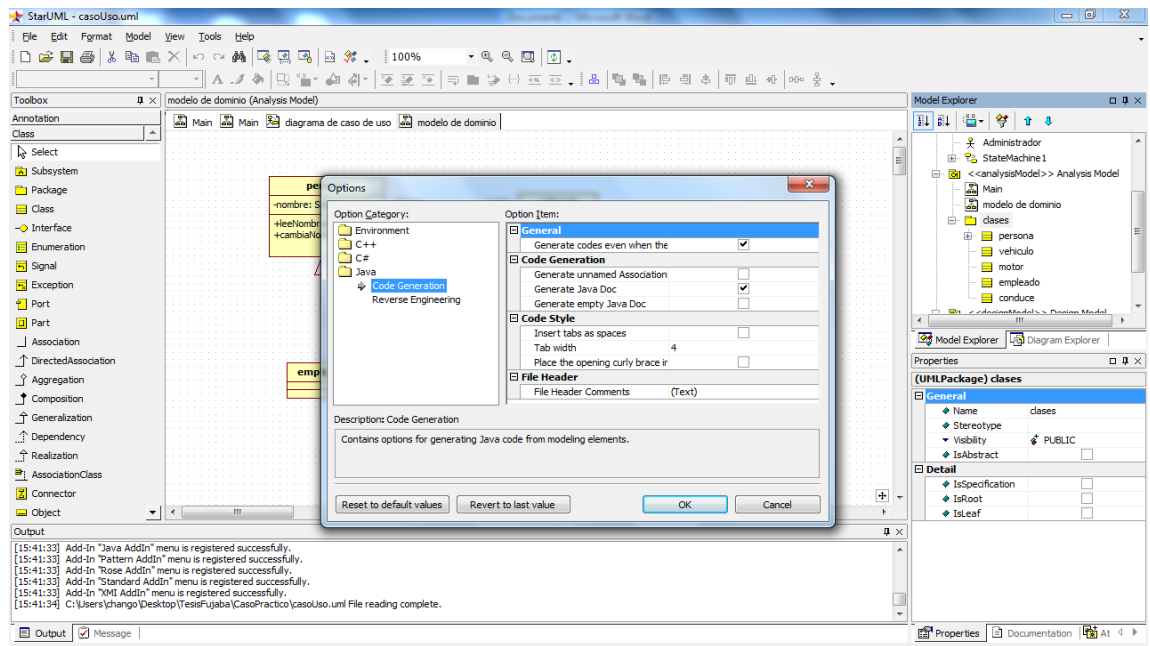
Realizado por: Fabián Medina, Orlando Cepeda

Generar código (starUML)

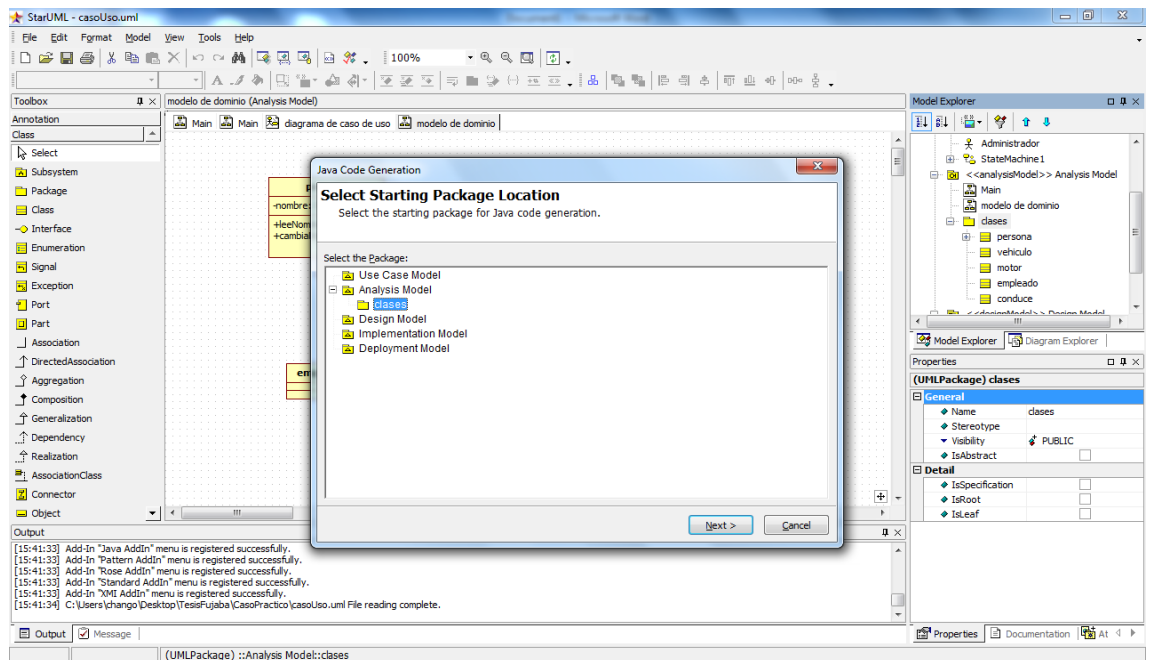


Realizado por: Fabián Medina, Orlando Cepeda

Generando Código Java a partir de un ejemplo.

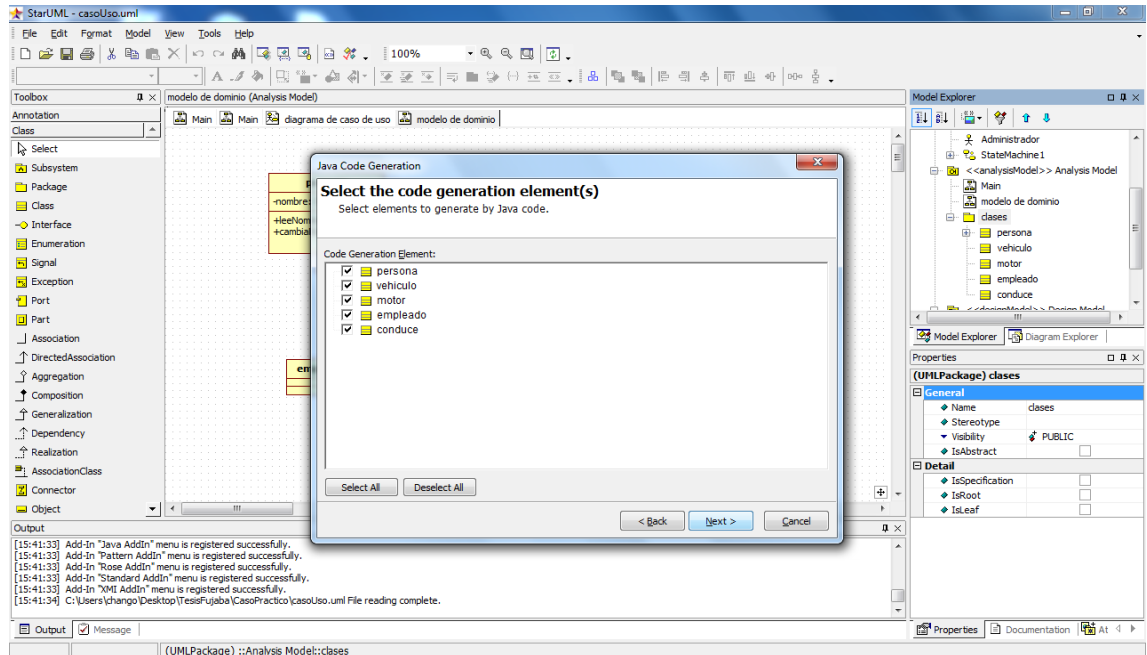


Realizado por: Fabián Medina, Orlando Cepeda



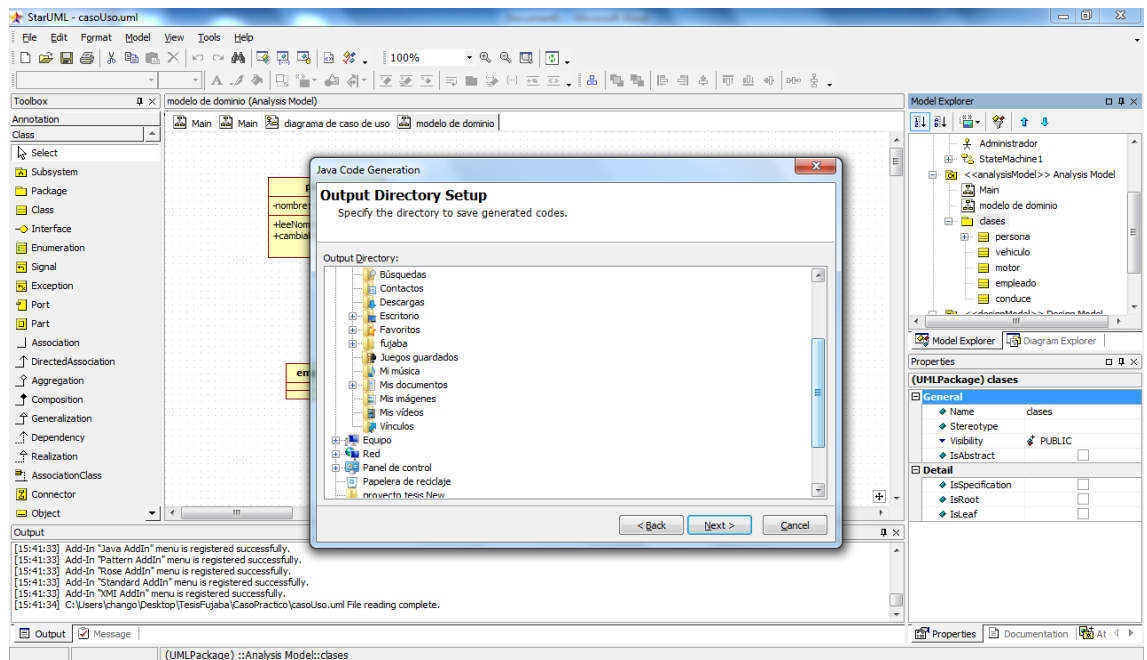
Realizado por: Fabián Medina, Orlando Cepeda

Seleccionamos los elementos para generar el código.

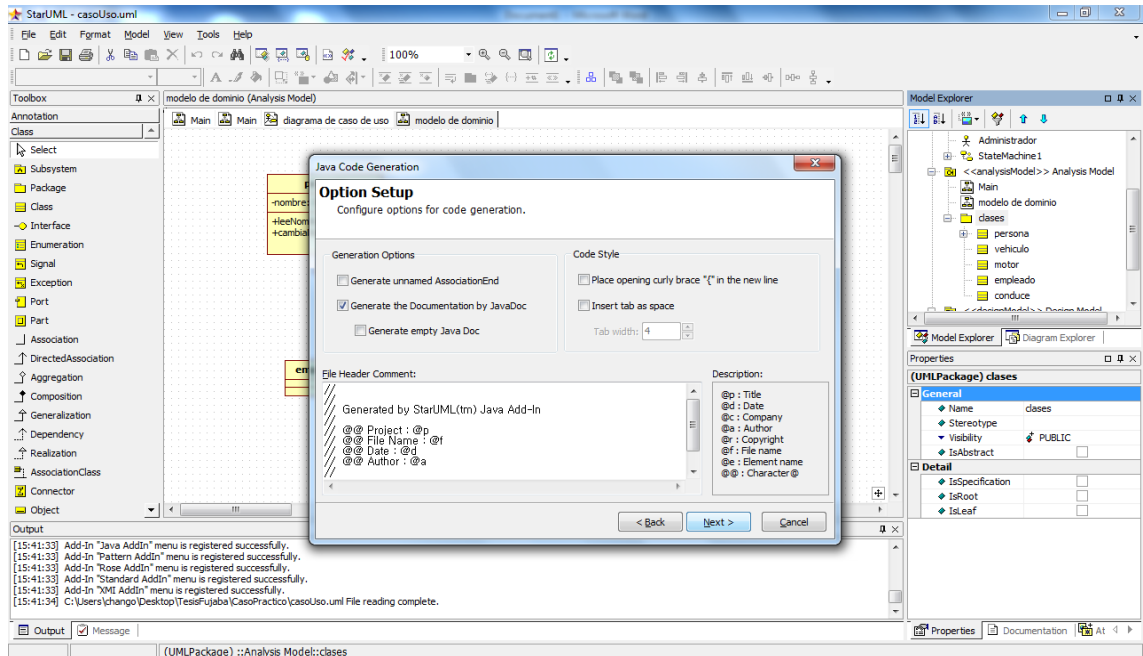


Realizado por: Fabián Medina, Orlando Cepeda

Seleccionamos el directorio donde deseamos guardar el código java que se va a generar.

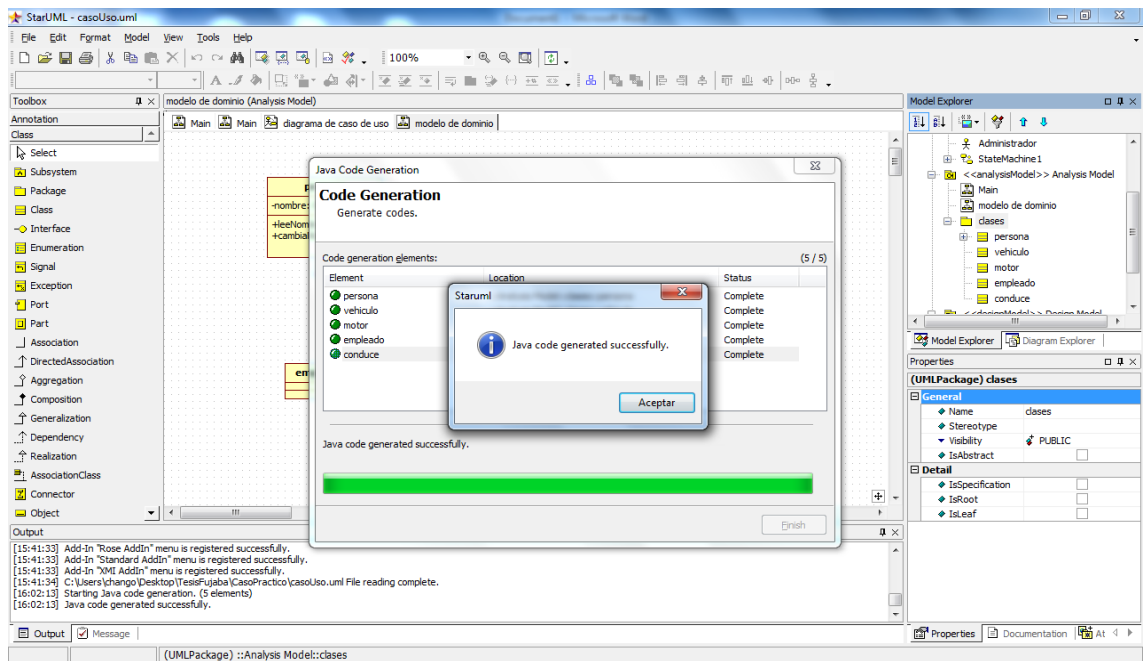


Realizado por: Fabián Medina, Orlando Cepeda



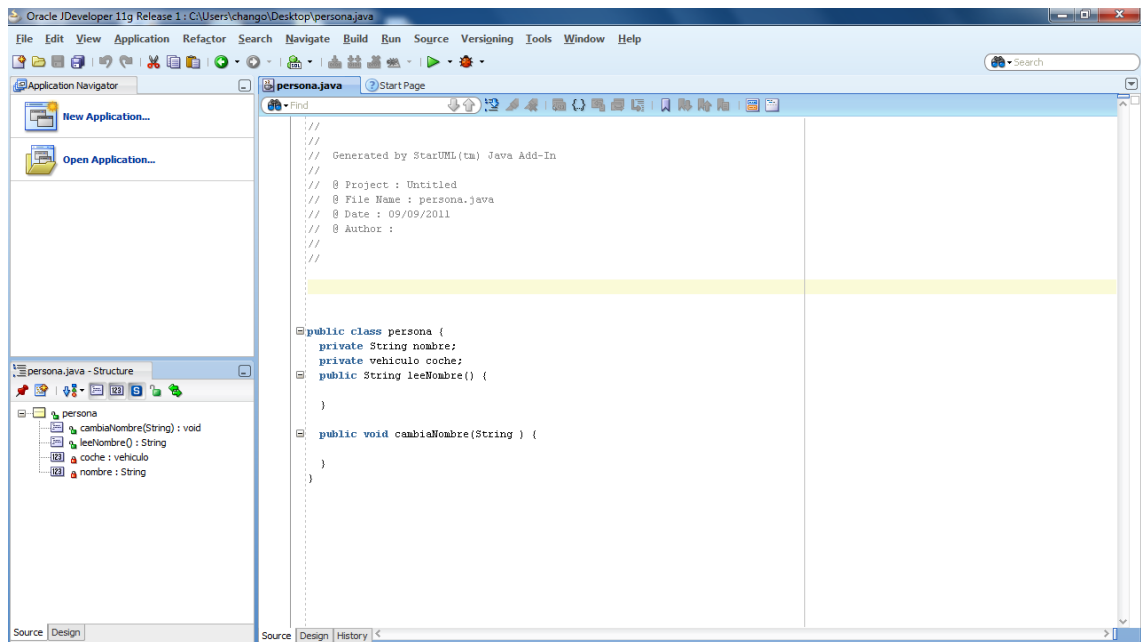
Realizado por: Fabián Medina, Orlando Cepeda

Código Java generado con éxito.



Realizado por: Fabián Medina, Orlando Cepeda

Luego tenemos el código generado en java.



Realizado por: Fabián Medina, Orlando Cepeda

Código Generado A Partir De La Reingeniería Inversa

```
//
//
// Generated by StarUML(tm) Java Add-In
//
// @ Project : Untitled
// @ File Name : persona.java
// @ Date : 09/09/2011
// @ Author :
//
//

public class persona {
    private String nombre;
    private vehiculo coche;
    public String leeNombre() {

    }

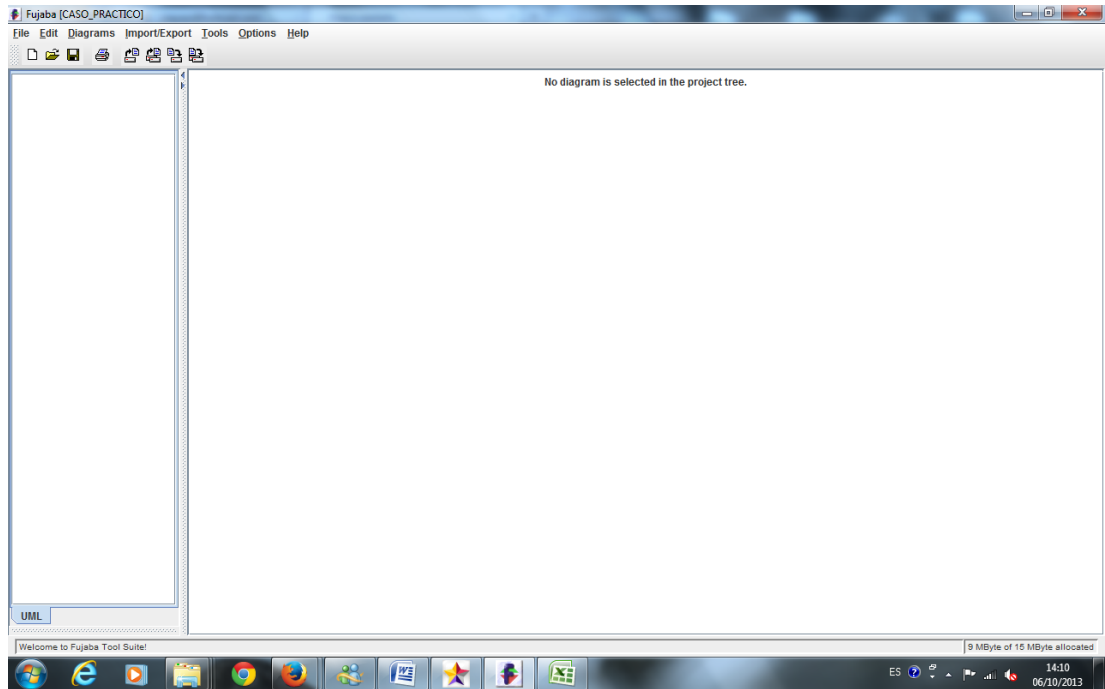
    public void cambiaNombre(String ) {

    }
}
}
```

Realizado por: Fabián Medina, Orlando Cepeda

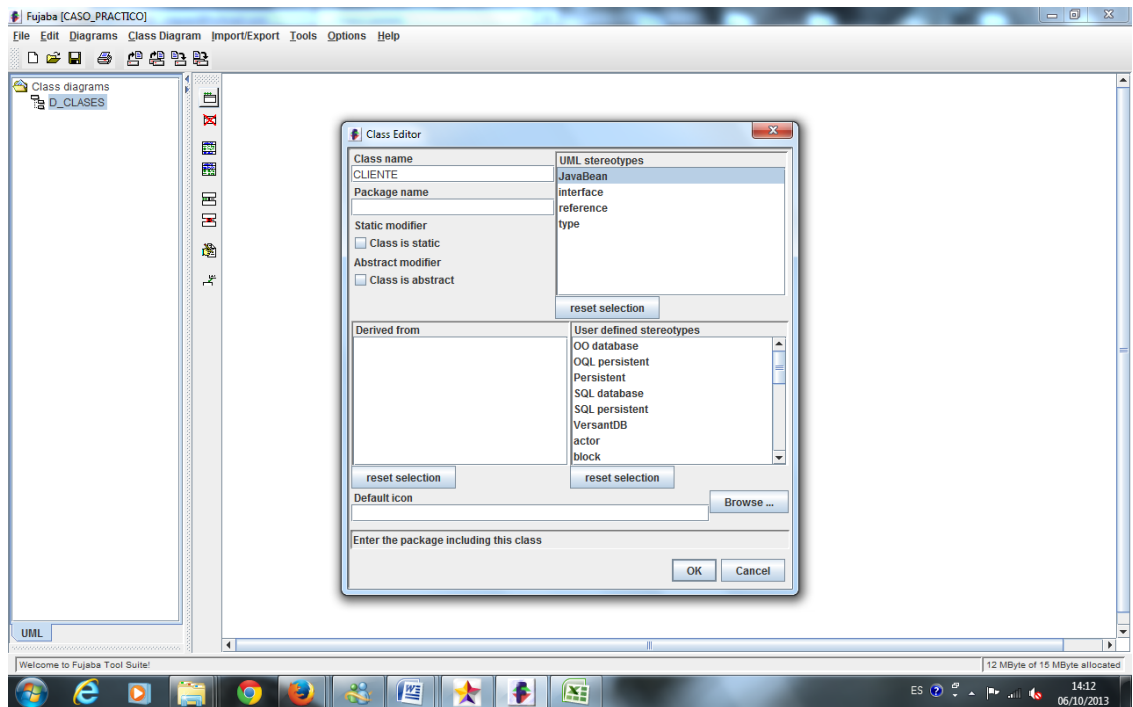
ANEXO 4

FUJABA

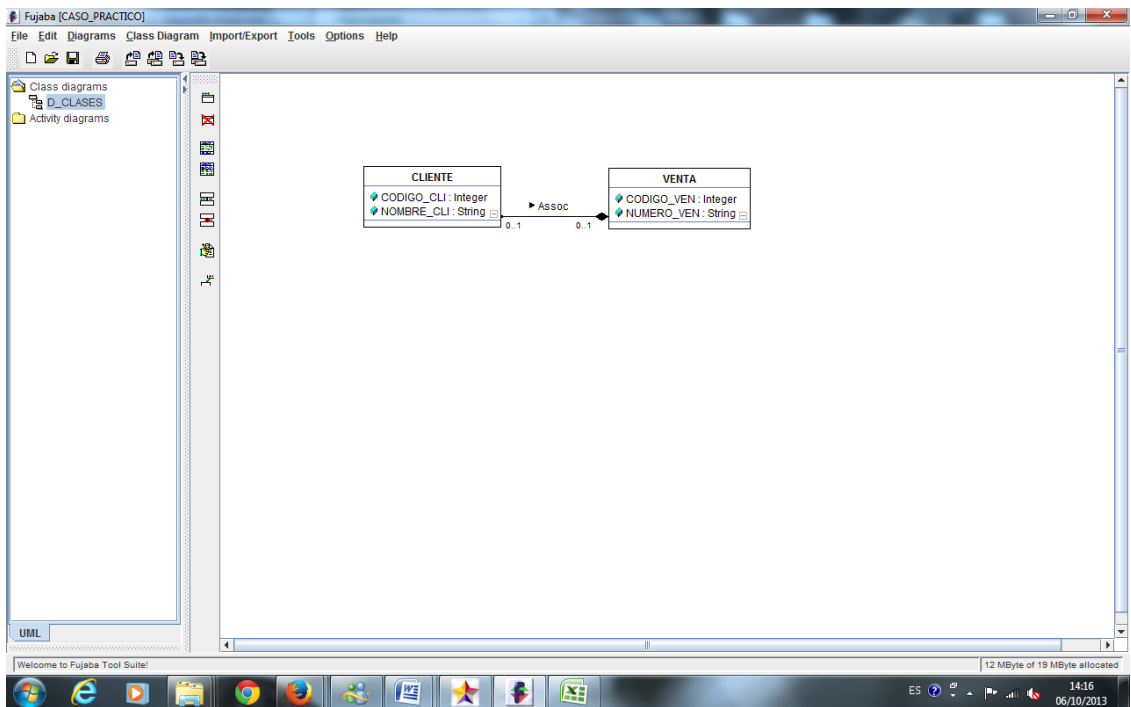


Realizado por: Fabián Medina, Orlando Cepeda

Diagrama de Clases

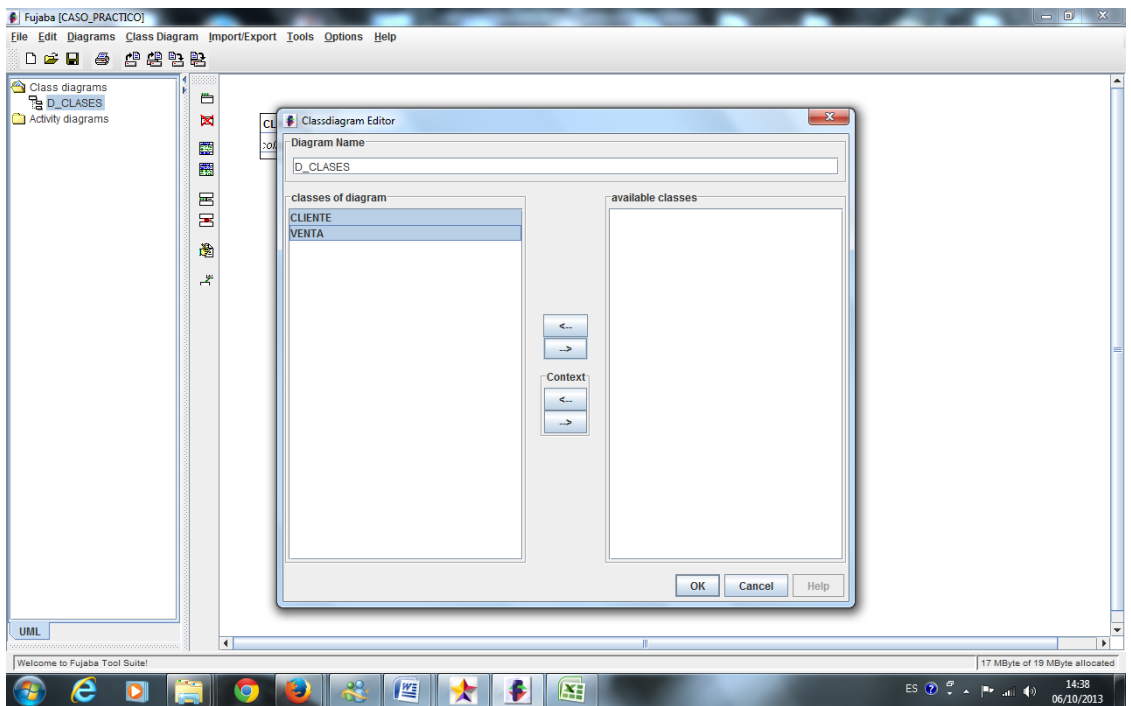


Realizado por: Fabián Medina, Orlando Cepeda



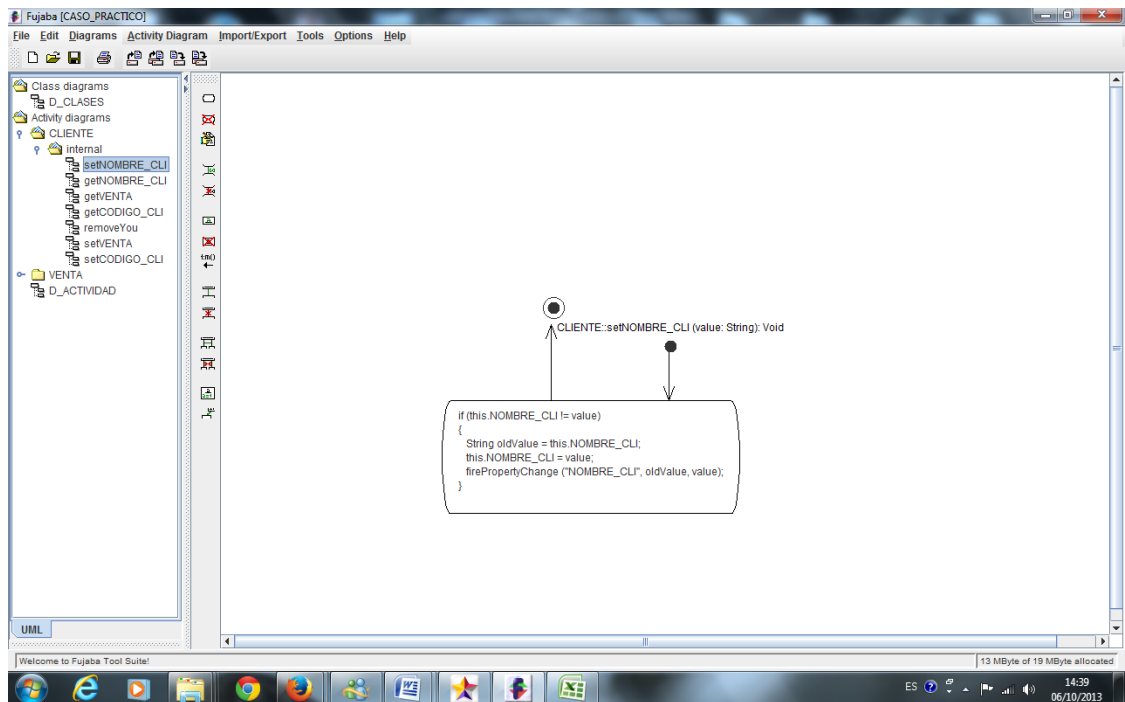
Realizado por: Fabián Medina, Orlando Cepeda

Diagrama de Actividad

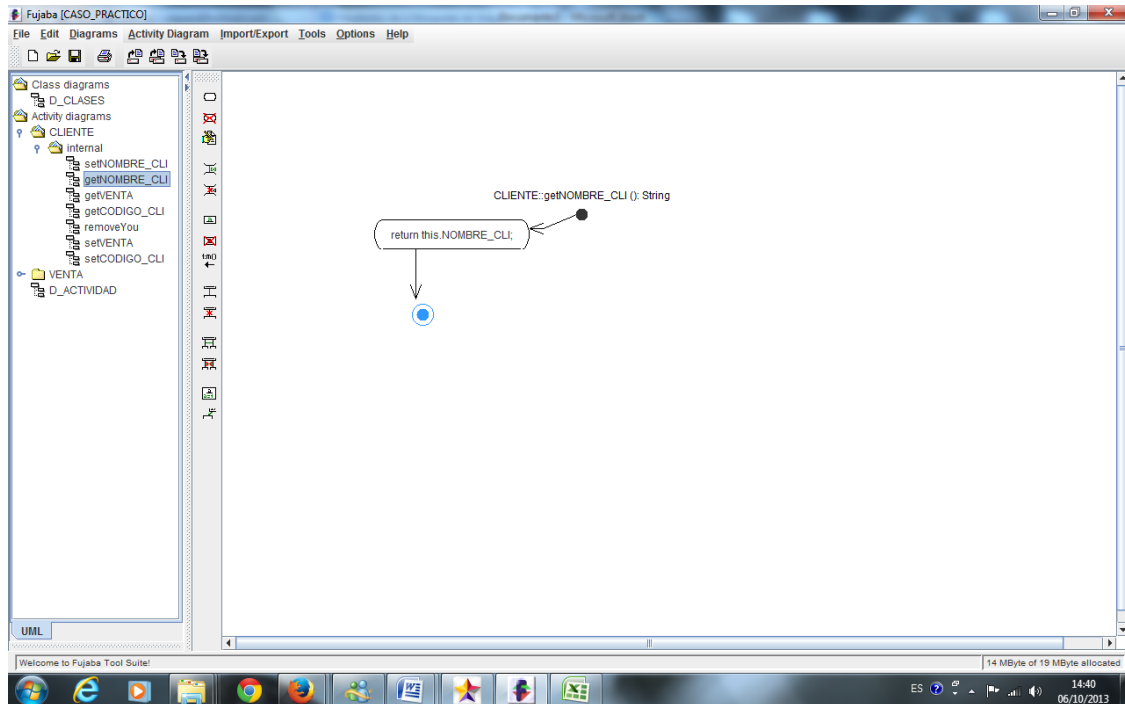


Realizado por: Fabián Medina, Orlando Cepeda

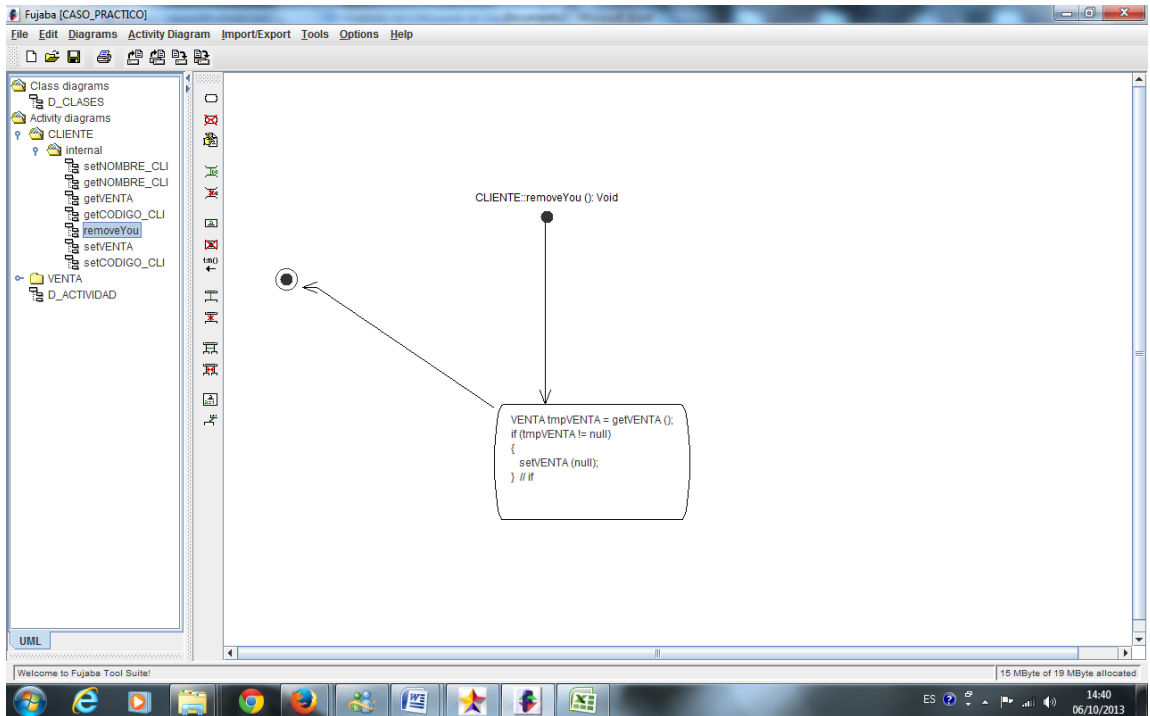
Genera Automáticamente el diagrama de Actividad



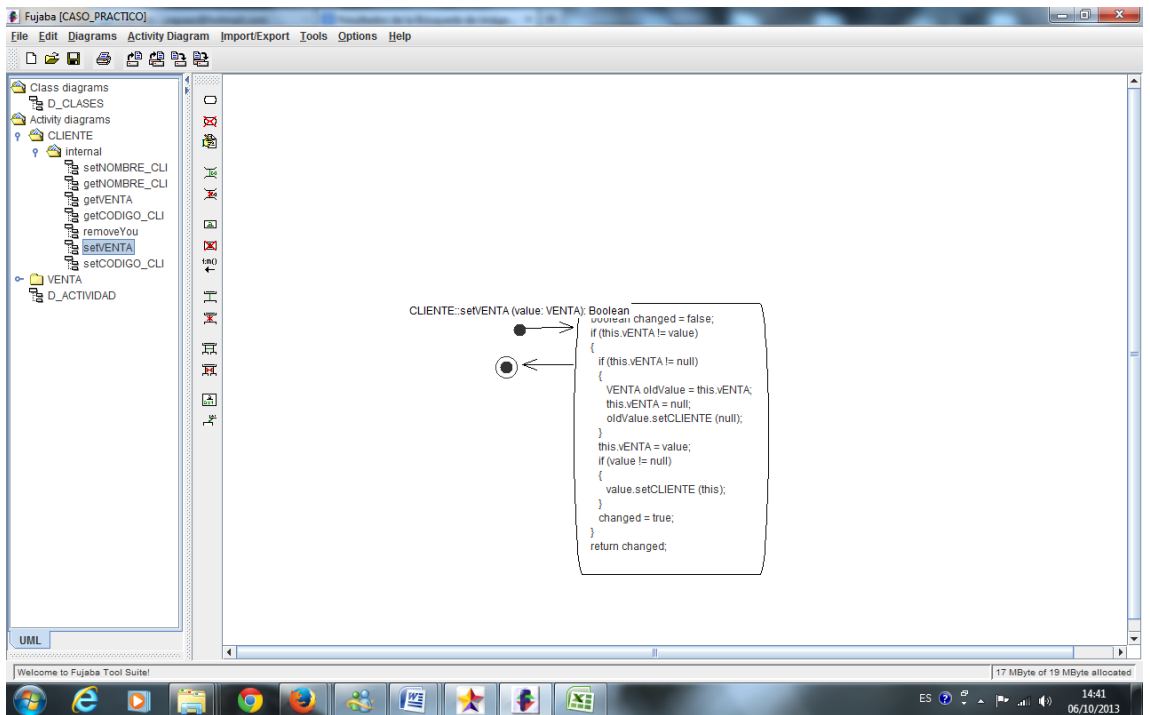
Realizado por: Fabián Medina, Orlando Cepeda



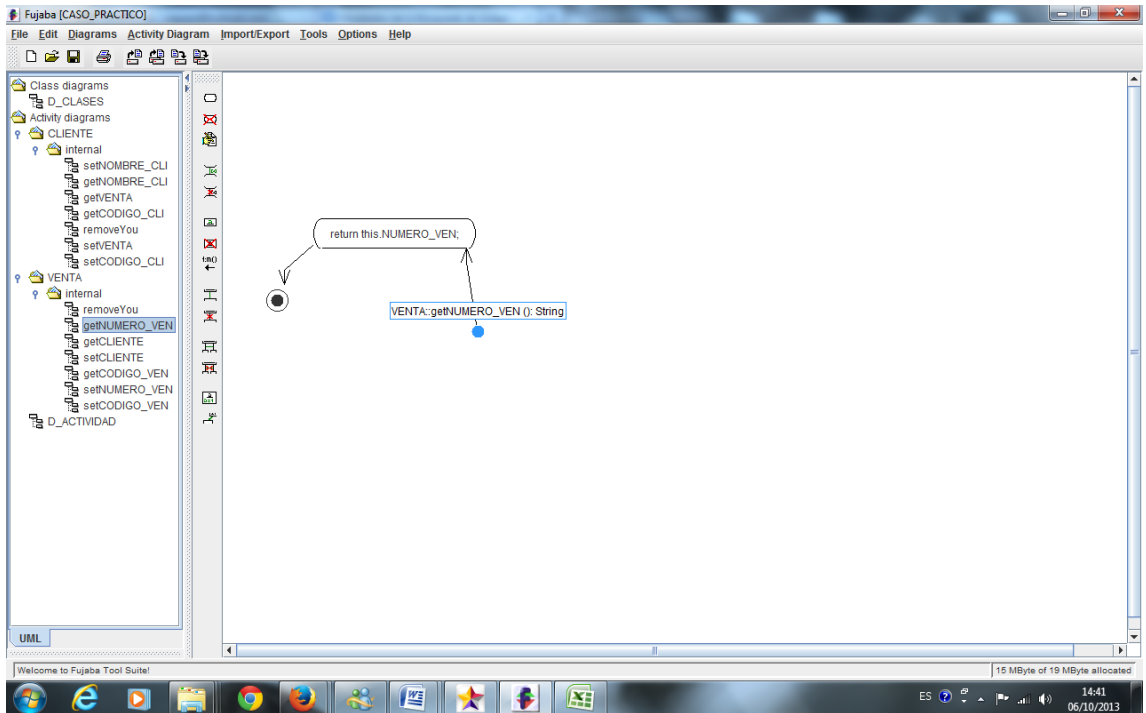
Realizado por: Fabián Medina, Orlando Cepeda



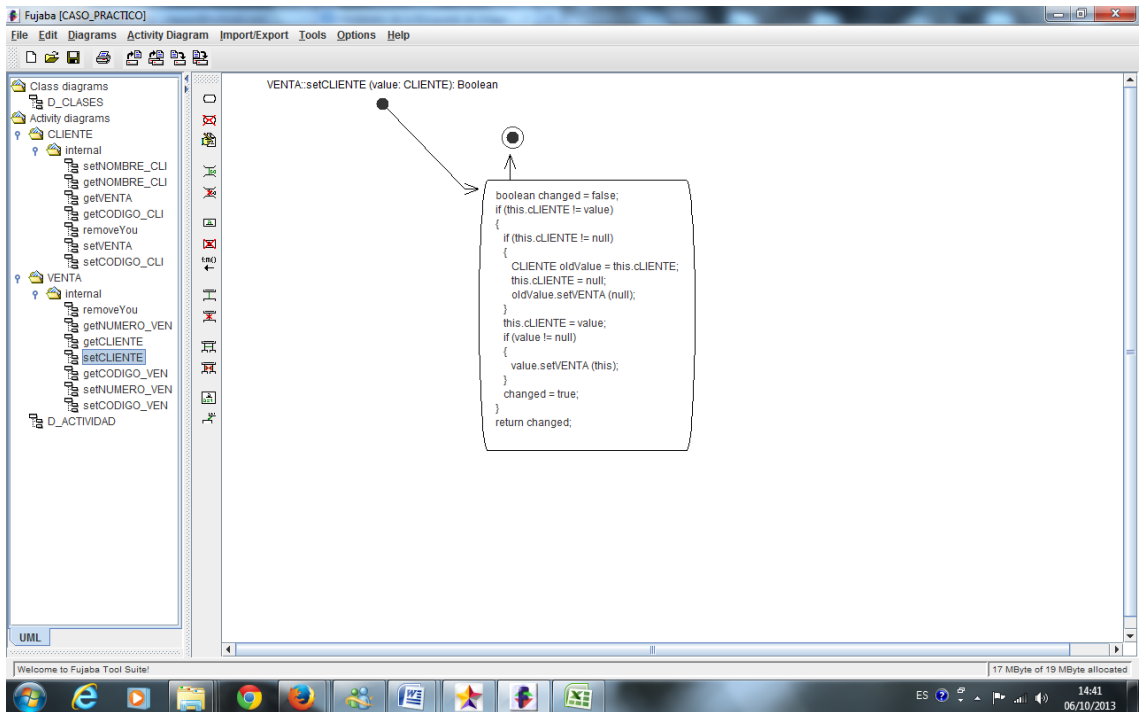
Realizado por: Fabián Medina, Orlando Cepeda



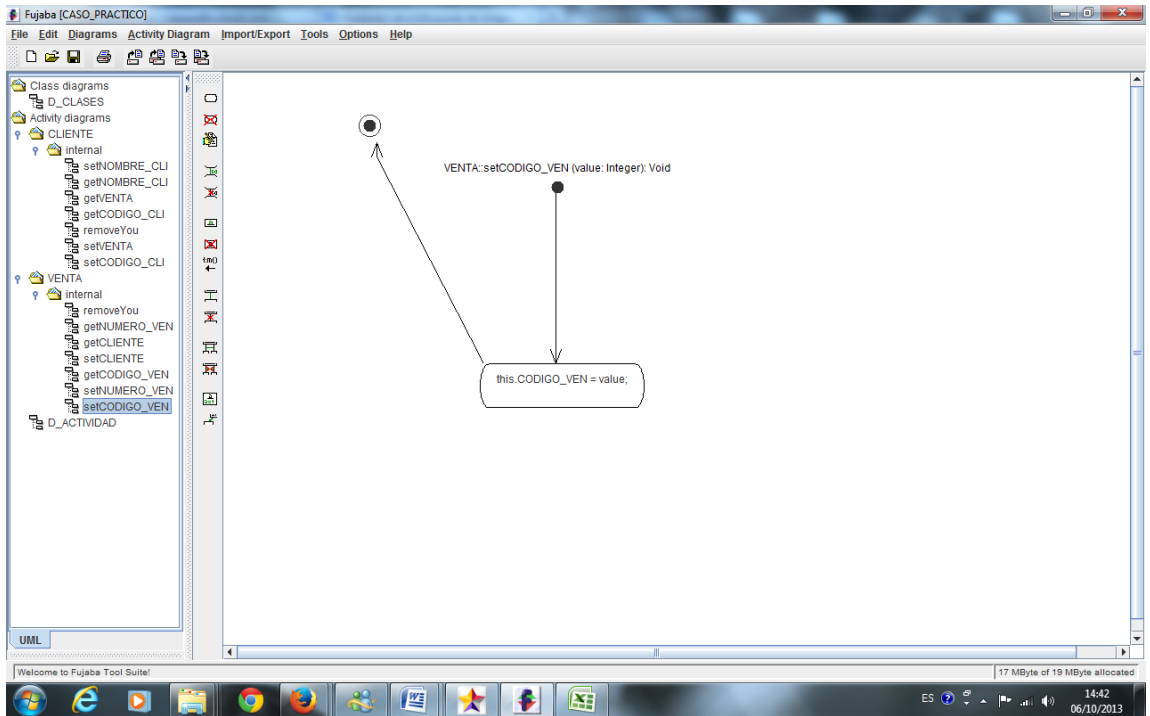
Realizado por: Fabián Medina, Orlando Cepeda



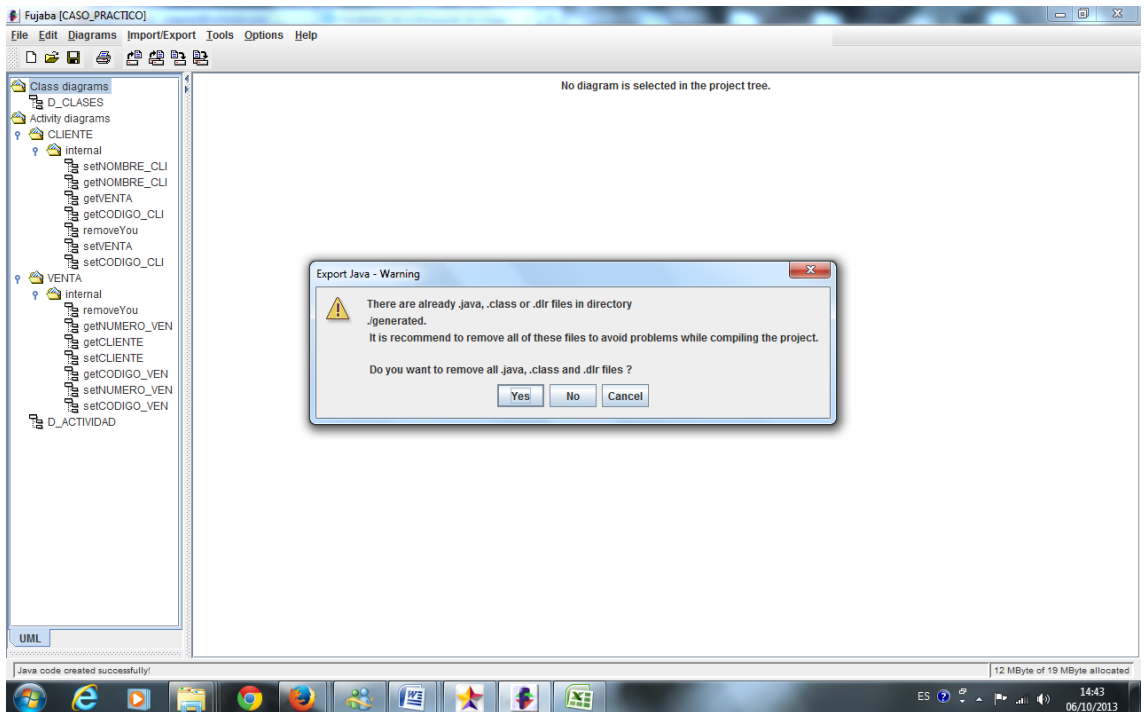
Realizado por: Fabián Medina, Orlando Cepeda



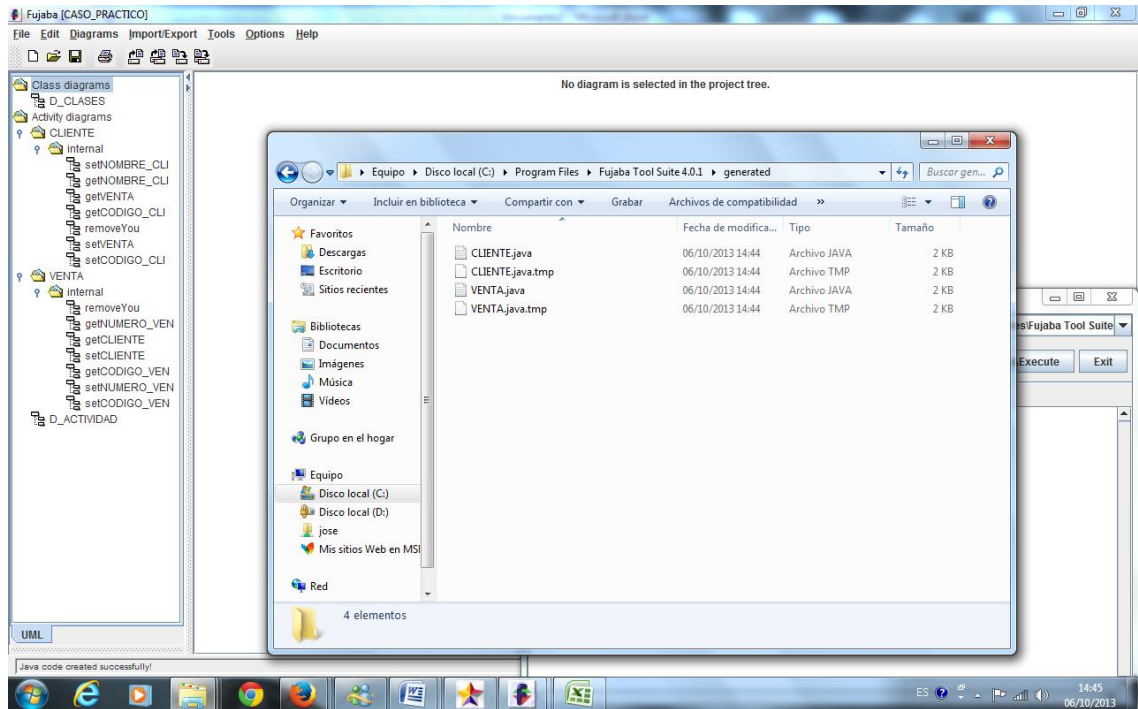
Realizado por: Fabián Medina, Orlando Cepeda



Realizado por: Fabián Medina, Orlando Cepeda

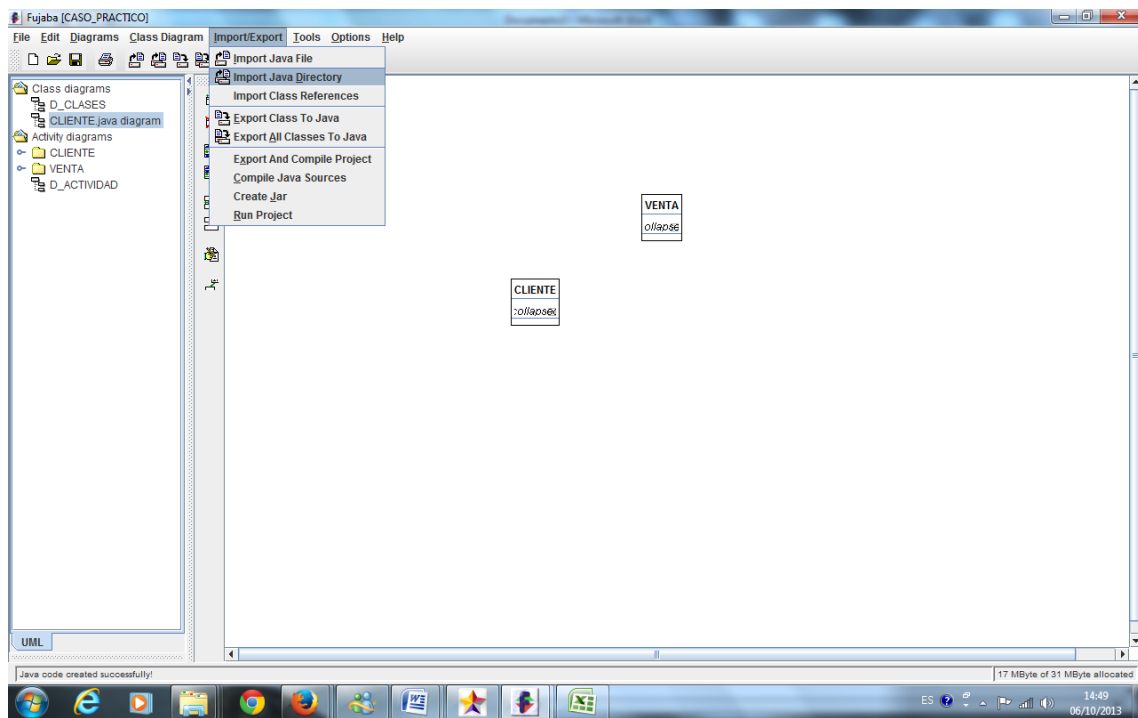


Realizado por: Fabián Medina, Orlando Cepeda



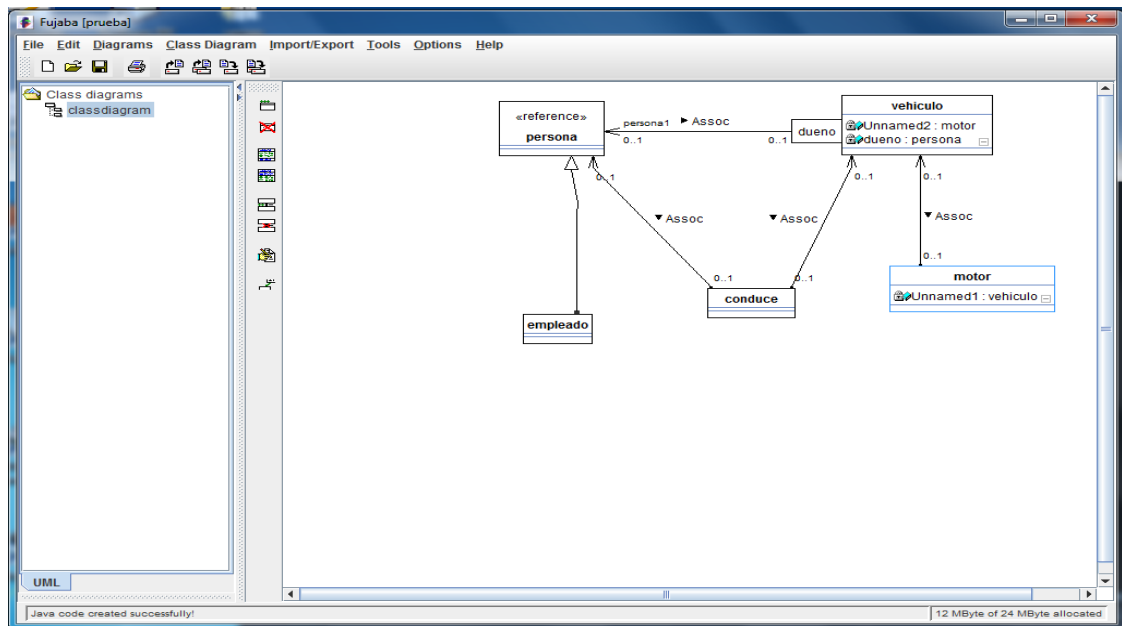
Realizado por: Fabián Medina, Orlando Cepeda

Ingeniería Inversa desde los archivos .java se regenera en los diagramas de clases.



Realizado por: Fabián Medina, Orlando Cepeda

Diagrama de Clases en Fujaba.



Realizado por: Fabián Medina, Orlando Cepeda



UNIVERSIDAD TECNICA DE COTOPAXI

UNIDAD ACADEMICA DE CIENCIAS DE LA INGENIERIA Y APLICADAS

Encuesta dirigida al equipo de análisis, desarrollo, control de calidad de software de la empresa **MERCADO LATACUNGA**

Indicaciones.

- ✓ Lea detenidamente el documento.
- ✓ Conteste las preguntas en base a su criterio profesional.

.....
.....

1.- ¿Ha utilizado usted el software libre?

SI NO

2.- ¿Conoce usted los beneficios del software libre?

SI NO

3.- ¿Utiliza herramientas libres para el desarrollo de software en su empresa?

En qué porcentaje.

%

4.- ¿Maneja usted herramientas de análisis para el desarrollo de software en su empresa?

SI NO

5.- ¿Qué metodología (as) utiliza usted para el desarrollo de software?

6.- ¿Conoce usted herramientas libres de análisis a más de las ya conocidas como Rational Rose, Power Designer?

SI NO

7.- ¿Usted implementaría herramientas libres para el análisis de desarrollo de software para su empresa?

SI NO

Porque.....

8.- ¿Ha escuchado usted sobre starUML y Fujaba?

SI NO

9.- ¿Al conocer las múltiples ventajas de las herramientas libres de análisis, que las tradicionales, usted implementaría en el desarrollo de software para su empresa?

SI

NO

Porque.....

GRACIAS POR SU COLABORACION.