

---

---

# Centro Nacional de Investigación y Desarrollo Tecnológico

Subdirección Académica

Departamento de Ciencias Computacionales

## TESIS DE MAESTRÍA EN CIENCIAS

Componentes de Acoplamiento para la Infraestructura de Soporte de  
Minería de Datos de Desarrollo de Software

presentada por

**Lic. Edgardo Pérez Luna**

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

Director de tesis  
**Dr. Moisés González García**

Cuernavaca, Morelos, México. Febrero de 2016.



Cuernavaca, Morelos a 26 de enero del 2016  
OFICIO No. DCC/032/2016

**Asunto:** Aceptación de documento de tesis

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

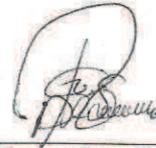
Por este conducto, los integrantes de Comité Tutorial del **C. Edgardo Pérez Luna**, con número de control M13CE059, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis profesional titulado **"Componentes de acoplamiento para la infraestructura de soporte de minería de datos de desarrollo de software"** y hemos encontrado que se han realizado todas las correcciones y observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

DIRECTOR DE TESIS



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

REVISOR 1



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

REVISOR 2



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

REVISOR 3



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

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

AMR/lmz



SEP

SECRETARÍA DE  
EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO  
Centro Nacional de Investigación y Desarrollo Tecnológico

Cuernavaca, Mor., 28 de enero de 2016  
OFICIO No. SAC/079/2016

**Asunto:** Autorización de impresión de tesis

**C. EDGARDO PÉREZ LUNA  
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS  
DE LA COMPUTACIÓN  
PRESENTE**

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado "**Componentes de acoplamiento para la infraestructura de soporte de minería de datos de desarrollo de software**", ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

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

**ATENTAMENTE**

"CONOCIMIENTO Y TECNOLOGÍA AL SERVICIO DE MÉXICO"

**DR. GERARDO VICENTE GUERRERO RAMÍREZ  
SUBDIRECTOR ACADÉMICO**

CENTRO NACIONAL DE  
INVESTIGACIÓN Y  
DESARROLLO  
TECNOLÓGICO  
SUBDIRECCIÓN  
ACADÉMICA

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

GVGR/mcr

**cenidet**<sup>®</sup>  
Centro Nacional de Investigación  
y Desarrollo Tecnológico

Interior Internado Palmira S/N, Col. Palmira. C.P. 62490 Cuernavaca, Mor.  
Tels. (01)777 362-77-70 Ext. 4106, e-mail: [direccion@cenidet.edu.mx](mailto:direccion@cenidet.edu.mx)  
[www.cenidet.edu.mx](http://www.cenidet.edu.mx)





## **Dedicatoria**

A mi familia...

A mi padre Efraín Pérez (QEPD), por su interés en siempre ver a su familia sana y feliz, por sus consejos, valores, inteligencia y buen sentido del humor, nunca te olvidaré. A mi madre Eugenia Luna, por su fortaleza, dedicación, consejos y por su coraje para afrontar adversidades. A mis hermanos, Julio, Álvaro, Teresa y Eduardo, por estar conmigo, por apoyarme en cada etapa de mi vida y por compartir risas y momentos muy felices. A mi tío Israel por todo el apoyo brindado a lo largo de mi vida, por ser como un padre para mis hermanos y para mí. A mi primo Osmar y mis sobrinos Jafet, Natalia y Estelita por hacerme sonreír.

Porque gracias a ustedes soy lo que soy, ¡Los amo!



## **Agradecimientos**

Al Consejo Nacional de Ciencia y Tecnología (CONACyT), por el apoyo económico recibido durante estos años permitiéndome así realizar mis estudios de posgrado.

Al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), por darme la oportunidad de formar parte de esta gran institución y por el apoyo brindado desde el primer día de mi estancia en este centro de investigación.

A mi director de tesis Dr. Moisés González García, por sus consejos, ayuda, orientación y paciencia brindada durante la realización de este proyecto y mi estancia en el CENIDET.

A mis revisores de tesis: Dr. René Santaolaya Salgado, Dra. Olivia G. Fragoso Díaz y M.C. Mario Guillen Rodríguez por su orientación y consejos para obtener un buen resultado de esta investigación.

Al Dr. Joaquín Pérez Ortega, por su apoyo y consejos durante estos años.

A mis compañeros y amigos de generación, Eduardo, Manuel, Sadher, Juan, Lupita, Yair, Mauricio, Bismarck, Lizbeth y Prometeo.

A las personas que de alguna u otra manera han sido parte de mi formación tanto personal como profesional.

Especialmente a mi familia, por confiar en mí y por la ayuda que siempre me han brindado.

¡Gracias!



## CONTENIDO

LISTA DE FIGURAS .....	VI
LISTA DE TABLAS.....	VIII
ABREVIATURAS.....	X
RESUMEN.....	XII
ABSTRACT .....	XIII
1. INTRODUCCIÓN.....	1
1.1. Introducción.....	3
1.2. Organización del documento.....	4
2. PROTOCOLO DE INVESTIGACIÓN .....	7
2.1. Planteamiento del problema .....	9
2.2. Objetivo de la tesis .....	9
2.3. Alcances y limitaciones.....	10
2.3.1. Alcances.....	10
2.3.2. Limitaciones.....	10
2.4. Solución propuesta.....	10
2.5. Aportaciones .....	11
2.6. Antecedentes .....	11
2.7. Trabajos relacionados.....	15
3. MARCO CONCEPTUAL.....	19
3.1. KDD y Minería de datos .....	21
3.1.1. Tareas de minería de datos .....	21
3.1.2. Metodología CRISP-DM .....	22
3.1.2.1. Fases de la metodología CRISP-DM .....	24
3.1.2.2. Fase de preparación de datos.....	25
3.1.2.3. Importancia de la preparación de datos.....	26
3.1.2.4. Etapas de la preparación de datos .....	27
3.2. Ingeniería de software .....	28
3.3. Minería de datos para ingeniería de software.....	28
3.4. Software libre .....	29
3.5. Integración de herramientas .....	30
3.6. Modularidad en Java.....	31

3.7.	OSGI.....	32
3.7.1.	Bundle .....	32
3.7.2.	Ciclo de vida.....	32
3.7.3.	Colaboración .....	33
4.	HERRAMIENTAS SELECCIONADAS PARA LA INTEGRACIÓN .....	35
4.1.	Metodología QSOS .....	37
4.2.	Herramientas seleccionadas .....	38
4.2.1.	Mantis Bug Tracker (MantisBT).....	38
4.2.2.	MySQL .....	39
4.2.3.	DotProject .....	39
4.2.4.	OpenRefine.....	40
4.2.5.	RapidMiner .....	40
4.2.6.	LibreOffice Calc .....	41
4.2.7.	LibreOffice Writer.....	42
4.2.8.	LibreOffice Impress .....	42
4.2.9.	LibreOffice Draw .....	43
5.	REQUERIMIENTOS Y DISEÑO .....	45
5.1.	Introducción.....	47
5.2.	Lenguajes utilizados en la especificación de requerimientos.....	47
5.3.	Perspectiva del producto.....	50
5.4.	Funciones del producto .....	50
5.5.	Características del usuario.....	51
5.6.	Restricciones.....	51
5.7.	Especificación de requerimientos .....	52
5.8.	Requerimientos obtenidos .....	57
5.8.1.	Requerimientos Funcionales .....	57
5.8.2.	Requerimientos no funcionales .....	57
5.9.	Diseño del ambiente AMDADS .....	57
6.	IMPLEMENTACIÓN .....	63
6.1.	Implementación.....	65
6.1.1.	Componentes de acoplamiento.....	65
6.1.2.	Interfaz de usuario.....	67
7.	PRUEBAS Y RESULTADOS.....	77
7.1.	Plan de pruebas.....	79

7.1.1.	Identificador .....	79
7.1.2.	Alcance .....	79
7.1.3.	Matriz de Trazabilidad de Pruebas.....	80
7.1.4.	Criterio para Suspensión y Reanudación de Pruebas.....	81
7.2.	Diseño de pruebas.....	81
7.2.1.	Características a Probar .....	81
7.2.2.	Estrategia de Pruebas .....	82
7.2.3.	Criterio de Aceptación del Caso de Prueba.....	83
7.3.	Casos de prueba.....	83
7.3.1.	Identificador del Caso de Prueba .....	83
7.3.2.	Objetivo.....	83
7.3.3.	Entradas.....	84
7.3.4.	Salidas.....	84
7.4.	Procedimiento de pruebas .....	85
7.4.1.	Alcance .....	86
7.4.2.	Entradas, Salidas, y requerimientos especiales.....	86
7.4.3.	Descripción del conjunto de pasos a seguir para ejecutar el caso de prueba CA-TC-02 .....	87
7.5.	Registro de pruebas .....	88
7.5.1.	Alcance .....	88
7.5.2.	Descripción .....	88
7.5.3.	Actividades y eventos de la prueba.....	88
7.5.4.	Anomalías.....	89
7.5.5.	Ejemplo de datos utilizados para la prueba.....	90
7.6.	Reporte de pruebas.....	90
7.6.1.	Alcance .....	90
7.6.2.	Detalles .....	90
7.6.2.1.	Resultados Generales.....	91
7.6.2.2.	Resultados Detallados .....	91
8.	CONCLUSIONES Y TRABAJOS FUTUROS .....	93
8.1.	Conclusiones.....	95
8.2.	Trabajos futuros .....	96
	REFERENCIAS.....	99
A.	ANEXO A COMPARACIÓN DE HERRAMIENTAS.....	105

1.	Metodología QSOS .....	107
2.	Mapa mental de criterios a evaluar.....	108
3.	Uso de plantilla .....	110
4.	Resultados.....	111
B.	ANEXO B PRUEBAS.....	115
1.	Plan de pruebas.....	117
1.1.	Introducción .....	117
1.2.	Identificador del documento.....	117
1.3.	Alcance.....	117
1.4.	Referencias.....	117
1.5.	Detalles del Plan de Pruebas de Sistema .....	118
1.6.	Administración de Pruebas .....	121
2.	Diseño de pruebas .....	123
2.1.	Introducción .....	123
2.2.	Detalles del Diseño de la Prueba.....	123
3.	Casos de prueba .....	126
3.1.	Introducción .....	126
3.2.	Caso de Prueba CA-TC-01 .....	126
3.3.	Caso de Prueba CA-TC-02.....	128
3.4.	Caso de Prueba CA-TC-03.....	130
3.5.	Caso de Prueba CA-TC-04.....	132
3.6.	Caso de Prueba CA-TC-05/06/07 .....	133
3.7.	Caso de Prueba CA-TC-08.....	134
4.	Procedimientos de prueba .....	136
4.1.	Introducción .....	136
4.2.	Procedimiento de Prueba CA-TPr-01.....	136
4.3.	Procedimiento de prueba CA-TPr-02 .....	138
4.4.	Procedimiento de Prueba CompAcop-TPr-03 .....	140
4.5.	Procedimiento de Prueba CompAcop-TPr-04 .....	142
4.6.	Procedimiento de Prueba CompAcop-TPr-05/06/07 .....	143
4.7.	Procedimiento de Prueba CompAcop-TPr-08 .....	145
5.	Registro de pruebas.....	147
5.1.	Introducción .....	147
5.2.	Registro de prueba CA-TL-01.....	147

5.3.	Registro de Prueba CA-TL-02 .....	149
5.4.	Registro de Prueba CA-TL-03 .....	151
5.5.	Registro de Prueba CA-TL-04 .....	153
5.6.	Registro de Prueba CA-TL-05/06/07 .....	154
5.7.	Registro de Prueba CA-TL-08 .....	156
6.	Reporte de pruebas.....	158
6.1.	Introducción .....	158
6.2.	Detalles .....	158

## LISTA DE FIGURAS

<b>Figura 2.1</b> Método de minería de datos de ingeniería de software. (Xie et al 2009).....	15
<b>Figura 3.1</b> La minería de datos como un paso en el descubrimiento de información. (Han et al 2006).....	21
<b>Figura 3.2</b> Niveles de abstracción CRISP-DM. (Chapman et al 2000).....	23
<b>Figura 3.3</b> Fases del modelo de referencia CRISP-DM. (Chapman, et al 2000).....	24
<b>Figura 3.4</b> Tareas genéricas (negrita) y salidas (cursiva) del modelo de referencia CRISP-DM (Chapman et al 2000).....	24
<b>Figura 3.5</b> Resumen de los datos de ingeniería de software. (Xie et al 2010).....	29
<b>Figura 3.6</b> Metodología para minería de datos de ingeniería de datos (Xie, et al 2009) ...	29
<b>Figura 3.7</b> Ciclo de vida de un Bundle .....	32
<b>Figura 3.8</b> Colaboración basada en servicios .....	33
<b>Figura 4.1</b> Herramientas propuestas para implementar QSOS (QSOS, 2014).....	38
<b>Figura 4.2</b> Logotipo MantisBT.....	39
<b>Figura 4.3</b> Logotipo MySQL .....	39
<b>Figura 4.4</b> Logotipo dotProject .....	40
<b>Figura 4.5</b> Logotipo OpenRefine .....	40
<b>Figura 4.6</b> Logotipo de RapidMiner .....	41
<b>Figura 4.7</b> Logotipo LibreOffice Calc.....	42
<b>Figura 4.8</b> Logotipo LibreOffice Writer .....	42
<b>Figura 4.9</b> Logotipo LibreOffice Impress.....	43
<b>Figura 4.10</b> Logotipo LibreOffice Draw .....	43
<b>Figura 5.1</b> Construcciones visuales de GRL herramienta JUCMNav v6.0.0 .....	48
<b>Figura 5.2</b> Construcciones visuales de UCM herramienta JUCMNav v6.0.0 .....	49
<b>Figura 5.3</b> GRL y UCM como una pieza faltante del rompecabezas UML.....	49
<b>Figura 5.4</b> Metodología MIDADS .....	51
<b>Figura 5.5</b> Diagrama GRL de los componentes de acoplamiento del ambiente .....	52
<b>Figura 5.6</b> Diagrama UCM de la meta funcional de integrar usuarios .....	54
<b>Figura 5.7</b> Diagrama UCM de la meta de integrar proyectos .....	55
<b>Figura 5.8</b> Diagrama UCM de la meta funcional de integrar tareas .....	56
<b>Figura 5.9</b> Diagrama UCM de la meta de integrar archivos .....	56
<b>Figura 5.10</b> Diagrama de componentes.....	59
<b>Figura 5.11</b> Diagrama de secuencia de la meta integrar proyectos .....	60
<b>Figura 5.12</b> Diagrama de clases de AMDADS.....	61
<b>Figura 6.1</b> Modelo de integración de AMDADS.....	66
<b>Figura 6.2</b> Diagrama GRL de la interfaz de usuario.....	68
<b>Figura 6.3</b> Diagrama UCM de la interfaz de usuario.....	69
<b>Figura 6.4</b> Diagrama de interacción de la interfaz de usuario.....	70
<b>Figura 6.5</b> Página de acceso .....	71
<b>Figura 6.6</b> Mensaje de bienvenida.....	72

<b>Figura 6.7</b> Menú principal de la interfaz.....	72
<b>Figura 6.8</b> Interfaz de proyectos asociados al usuario .....	73
<b>Figura 6.9</b> Información sobre metodología MIDADS .....	73
<b>Figura 6.10</b> Menú herramientas .....	74
<b>Figura 6.11</b> Herramientas que cubren y ayudan en la preparación de datos .....	75
<b>Figura 6.12</b> Enlaces a tutoriales en la sección Ayuda .....	75
<b>Figura A.1</b> Herramientas propuestas para implementar QSOS.....	107
<b>Figura A.2</b> Mapa mental de los criterios a evaluar.....	108
<b>Figura A.3</b> Pesos para los criterios relacionados a la gobernabilidad .....	108
<b>Figura A.4</b> Pesos para los criterios relacionados a la actividad.....	109
<b>Figura A.5</b> Pesos para los criterios relacionados a la industrialización .....	109
<b>Figura A.6</b> Pesos para los criterios de algunas características de las herramientas .....	110
<b>Figura A.7</b> Interfaz XulEditor para cargar la plantilla .....	110
<b>Figura A.8</b> Lista de criterios para seleccionar valores.....	111

## LISTA DE TABLAS

<b>Tabla 2.1</b>	Resumen de antecedentes.....	14
<b>Tabla 7.1</b>	Identificador del documento.....	79
<b>Tabla 7.2</b>	Requerimiento y casos de prueba correspondiente .....	80
<b>Tabla 7.3</b>	Interfaz y caso de prueba correspondiente .....	80
<b>Tabla 7.4</b>	Matriz de trazabilidad.....	81
<b>Tabla 7.5</b>	Matriz de trazabilidad para la interfaz .....	82
<b>Tabla 7.6</b>	Caso de prueba Integrar Proyectos.....	83
<b>Tabla 7.7</b>	Datos de entrada para dotProject.....	84
<b>Tabla 7.8</b>	Salida en herramienta MantisBT.....	84
<b>Tabla 7.9</b>	Salida en herramienta OpenRefine .....	85
<b>Tabla 7.10</b>	Salida en herramienta RapidMiner .....	85
<b>Tabla 7.11</b>	Tabla de relaciones de proyectos integrados .....	85
<b>Tabla 7.12</b>	Registro de eventos para CA-TPr-02.....	88
<b>Tabla 7.13</b>	Anomalías de CA-TPr-02.....	89
<b>Tabla 7.14</b>	Ejemplo de datos utilizados para CA-TPr-02.....	90
<b>Tabla 7.15</b>	Resultados generales de las pruebas .....	91
<b>Tabla A.1</b>	Comparación de herramienta .....	112
<b>Tabla A.2</b>	Comparación de herramientas .....	112
<b>Tabla A.3</b>	Comparación de herramientas .....	112
<b>Tabla A.4</b>	Comparación de herramientas .....	113
<b>Tabla B.1</b>	Nomenclatura de documentos.....	117
<b>Tabla B.2</b>	Matriz de trazabilidad de pruebas.....	120
<b>Tabla B.3</b>	Características a probar Interfaz.....	121
<b>Tabla B.4</b>	Matriz de trazabilidad de pruebas.....	123
<b>Tabla B.5</b>	Características a probar Interfaz.....	125
<b>Tabla B.6</b>	Caso de prueba 01 .....	126
<b>Tabla B.7</b>	Entradas a dotProject CA-TC-01 .....	127
<b>Tabla B.8</b>	Salida MantisBT CA-TC-01.....	127
<b>Tabla B.9</b>	Salida en tabla relaciones CA-TC-01 .....	127
<b>Tabla B.10</b>	Caso de prueba 02.....	128
<b>Tabla B.11</b>	Entrada en dotProject CA-TC-02 .....	128
<b>Tabla B.12</b>	Salidas en MantisBT CA-TC-02 .....	129
<b>Tabla B.13</b>	Salidas en OpenRefine CA-TC-02.....	129
<b>Tabla B.14</b>	Salida en RapidMiner CA-TC-02 .....	129
<b>Tabla B.15</b>	Salida en tabla relaciones CA-TC-02.....	130
<b>Tabla B.16</b>	Caso de prueba 03.....	130
<b>Tabla B.17</b>	Entrada en dotProject CA-TC-03 .....	131
<b>Tabla B.18</b>	Salida en dotProject CA-TC-03.....	131
<b>Tabla B.19</b>	Salida en tabla relaciones CA-TC-03.....	132
<b>Tabla B.20</b>	Caso de prueba 04.....	132

<b>Tabla B.21</b>	Entrada en MantisBT o dotproject CA-TC-04 .....	132
<b>Tabla B.22</b>	Salida en dotProject o MantisBT CA-TC-04.....	133
<b>Tabla B.23</b>	Salida en tabla relaciones CA-TC-04.....	133
<b>Tabla B.24</b>	Caso de prueba 05-06-07 .....	134
<b>Tabla B.25</b>	Caso de prueba 08.....	134
<b>Tabla B.26</b>	Entradas en interfaz.....	135
<b>Tabla B.27</b>	Registro de eventos y actividades CA-TPr-01 .....	147
<b>Tabla B.28</b>	Anomalías CA-TPr-01.....	148
<b>Tabla B.29</b>	Ejemplo de datos utilizados CA-TPr-01 .....	148
<b>Tabla B.30</b>	Registro de eventos y actividades CA-TPr-02.....	149
<b>Tabla B.31</b>	Anomalías CA-TPr-02.....	150
<b>Tabla B.32</b>	Ejemplo de datos utilizados CA-TPr-02.....	151
<b>Tabla B.33</b>	Registro de eventos y actividades CA-TPr-03.....	151
<b>Tabla B.34</b>	Anomalías CA-TPr-03.....	152
<b>Tabla B.35</b>	Ejemplo de datos utilizados CA-TPr-03.....	152
<b>Tabla B.36</b>	Registro de eventos y actividades CA-TPr-04.....	153
<b>Tabla B.37</b>	Anomalías CA-TPr-04.....	154
<b>Tabla B.38</b>	Ejemplo de datos utilizados CA-TPr-04.....	154
<b>Tabla B.39</b>	Registro de eventos y actividades CA-TPr-05-06-07 .....	155
<b>Tabla B.40</b>	Anomalías CA-TPr-05-06-07 .....	156
<b>Tabla B.41</b>	Ejemplo de datos utilizados CA-TPr-05-06-07 .....	156
<b>Tabla B.42</b>	Registro de eventos y actividades CA-TPr-08.....	156
<b>Tabla B.43</b>	Anomalías CA-TPr-08.....	157
<b>Tabla B.44</b>	Ejemplo de datos utilizados CA-TPr-08.....	157
<b>Tabla B.45</b>	Resultados generales .....	158

## ABREVIATURAS

AGD	<i>Architectural and Group Development</i> / Desarrollo Arquitectónico en Grupo
API	Application Programming Interface / Interfaz de Programación de Aplicaciones
AMDADS	Ambiente de Soporte de Minería de Datos de Desarrollo de Software
CENIDET	Centro Nacional de Investigación y Desarrollo Tecnológico
CINVESTAV	Centro de Investigación y de Estudios Avanzados
CRISP-DM	<i>Cross Industry Standard Process for Data Mining</i> / Proceso Estándar de Industria Cruzada para Minería de Datos
FLOSS	Free - <i>Libre and Open Source Software</i> / Software Libre y de Código Abierto
GPL	<i>General Public License</i> / Licencia Publica General
GRL	<i>Goal-oriented Requirements Language</i> / Lenguaje de Requerimientos Orientados a Metas
IEEE	<i>Institute of Electrical and Electronics Engineers</i> / Instituto de Ingenieros Eléctricos y Electrónicos
IS	Ingeniería de Software
JAR	<i>Java Archive</i> / Archivo Java
KDD	<i>Knowledge Discovery and Data Bases</i> / Descubrimiento de Conocimiento en Bases de Datos
MIDADS	Metodología de Minería de Datos de Desarrollo de Software
OMM	<i>OpenSource Maturity Model</i> / Modelo de Madurez de Código Abierto
OpenBRR	<i>Open Business Readiness Rating</i> / Calificación de Disposición de Negocio Abierto
OSGI	<i>Open Services Gateway initiative</i> / Iniciativa de Enlace de Servicios Abiertos
QSOS	<i>Qualification and Selection of Open Source software</i> / Calificación y Selección de Software de Código Abierto
SRS	<i>Software Requirements Specification</i> / Especificación de

Requerimientos de Software

SDS *Software Design Specification* / Especificación de Diseño de Software

UCM *Use Case Maps* / Mapas de Casos de Uso

UML *Unified Modeling Language* / Lenguaje de Modelado Unificado

URN *User Requirements Notation* / Notación de Requerimientos de Usuario

## RESUMEN

La aplicación del proceso de minería de datos ha dado buenos resultados, esto hace que cada vez sean más los campos en los que se aplica dicho proceso. Uno de los campos de creciente aplicación de la minería de datos es en la ingeniería de software. Los datos generados durante el proceso de desarrollo de software son una fuente potencial para la obtención de conocimiento, que aplicado a proyectos futuros genere mejores productos y procesos de software.

Existen herramientas de software que soportan cada una de las fases del proceso de minería de datos y que trabajan de forma aislada cubriendo las necesidades de alguna de sus fases, esto ocasiona que algunas tareas tengan que realizarse más de una vez (ej. registrar un nuevo proyecto). En este trabajo de investigación no se encontró algún tipo de integración de herramientas libres y de código abierto FLOSS que soporten el proceso de minería de datos.

Debido a esta carencia, este trabajo de investigación se enfoca en la creación de *componentes de acoplamiento* desarrollados siguiendo las especificaciones OSGI, que integren herramientas FLOSS en un ambiente que soporte el proceso de minería de datos de desarrollo de software (que denominamos AMDADS), con el fin de evitar la captura repetitiva de los mismos datos en diferentes herramientas y disminuir la carga de trabajo al usuario. AMDADS está enfocado principalmente en la etapa de preparación de datos, que es la etapa que implica mayor atención porque se ocupa el mayor esfuerzo en ella.

Las pruebas, del acoplamiento de las herramientas, se realizaron usando datos ficticios sobre los tipos de registros a integrar (usuarios, proyectos, incidencias/tareas y archivos), al final se inspeccionó visualmente la creación de los registros en las demás herramientas, además se validó la creación de un repositorio que incluye el id de los registros relacionados.

La implementación de AMDADS arrojó resultados satisfactorios, por lo que se sugiere continuar en esta línea de investigación ampliando sus funcionalidades.

## ABSTRACT

Data mining process application has been given great results, so the data mining application is increasing to other fields. One of them is software engineering. Generated data during development process are a potential source of knowledge, applied to future projects generates better software processes and products.

There are software tools that support every phase of the data mining process, working in isolation, and covering the needs from some process phase, it causes that some tasks need to be done more than once (i.e. new project registration). As part of this research, some type of FLOSS tool integration that supports data mining process has not been detected.

This research is focused on the creation of *coupling components* using OSGI specification, which integrate FLOSS tools in an environment that supports software development data mining (SDDME), avoiding repetitive typed of the same data in every tool, and taking some work away to user. SDDME is mainly focused on data preparation phase, being the one that requires more attention due its important.

Tests were applied using fictitious data from records that will be integrated (users, projects, issues/tasks and files), at the end, a visual inspection was done for validating the records creation. Also a validation was applied for the creation of a related repository that includes the id from related records.

SDDME implementation gives satisfactory results, due to this, is suggested continuing with this research line and adding more functionality to SDDME.



En este capítulo se presenta la introducción y una descripción de forma breve de la organización de la tesis.



## 1.1. Introducción

En la actualidad el software forma una parte importante en la vida diaria de las personas, debido a esto, el desarrollo de software de calidad es una preocupación para los ingenieros de software.

Durante el proceso de desarrollo de software se generan datos en cada una de las etapas que conforman este proceso, el estudio de los datos de cada proyecto de desarrollo es de gran importancia, ya que a partir de ello se puede generar conocimiento que es aplicado a futuros proyectos con el fin de mejorar la calidad de los procesos y los productos de software. Para generar este conocimiento se usan diferentes técnicas, una de las más usadas es la minería de datos.

La aplicación del proceso de minería de datos se realiza en diferentes etapas, la más importante, que requiere mayor atención y en la que se invierte más tiempo es la preparación de datos. Además a lo largo del proceso de minería de datos se realizan tareas repetitivas que aumentan el esfuerzo y el tiempo invertido en este proceso. Es por esta razón que se decidió desarrollar un ambiente en donde se elimine la repetición de ciertas tareas (ej. El mismo proyecto se tiene que registrar en cada una de las herramientas).

En las investigaciones realizadas como parte de este trabajo no se ha encontrado algún tipo de integración de herramientas libres y de código abierto que soporten el proceso de minería de datos. Por esta razón se diseñó y construyó un ambiente de minería de datos de desarrollo de software (que llamamos AMDADS), basado en el diseño de alto nivel de la tesis “Especificación de Requerimientos para un Ambiente de Minería Soporte al Proceso de Minería de Datos Aplicado a Repositorios de Datos de Desarrollo de Software” desarrollada en el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) por la M.C. María del Rosario Meza Bazán.

AMDADS está integrado por diferentes herramientas que soportan las necesidades de algunas de las etapas del proceso de minería de datos: dotProject, MantisBT, OpenRefine, RapidMiner y se ponen a disposición las herramientas incluidas en el paquete de ofimática de LibreOffice. Principalmente se atiende las etapas iniciales y la etapa de preparación de datos. La intención del ambiente es reducir la repetición de

tareas repetitivas por parte del usuario, a través de la implementación de *componentes de acoplamiento* que se encargan de duplicar automáticamente los datos que se repiten en las herramientas.

Para facilitar la creación de las actividades a realizar en el proceso de minería, se incluye el uso de plantillas para la creación de proyectos basados: 1) en el modelo CRISP-DM (por sus siglas en inglés *Cross Industry Standard Process for Data Mining*), que es una metodología general, aplicable a diferentes dominios y 2) en MIDADS (Metodología para Minería de Datos de Desarrollo de Software) que es una metodología híbrida en la que se integran aspectos relacionados a la ingeniería de software y a la preparación de datos.

## **1.2. Organización del documento**

Este documento se organiza de la siguiente manera:

Capítulo 2. Protocolo de investigación. En este capítulo se presenta el planteamiento del problema que dio origen a esta investigación, así como el objetivo de la tesis, alcances y limitaciones, solución propuesta, las aportaciones, así como los antecedentes y trabajos relacionados a esta investigación.

Capítulo 3. Marco conceptual. En este capítulo se describen los conceptos ligados a esta investigación que corresponden a campos como son: descubrimiento de conocimiento en bases de datos, ingeniería de software, modularidad en Java y la plataforma OSGI. Esto con el fin de facilitar el entendimiento de esta investigación.

Capítulo 4. Herramientas seleccionadas. En este capítulo se describen las herramientas seleccionadas para formar parte del ambiente de soporte a minería de datos de desarrollo de software y la forma en la estas herramientas se eligieron.

Capítulo 5. Requerimientos y diseño. Como parte de la investigación se especificaron los requerimientos para la construcción del ambiente, así como su diseño. En este capítulo se describen estas actividades.

Capítulo 6. Implementación. En este capítulo se describe de forma general la forma en la que se construyó el ambiente de soporte a minería de datos de desarrollo de software.

Capítulo 7. Pruebas y resultados. Este capítulo contiene las pruebas hechas al ambiente y los resultados obtenidos a partir de ellas.

Capítulo 8. Conclusiones y trabajos futuros. En este capítulo se incluyen las conclusiones a las que se llegaron después de haber realizado este trabajo, las aportaciones y los trabajos futuros que permitan la continuación de esta investigación.



En este capítulo se presenta el protocolo de investigación lo que incluye: el planteamiento del problema que dio origen a esta investigación, así como el objetivo de la tesis, los alcances y limitaciones, la solución propuesta, las aportaciones, así como los antecedentes y trabajos relacionados a esta investigación.



En este capítulo se abordan temas que describen la investigación, se inicia con la descripción o planteamiento del problema que dio inicio a esta investigación, posteriormente se describe el objetivo de la tesis, a continuación se mencionan los alcances y las limitaciones de esta investigación, la solución propuesta y las aportaciones. Por último se describen los antecedentes y trabajos relacionados a este tema de investigación.

### **2.1. Planteamiento del problema**

Debido a que el software juega un papel importante en nuestras actividades cotidianas, mejorar la productividad y su calidad es un objetivo importante para los ingenieros de software. La minería de datos de Ingeniería de Software es un medio prometedor para cumplir ese objetivo (Xie et al. 2009).

La aplicación del proceso de minería de datos implica el uso de diferentes herramientas que generan archivos con formatos incompatibles, también se realizan actividades o tareas repetitivas que generan trabajo extra.

Además, para mejorar el proceso se hace necesaria la planificación de proyectos y el seguimiento de incidencias con el fin de monitorearlos hasta que se le dé alguna solución.

Debido a esto se tiene la necesidad de desarrollar componentes de software que permitan la integración de herramientas de software para evitar la incompatibilidad de las herramientas y las tareas repetitivas, así como proporcionar una forma de gestionar el proyecto y dar seguimiento a las incidencias encontradas.

### **2.2. Objetivo de la tesis**

Integrar herramientas mediante el uso de sus bases de datos, con el fin de evitar la captura de datos repetitiva y eliminar la incompatibilidad entre formatos de entrada y salida.

## 2.3. Alcances y limitaciones

### 2.3.1. Alcances

- Selección de herramientas libres utilizando la metodología QSOS (*Qualification and Selection of Open Source Software*)
- Especificar los requerimientos de los elementos de acoplamiento
- Realizar el diseño detallado de los elementos de acoplamiento
- Desarrollo de los componentes necesarios para la integración
- Realizar pruebas para validar el funcionamiento del ambiente

### 2.3.2. Limitaciones

- Las herramientas y repositorios a utilizar son únicamente los disponibles de forma libre
- La fase del proceso de minería de datos a abarcar en este trabajo será la fase de preparación de datos
- Se abarca únicamente el acoplamiento de las herramientas usando los datos generados en el registro de algún elemento (usuario, proyecto, incidencia-tarea o archivo)

## 2.4. Solución propuesta

Con el fin de resolver la problemática planteada anteriormente, se propone desarrollar componentes (módulos) Java que monitoreen cada cierto tiempo la base de datos en espera de la inserción de un nuevo registro para posteriormente replicar dicho registro en las demás herramientas. Los módulos reciben el nombre de *componentes de acoplamiento*, se propone desarrollarlos usando la plataforma OSGI para aprovechar las ventajas que proporciona su implementación. Habrá un componente estará encargado para cada uno de los tipos de registro que se manejarán (usuarios, proyectos, incidencias/tareas o archivos).

Cada componente de acoplamiento extraerá los datos de la base de datos de una herramienta y si es conveniente los adaptará a las características necesarias de la base de datos de la herramienta en donde se realizará la réplica. Al realizarse esta

replica el componente extraerá el id del registro replicado, con el fin de almacenarlo en un *repositorio de relaciones* para mantener enlazados los registros equivalentes.

Para validar la funcionalidad de los componentes se realizarán pruebas que verifiquen que un nuevo registro sea replicado en las demás herramientas y mantenga sus relaciones correspondientes, estas pruebas serán con registros sintéticos.

## 2.5. Aportaciones

Las aportaciones de esta investigación son:

La implementación exitosa de AMDADS

El uso de la metodología de minería de datos de desarrollo de software (MIDADS) para la creación de plantillas de proyectos

La implementación de las especificaciones OSGI para el desarrollo de los componentes de acoplamiento

La integración heterogénea de herramientas, ya que durante el proceso de desarrollo de AMDADS se hizo uso de dos tipos de integración, por datos y API's

## 2.6. Antecedentes

En seguida se describen de manera general los trabajos de investigación hechos por profesores y/o alumnos del CENIDET, que preceden a esta propuesta de tesis.

**Documento de Especificación de Requerimientos de Software: para un Ambiente de Soporte al Proceso de Minería de Datos Aplicado a Repositorios de Datos de Desarrollo de Software**, V0.1, (Meza, 2013-a). Desarrollado en la investigación de tesis de maestría en Ciencias de la Computación, por María del Rosario Meza Bazán en el Centro de Investigación y Desarrollo Tecnológico, de abril del 2013. En este trabajo se especifican los requerimientos de un ambiente que soporte al proceso de minería de datos aplicado a repositorios de software, el cual sirve como un punto de partida para la realización de este trabajo de tesis.

**Documento de Diseño de Software: para un Ambiente de Soporte al Proceso de Minería de Datos Aplicado a Repositorios de Datos de Desarrollo de Software**, V0.1, (Meza, 2013-b). Desarrollado en la investigación de tesis de maestría en Ciencias de la Computación, por María del Rosario Meza Bazán en el Centro de Investigación y Desarrollo Tecnológico, de abril del 2013. En este trabajo se reporta el diseño de alto nivel de un ambiente que soporte al proceso de minería de datos aplicado a repositorios de software, el cual sirve como punto de partida para la realización de este trabajo de tesis.

Tesis de maestría **“Metodología de Preparación de Datos Orientada a aplicaciones de Epidemiología Basada en el Modelo CRISP-DM”**, (Iturbide, 2013), desarrollada por Gregorio Emmanuel Iturbide Domínguez en el Centro Nacional de Investigación y Desarrollo Tecnológico, concluida en Febrero del 2013. En donde presenta una metodología para la fase de preparación de datos, con un nivel de detalle mayor al propuesto en la metodología CRISP-DM. Establece la división en la fase de preparación de datos, en preparación general enfocada al proceso y preparación específica enfocada al objetivo del proceso de minería de datos. La metodología presentada se validó mediante una aplicación en el área de epidemiología, con resultados satisfactorios.

Tesis de maestría **“Evaluación de técnicas de Comparación de Diferentes Grupos de Características de Proyectos de Software”**, (Sánchez, 2010), desarrollada por la M.C. Miriam Sánchez Santamaria en el Centro nacional de Investigación y Desarrollo Tecnológico, concluida en Enero del 2010. En esta investigación se propone un procedimiento, que se utilizará en un trabajo futuro de investigación, para llevar a cabo las comparaciones de proyectos de software, utilizando técnicas de minería de datos. Por esta razón se realizó un estudio, análisis y evaluación teórica de diferentes técnicas de minería de datos, para determinar cuáles son las adecuadas para comparar los proyectos.

Tesis doctoral **“Método de Desarrollo Arquitectónico en Grupo”**, (González 2006), desarrollada por el Dr. Moisés González García, en el Centro de Investigaciones y

Estudios Avanzados (CINVESTAV) del Instituto Politécnico Nacional, en la Sección de Computación del Departamento de Ingeniería Eléctrica, concluida en Agosto de 2006. En esta investigación se propone la construcción del ambiente de soporte AGDE, que tiene como uno de sus componentes el de Adecuación del Desarrollo de Software, que especifica usar la minería de datos, de repositorios de datos acerca de desarrollos, previos, de software. Esto, con la finalidad de estimar y proponer los atributos del método de desarrollo a utilizar, en un proyecto que se realizará.

Tesis de maestría “**Caracterización de Proyectos de Software para Configurar su Desarrollo y Habilitar la Comparación entre Casos Almacenados en la Memoria Organizacional**”, (Delgado 2008), realizada por la M.C. Cindy Delgado Solís en el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), finalizada en Enero del 2008. Este trabajo constituye el primer resultado en la construcción de la herramienta de adecuación, incluida en el método AGD (antecedente 4), la cual tiene como objetivo adaptar aspectos presentes en el desarrollo de un proyecto considerando las características particulares del proyecto a desarrollar. Además se presenta un marco para la caracterización de proyectos de software, constituido por un conjunto de factores y características establecidas en base a trabajos relacionados, a la literatura sobre el producto y el proceso de software y a la recolección de información histórica de dos organizaciones.

Tesis de maestría “**Componentes de acoplamiento para la infraestructura de un ambiente libre para desarrollo y mejora incremental de productos de software**” (Correa, 2015), realizada por la M.C. Manuel José Correa Delval en el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), finalizada en febrero del 2015.

En este trabajo se desarrolla un ambiente integrado usando la integración por datos, el fin de este ambiente es dar soporte al proceso de desarrollo y mejora incremental de productos de software. La forma de solución es la creación de componentes de acoplamiento desarrollados con el lenguaje de programación Java y siguiendo las especificaciones OSGI, teniendo resultados exitosos. A partir de este trabajo se

obtuvo la idea de la creación de un repositorio de relaciones para mantener enlazados los registros entre las herramientas.

**Tabla 2.1** Resumen de antecedentes

Trabajo	Objetivo	Producto	Técnicas de MD	Herramientas de MD	Metodología
<b>González, 2006</b>	Crear un método de elaboración, asociando un proceso de grupo a cada uno de los productos-del-trabajo	Método AGD	*		*
<b>Delgado, 2008</b>	Caracterizar proyectos y el diseño de una estructura de almacenamiento de información relativa al desarrollo de proyectos pasados	Marco para caracterizar proyectos de software	*	*	*
<b>Sánchez, 2010</b>	Evaluar técnicas de comparación de proyectos	Procedimiento para evaluar técnicas de MD Propuesta de técnicas de MD para comparar proyectos de software	*	*	*
<b>Iturbide, 2013</b>	Contribuir a la sistematización del proceso de MD en el dominio de la salud, particularmente en la etapa de preparación de datos	Metodología de preparación de datos	*	*	*

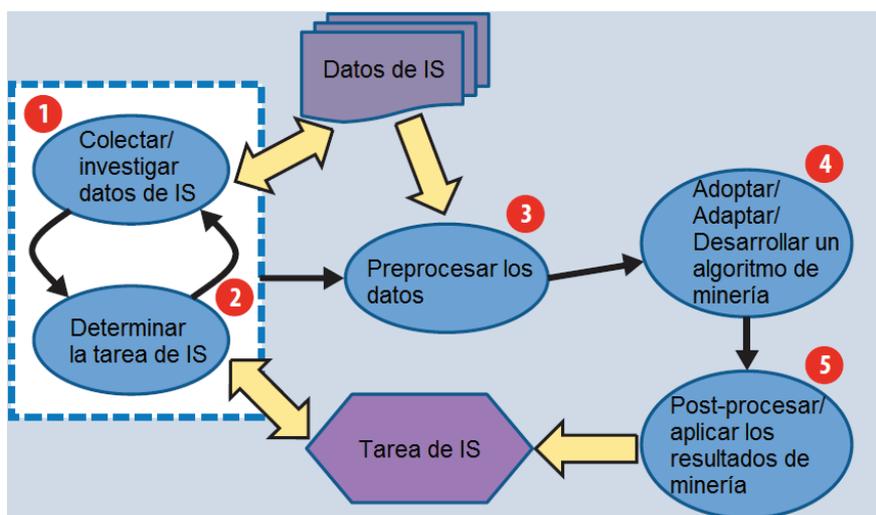
<b>Correa, 2015</b>	Integrar herramientas para dar soporte al proceso de desarrollo y de mejora incremental de productos de software	Ambiente de soporte al proceso de desarrollo y de mejora incremental de productos de software
---------------------	--	---

### 2.7. Trabajos relacionados

#### ***“Data mining for software engineering” (Xie, et al 2009)***

En este trabajo, se explica que, para mejorar la productividad y la calidad del software, los ingenieros de software están aplicando la minería de datos a diferentes tareas de Ingeniería de Software (IS). Se presentan ejemplos de datos de IS generados por Personas, Procesos y Productos, algoritmos de minería de datos aplicados a estos datos y tareas de la IS a las que apoya la minería de datos. También muestra un método para minar datos de IS, que es la principal aportación a este trabajo de investigación en el cual no habla sobre herramientas con las cuales se pueda trabajar la minería.

En la metodología para minar datos de IS los dos primeros pasos (Figura 2.1) los muestra delimitados por un alineo punteado ya que en la práctica en la ingeniería de software adopta una mezcla entre ambos. El paso 3 es el procesamiento de los datos de IS, el paso 4 es la aplicación del algoritmo de minería y por último aplicar los resultados de minería en alguna de las tareas de IS.



**Figura 2.1** Método de minería de datos de ingeniería de software. (Xie et al 2009)

**“Comparing OpenBRR, QSOS, and OMM Assessment Models” (Petrinja, et al 2010)**

El objetivo de este estudio fue investigar la calidad y usabilidad de tres modelos de evaluación de software libre de código abierto por sus siglas en inglés FLOSS: el modelo **OpenBRR** (*Open Business Readiness Rating*), el **QSOS** (*Qualification and Selection of Open Source Software*) y **OMM** (*QualiPSo OpenSource Maturity Model*). El estudio identificó los aspectos negativos y positivos de cada uno de ellos. Los modelos fueron usados para evaluar dos proyectos libres de código abierto Firefox y Chrome (*Chromium*). Los resultados revelaron que los tres modelos proveen evaluaciones comparables para los dos proyectos evaluados. La conclusión principal fue que los tres modelos contienen algunas respuestas propuestas y preguntas que no son claras para los asesores, por lo tanto debían reescribirse o explicarse mejor. La información de este trabajo de investigación nos ayudó a seleccionar la metodología correcta para comparar herramientas FLOSS a integrar en el ambiente de soporte a minería de datos propuesto.

**“Data mining in software engineering” (Halkidi, et al 2011)**

Este trabajo explica cómo el aumento en la disponibilidad de datos como parte del proceso de desarrollo de software nos permite aplicar técnicas de análisis novedosas a los datos y usar los resultados para guiar la optimización del proceso. En este artículo se describen varias fuentes de datos y se discuten los principios y las técnicas de minería de datos de ingeniería de software (IS). Hay muchos retos que surgen en la minería de repositorios de software: la complejidad de los datos, datos a gran escala, etc. Las fuentes de datos a las que ha sido aplicada la minería de datos son: la documentación, datos de la administración de la configuración del software, el código fuente, código compilado y rastros de ejecución y las listas de correos (*mailing lists*).

Los autores nos aportan una descripción de los datos y los retos a los que podemos enfrentarnos al aplicar la minería de datos de la IS. Esta información nos ayudó para adaptar la integración del ambiente de soporte a minería de datos respecto a los datos de ingeniería de software

***“Software Intelligence: The Future of Mining Software Engineering Data” (Xie, et al 2010)***

En este trabajo se propone la Inteligencia de Software como el futuro de la minería de datos de ingeniería de software debido a la dirección exitosa en la que se dirige. Se acuña el nombre Inteligencia de Software como una inspiración del campo de la inteligencia de negocios, que ofrece técnicas y conceptos para mejorar las decisiones relativas al negocio. La inteligencia de software debe apoyar el proceso de toma de decisiones a través del tiempo de vida del software no sólo durante la fase de desarrollo. Esta investigación proporciona argumentos para el trabajo que se está realizando como parte de esta propuesta.

Los antecedentes y trabajos relacionados aclaran el panorama de esta investigación proporcionando ideas y manteniendo la línea de la investigación.

En este capítulo se incluyeron investigaciones que tienen aportaciones y que argumentan esta investigación. Se mencionan los trabajos que forman parte de la línea de investigación en la que está situada esta tesis.

En el siguiente capítulo se definen los conceptos relacionados al tema de investigación con el fin de tener una mejor comprensión del tema.



En este capítulo se presenta los conceptos acerca del *Knowledge Discovery in Databases* (KDD) y Minería de datos; tareas de minería de datos; metodología CRISP-DM; ingeniería de software; minería de datos para ingeniería de software; integración de herramientas; modularidad en Java, y el *Open Services Gateway Initiative* (OSGI).



A continuación se describen los conceptos que ayudarán en la comprensión del tema de investigación.

### 3.1. KDD y Minería de datos

KDD (*Knowledge Discovery in Databases*) descubrimiento de conocimiento en bases de datos, se define comúnmente como “el proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y entendibles en los datos”, (Ohrn, 1999).

Minería de datos es el descubrimiento semi-automático de patrones, asociaciones, cambios, anomalías y estructuras estadísticamente significantes y eventos en los datos, (Grossman, 1998).

La minería de datos es una etapa del proceso KDD que se encarga de aplicar algoritmos a los datos con la intención de descubrir patrones de comportamiento en dichos datos (Figura 3.1).

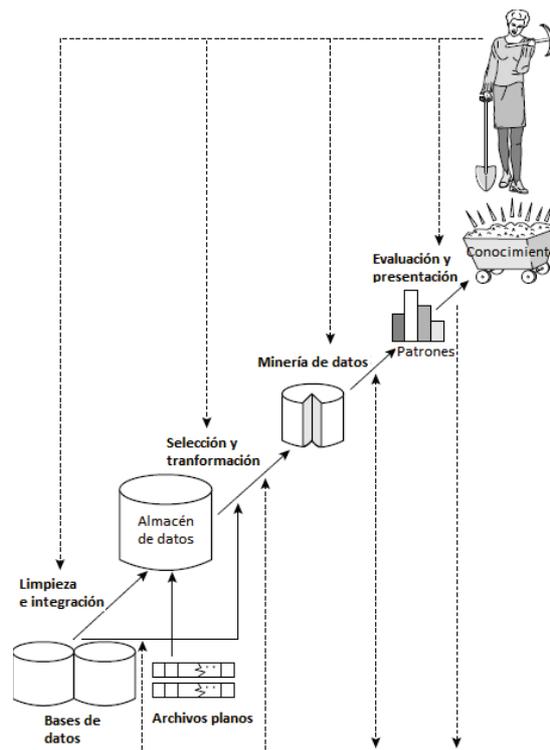


Figura 3.1 La minería de datos como un paso en el descubrimiento de información. (Han et al 2006)

#### 3.1.1. Tareas de minería de datos

Según (Riquelme, et al 2006) entre las tareas de minería de datos más comunes se encuentran las que se describen en seguida, acompañadas de un breve ejemplo:

- **Clasificación:** en esta tarea se clasifica un dato dentro de una de las clases categóricas predefinidas. Responde a preguntas tales como, ¿Cuál es el riesgo de conceder un crédito a este cliente? ¿Dado este nuevo paciente qué estado de la enfermedad indican sus análisis?
- **Regresión:** el propósito de esta tarea es usar el modelo de regresión para hacer corresponder un dato con un valor real de una variable. Responde a cuestiones como ¿Cuál es la previsión de ventas para el mes que viene? ¿De qué depende?
- **Agrupamiento (*Clustering*):** se refiere a la agrupación de registros, observaciones, o casos en clases de objetos similares. Un agrupamiento (clúster) es una colección de registros que son similares entre sí, y distintos a los registros de otro clúster. ¿Cuántos tipos de clientes vienen a mi negocio? ¿Qué perfiles de necesidades se dan en un cierto grupo de pacientes?
- **Generación de reglas:** en esta tarea se extraen o generan reglas de los datos. Estas reglas hacen referencia al descubrimiento de relaciones de asociación y dependencias funcionales entre los diferentes atributos. ¿Cuánto debe valer este indicador en sangre para que un paciente se considere grave? ¿Si un cliente de un hipermercado compra pañales también compra cerveza?
- **Resumen o sumarización:** en esta tarea se usan modelos que proporcionan una descripción compacta de un subconjunto de datos. ¿Cuáles son las principales características de mis clientes?
- **Análisis de secuencias:** en esta tarea se modelan patrones secuenciales, como análisis de series temporales, secuencias de genes, etc. El objetivo es modelar los estados del proceso, o extraer e informar de la desviación y tendencias en el tiempo. ¿El consumo de energía eléctrica de este mes es similar al del año pasado? Dados los niveles de contaminación atmosférica de la última semana cuál es la previsión para las próximas 24 horas.

### 3.1.2. Metodología CRISP-DM

La metodología de minería de datos CRISP-DM (*Cross Industry Standard Process for Data Mining*) es la más usada según kdnuggets (kdnuggets.com, 2014). CRISP-DM se describe en términos de un modelo de proceso jerárquico, consiste en un conjunto de tareas descritas en cuatro niveles de abstracción (del general al específico): fase,

tarea genérica, tarea especializada e instancia del proceso (Figura 3.2). El nivel superior el proceso de minería de datos está organizado en un número de fases; cada fase consiste en diferentes tareas genéricas del segundo nivel y son llamadas genéricas porque su intención es cubrir todas las posibles situaciones de minería. El tercer nivel es de tareas especializadas donde se describe cómo, las acciones genéricas, se llevarán a cabo, en ciertas situaciones específicas. El cuarto nivel es un registro de las acciones, decisiones y resultados de la tarea de minería de datos.

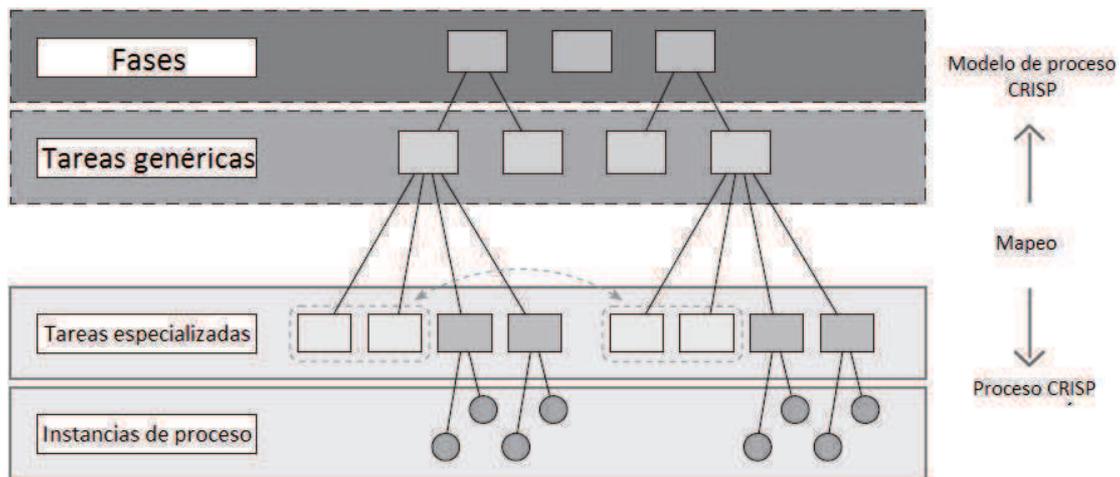


Figura 3.2 Niveles de abstracción CRISP-DM. (Chapman et al 2000)

El modelo de proceso común, para minería de datos, provee una visión del ciclo de vida de un proyecto de minería de datos. Contiene las fases de un proyecto, sus respectivas tareas y la relación entre ellas. El proceso consiste en 6 etapas (figura 3.3), la salida de cada etapa determina qué fase o tarea específica de la fase tiene que ser realizada a continuación. El círculo externo simboliza la naturaleza cíclica de la misma minería de datos, Genéricas.

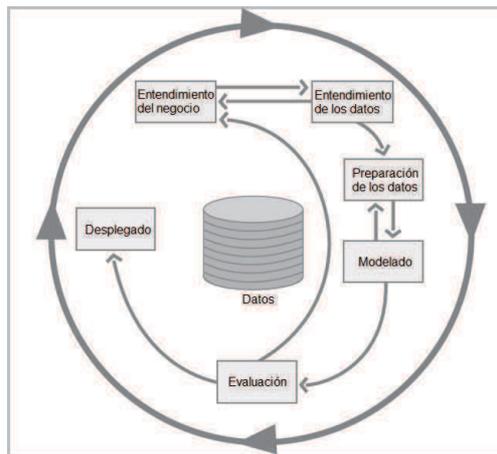


Figura 3.3 Fases del modelo de referencia CRISP-DM. (Chapman, et al 2000)

### 3.1.2.1. Fases de la metodología CRISP-DM

En la figura 3.4 se presentan las fases de la metodología CRISP-DM, las tareas específicas de cada etapa acompañadas de sus salidas (cursiva) correspondientes:

Entendimiento del negocio	Entendimiento de los datos	Preparación de datos	Modelado	Evaluación	Desplegado
<b>Determinar los objetivos del negocio</b> <i>Antecedentes</i> <i>Objetivos del negocio</i>	<b>Colectar los datos iniciales</b> <i>Reporte de la colección de datos iniciales</i>	<b>Seleccionar los datos</b> <i>Inclusión y exclusión racional</i>	<b>Seleccionar las técnicas de modelado</b> <i>Técnica de modelado</i> <i>Supuestos del modelado</i>	<b>Evaluar resultados</b> <i>Evaluación de los resultados de la minería de datos con respecto a los criterios de éxito del negocio</i>	<b>Desplegado del proyecto</b> <i>Plan desplegado</i>
<b>Evaluar la situación</b> <i>Inventario de los recursos</i> <i>Requerimientos, supuestos y restricciones</i> <i>Riesgos y contingencias</i> <i>Terminología</i> <i>Costos y beneficios</i>	<b>Describir los datos</b> <i>Reporte de descripción de datos</i>	<b>Limpiar los datos</b> <i>Reporte de la limpieza de los datos</i>	<b>Generar prueba de diseño</b> <i>Prueba de diseño</i>	<b>Modelos aprobados</b>	<b>Monitoreo y mantenimiento del plan</b> <i>Monitorear y mantener el plan</i>
<b>Determinar los objetivos de la minería de datos</b> <i>Objetivos de minería de datos</i> <i>Criterios de éxito de la minería</i>	<b>Explorar datos</b> <i>Reporte de exploración de datos</i>	<b>Construir datos</b> <i>Derivar atributos</i> <i>Generar registros</i>	<b>Construir el modelo</b> <i>Configuración de los parámetros</i> <i>Modelos</i> <i>Descripción de los modelos</i>	<b>Proceso de revisión</b> <i>Revisión del proceso</i>	<b>Producir el reporte final</b> <i>Reporte final</i> <i>Presentación final</i>
<b>Producir el plan del proyecto</b> <i>Plan del proyecto</i> <i>Evaluación inicial de técnicas y herramientas</i>	<b>Verificar la calidad de los datos</b> <i>Reporte de calidad de los datos</i>	<b>Integrar datos</b> <i>Mezcla de datos</i>	<b>Evaluar modelo</b> <i>Evaluación del modelo</i> <i>Configuración de los parámetros revisados</i>	<b>Determinar los pasos siguientes</b> <i>Lista de las posibles acciones de decisión</i>	<b>Revisión del proyecto</b> <i>Documentar la experiencia</i>
		<b>Formatear datos</b> <i>Datos</i> <i>Reformateados</i> <i>Conjuntos de datos</i> <i>Descripción de los conjuntos de datos</i>			

Figura 3.4 Tareas genéricas (negrita) y salidas (cursiva) del modelo de referencia CRISP-DM (Chapman et al 2000)

A continuación se describen las fases que conforman la metodología CRISP-DM.

- **Entendimiento del negocio:** La fase inicial se enfoca en entender los objetivos del proyecto y requerimientos desde una perspectiva del negocio, para luego convertir este conocimiento en una definición de problema de minería de datos y un diseño preliminar de un plan para alcanzar estos objetivos.
- **Entendimiento de los datos:** Esta fase empieza con la colección de datos iniciales y procede con las actividades que permiten familiarizarse con los datos, identificar los problemas de calidad de los datos, hacer las primeras intuiciones y/o detectar subconjuntos interesantes y formar hipótesis con respecto a la información oculta.
- **Preparación de los datos:** Esta fase cubre todas las actividades necesarias para construir el conjunto final de datos de los datos en bruto iniciales.
- **Modelado:** En esta fase, varias técnicas de modelado se seleccionan y aplican, y sus parámetros se calibran a sus valores óptimos. Normalmente hay muchas técnicas para el mismo tipo de problema de minería de datos, algunas técnicas tienen requerimientos específicos en la forma de los datos, por lo tanto, regresar a la fase de preparación de datos es a menudo necesario.
- **Evaluación:** En esta fase del proyecto se tiene que construir un modelo (o modelos) que parezcan tener calidad alta desde una perspectiva de análisis de datos. Antes de proceder al desplegado final del modelo, es importante evaluar a fondo y revisar los pasos ejecutados para crearlo, para asegurar que el modelo alcanza propiamente los objetivos del negocio.
- **Despliegue:** Esta fase se requiere porque generalmente la creación del modelo no es el fin del proyecto, aún si el propósito del modelo es incrementar el conocimiento de los datos, el conocimiento ganado se necesita presentar de una manera que el cliente lo pueda usar.

En las sub-secciones siguientes se describe la fase de preparación de datos, que es el tema principal de esta investigación.

#### 3.1.2.2. Fase de preparación de datos

Esta fase cubre todas las actividades necesarias para construir el conjunto final de datos de los datos en bruto iniciales. Las tareas de preparación de datos se

desarrollan probablemente múltiples veces y no en cierto orden. Las tareas incluyen, uso de tablas, registros y selección de atributos, así como la transformación y limpieza de los datos para que las herramientas de modelado.

La preparación de datos es una etapa fundamental en el análisis de los datos, mucha información de mala calidad está disponible en la Web y debido a esto muchas organizaciones están interesadas en la forma de transformarla en datos limpios que puedan usarse para propósitos de alta rentabilidad.

### **3.1.2.3. Importancia de la preparación de datos**

Según Zhang (Zhang, et al 2013) con el transcurso de los años ha habido un avance significativo en las técnicas de la minería de datos. Este avance no ha correspondido al mismo progreso en la preparación de los datos. Por lo tanto, ahora hay una fuerte necesidad de diseñar nuevas técnicas y herramientas automatizadas que puedan asistirnos significativamente en la preparación de datos de calidad. La preparación de datos puede consumir más tiempo que la minería de datos y puede presentar igual o aún más retos que la minería de datos. La importancia de la preparación de los datos se argumenta en tres aspectos:

1. Los datos del mundo real pueden ser incompletos, sucios e inconsistentes y pueden ocultar patrones útiles
  - Datos incompletos: falta de valores, falta de ciertos atributos de interés, contener datos agregados
  - Datos sucios: contener datos erróneos o fuera de rango
  - Datos inconsistentes: contienen discrepancias en códigos o nombres
2. La preparación de datos genera un conjunto de datos más pequeño que el original, que significativamente mejore la eficiencia de la minería de datos. Esta tarea incluye:
  - Seleccionar datos relevantes
  - Reducción de datos
3. La preparación de datos genera datos de calidad, que producen patrones de calidad

A partir de estos tres aspectos se puede entender que el pre-procesamiento, limpieza y preparación de datos no es una tarea pequeña.

#### 3.1.2.4. Etapas de la preparación de datos

Dentro de la fase de preparación de datos existen diferentes tareas generales, a continuación se explica cada una de ellas:

**Selección de datos:** En esta tarea general se decide qué datos se usarán para el análisis. El criterio incluye la relevancia de los objetivos de la minería de datos, la calidad y restricciones técnicas tales como los límites en el volumen y tipo de los datos. Se hace notar que la selección de datos cubre la selección de atributos (columnas) al igual que la selección de los registros en una tabla (filas).

*Salida:* Lista de los datos a ser incluidos/excluidos y las razones de esta decisión.

**Limpieza de datos:** En esta tarea general se eleva la calidad de los datos al nivel requerido por las técnicas de análisis seleccionadas, la inserción de valores predeterminados adecuados, o más técnicas ambiciosas tales como la estimación de falta de datos a modelar.

*Salida:* Reporte de limpieza de datos. Describe las decisiones y acciones que fueron tomadas para direccionar los problemas reportados sobre la calidad de los datos.

**Construcción de datos:** Esta tarea incluye las operaciones de preparación de datos constructivas, tales como la producción de atributos derivados o el registro de nuevos datos completos, o valores transformados para atributos existentes.

*Salida:* 1) Atributos derivados. Son nuevos atributos que se construyen de uno o más atributos existentes en el mismo registro. Ejemplo:  $\text{área} = \text{longitud} * \text{anchura}$ . 2) Registros generados. Son registros completamente nuevos.

**Integración de datos:** En esta tarea general se usan métodos mediante los cuales la información se combina desde múltiples tablas y registros para crear nuevos registros o valores.

*Salida:* Datos combinados. Estos datos combinan dos o más tablas que tienen diferente información sobre los mismos objetos.

**Formatear datos:** En esta tarea general se formatean transformaciones que se refieren primariamente a modificaciones sintácticas hechas a los datos que no cambian su significado, pero podrían ser requeridas por la herramienta de modelado.

*Salida:* Datos reformateados. Algunas herramientas tienen requerimientos del orden de los atributos, tales como que el primer campo sea el único identificador para cada registro o el último campo sea el resultado del modelo.

### **3.2. Ingeniería de software**

La definición que propone la IEEE para la ingeniería de software es: la aplicación de un enfoque sistemático (ordenado), disciplinado y cuantificable al desarrollo, operación y mantenimiento de software, esto es, la aplicación de la ingeniería en el área del software.

Otras definiciones de ingeniería de software son las siguientes:

- Ingeniería de software es el estudio de principios y metodologías para el desarrollo y mantenimiento de sistemas software (Zelkovitz, 1978).
- Ingeniería de software es la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software (Bohem, 1976).
- La ingeniería de software trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable, que sea fiable y trabaje en máquinas reales (Bauer, 1972).

### **3.3. Minería de datos para ingeniería de software**

Según Xie (Xie, et al 2010) los datos de ingeniería de software son ricos en información acerca del progreso y evolución de un proyecto. Muchos estudios han podido comprobar que usar estos datos ayuda en varios aspectos del desarrollo de software dentro de ajustes industriales y de código abierto.

Se pueden tomar los datos generados por el proceso de desarrollo de software, aplicarles técnicas de minería de datos y aplicar los resultados en las diferentes tareas de ingeniería de software como se muestra en la figura 3.5.

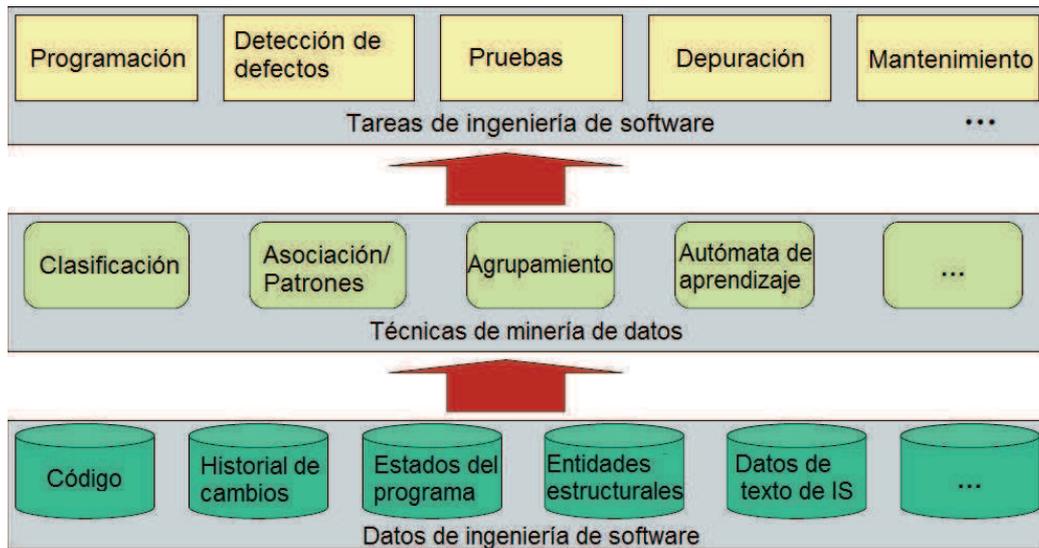


Figura 3.5 Resumen de los datos de ingeniería de software. (Xie et al 2010)

Según Xie (Xie, et al 2009) la metodología para la minería de datos de ingeniería de software incluye cinco pasos básicos. En la práctica los ingenieros de software adoptan una combinación de los dos primeros pasos (Figura 3.6).

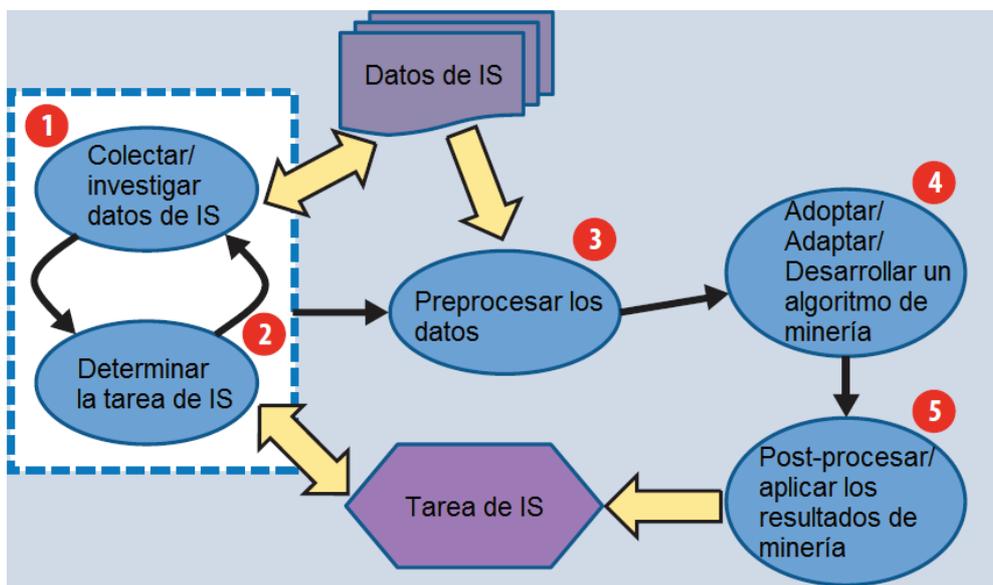


Figura 3.6 Metodología para minería de datos de ingeniería de datos (Xie, et al 2009)

### 3.4. Software libre

Según la *Free Software Foundation* el software libre es el software que respeta la libertad de los usuarios y la comunidad. En grandes líneas, significa que los usuarios tienen la libertad para ejecutar, copiar, distribuir, estudiar, modificar y mejorar el

software. Es decir, el «software libre» es una cuestión de libertad, no de precio. Para entender el concepto, piense en «libre» como en «libre expresión», no como en «barra libre».

Un programa es software libre si los usuarios tienen las cuatro libertades esenciales:

- La libertad de ejecutar el programa para cualquier propósito (libertad 0)
- La libertad de estudiar cómo funciona el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello
- La libertad de redistribuir copias para ayudar a su prójimo (libertad 2)
- La libertad de distribuir copias de sus versiones modificadas a terceros (libertad 3). Esto le permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello

### 3.5. Integración de herramientas

La integración de herramientas según Thomas (Thomas, 1992) no es una propiedad únicamente de un sola herramienta pero si de sus relaciones con otros elementos en el ambiente.

Wasserman (Wasserman, 1990) divide la forma de integración de herramientas en 5 categorías, las cuales son por: plataforma, presentación, datos, control y proceso. A continuación se describe brevemente cada categoría:

**Plataforma:** Se refiere a los servicios comunes del marco de trabajo o plataforma usados por las herramientas. Ya que pueden estar distribuidas en diferentes máquinas y diferentes sistemas operativos.

**Presentación:** Se refiere a la interacción con el usuario. Las herramientas deben compartir un “ver y sentir (*look and feel*)” común desde la perspectiva del usuario.

**Datos:** Se refiere al intercambio de datos entre las herramientas, este tipo de integración requiere tanto compartir datos entre las herramientas como

manejar las relaciones entre los objetos de datos producidos por las herramientas.

**Control:** Se refiere a la interoperabilidad de las herramientas, las cuales deben ser capaces de notificar a otra de eventos sucedidos y a su vez tener la habilidad de activar otra herramienta.

**Proceso:** En este tipo de integración se considera el rol de las herramientas dentro del proceso de software completo.

Además de estos tipos de integración existe la integración usando API's (de las siglas en inglés de *Application Programming Interface*). Así, algunas herramientas proveen una API, la cual puede usarse para comunicarse con la herramienta de forma automática. En este caso, las herramientas pueden comunicarse entre sí usando este método de integración.

Para el desarrollo del ambiente de soporte a minería de datos de desarrollo de software, que se reporta en este documento, se utilizan la integración por datos y API's, lo que genera un ambiente heterogéneo.

### 3.6. Modularidad en Java

La modularidad, se define en el sentido tradicional de las ciencias de la computación, donde el código del software está dividido en partes lógicas que representan intereses separados (Hall, 2011). Estas partes lógicamente independientes se llaman módulos.

La modularidad permite:

- Facilidad de cambio
- Facilidad de comprensión
- Desarrollo en paralelo
- Aplicaciones fáciles de testear
- Reutilización

### 3.7. OSGI

*Open Services Gateway Initiative (OSGI)*, creado por la *OSGI Alliance* en 1999, es un consorcio de empresas líderes en tecnología.

La iniciativa OSGI define un sistema modular dinámico para Java, dando mejor control sobre la estructura del código, la habilidad de dinámicamente manejar su ciclo de vida y un enfoque de bajo acoplamiento para la colaboración del código.

Desde el enfoque de OSGI cada módulo se llama *bundle*.

#### 3.7.1. Bundle

*Bundle* es una unidad física de modularidad en la forma de un archivo JAR (por sus siglas en inglés *Java ARchive*) que contiene código, recursos y metadatos (Hall, 2011).

#### 3.7.2. Ciclo de vida

Cada *Bundle* inicia su ciclo de vida (ver Figura 3.7) en el estado instalado, si todas sus dependencias se cumplen pasa al estado *resuelto* y sus clases pueden cargarse y ejecutarse, si se inicia pasa al estado *activo* y puede detenerse. Estando en el estado resuelto puede reiniciarse o *desinstalarse*.

Todo este ciclo de vida lo hace dinámicamente, esto significa que no necesita detenerse la operación del sistema por completo para instalar, desinstalar o cambiar el estado de un *Bundle*.

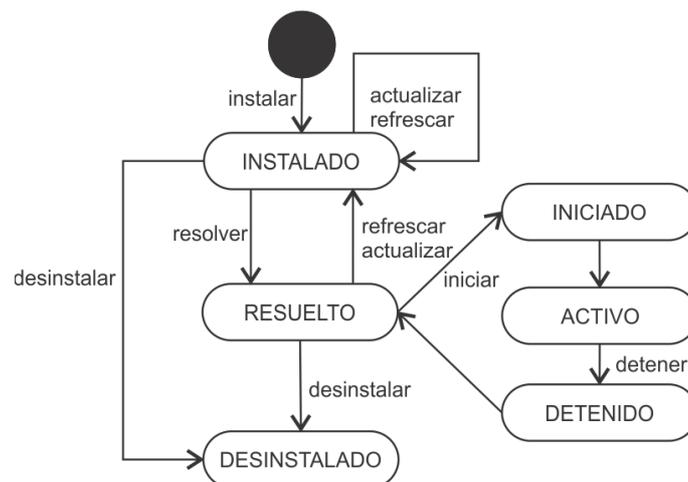


Figura 3.7 Ciclo de vida de un *Bundle*

### 3.7.3. Colaboración

El mecanismo de colaboración de OSGI es el registro de servicios que actúa como un tablón de anuncios de funciones que coordina tres partes: *Bundles* que definen interfaces de servicio, *Bundles* que implementan y registran objetos de servicio y *Bundles* que descubren y usan esos servicios. El servicio de servicios hace esta colaboración anónima, el *Bundle* que provee el servicio no sabe quién lo consume y el *Bundle* que consume el servicio no sabe quién lo provee. En la figura 3.8 se presentan 3 *bundles*: el *bundle C* que declara una interfaz usada por el *bundle B* para registrar un servicio y el *bundle A* que descubre y usa el servicio, de esta forma el *bundle A* solo depende del *bundle C*.

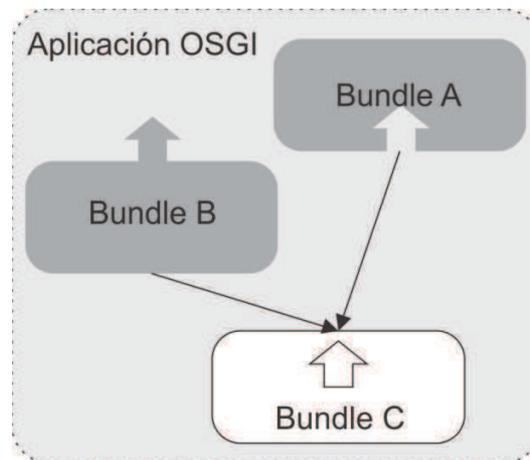


Figura 3.8 Colaboración basada en servicios

En este capítulo, se definieron conceptos relacionados a KDD y minería de datos, las fases del proceso CRISP-DM, y se define a más detalle la fase de preparación de datos ya que es la fase a cubrir en esta investigación, también se define el término ingeniería de software, pues este es el campo de aplicación de la minería de datos en esta investigación.

Se define la integración de herramientas y algunos tipos que existen, siendo los elegidos para esta investigación integración por: datos y APIs.

El ambiente de soporte a minería de datos de desarrollo de software (AMDADS) se diseñó por módulos Java siguiendo las especificaciones OSGI.

En el siguiente capítulo se define la forma de evaluar las herramientas candidatas a integrar y se listan las elegidas.

## HERRAMIENTAS SELECCIONADAS PARA LA INTEGRACIÓN

# Capítulo 4

En este capítulo se presenta la metodología utilizada para evaluar y seleccionar las herramientas a integrar en el ambiente de soporte a minería de datos de desarrollo de software, además se describen de forma breve cada una de estas herramientas.



En este capítulo se incluye la forma de seleccionar las herramientas a integrar en el ambiente de soporte a minería de datos de desarrollo de software. Se inicia el capítulo explicando la metodología utilizada para hacer la evaluación y selección de herramientas, después se describe cada una de las herramientas seleccionadas, las cuales son: Mantis BT, MySQL, DotProject, OpenRefine y RapidMiner. También se definen las herramientas que no fueron integradas pero tienen un soporte potencial al proceso de minería de datos de desarrollo de software, estas herramientas forman parte del paquete de ofimática LibreOffice y son las siguientes: Calc, Writer, Impress y Draw.

#### **4.1. Metodología QSOS**

Para hacer la comparación y evaluación de las herramientas FLOSS a integrar en este ambiente de soporte a minería de datos de repositorios de software, se utilizó la metodología QSOS (del inglés *Qualification and Selection of Open Source software*), su objetivo es evaluar herramientas FLOSS (del inglés *Free/Libre Open Source Software*). (QSOS, 2014)

La metodología QSOS consiste en una lista de cuatro pasos dentro de un proceso iterativo:

- Definir - y organizar lo que será evaluado (criterios)
- Evaluar - el software contra los criterios definidos
- Calificar - establecer pesos relacionados al contexto
- Seleccionar - la herramienta apropiada

Para la implementación de esta metodología sus creadores proponen el uso de herramientas para llevar a cabo este proceso (Figura 4.1).

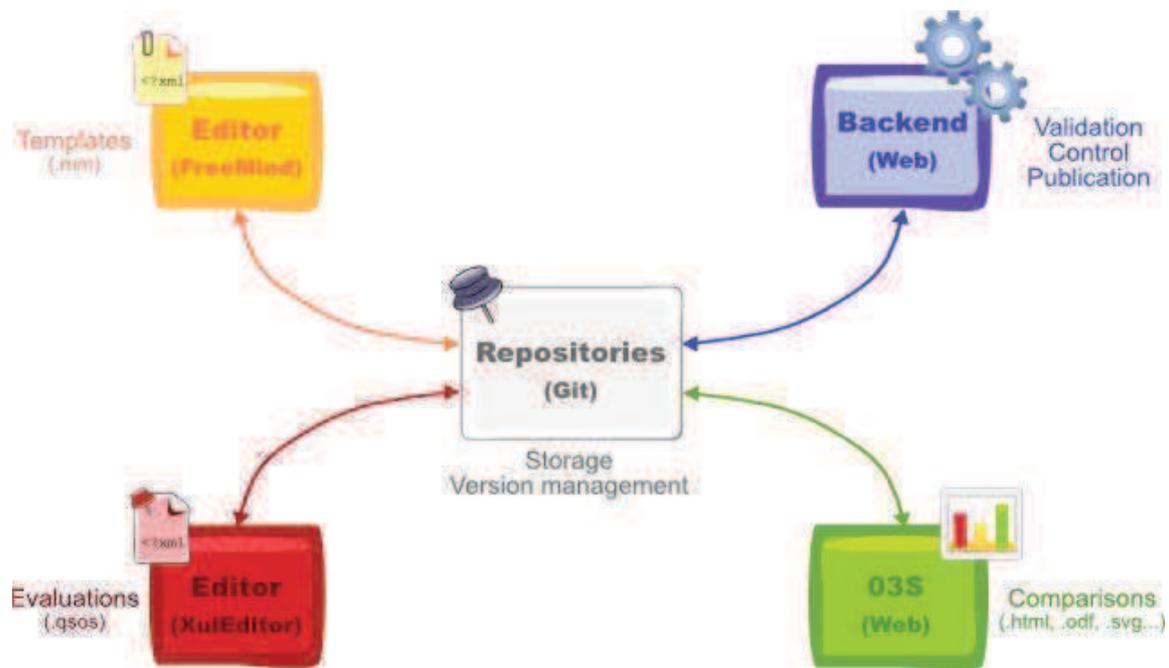


Figura 4.1 Herramientas propuestas para implementar QSOS (QSOS, 2014)

## 4.2. Herramientas seleccionadas

Con base en los resultados de la comparación y evaluación de las herramientas (Anexo A) se eligieron las siguientes herramientas para ser parte del ambiente de soporte a minería de datos de desarrollo de software.

### 4.2.1. Mantis Bug Tracker (MantisBT)

MantisBT es un software popular de código abierto para el seguimiento de incidencias basado en web que provee balance delicado entre simplicidad y poder. Su logotipo es un dibujo del insecto que origina su nombre (Figura 4.2). Construido en PHP, soporta Linux, Windows y Mac OS X del lado del servidor. Compatible con Google Chrome, Firefox, Safari, Opera e Internet Explorer 7+. Lanzado bajo los términos de la Licencia Pública Genérica (GPL por sus siglas en inglés).

MantisBT tiene soporte para múltiples bases de datos incluidas MySQL, MS SQL, PostgreSQL y DB2.



Figura 4.2 Logotipo MantisBT

#### 4.2.2. MySQL

MySQL es el sistema de gestión de base de datos de código abierto más popular, es desarrollado, distribuido y apoyado por Oracle Corporation. Su logotipo está compuesto por el nombre del software acompañado de un delfín (Figura 4.3).

El sitio web de MySQL (<http://www.mysql.com/>) provee la información más reciente sobre el software MySQL

Algunas características de MySQL son:

- MySQL es un sistema de gestión de base de datos
- Las bases de datos MySQL son relacionales
- MySQL es un software de código abierto
- EL servidor de base de datos MySQL es muy rápido, confiable, escalable y fácil de usar
- MySQL trabaja en cliente/servidor o en sistemas embebidos
- Una gran cantidad de software MySQL contribuido está disponible



Figura 4.3 Logotipo MySQL

#### 4.2.3. DotProject

DotProject es una aplicación para administración de proyectos soportada por voluntarios. Es una aplicación basada en web, multiusuario, soporta varios lenguajes y es software libre. Su logotipo es una representación de sus iniciales: *d* y *p* (ver Figura 4.3).

Algunas de sus características son:

- Basado en web
- Escrito en PHP
- Comunidad activa que promueve la ayuda entre usuarios
- Código abierto y libre
- Crear proyectos a partir de plantillas
- Generación de reportes
- Soporta varios idiomas



**Figura 4.4** Logotipo dotProject

#### **4.2.4. OpenRefine**

OpenRefine es una herramienta poderosa para trabajar con datos desordenados: limpiarlos; transformarlos; de un formato a otro; extenderlos con servicios web; y unirlos a bases de datos. Su logotipo es un diamante (Figura 4.5).

Características:

- Importar datos en varios formatos
- Explorar conjuntos de datos en cuestión de segundos
- Aplicar transformación de celdas básica y avanzadas
- Crear enlaces instantáneos entre los conjuntos de datos
- Limpiar y transformar datos



**Figura 4.5** Logotipo OpenRefine

#### **4.2.5. RapidMiner**

RapidMiner (Anteriormente llamado YALE) es una poderosa herramienta visual para minería de datos, máquinas de aprendizaje y análisis predictivo. En la figura 4.6 se muestra el logotipo de la herramienta.

El encadenamiento de operadores fue su característica desde el inicio. Esto permite tener un número pequeño de operadores en un nivel.

La validación cruzada es uno de los operadores más populares, además permite el uso de ciclos lo que lo hace similar a las nociones de programación, pero la programación no es necesaria, es solo arrastrar y soltar lo que lo hace fácil de entender.

Características:

- Ambiente visual fácil de usar para análisis predictivo. No se requiere programación
- Abierto y extensible
- Análisis avanzado a todas las escalas, perfecto para Big Data
- Cuenta con plantillas
- Corre bajo las mejores plataformas y sistemas operativos



**Figura 4.6** Logotipo de RapidMiner

Las herramientas enlistadas a continuación no se integran por algún componente de acoplamiento pero se agregan al ambiente debido a su soporte potencial al proceso de minería de datos de desarrollo de software.

#### **4.2.6. LibreOffice Calc**

Es la herramienta de hoja de cálculo de la paquetería LibreOffice. Creado por *The Document Foundation*. Calc permite manipular datos numéricos y alfanuméricos. Según sus creadores es fácil de aprender, intuitiva e ideal para minería de datos. Su logotipo representa una hoja de cálculo (Figura 4.7).

Esta herramienta no está integrada pero se incluye ya que es necesaria para la conversión de archivos y manejo de datos.

Características:

- Plantillas integradas
- Administrador de escenarios
- Facilidad para manejar datos
- Trabajo colaborativo con otros usuarios
- Compatibilidad con Microsoft Excel
- Soporta diversos tipos de archivos (csv, xls, ods, xml, html, dif, etc.)



**Figura 4.7** Logotipo LibreOffice Calc

#### **4.2.7. LibreOffice Writer**

Es la herramienta de procesador de texto y autoedición. Su logotipo representa un documento de texto con una imagen insertada (Figura 4.8.).

Características:

- Simple
- Diccionario de autocorrección



**Figura 4.8** Logotipo LibreOffice Writer

#### **4.2.8. LibreOffice Impress**

Herramienta para la creación de presentaciones multimedia. Su logotipo está compuesto por una ilustración de una presentación (Figura 4.9).

Características:

- Notas
- Clasificador de diapositivas

- Dibujo y diagramación
- Fontworks
- Plantillas



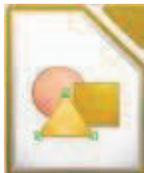
**Figura 4.9** Logotipo LibreOffice Impress

#### **4.2.9. LibreOffice Draw**

Herramienta para la creación de gráficos y diagramas. Su logotipo muestra figuras geométricas representando la creación de gráficos (Figura 4.10).

Características:

- Conectores inteligentes
- Organigramas
- Manipulación de imágenes



**Figura 4.10** Logotipo LibreOffice Draw

Con la inclusión del paquete de ofimática LibreOffice se complementa la integración de las herramientas seleccionadas mediante la metodología QSOS (del inglés *Qualification and Selection of Open Source software*).

En el siguiente capítulo se presentan los requerimientos y el diseño del ambiente de soporte a minería de datos de desarrollo de software.



En este capítulo se presentan los resultados principales de la investigación, que se obtuvieron para la especificación de requerimientos y el diseño del Ambiente de Soporte a Minería de Datos de Desarrollo de Software.



### 5.1. Introducción

Este capítulo contiene los resultados principales de la especificación de requerimientos necesarios y del diseño de software, para la realización de los componentes de acoplamiento de herramientas libres y de código abierto (ver Capítulo 4 inciso 4.2. Herramientas seleccionadas), que integradas formen un ambiente que soporte el proceso de minería de datos de desarrollo de software, para específicamente la etapa de preparación de datos.

Los componentes de acoplamiento hacen que las herramientas a integrar trabajen de una forma conjunta, evitando la captura repetida de los mismos datos.

En este capítulo se definen los lenguajes utilizados para la especificación de requerimientos, la perspectiva y funciones del producto, además de las características de los usuarios. Posteriormente se describen los requerimientos funcionales y no funcionales. Como parte del diseño se incluye el diagrama de componentes, de secuencia y de clases.

### 5.2. Lenguajes utilizados en la especificación de requerimientos

Para la especificación de requerimientos se siguió la recomendación ITU-T Z.151 (10/12), donde se define la Notación de Requerimiento de Usuario URN (por las siglas en inglés para *User Requirements Notation*), que está destinada a la elicitación, análisis, especificación y validación de requerimientos. URN combina conceptos de modelado y notaciones para metas (principalmente para requerimientos no funcionales y atributos de calidad) y escenarios (principalmente para requerimientos operacionales, requerimientos funcionales, rendimiento y razonamiento de la arquitectura). Para lo anterior hace uso de dos lenguajes de la URN, los cuales son:

- GRL (*Goal-oriented Requirements Language*)
- UCM (*Use Case Maps*)

La recomendación ITU-T Z.151 (10/12) define al lenguaje GRL como un lenguaje para soportar modelado orientado a metas y razonamiento sobre requerimientos, especialmente requerimientos funcionales y no funcionales y atributos de calidad. Provee construcciones visuales (Figura 5.1) para expresar varios tipos de conceptos que aparecen durante el proceso de requerimientos. Varias figuras de este capítulo, a partir de la figura 5.5 se desarrollaron con el lenguaje GRL.

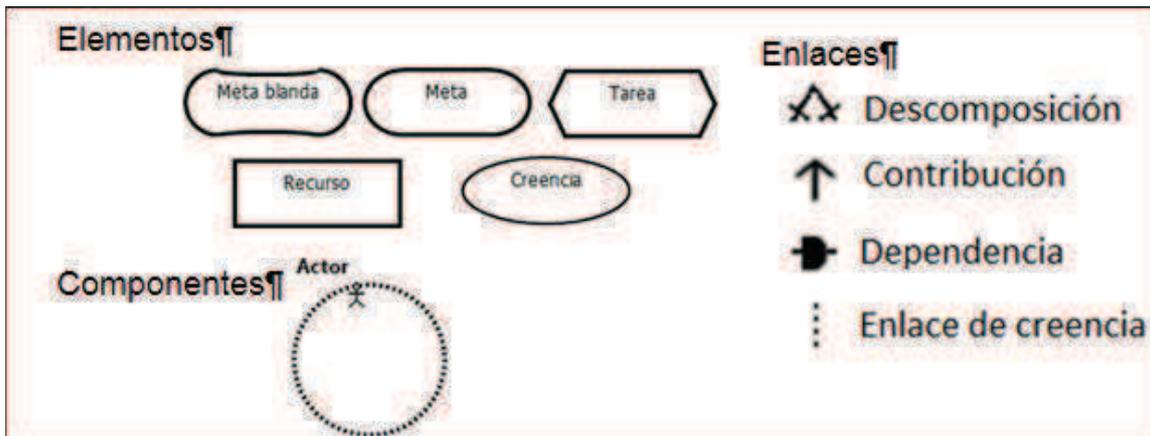


Figura 5.1 Construcciones visuales de GRL herramienta JUCMNav v6.0.0

Además, según la recomendación ITU-T Z.151 (10/12) las especificaciones representadas mediante el lenguaje UCM emplean caminos de escenarios para ilustrar relaciones casuales entre las responsabilidades. Así, mediante el lenguaje UCM se provee una vista integrada del comportamiento y estructura que permite la superposición de caminos de escenario en una estructura de componentes abstractos. La combinación del comportamiento y la estructura permite el razonamiento arquitectónico que hace posible refinar las especificaciones UCM en modelos de escenario más detallados y finalmente en implementaciones concretas. Así, la notación UCM permite una transición fluida de lo informal a lo formal uniendo la brecha de modelado entre los modelos de metas (representados con el lenguaje GRL) y los requerimientos en lenguaje natural (por ejemplo los casos de uso) y el diseño en una forma visual y explícita. Las construcciones visuales del UCM se presentan en la figura 5.2. Varias figuras de este capítulo, a partir de la figura 5.6 se desarrollaron con el lenguaje UCM.

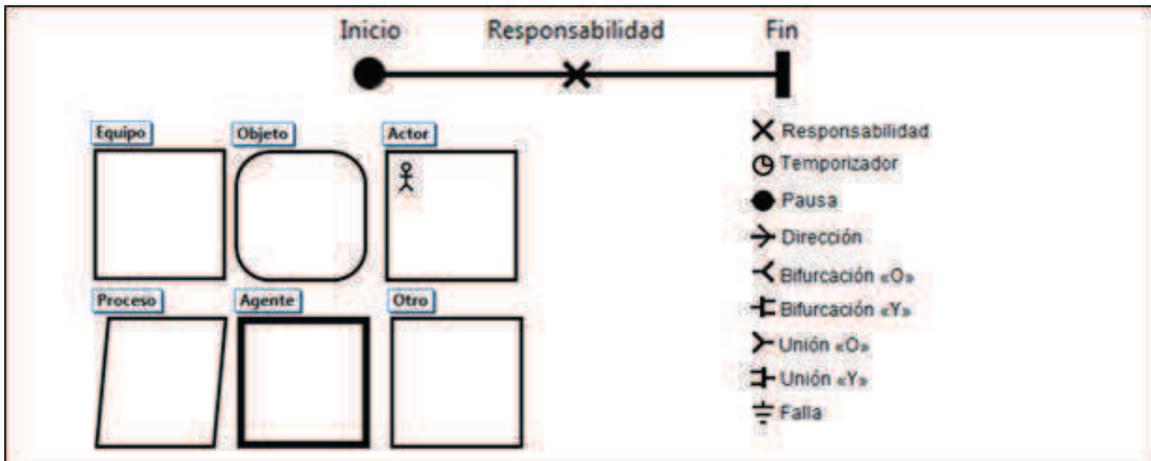


Figura 5.2 Construcciones visuales de UCM herramienta JUCMNav v6.0.0

Según [Amyot, 2001] los modelos realizados con GRL y UCM representan una pieza útil, en el rompecabezas (Figura 5.3) de lenguajes visuales usados en el desarrollo de software, que ayuda a unir la brecha entre requerimientos y diseño.

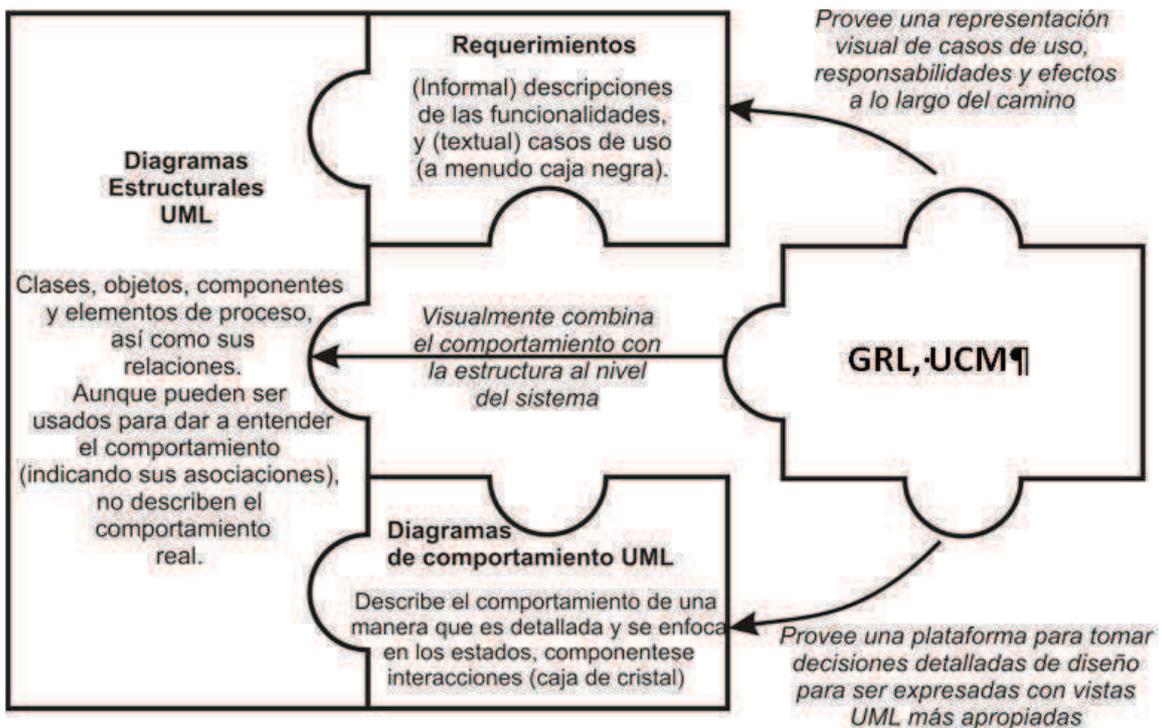


Figura 5.3 GRL y UCM como una pieza faltante del rompecabezas UML

La realización de esta sección de especificación, además de utilizar los lenguajes GRL y UCM, se estructuran siguiendo los lineamientos del estándar IEEE-STD-830-1998, en el que se presenta el contenido de la Especificación de Requerimientos de Software (SRS).

### 5.3. Perspectiva del producto

El ambiente de soporte a minería de datos de desarrollo de software (AMDADS) es un producto que estará integrado por diferentes herramientas FLOSS que soporten o sean de utilidad al proceso de MD.

### 5.4. Funciones del producto

Las funciones de los componentes de acoplamiento y del ambiente son:

- Integrar las herramientas que se seleccionaron para formar parte del ambiente de soporte a minería de datos de desarrollo de software
- Se soportará principalmente la fase de preparación de datos, pero el ambiente estará abierto a futuras integraciones de herramientas, que soporten el resto de las fases del proceso de MD
- Los usuarios tendrán que registrarse para obtener un identificador de usuario y una contraseña, para poder hacer uso del ambiente

Las fases y tareas del proceso de minería de datos del desarrollo de software, que se usarán son las establecidas en la metodología híbrida MIDADS (Figura 5.4) propuesta por [Meza, 2014], en donde se integran tres metodologías, y tareas adicionales a las establecidas en dichas metodología. Las tres metodologías mencionadas son: 1) la metodología base es la CRISP-DM (Chapman et al. 2000) la cual es la más usada; 2) la metodología propuesta por Xie (Xie et al. 2009) que se enfoca en datos de ingeniería de software; y 3) la propuesta por Iturbide (Iturbide 2013) en la que la fase de preparación de datos la divide en preparación general y específica, con el fin de aplicarla a epidemiología. Según (Meza, 2014), la metodología MIDADS es aplicable al dominio de datos de desarrollo de software.

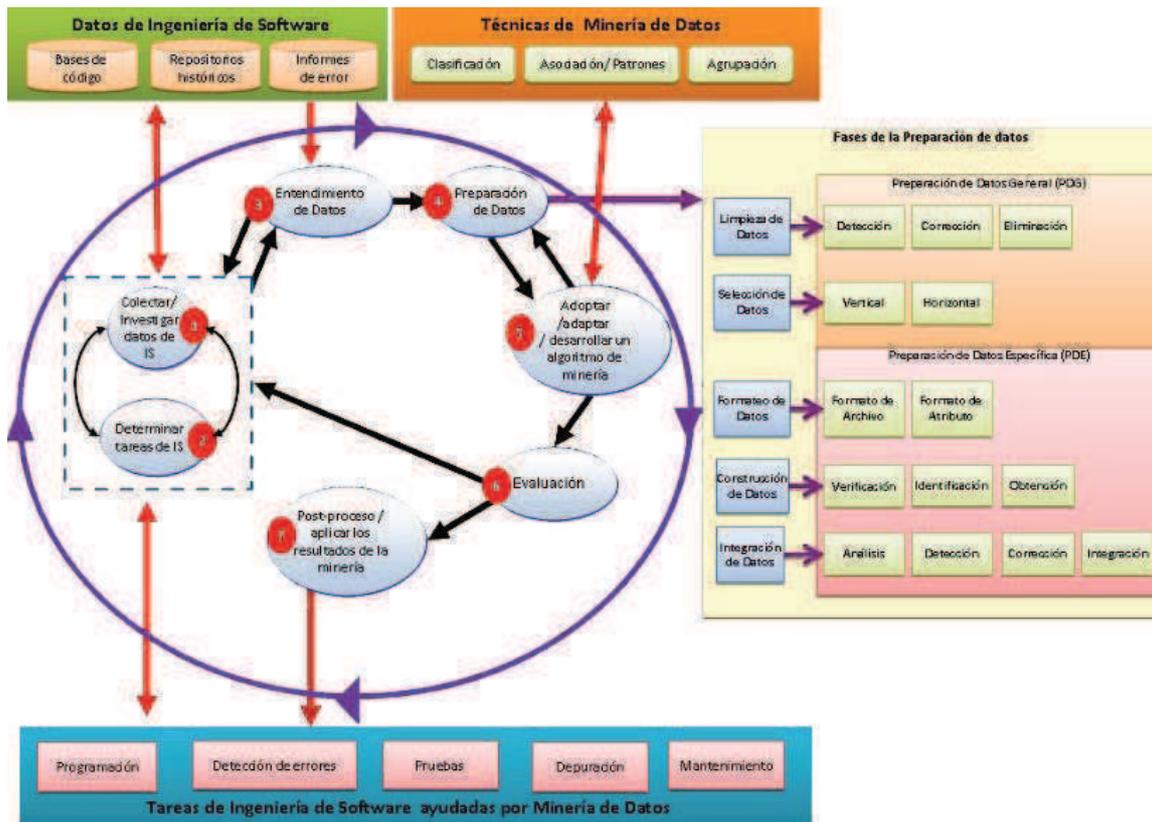


Figura 5.4 Metodología MIDADS

### 5.5. Características del usuario

La funcionalidad de las herramientas que se integren, no se verá afectada en sus interfaces de usuario, así que los usuarios no necesitan conocimientos extra y el aprendizaje para utilizar el ambiente será mínimo. Los usuarios solo requerirán familiarizarse con la manera de utilizar las herramientas integradas, y usar los conocimientos previos acerca del uso de las interfaces de cada herramienta integrada.

### 5.6. Restricciones

Las restricciones que se consideran son las siguientes:

1. Se asume que los requisitos descritos son estables
2. El usuario deberá tener instalado un software virtualizador (ej. Oracle VirtualBox)

### 5.7. Especificación de requerimientos

En esta sección se presentan los diagramas GRL y UCM que se usaron para identificar los requerimientos para los componentes de acoplamiento.

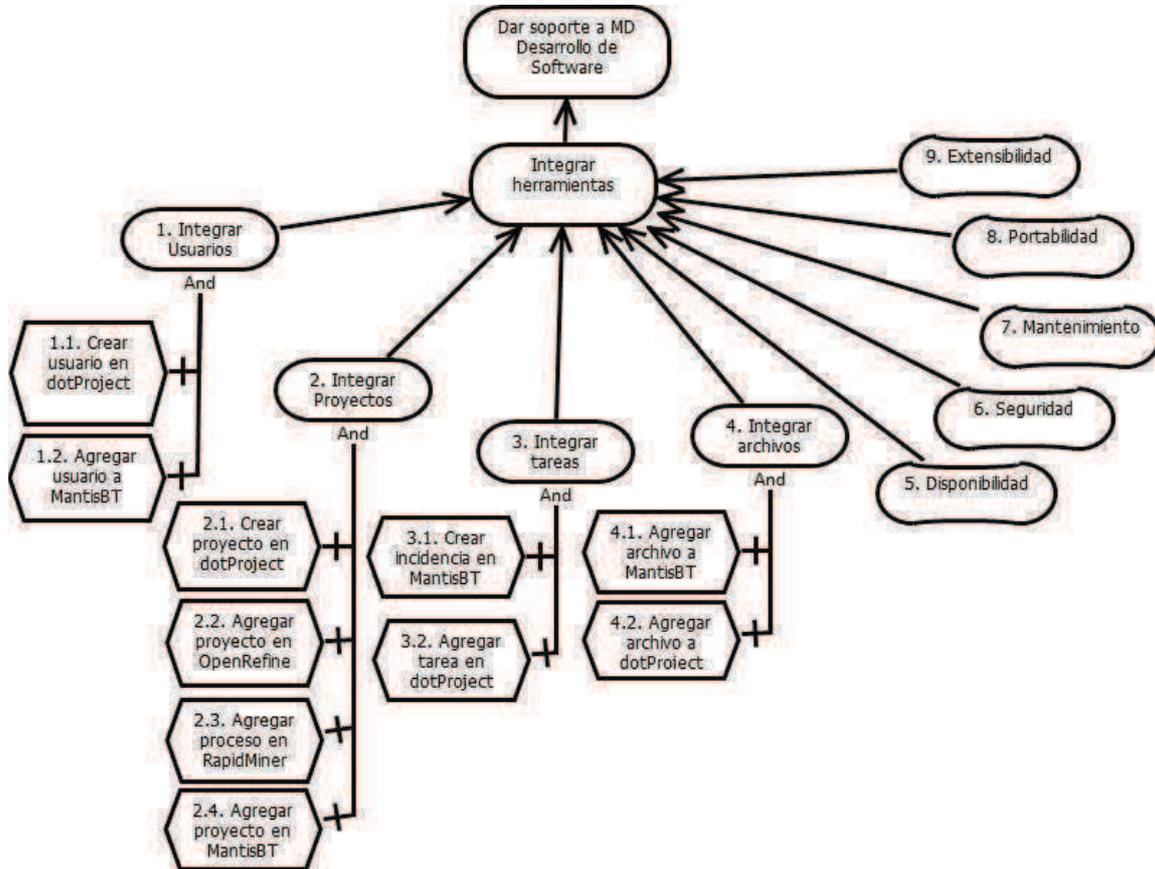


Figura 5.5 Diagrama GRL de los componentes de acoplamiento del ambiente

En la figura 5.5 se muestra el diagrama GRL, con las metas funcionales que nos ayudaron a definir la necesidad de los componentes de acoplamiento. Dichas metas funcionales son: integrar usuarios, integrar proyectos, integrar tareas e integrar archivos. En conjunto estas metas tienen el objetivo de integrar las herramientas con el fin de dar soporte al proceso de minería de datos, en particular a la fase de preparación de datos.

La fase en la que se pone énfasis, en esta investigación, es la preparación de los datos y se considera también que se pueda extender el ambiente implementado, con otras herramientas que soporten las demás fases del proceso de minería de datos.

Los requerimientos no funcionales o metas blandas, mostradas en la figura 5.5 son: extensibilidad, portabilidad, facilidad de mantenimiento, seguridad y disponibilidad. Para esta investigación se definen de la siguiente manera:

**Disponibilidad:** Es la capacidad de un producto de software de permanecer en un estado para realizar una función requerida en cualquier momento, bajo condiciones de uso específicas (ISO/IEC 9126-1, 2001).

**Seguridad:** Es la capacidad de un producto de software para proteger la información y datos de forma que personas o sistemas no autorizados no puedan leerlos o modificarlos. Además, que personas y sistemas autorizados tengan acceso a los datos (ISO/IEC 9126-1, 2001).

**Facilidad de mantenimiento:** Son las características de un producto de software que lo hacen fácil de modificar. Las modificaciones pueden incluir correcciones, mejoras o adaptaciones del software, para responder a cambios en el ambiente, las especificaciones funcionales y los requerimientos (ISO/IEC 9126-1, 2001).

**Portabilidad:** Son las características de un producto de software que facilitan ser transferido de un ambiente a otro (ISO/IEC 9126-1, 2001).

**Extensibilidad:** Son las características que permiten extender un sistema de software con nuevas características y componentes, sin pérdida de funcionalidad o cualidades especificadas como requerimientos. (Henttonen 2007).

A continuación se muestran los diagramas UCM de cada uno de las metas definidas anteriormente en la figura 5.5.

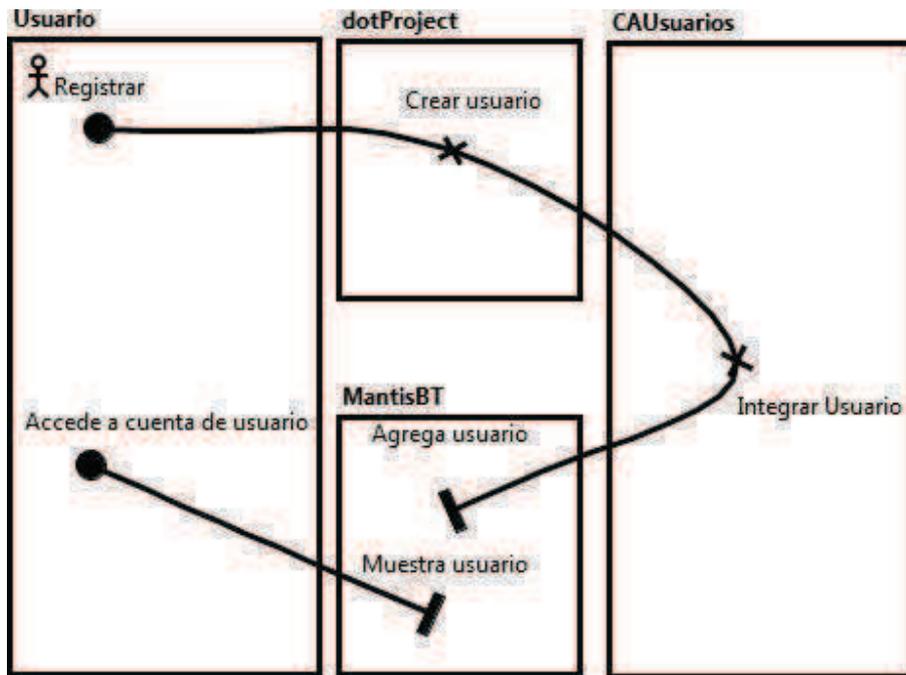


Figura 5.6 Diagrama UCM de la meta funcional de integrar usuarios

En el diagrama UCM de la meta de integrar usuarios (Figura 5.6) se observa que el usuario se registra en la herramienta dotProject y posteriormente el componente encargado de la integración de los usuarios CAUsuarios lo da de alta en la base de datos de la herramienta MantisBT. DotProject y MantisBT son las únicas herramientas que permiten la creación de usuarios.

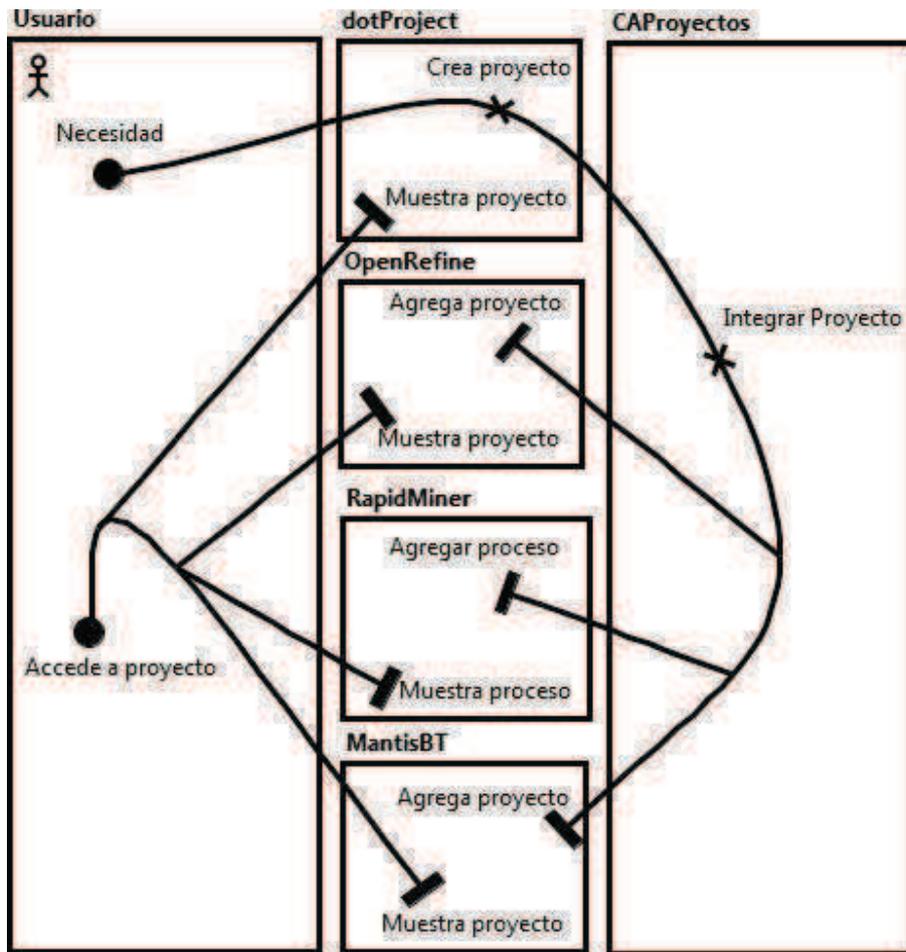


Figura 5.7 Diagrama UCM de la meta de integrar proyectos

En el diagrama UCM de la meta de integrar proyectos (Figura 5.7) se observa que se parte de la necesidad de crear el proyecto, que el usuario crea en dotProject. El componente de acoplamiento registra el proyecto en OpenRefine, RapidMiner y MantisBT. La herramienta RapidMiner no trabaja con un proyecto, pero hace uso de procesos, que consideramos equivaldría al proyecto en las demás herramientas.

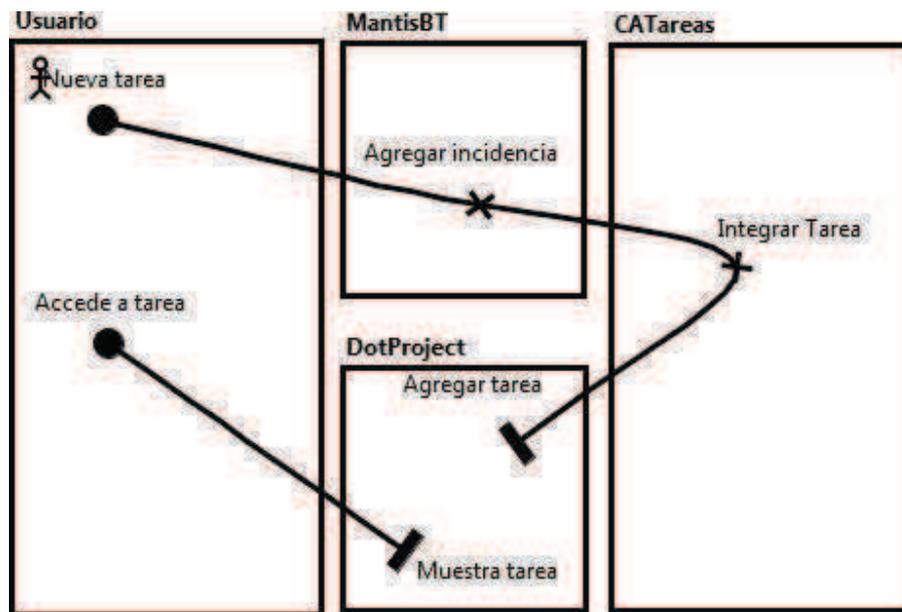


Figura 5.8 Diagrama UCM de la meta funcional de integrar tareas

En el diagrama UCM de la meta de integrar tareas (Figura 5.8) se observa que el usuario crea una nueva incidencia en MantisBT, la asigna a un usuario y proyecto existente, entonces el componente encargado de la integración de las tareas CATareas la agregará al proyecto y usuario en la base de datos de dotProject.

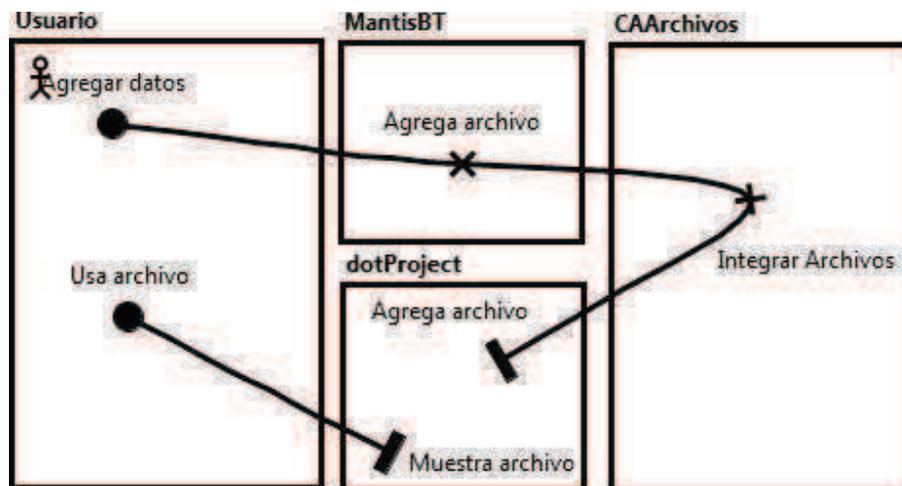


Figura 5.9 Diagrama UCM de la meta de integrar archivos

En el diagrama UCM de la meta de integrar archivos (Figura 5.9) se observa que el usuario agrega un archivo a MantisBT, después el componente encargado de integrar archivos agrega el archivo a dotProject en la tarea equivalente.

A continuación se enlistan los requerimientos funcionales para los componentes de acoplamiento.

## **5.8. Requerimientos obtenidos**

Del análisis de los diagramas GRL se identificaron y se enlistan a continuación, los requerimientos funcionales para los componentes de acoplamiento.

### **5.8.1. Requerimientos Funcionales**

1. Permitir la integración de usuarios
  - 1.1. Permitir la creación una cuenta de usuario en la herramienta dotProject
  - 1.2. Permitir agregar una cuenta de usuario en la herramienta MantisBT
2. Permitir la integración de proyectos
  - 2.1. Permitir la creación de un proyecto en la herramienta dotProject
  - 2.2. Permitir agregar un proyecto en OpenRefine
  - 2.3. Permitir agregar un proceso en la herramienta RapidMiner
  - 2.4. Permitir agregar un proyecto en la herramienta MantisBT
3. Permitir la integración de tareas
  - 3.1. Permitir la creación de una incidencia en MantisBT
  - 3.2. Permitir agregar una tarea en dotProject
4. Permitir la integración de archivos
  - 4.1. Permitir agregar un archivo en la herramienta MantisBT
  - 4.2. Permitir agregar un archivo en la herramienta dotProject

### **5.8.2. Requerimientos no funcionales**

Los atributos del sistema de software son los requerimientos no funcionales que deben cumplir los componentes de acoplamiento, y que se mostraron en la figura 5.5.

El ambiente estará abierto a futuras ampliaciones, otros componente podrán agregarse sin problema alguno.

## **5.9. Diseño del ambiente AMDADS**

En esta sección se describe el diseño de software de los componentes de acoplamiento para integrar herramientas FLOSS, en el ambiente de soporte a minería

de datos de desarrollo de software (AMDADS), que soportar principalmente la fase de preparación de datos.

Se muestra aquí el diseño del ambiente de soporte a minería de datos de desarrollo de software poniendo especial atención a la etapa de preparación de datos. Para este diseño se siguieron los requisitos definidos en el SRS y la estructura del documento de especificación de diseño resultante se apega a la propuesta en el estándar IEEE 1016.

A continuación se presentan los diagramas de componentes, secuencia y de clases, que se desarrollaron para representar elementos de la arquitectura o el diseño del ambiente AMDADS.

### **Diagrama de componentes**

En el diagrama de componentes de la figura 5.10, el componente Integrar usuarios es el encargado de monitorear la base de datos en dotProject y posteriormente agregar los usuarios en las bases de datos de las herramientas que lo utilicen.

El componente Integrar Proyectos es el encargado de monitorear la base de datos de dotProject para detectar la inserción y sincronizar sus datos con las bases de datos de las demás herramientas.

El componente Integrar Tareas es el encargado de monitorear la base de datos de MantisBT para detectar los cambios sobre las tareas y sincronizar los datos de las tareas con la base de datos de la herramienta dotProject.

El componente Integrar Archivos es el encargado de monitorear la base de datos de dotProject o MantisBT para detectar cuando se adjunta un archivo y hacer una copia en la herramienta complementaria.

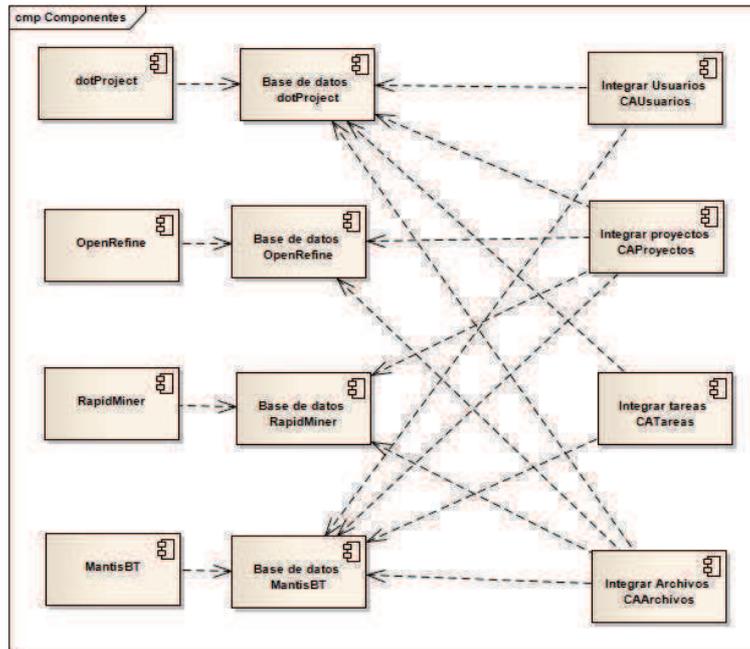


Figura 5.10 Diagrama de componentes

### Diagrama de secuencia

En el diagrama de secuencias de la figura 5.11 se describe la secuencia para el uso del componente CAProyectos. Se inicia con la creación de un nuevo proyecto por parte del usuario Administrador, la herramienta dotProject guarda el proyecto. El componente CAProyectos se encuentra monitoreando la base de datos para identificar la inserción de un nuevo registro, al detectarlo, consulta los datos del nuevo registro para crear un proyecto en cada una de las herramientas.

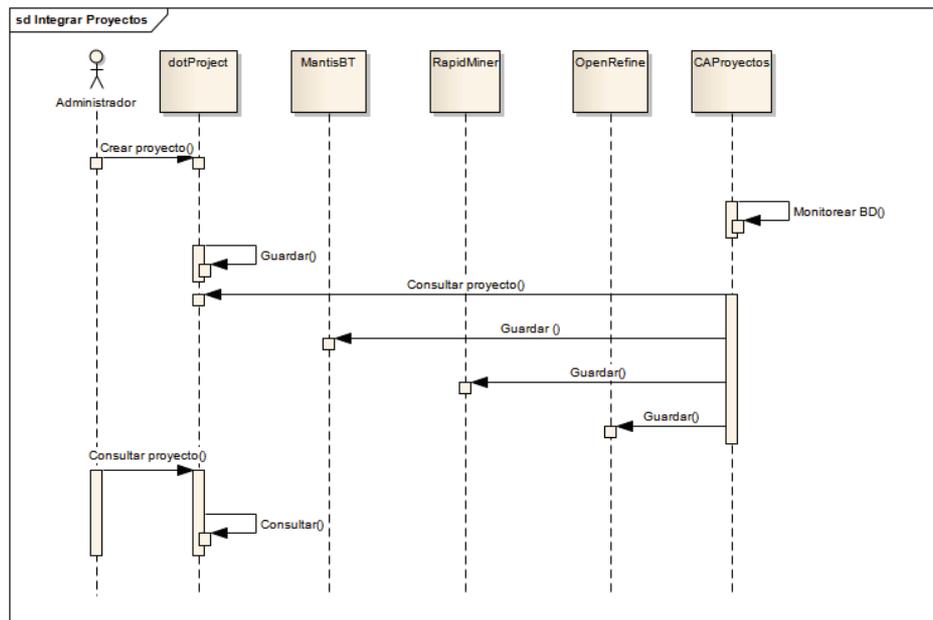


Figura 5.11 Diagrama de secuencia de la meta integrar proyectos

### Diagrama de clases

En el diagrama de la figura 5.12 se muestra la estructura de las clases que conforman los componentes de acoplamiento, la estructura se basa en el sistema de colaboración basada en servicios especificada por OSGI (del inglés *Open Services Gateway Initiative*), El componente CAStarter es el encargado de iniciar el sistema mediante la clase Activator. El componente CANúcleo es el encargado de recolectar en una lista las implementaciones de la interfaz CAI. Los componentes CAUsuarios, CAProyectos, CATareas y CAArchivos a través de su clase del mismo nombre, implementan la interfaz CAI.

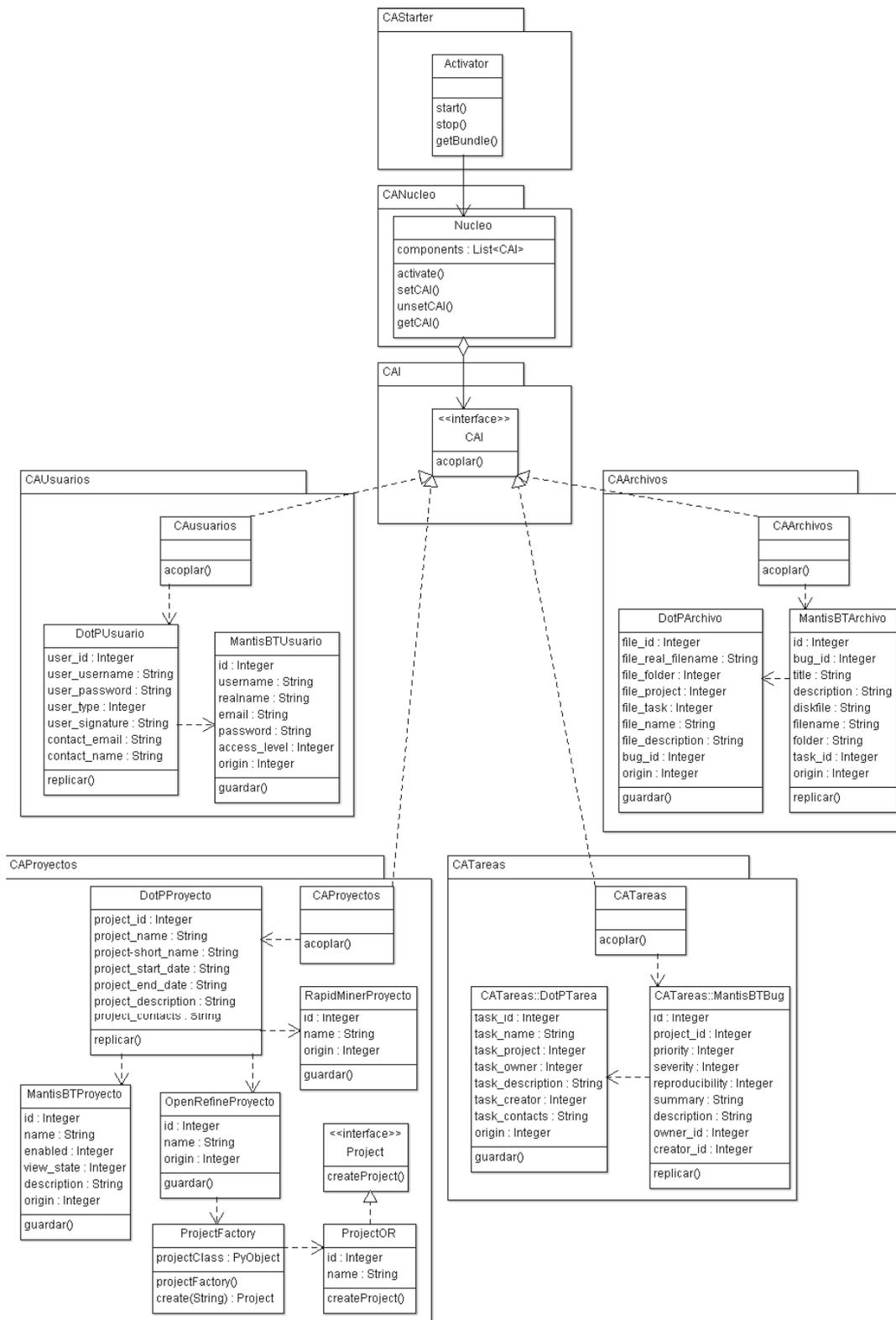


Figura 5.12 Diagrama de clases de AMDADS

Dentro de este capítulo se definieron los lenguajes GRL y UCM bajo los cuales se desarrollaron algunos diagramas presentados para definir los requerimientos funcionales y no funcionales, además se presentaron algunos diagramas UML tales como: diagrama de componentes, diagrama de secuencia y diagrama de clases. A partir de estos diagramas se desarrollaron los componentes de acoplamiento para integrar las herramientas que den soporte al proceso de minería de datos de desarrollo de software

En esta sección se presentan los resultados principales de la investigación, que se obtuvieron de la implementación del Ambiente de Soporte a Minería de Datos de Desarrollo de Software.



## 6.1. Implementación

Con base en los requerimientos y diseño del ambiente de soporte a minería de datos de desarrollo de software (AMDADS), presentados en el capítulo anterior, se realizó su implementación.

A continuación se describe la forma en que se implementaron los componentes de acoplamiento, así como la descripción de la interfaz de usuario, que es sencilla, presenta información y ayuda sobre el ambiente AMDADS.

### 6.1.1. Componentes de acoplamiento

La implementación de los componentes de acoplamiento se hizo siguiendo las especificaciones de OSGI, destinadas a desarrollar aplicaciones modulares usando el lenguaje de programación Java y que a su vez facilitan principalmente el desarrollo, mantenimiento y escalabilidad del sistema. Otra ventaja del uso de OSGI es el poder usar componentes versionados, así en el caso de que se requiera la ejecución de dos componentes pero diferentes versiones no haya problemas.

Los componentes de acoplamiento están desarrollados totalmente usando Java a excepción del componente encargado de la integración de proyectos (ver figura 5.12), que además de estar desarrollado en Java, fue necesario utilizar el lenguaje de programación Python debido al uso de la API (del inglés *Application Programming Interface*) de la herramienta OpenRefine, que no cuenta con una API oficial. Debido a esto, algunos usuarios han hecho aportaciones a la comunidad con API's personales escritas en los lenguajes de programación Python y Ruby.

Debido a esta situación, el componente encargado de integrar proyectos se desarrolló principalmente en Java, con clases en Python y Jython, (Jython es la implementación del lenguaje Python que corre sobre la plataforma Java).

Así, AMDADS está compuesto por diferentes herramientas libres y de código abierto que a través de los componentes de acoplamiento comparten datos automáticamente, además de mantener un repositorio de relaciones, el cual como su nombre lo indica, almacena las relaciones entre los elementos de las bases de datos que fueron replicados en las herramientas destino y el elemento de la base de datos de la herramienta original.

En la figura 6.1 se observa el flujo que inicia con los datos que ingresa el usuario en la herramienta inicial y son almacenados en su base de datos. Periódicamente el componente de acoplamiento detecta el nuevo registro y procede a extraer los datos para adaptarlos y almacenarlos en las bases de datos de las demás herramientas. En seguida se obtiene el id de cada nuevo registro para poder guardarlo en el repositorio de relaciones y así saber a qué registro corresponde cada replica. De esta forma los datos ingresados una sola vez por el usuario, se distribuyen y estarán disponibles en las bases de datos de las otras herramientas, en diferentes contextos.

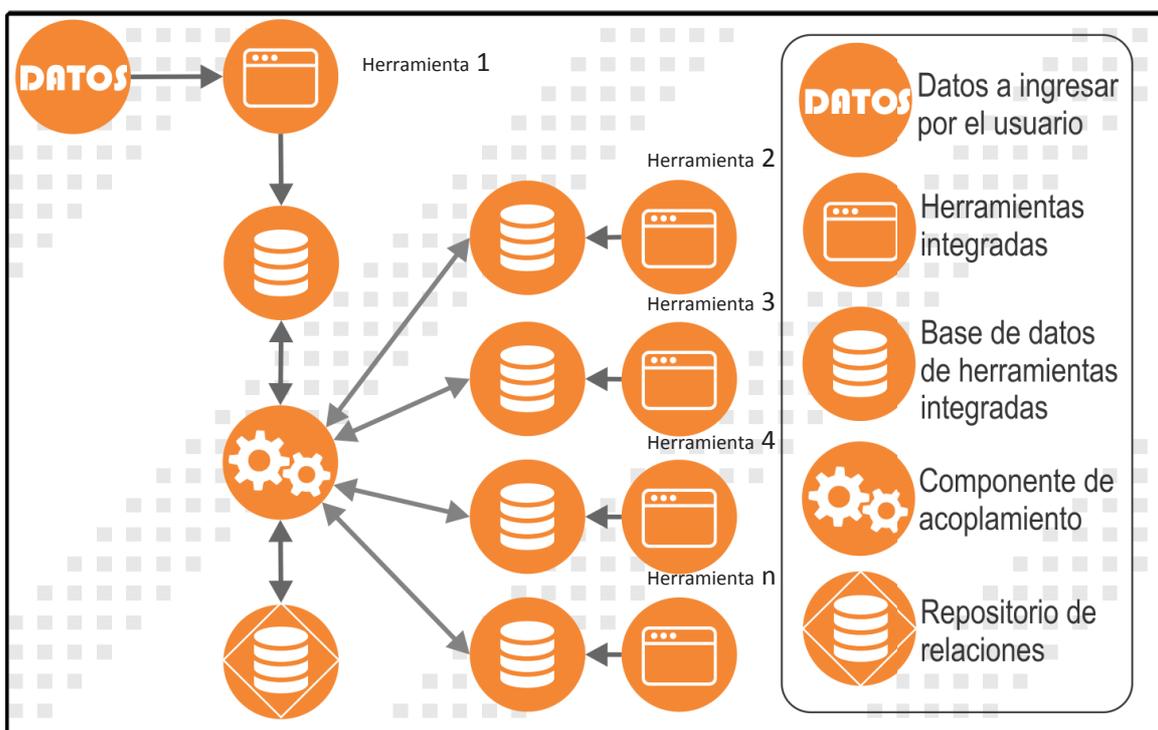


Figura 6.1 Modelo de integración de AMDADS

Con la implementación de estos componentes de acoplamiento se integraron las herramientas seleccionadas, de manera heterogénea debido a que se usaron dos tipos de integración: integración por datos e integración usando API's.

La integración usando datos se logra ya que las herramientas dotProject y MantisBT usan el gestor de base de datos MySQL para almacenar sus datos. Por su parte OpenRefine y RapidMiner generan directorios y archivos en donde se almacenan su información, como consecuencia de esto se procede al uso de sus respectivas API's para manipular desde código la creación de proyectos en estas herramientas.

### 6.1.2. Interfaz de usuario

Como parte del desarrollo de AMDADS se implementó una interfaz de usuario que sirve para dar información y ayuda sobre AMDADS. La interfaz consiste en una página de acceso (Figura 6.5) que está relacionada a la base de datos de usuarios de la herramienta dotProject. Esta desarrollada con las tecnologías JSF (del inglés Java Server Faces), que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE, Hibernate que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación java y PrimeFaces componente para JSF con el fin de crear aplicaciones RIA (del inglés *Rich Internet Applications*).

La interfaz de usuario es de gran ayuda, ya que proporciona información general del usuario y los proyectos que tiene asociados al momento además de la ayuda e información proporcionada como los son; los manuales de usuario y tutoriales tanto de las herramientas con del ambiente de soporte a minería de datos de desarrollo de software; información sobre las metodologías usadas para la elaboración de las plantillas; y acceso a las herramientas integradas.

A continuación se presentan los diagramas GRL y UCM relacionados a la interfaz de usuario.

El diagrama GRL de la figura 6.2 representa el objetivo principal de la interfaz que es ofrecer información sobre AMDADS, al que se puede llegar dando información sobre: el usuario, sus proyectos asociados, metodologías de minería de datos, las herramientas integradas e información general y ayuda

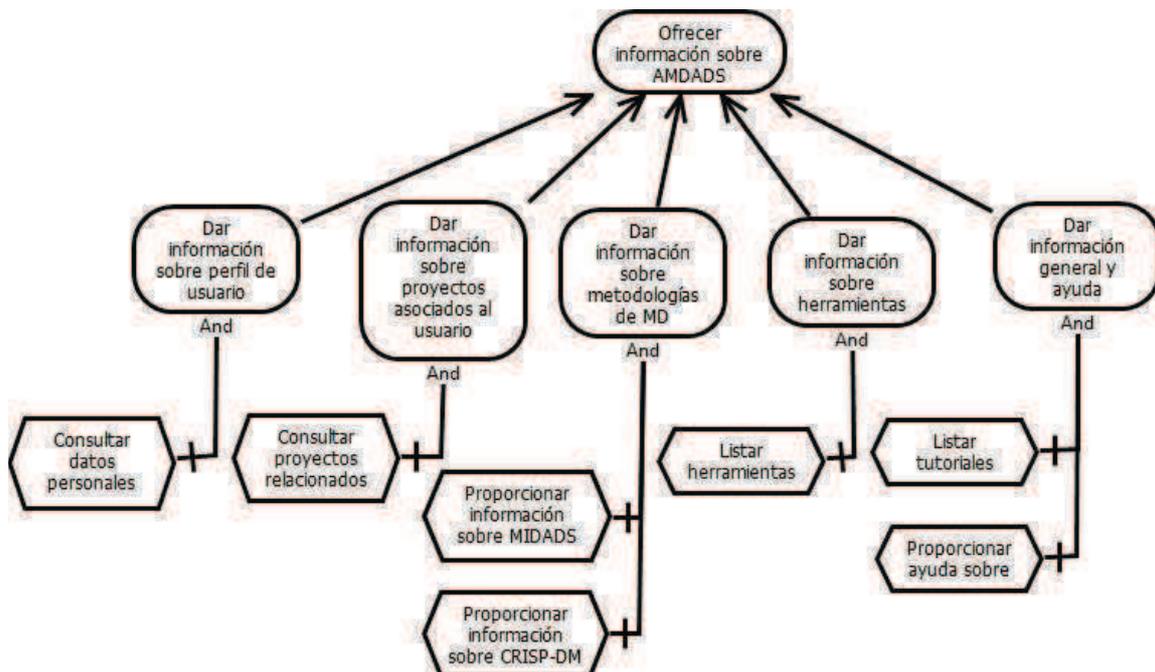


Figura 6.2 Diagrama GRL de la interfaz de usuario

El diagrama UCM de la figura 6.3 muestra que el usuario inicia ingresando sus credenciales para poder ingresar al sistema, una vez validados sus datos le muestra su perfil y el usuario puede elegir cualquiera de las opciones que muestra el menú, ya sea la opción Proyectos, Metodologías MD (Minería de Datos), Herramientas o Ayuda

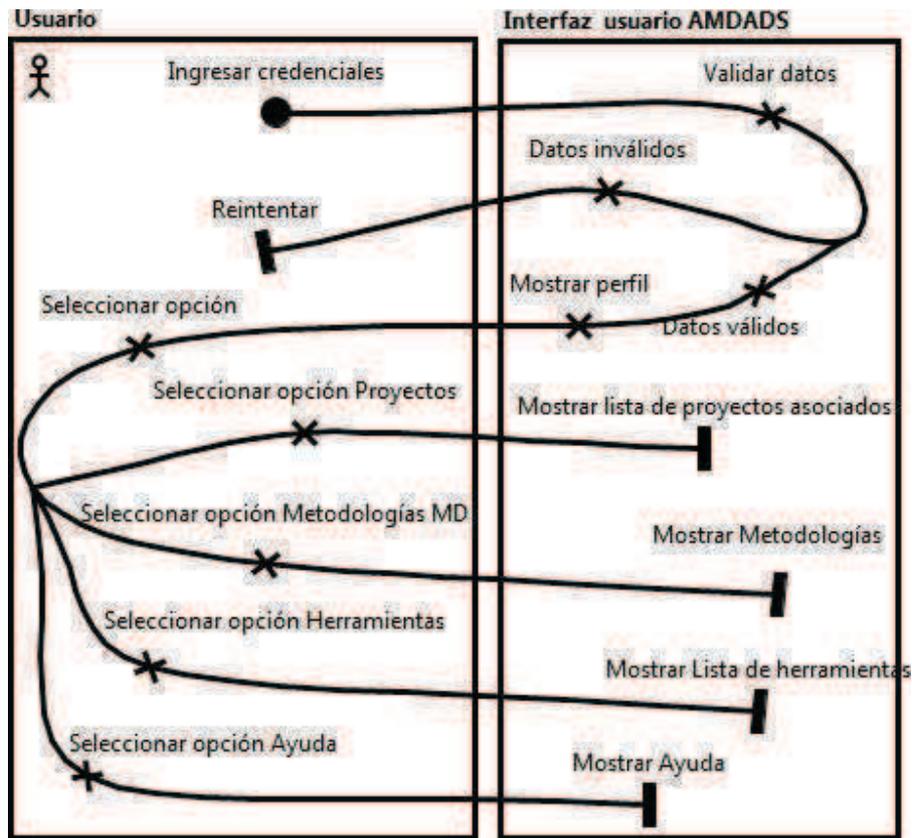


Figura 6.3 Diagrama UCM de la interfaz de usuario

Para comprender mejor el funcionamiento de la interfaz en la figura 6.4 se presenta un diagrama UML que corresponde a la interacción del usuario con la interfaz y esta a su vez se comunica con su base de datos.

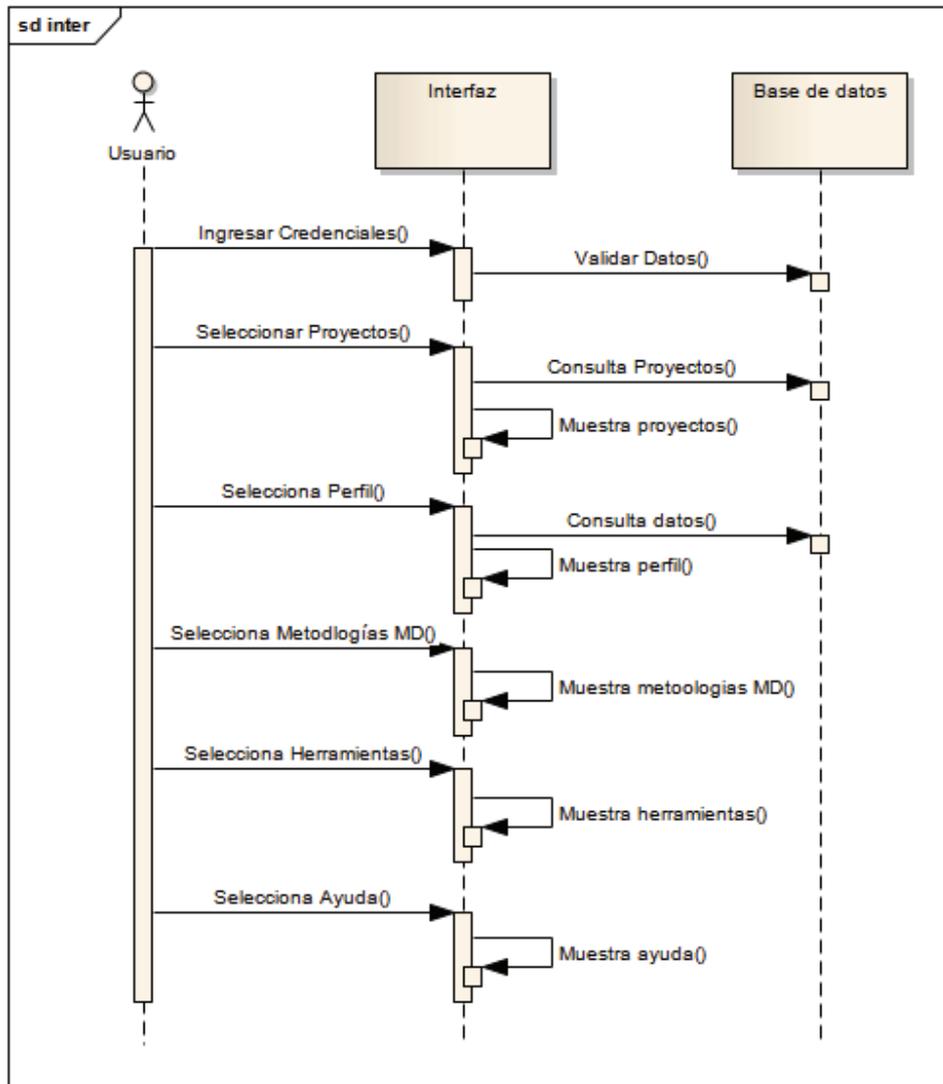
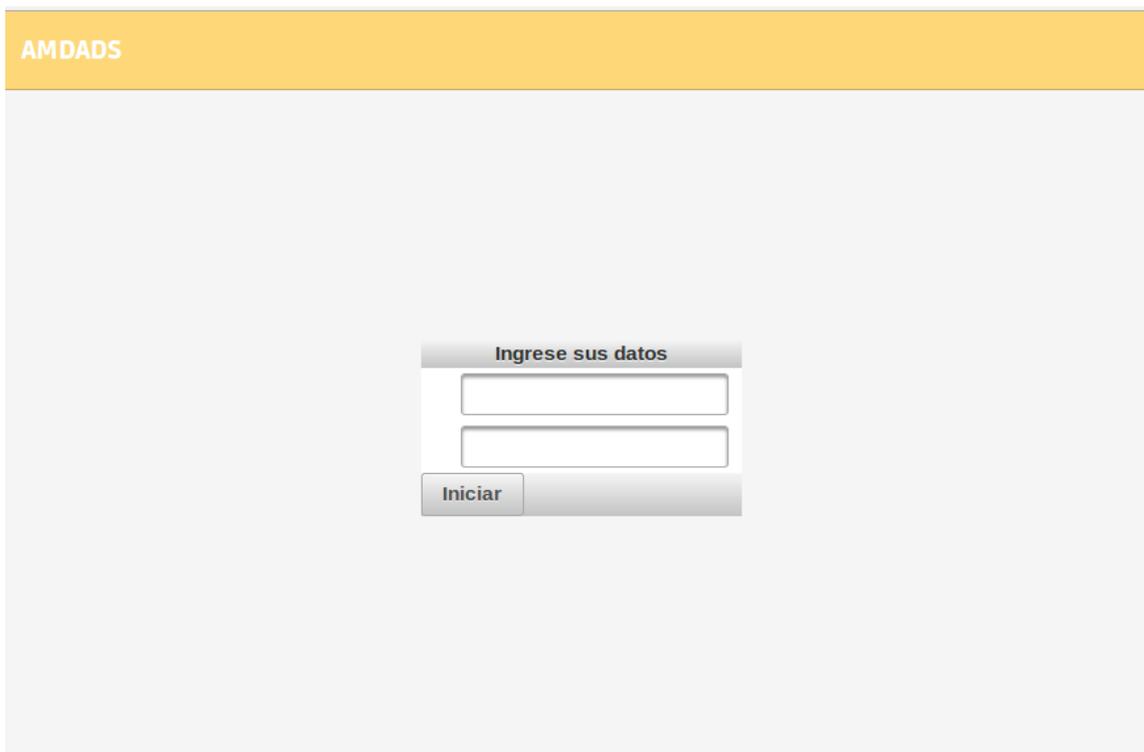


Figura 6.4 Diagrama de interacción de la interfaz de usuario



**Figura 6.5** Página de acceso

A continuación muestra un mensaje de bienvenida y la información relevante del usuario (ver Figura 6.6), cuenta con un menú (ver Figura 6.7) desde el cual puede consultar el perfil general del usuario, los proyectos que tiene asociados (ver figura 6.8), las metodologías sobre las que se basan las plantillas propuestas para la creación de proyectos, las herramientas que conforman AMDADS y ayuda sobre el ambiente y sobre cada una de las herramientas que lo conforman, cuenta con tutoriales y enlaces a páginas de ayuda específica. Respecto al diagrama UCM (ver Figura 6.3) la página de acceso (ver Figura 6.5) representa el primer paso, ingresar credenciales. La figura 6.6 representa la acción realizada en caso de ingresar credenciales válidas, muestra el perfil del usuario y un mensaje de bienvenida.

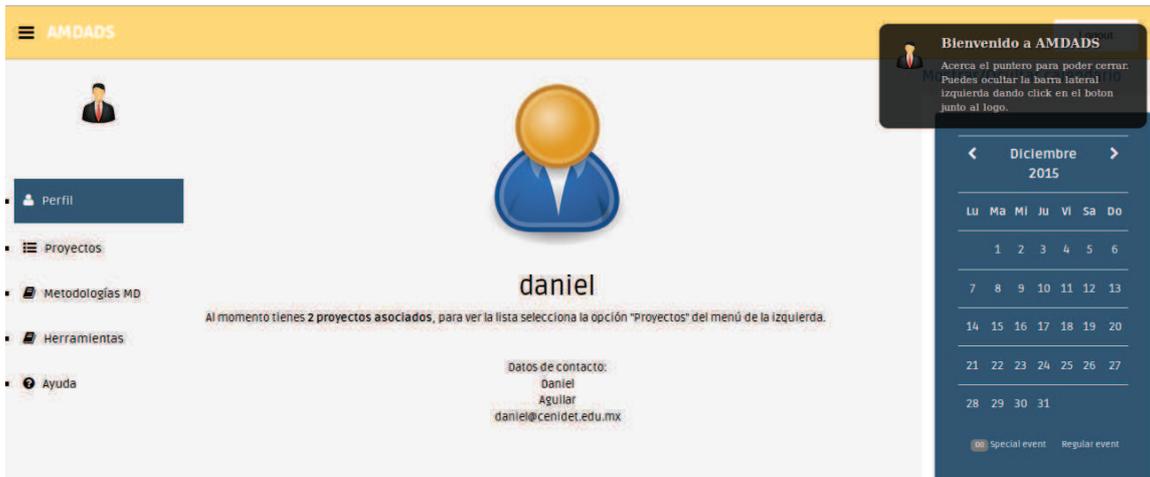


Figura 6.6 Mensaje de bienvenida

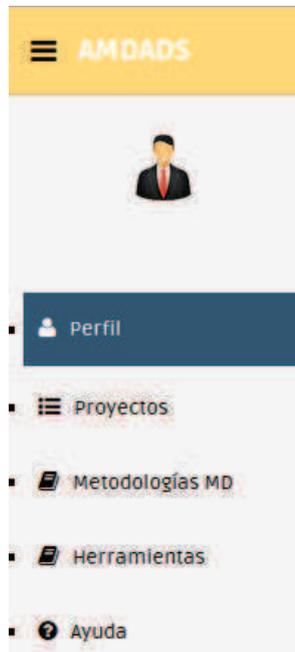


Figura 6.7 Menú principal de la interfaz

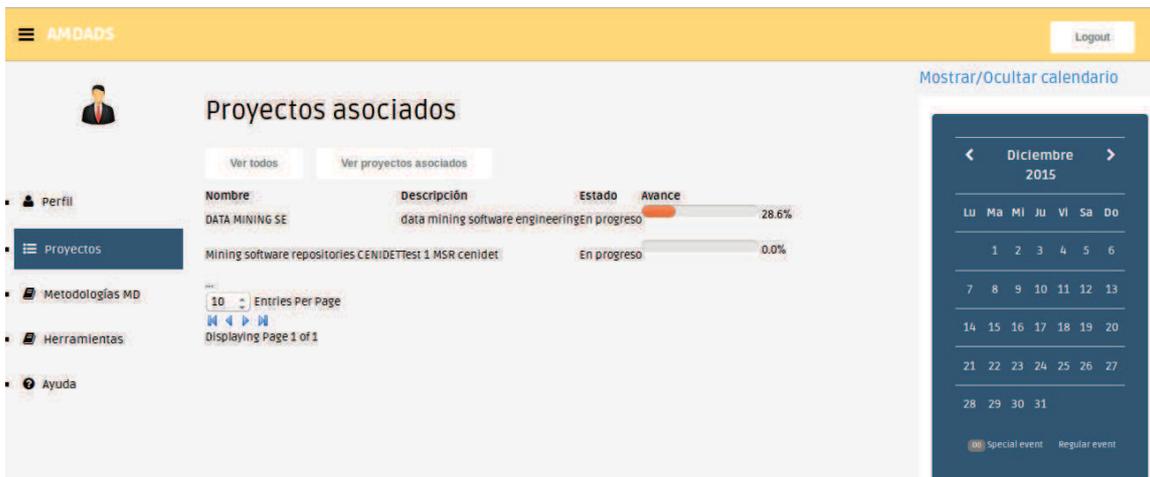


Figura 6.8 Interfaz de proyectos asociados al usuario

Según el diagrama UCM, (ver Figura 6.3) la siguiente opción disponible una vez validados sus datos es *seleccionar Metodologías MD*. Las metodologías sobre las que se basan las plantillas sugeridas con CRISP-DM y MIDADS, sobre estas plantillas se puede obtener información seleccionando la opción Metodologías MD del menú de la interfaz y se despliega la información sobre cada una de las plantillas, en la figura 6.9 se muestra la información sobre la metodología MIDADS que se recomienda usar.



Figura 6.9 Información sobre metodología MIDADS

Según el diagrama UCM, (ver Figura 6.3) la siguiente opción disponible una vez validados sus datos es *seleccionar Herramientas*. Las herramientas se listan desde la opción del menú *Herramientas* (Figura 6.10), se muestra la lista de fases de proceso de minería de datos CRISP-DM y las herramientas que cubren cada una de sus etapas (Figura 6.11).

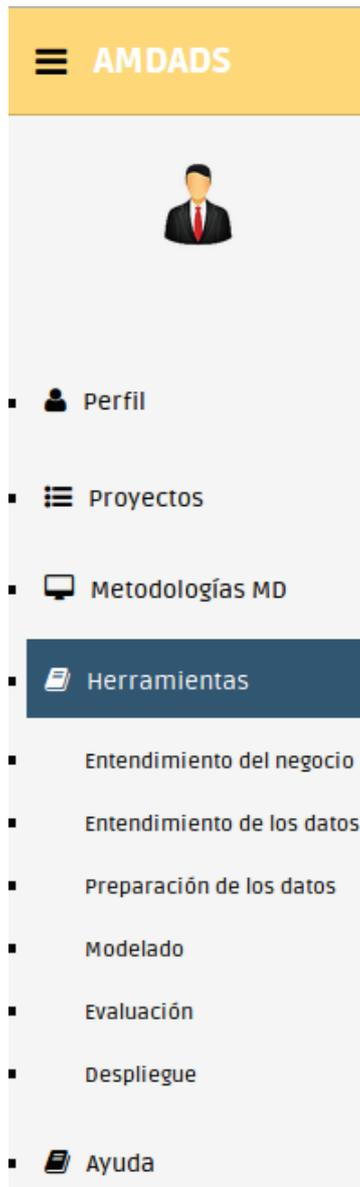


Figura 6.10 Menú herramientas

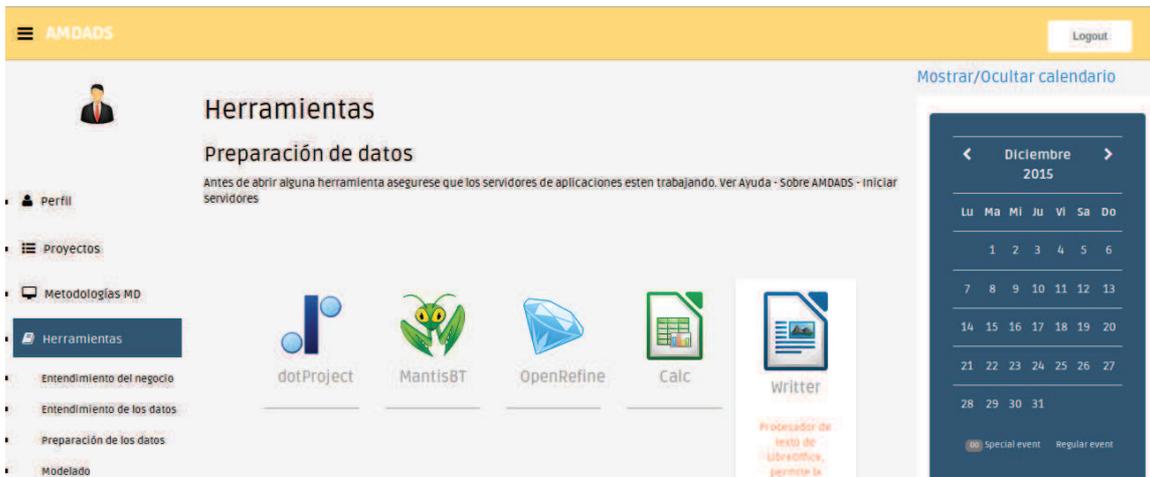


Figura 6.11 Herramientas que cubren y ayudan en la preparación de datos

Según el diagrama UCM, (ver Figura 6.3) la siguiente opción disponible una vez validados sus datos es *seleccionar Ayuda*. Sobre la ayuda proporciona información sobre AMDADS y tutoriales de cada una de las herramientas que integran el ambiente (Figura 6.12).



Figura 6.12 Enlaces a tutoriales en la sección Ayuda

En este capítulo se incluyeron diagramas GRL, UCM y UML que sirvieron para diseñar la interfaz de usuario, también se muestran imágenes de la interfaz para explicar la forma de interacción con el usuario.

En el siguiente capítulo se describen las pruebas y los resultados obtenidos.



En este capítulo se presentan los resultados obtenidos de la aplicación de las pruebas principales al ambiente de soporte a minería de datos de desarrollo de software.



Con el fin de validar el funcionamiento del ambiente de soporte a minería de datos de desarrollo de software (que denominamos AMDADS) se desarrollaron pruebas que se aplicaron a los resultados obtenidos. Dentro de este capítulo se incluye: el plan de pruebas, el diseño de las pruebas, los casos de pruebas, el procedimiento de pruebas, el registro de pruebas y el reporte de las pruebas.

### 7.1. Plan de pruebas

El plan de pruebas tiene como objetivo identificar los requisitos para las pruebas y desarrollar la estrategia para las pruebas. Esta sección está elaborada a partir del documento de plan de pruebas, este documento se encuentra en el anexo B.

#### 7.1.1. Identificador

A continuación, en la tabla 7.1, se lista la nomenclatura utilizada para identificar los documentos en la etapa de pruebas del ambiente de soporte a minería de datos de desarrollo de software AMDADS.

Tabla 7.1 Identificador del documento

Nomenclatura	Identificador	Descripción
CA	Nombre del proyecto	<b>CA</b> = Componentes de Acoplamiento para la infraestructura de soporte a minería de datos de desarrollo de software
AB	Tipo de documento	<b>TP</b> - Plan de Pruebas (Test Plan) <b>TD</b> - Diseño de Pruebas (Test Design) <b>TC</b> - Casos de Prueba (Test Case) <b>TPr</b> - Procedimientos de Pruebas (Test Procedure) <b>TL</b> - Registro de Pruebas (Test Log) <b>TR</b> - Reporte de Pruebas (Test Report)
XY	Numeración	Representa el número correspondiente al documento.

Por ejemplo para el primer caso de prueba se tiene la nomenclatura CA-TC-01, para el primer procedimiento de prueba CA-TPr-01 y así dependiendo del tipo de documento y su numeración.

#### 7.1.2. Alcance

El alcance del plan de pruebas y la realización de las tareas programadas, es verificar la funcionalidad, comportamiento y cumplimiento de los requerimientos funcionales

que se especificaron para la implementación de los siete componentes de acoplamiento los cuales son: CAProyectos, CAUsuarios, CATareas, CAArchivos, CANucleo, CAI y CASTarter. Además de la interfaz de usuario.

### 7.1.3. Matriz de Trazabilidad de Pruebas

Se provee la lista de requerimientos (Tabla 7.2) que debe cumplir el sistema y el componente de acoplamiento correspondiente a ese requerimiento, así como el identificador de su caso de prueba que se usará a lo largo de las pruebas. Los requerimientos Integrar Usuarios, Integrar Proyectos, Integrar Tareas e Integrar Archivos son los obtenidos en la especificación de requerimientos (ver Capítulo 5, figura 5.5). En la tabla 7.2 se incluyen espacios vacíos ya que el Componente de Acoplamiento relacionado no atiende directamente a ningún requerimiento.

**Tabla 7.2** Requerimiento y casos de prueba correspondiente

Requerimiento	Componente de Acoplamiento	Caso de Prueba
Integrar Usuarios	CAUsuarios	CA-TC-01
Integrar Proyectos	CAProyectos	CA-TC-02
Integrar Tareas	CATareas	CA-TC-03
Integrar Archivos	CAArchivos	CA-TC-04
-	CANucleo	CA-TC-05
-	CAI	CA-TC-06
-	CASTarter	CA-TC-07

Otro elemento a probar es la interfaz de usuario de AMDADS. Debido a la importancia de dar información al usuario sobre AMDADS, lo que implica más casos de prueba (ver Figura 7.3)

**Tabla 7.3** Interfaz y caso de prueba correspondiente

Elemento a probar	Características	Caso de prueba
Interfaz	<ul style="list-style-type: none"> <li>Autenticación de usuarios</li> <li>Listado de proyectos asociados</li> </ul>	CA-TC-08

**7.1.4. Criterio para Suspensión y Reanudación de Pruebas**

La suspensión de las pruebas está regida por la presencia de fallas en los componentes que impidan el cumplimiento de sus funcionalidades, al registrarse una falla se suspende la prueba y se reanuda al ser corregida. La prueba es finalizada cuando se tiene el resultado de la ejecución del componente de acoplamiento respectivo.

**7.2. Diseño de pruebas**

Esta sección contiene las siguientes subsecciones: características a probar, los casos de prueba así como los criterios de aceptación. Esta sección está elaborada a partir del documento de diseño de pruebas, este documento se encuentra en el anexo B.

**7.2.1. Características a Probar**

El diseño de pruebas abarca las características referentes a los siguientes requerimientos: integrar usuarios, integrar proyectos, integrar tareas e integrar archivos. A continuación en la figura 7.4 se muestra una matriz de trazabilidad de pruebas, en donde se muestran las características que van a probarse, así como su caso y procedimiento de prueba correspondiente, ubicados en las secciones 7.3 y 7.4 de este capítulo respectivamente.

Tabla 7.4 Matriz de trazabilidad

Requerimiento	Componente de Acoplamiento	Característica a probar	Caso de Prueba	Procedimiento de Prueba
Integrar Usuarios	CAUsuarios	Crear cuenta de usuario dotProject Crear cuenta de usuario en MantisBT Registrar la relación entre cuentas de usuario	CA-TC-01	CA-TPr-01
Integrar Proyectos	CAProyectos	Crear un proyecto en dotProject Crear proyecto en MantisBT Crear proyecto en OpenRefine Crear proceso en RapidMiner Registrar la relación	CA-TC-02	CA-TPr-02

		entre proyectos		
Integrar Tareas	CATareas	Crear defecto en MantisBT Crear tarea en dotProject Registrar la relación entre defecto-tarea	CA-TC-03	CA-TPr-03
Integrar Archivos	CAArchivos	Adjuntar archivo a defecto de MantisBT o tarea de dotProject Adjuntar archivo a defecto de MantisBT o tarea de dotProject Registrar la relación entre archivos	CA-TC-04	CA-TPr-04
-	CANucleo	Este componente es el encargado de coleccionar los componentes que implementen la interfaz del componente CAI.	CA-TC-05	CA-TPr-05
-	CAI	Proporciona la interfaz que enlazará a los componentes que implementan dicha interfaz con el componente CANucleo.	CA-TC-06	CA-TPr-06
-	CASarter	Iniciar, detener o reanudar el sistema.	CA-TC-07	CA-TPr-07

Dentro de los elementos a probar se incluye la interfaz de usuario, en la tabla 7.5 se muestra la matriz de trazabilidad para la interfaz de usuario.

Tabla 7.5 Matriz de trazabilidad para la interfaz

Componente a probar	Característica	Caso de Prueba	Procedimiento de Prueba
Interfaz de usuario	<ul style="list-style-type: none"> <li>Autenticación de usuarios</li> <li>Listado de proyectos asociados</li> </ul>	CA-TC-08	CA-TPr-08

### 7.2.2. Estrategia de Pruebas

La realización de las pruebas será de la siguiente manera: para los componentes CAUsuarios y CAProyectos se utilizarán datos de proyectos/usuarios ficticios y se

registraran en la herramienta dotProject, para probar el componente CATareas, al igual se usarán datos de defectos ficticios y se registrarán en MantisBT, para probar el componente CAArchivos se usarán archivos en diferentes formatos. El resultado esperado es el cumplimiento de cada uno de los requerimientos establecidos anteriormente (ver Capitulo 5, sección 5.8.1 requerimientos funcionales).

### 7.2.3. Criterio de Aceptación del Caso de Prueba

Para que las pruebas se aprueben deben cumplir con cada una de las funcionalidades especificadas en los requerimientos sin ninguna clase de contratiempo.

### 7.3. Casos de prueba

En este apartado se incluyen las subsecciones: identificador del caos de prueba, las entradas necesarias y las salidas anticipadas. Esta sección está elaborada a partir del documento de casos de pruebas, este documento se encuentra en el anexo B.

Se describe el caso de prueba CA-TC-02 correspondiente al componente de acoplamiento CAProyectos que cubre el requerimiento integrar proyectos. Los demás casos de prueba se presentan en anexo B, de este documento.

#### 7.3.1. Identificador del Caso de Prueba

En la tabla 7.6 se describe el requerimiento funcional, el componente de acoplamiento al que corresponde y el caso de prueba relacionado.

Tabla 7.6 Caso de prueba Integrar Proyectos

Requerimiento funcional	Componente de Acoplamiento	Caso de Prueba
1. Integrar Proyectos	CAProyectos	CA-TC-02

#### 7.3.2. Objetivo

El objetivo de este caso de prueba es validar el cumplimiento del requerimiento **integrar proyectos** que corresponde al componente de acoplamiento **CAProyectos**.

### 7.3.3. Entradas

La entrada son los datos ingresados en la herramienta dotProject (ver Tabla 7.7), en este caso son datos relacionado a un proyecto, para posteriormente extraerla y almacenarla en las demás herramientas.

**Tabla 7.7** Datos de entrada para dotProject

<b>Nombre del campo</b>	<b>Tipo</b>
Nombre	Cadena
Propietario	Cadena
Empresa/compañía	Cadena
Fecha de inicio	Fecha
Fecha de fin objetivo	Fecha
Prioridad	Cadena
Nombre corto	Cadena
Color de identificación	Cadena
Tipo de proyecto	Cadena
Estado del proyecto	Cadena
Descripción del proyecto	Texto

### 7.3.4. Salidas

La salida de CAProyectos es el registro automático de un proyecto en cada una de las demás herramientas MantisBT, OpenRefine y un proceso en RapidMiner, además de registrar la relación de proyectos en la base de datos CABD.

En seguida se describen las salidas esperadas en cada herramienta y en la base de datos CABD.

#### 7.3.4.1. MantisBT

La salida esperada en MantisBT es el registro de un nuevo proyecto con los datos del registrado en dotProject, en la tabla 7.8 se muestran los campos esperados en la herramienta MantisBT.

**Tabla 7.8** Salida en herramienta MantisBT

<b>Nombre del campo</b>	<b>Tipo</b>
Nombre	Cadena
Descripción	Texto
Estado	Cadena

### 7.3.4.2. OpenRefine

La salida esperada en OpenRefine es el registro de un nuevo proyecto con los datos del registrado en dotProject, en la tabla 7.9 se muestran los campos esperados en la herramienta OpenRefine.

**Tabla 7.9** Salida en herramienta OpenRefine

Nombre del campo	Tipo
Nombre	Cadena

### 7.3.4.3. RapidMiner

La salida esperada en RapidMiner es el registro de un nuevo proceso con los datos del registrado en dotProject, en la tabla 7.10 se muestran los campos esperados en la herramienta MantisBT.

**Tabla 7.10** Salida en herramienta RapidMiner

Nombre	Tipo
Directorio	Cadena
Nombre	Cadena

### 7.3.4.4. Tabla de relaciones

La salida esperada en la base de datos de relaciones (CABD) y específicamente en la tabla proyectos es el registro de los valores del id de cada nuevo proyecto creado en la tabla 7.11 se muestran los campos esperados en la tabla de relaciones.

**Tabla 7.11** Tabla de relaciones de proyectos integrados

Nombre	Tipo
id	Entero
dotproject	Entero
rapidminer	Entero
openrefine	Long
bugtracker	Entero

## 7.4. Procedimiento de pruebas

En esta sección se especifica la secuencia de acciones para la ejecución del conjunto de pruebas usadas para analizar los componentes de software con el fin de evaluar el

conjunto de características. Lo que incluye el alcance del procedimiento de pruebas, las entradas y salidas. Esta sección está elaborada a partir del documento de procedimiento de pruebas, este documento se encuentra en el anexo B.

#### **7.4.1. Alcance**

En el alcance de los procedimientos de prueba se encuentran los casos de prueba CA-TC-01, CA-TC-02, CA-TC-03, CA-TC-04, CA-TC-05, CA-TC-06, CA-TC-07, CA-TC-08 que se definieron en el documento de casos de prueba. Como ejemplo, a continuación se muestra el procedimiento correspondiente al CA-TC-02. Los demás procedimientos se anexan a este documento, en el anexo B.

#### **7.4.2. Entradas, Salidas, y requerimientos especiales**

Para la ejecución adecuada del caso de prueba CA-TC-02 son necesarios los siguientes requisitos: la documentación para poder llevar a cabo las pruebas, los servidores para poder ejecutar AMDADS, las aplicaciones (herramientas) y los datos a usar, a continuación se enlistan estos requisitos.

##### **Documentos:**

- Plan de Pruebas
- Diseño de Pruebas
- Procedimiento de Pruebas
- Reporte de Pruebas

##### **Servidores:**

- Apache
- OSGI

##### **Aplicaciones:**

- Firefox
- dotProject
- MantisBT
- OpenRefine
- RapidMiner

- Componente de acoplamiento CAProyectos

**Datos:**

- Datos sobre proyectos

### 7.4.3. Descripción del conjunto de pasos a seguir para ejecutar el caso de prueba CA-TC-02

**Preparar herramientas, servidores y datos a usar:** Antes de iniciar la ejecución del procedimiento se deben iniciar las aplicaciones (herramientas) y servidores. Además se deben tener listos los datos del nuevo proyecto que se va a registrar.

**Iniciar:** Usando el navegador Firefox se debe registrar un nuevo proyecto en la herramienta dotProject.

**Proceso:** En este caso es el componente CAProyectos el que participa, identifica el nuevo registro en la base de datos y recolecta los datos necesarios para registrar el proyecto en las demás herramientas siguiendo esta secuencia: MantisBT, RapidMiner y OpenRefine. Una vez hecho esto el componente registra la relación entre los proyectos en la base de datos CABD.

**Medir:** Las herramientas MantisBT, RapidMiner y OpenRefine deben tener registrado el nuevo proyecto.

**Cierre:** La prueba termina con el registro en el documento de reporte de pruebas.

La elaboración de los reportes es de forma manual al hacer una inspección visual de los eventos que ocurren a lo largo de la ejecución de la prueba y serán registrados en el documento de reporte de pruebas, de esta misma forma los errores, problemas o anomalías serán registrados en el mismo documento.

Al presentarse anomalías la prueba tiene que ser suspendida y el error o fallo corregidos. Para poder reiniciar la prueba de debe haber corregido la anomalía.

La prueba se puede reiniciar en cualquier momento ejecutando el componente CAProyectos teniendo en cuenta el paso de preparación de esta sección.

Debido al uso de diferentes API's y librerías usadas en la creación de proyectos, la ejecución del componente CAProyectos tiende a tardar más de 30 segundos, la prueba puede ser suspendida después de este tiempo.

### 7.5. Registro de pruebas

Esta sección tiene la función de proveer un registro cronológico de detalles relevantes sobre la ejecución de las pruebas. Esta sección está elaborada a partir del documento de registro de pruebas, este documento se encuentra en el anexo B.

#### 7.5.1. Alcance

El alcance de los registros de pruebas abarca los procedimientos de prueba CA-TPr-01, CA-TPr-02, CA-TPr-03, CA-TPr-04, CA-TPr-05, CA-TPr-06, CA-TPr-07, CA-TPr-08 definidos en el documento de Procedimientos de Pruebas. A continuación se describe el registro para el procedimiento de prueba CA-TPr-02, así como sus actividades y eventos. Se anexan a este documento los procedimientos de prueba restantes, en el anexo B. A continuación se da una breve descripción del registro de prueba, las actividades y eventos realizados durante la prueba, las anomalías encontradas (en caso de existir) y ejemplos de datos utilizados.

#### 7.5.2. Descripción

El registro de prueba CA-TL-02 cubre el procedimiento de prueba CA-TPr-02 correspondiente al caso de prueba CA-TC-02. En la ejecución de la prueba se usaron datos ficticios.

#### 7.5.3. Actividades y eventos de la prueba

A continuación, en la tabla 7.12 se describe el registro de eventos y actividades al ejecutar el procedimiento de prueba CA-TPr-02.

**Tabla 7.12** Registro de eventos para CA-TPr-02

Paso	Actor	Descripción	Verificado
1	Encargado de pruebas	Prepara datos a ingresar y documentación de pruebas.	Si
2	Servidor Apache	Inicia servidor web	Si
3	MySQL	Inicia gestor de bases de datos	Si
4	OSGI	Inicia servidor OSGI y componente CAProyectos	Si

5	OpenRefine	Iniciar el servidor OpenRefine	
6	Encargado de pruebas	Entra a dotProject a través de Mozilla Firefox	Si
7	Encargado de pruebas	Registrar un nuevo proyecto a través del formulario proporcionado por dotProject	Si
8	CAProyectos	Identifica el registro de un nuevo proyecto en dotProject.	Si
9	CAProyectos	Extrae los datos necesarios del nuevo proyecto	Si
10	CAProyectos	Agrega un nuevo proyecto en MantisBT con los datos extraídos	Si
11	CAProyectos	Agrega un nuevo proyecto en OpenRefine con los datos extraídos	Si
12	CAProyectos	Agrega un nuevo proceso en RapidMiner con los datos extraídos	Si
13	CAProyectos	Registra la relación entre los proyectos en la base de datos CABD.	Si
14	CAProyectos	Se pone en modo de espera	Si
15	Encargado de pruebas	Verifica la existencia del nuevo proyecto en MantisBT	Si
16	Encargado de pruebas	Verifica la existencia del nuevo proyecto en OpenRefine	Si
17	Encargado de pruebas	Verifica la existencia del nuevo proceso en RapidMiner	Si
18	Encargado de pruebas	Documentar la prueba	Si

#### 7.5.4. Anomalías

A continuación, en la tabla 7.13 se describen las anomalías encontradas en las pruebas. La columna Paso contiene el número del paso de la lista anterior en el que se originó la anomalía. Se presenta una breve descripción, así como la solución implementada.

**Tabla 7.13** Anomalías de CA-TPr-02

Paso	Descripción	Solución
13	Se agregó el nuevo registro con los valores de los id's correctos a excepción del id correspondiente al campo OpenRefine, aquí se agregó un -1	Al crear el proyecto en OpenRefine el método setID se llamaba dos veces, una pasando el id correcto y después se le asignaba un valor de -1, así que se eliminó la segunda llamada

### 7.5.5. Ejemplo de datos utilizados para la prueba

A continuación, en la tabla 7.14 se muestran los datos de un proyecto registrado con el fin de probar el requerimiento funcional *Integrar Proyectos*.

**Tabla 7.14** Ejemplo de datos utilizados para CA-TPPr-02

Nombre del campo	Tipo	Valor
Nombre	Cadena	Mining software repositories
Propietario	Cadena	Admin
Empresa/compañía	Cadena	CENIDET
Fecha de inicio	Fecha	03/09/2015
Fecha de fin objetivo	Fecha	08/10/2015
Prioridad	Cadena	normal
Nombre corto	Cadena	msr
Color de identificación	Cadena	990000
Tipo de proyecto	Cadena	Operative
Estado del proyecto	Cadena	In Planning
Importar tareas de	Cadena	1 Plantilla MIDADS (recomendado)
Descripción del proyecto	Texto	Test – Mining software repositories

### 7.6. Reporte de pruebas

En esta sección se resumen las actividades de prueba y sus resultados, además contiene evaluaciones de las correspondientes pruebas. Se define el alcance del reporte de pruebas, los detalles y resultados de las pruebas. Esta sección está elaborada a partir del documento de reporte de pruebas, este documento se encuentra en el anexo B.

#### 7.6.1. Alcance

El alcance de esta documentación se centra en los casos de prueba descritos en el documento de casos de prueba y son los siguientes: CA-TC-01, CA-TC-02, CA-TC-03, CA-TC-04, CA-TC-05/06/07 (se incluyen tres casos de prueba en uno debido a que estos tres casos están relacionados a los componentes CAStarter, CANucleo y CAI, para la ejecución de estos componentes solo es necesario iniciar o detener cada uno) y CA-TC-08.

#### 7.6.2. Detalles

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

### 7.6.2.1. Resultados Generales

La tabla 7.15 incluye el requerimiento a probar, el componente de acoplamiento encargado de dicho requerimiento, el caso de prueba correspondiente y el resultado de la prueba, se incluyen espacios vacíos ya que el componente de acoplamiento no está relacionado directamente con ningún requerimiento.

**Tabla 7.15** Resultados generales de las pruebas

Requerimiento	Componente de Acoplamiento	Caso de Prueba	Resultado de la Prueba
Integrar Proyectos	CAProyectos	CA-TC-01	Exitoso
Integrar Usuarios	CAUsuarios	CA-TC-02	Exitoso
Integrar Tareas	CATareas	CA-TC-03	Exitoso
Integrar Archivos	CAArchivos	CA-TC-04	Exitoso
-	CANucleo	CA-TC-05	Exitoso
-	CAI	CA-TC-06	Exitoso
-	CASstarter	CA-TC-07	Exitoso
-	Interfaz de usuario	CA-TC-08	Exitoso

Después de corregir las anomalías internas fueron corregidas. Las pruebas fueron realizadas bajo condiciones normales y no se presentaron anomalías externas. Al final todas las pruebas fueron exitosas.

### 7.6.2.2. Resultados Detallados

A lo largo de las pruebas se presentaron anomalías que van desde pequeñas que permitían al sistema seguir su ejecución normal (en la tabla 7.13 se puede ver un ejemplo de esta anomalía) hasta anomalías mayores que detenían por completo la ejecución del sistema (ver Anexo B tabla B.37 para ver un ejemplo de esta anomalía), al final todas fueron corregidas. Todo esto resultó en el cumplimiento de los requerimientos por completo.

Los proyectos generados en cada herramienta son complementarios para cumplir con el proceso de minería de datos de desarrollo de software, es decir, existe un proyectos destinado a la planeación de proyectos relacionado a un proyecto de seguimiento de incidencias, a su vez relacionado a un proyecto de tratamiento de datos y un proyecto relacionado de minería de datos, pero aún faltan herramientas que harían el sistema más completo.

El componente *CAProyectos* encargado del requerimiento *Integrar Proyectos* tarda más que los otros componentes debido a la implementación de API's, aun así no afecta en la ejecución del componente, solo se retrasa unos segundos más la creación de los proyectos en las herramientas RapidMiner y OpenRefine.

Dentro de este capítulo se incluyó la forma de realizar las pruebas al ambiente soporte a minería de datos de desarrollo de software, así como los resultados obtenidos de esas pruebas. En el siguiente capítulo se plasman las conclusiones a las que se llega después de realizar este trabajo de investigación además de hacer mención de posibles trabajos futuros sugeridos.

## CONCLUSIONES Y TRABAJOS FUTUROS

## Capítulo 8

En este capítulo se detallan las conclusiones a las que se llega además de los trabajos futuros que se originan a partir de los resultados obtenidos al término de esta investigación.



## 8.1. Conclusiones

En esta investigación se demuestra la factibilidad de la implementación del ambiente de soporte a minería de datos de desarrollo de software (AMDADS). Este ambiente se integra por: herramientas que cubren las necesidades de una o más etapas del proceso de minería de datos; componentes de acoplamiento que comparten datos entre las herramientas integradas; y por una interfaz de usuario de información y ayuda sobre AMDADS.

El ambiente resultante, de esta investigación, no incluye modificación o eliminación de registros (especifica los registros a que te refieres), pero se muestra el beneficio del uso de un ambiente integrado, pues facilita la realización de tareas como las siguientes: incluye ejemplos. Así también se demuestran los beneficios de implementar los componentes de acoplamiento basados en las especificaciones OSGI, entre los que están: el versionado de componentes, provee una clara definición de su modularidad, etc.

Todo el proceso realizado en la investigación se desarrolló siguiendo metodologías y se documentó. Desde la selección de herramientas hasta las pruebas del ambiente.

Durante el proceso de desarrollo de AMDADS se afrontaron varios retos, entre los que están:

- a) el cifrado que manejan las herramientas para almacenar contraseñas, para esto necesitó el almacenamiento de las contraseñas usando el tipo de cifrado AES de MySQL;
- b) la integración de un módulo Python con el componente de acoplamiento CAProyectos desarrollado en Java. Para esto se recurrió al uso de Jython que es una implementación del lenguaje de programación Python para la plataforma Java. De esta forma se pudo hacer uso de la API de OpenRefine y poder integrarla al ambiente;
- c) la extracción de datos acerca de los usuarios y sus proyectos, así como su preparación para hacerlos coincidir con los registros en las diferentes bases de datos, se realizó mediante el estudio de las bases de datos de las

herramientas a integrar y mediante las funcionalidades que brinda el lenguaje de programación Java (cambiar tipos de datos) bajo el que están desarrollados los componentes de acoplamiento.

Pese a los retos y las complicaciones encontradas durante la implementación de AMDADS se llegó exitosamente al objetivo planteado al inicio de la investigación. El cual es: Integrar herramientas mediante el uso de sus bases de datos con el fin de evitar la captura de datos repetitiva y eliminar la incompatibilidad entre formatos de entrada y salida.

Las aportaciones de esta investigación son:

- La implementación exitosa del ambiente AMDADS, que integra de manera heterogénea, mediante datos e interfaz de aplicación (API), las herramientas: enumera las herramientas.
- El uso de la metodología de minería de datos de desarrollo de software (MIDADS) para la creación de plantillas de proyectos que incluyen tareas previamente definidas en la herramienta dotProject.
- La implementación de las especificaciones OSGI para el desarrollo de los componentes de acoplamiento. El uso de OSGI proporcionan una clara definición de su modularidad, así como permite el versionado de un componente y así poder usar las dos versiones del componente al mismo tiempo.

## 8.2. Trabajos futuros

La implementación del ambiente de soporte al proceso de minería de datos de desarrollo de software (AMDADS) fue exitosa por lo que se recomienda seguir en esta línea de investigación y se proponen como trabajos futuros las siguientes investigaciones:

1. Debido a que AMDADS solo se encarga de replicar los datos al momento de dar de alta un registro, se recomienda ampliar las funcionalidades necesarias para modificar o eliminar registros.
2. Dado que la etapa cubierta principalmente es la preparación de datos, así como en menor grado etapas iniciales del proceso de minería de datos. Se

pueden incluir herramientas adicionales, que cubran en su totalidad otras etapas del proceso de minería de datos de desarrollo de software.

3. Cuando se incluyan nuevas herramientas será necesario realizar el estudio de factibilidad para usar tipos de integración diferentes a los usados en este proyecto.
4. También será necesario promover el uso del ambiente AMDAS, en trabajos de investigación acerca de minería de datos en datos de proyectos de software.



## REFERENCIAS

- (Meza 2014) Meza Bazán, María del Rosario. (2014, Junio). Especificación de Requerimientos para un Ambiente de Soporte al Proceso de Minería de Datos Aplicado a Repositorios de Datos de Desarrollo de Software. Tesis de maestría en Ciencias Computacionales, CENIDET, Cuernavaca, Morelos.
- [Sánchez 2010] Sánchez Santamaría, Miriam. (2010, Enero). Evaluación de Técnicas de Comparación de Diferentes Grupos de Características de Proyectos de Software. Tesis de maestría en Ciencias Computacionales, CENIDET, Cuernavaca, Morelos.
- (Iturbide 2013) Iturbide Domínguez, Gregorio Emmanuel. (2013, Febrero) Metodología de Preparación de Datos Orientada a Aplicaciones de Epidemiología Basada en el Modelo CRISP-DM. Tesis de maestría en Ciencias Computacionales, CENIDET, Cuernavaca, Morelos.
- (Sánchez 2010) Sánchez Santamaría, Miriam. (2010, Enero). Evaluación de Técnicas de Comparación de Diferentes Grupos de Características de Proyectos de Software. Tesis de maestría en Ciencias Computacionales, CENIDET, Cuernavaca, Morelos.
- (González 2006) González García, Moisés. (2006, Agosto). Método de Desarrollo Arquitectónico en Grupo. Tesis doctoral, Centro de Investigaciones y Estudios Avanzados (CINVESTAV) del Instituto Politécnico Nacional.

- (Delgado 2008) Delgado Solís, Cindy (2008, Enero). Caracterización de Proyectos de Software para Configurar su Desarrollo y Habilitar la Comparación entre Casos Almacenados en la Memoria Organizacional. Tesis de maestría en Ciencias Computacionales, CENIDET, Cuernavaca, Morelos.
- (Petrinja, et al 2009) Petrinja, E., Nambakam, R., & Sillitti, A. (2009, May). Introducing the opensource maturity model. In Proceedings of the 2009 ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development (pp. 37-41). IEEE Computer Society.
- (Halkidi, et al 2011) Halkidi, M., Spinellis, D., Tsatsaronis, G., & Vazirgiannis, M. (2011). Data mining in software engineering. *Intelligent Data Analysis*, 15(3), 413-441.
- (Xie, et al 2009) Xie, T., Thummalapenta, S., Lo, D., & Liu, C. (2009). Data mining for software engineering. *Computer*, 42(8), 55-62.
- (Hassan, et al 2010) Hassan, A. E., & Xie, T. (2010, November). Software intelligence: the future of mining software engineering data. In Proceedings of the FSE/SDP workshop on Future of software engineering research (pp. 161-166). ACM.
- (Correa 2015) Correa Delval, Manuel José. (2015, Febrero). Componentes de Acoplamiento para la Infraestructura de un Ambiente Libre para el Desarrollo y Mejora Incremental de Productos de Software. Tesis de maestría en Ciencias Computacionales, CENIDET, Cuernavaca, Morelos.

- (Xie et al 2010) Hassan, A. E., & Xie, T. (2010, November). Software intelligence: the future of mining software engineering data. In Proceedings of the FSE/SDP workshop on Future of software engineering research (pp. 161-166). ACM.
- (Ohrn 1999) Øhrn, A. (2000). Discernibility and rough sets in medicine: tools and applications.
- (Grossman 1998) Grossman, R., Kasif, S., Moore, R., Rocke, D., & Ullman, J. (1998 January). Data mining research: Opportunities and challenges. In Report of three NSF workshops on mining large, massive, and distributed data.
- (Han et al 2006) Han, J., Kamber, M., & Pei, J. (2006). Data mining: concepts and techniques. Morgan kaufmann.
- (Riquelme et al 2006) Riquelme, J. C., Ruiz, R., & Gilbert, K. (2006). Minería de datos: Conceptos y tendencias. Revista Iberoamericana de Inteligencia Artificial, 10(29), 11-18.
- (KDNuggets, 2014) *Portal dedicado a Minería de datos, análisis de negocios y Big Data*  
*<http://www.kdnuggets.com>*  
*Fecha de consulta: Septiembre 2014*
- (Chapman et al 2000) Pete Chapman (NCR), Julian Clinton (SPSS), Randy Kerber (NCR), Thomas Khabaza (SPSS), Thomas Reinartz (DaimlerChrysler), Colin Shearer (SPSS) and Rüdiger Wirth (DaimlerChrysler) "CRISP-DM 1.0 Step-by-step data mining guide" SPSS Inc. CRISPMWP-1104 2010

- (Zhang et al 203) Zhang, S., Zhang, C., & Yang, Q. (2003). Data preparation for data mining. *Applied Artificial Intelligence*, 17(5-6), 375-381.
- (Thomas 1992) Thomas, I., & Nejme, B. A. (1992). Definitions of tool integration for environments. *Software, IEEE*, 9(2), 29-35.
- (Wasserman 1990) Wasserman, A. I. (1990, January). Tool integration in software engineering environments. In *Software Engineering Environments* (pp. 137-149). Springer Berlin Heidelberg.
- (Hall 2011) Hall, R. S., Pauls, K., McCulloch, S., & Savage, D. (2011). OSGi in action. *Creating Modular Applications in Java*.
- (QSOS, 2014) *Metodología de comparación y evaluación de herramientas FLOSS*  
<http://www.qsos.org>  
 Fecha de consulta: Septiembre 2014
- (Amyot, 2003) Daniel Amyot  
 Introduction to the User Requirements Notation: Learning by Example  
 SITE, University of Ottawa 800 King Edward  
 Ottawa (ON), Canada  
 Revista Computer Networks pág. 285-301  
 2003/06/21
- (IEEE, 1998) The Institute of Electrical and Electronics Engineers, Inc.  
 IEEE Computer Society

---

IEEE Std 830-1998. IEEE Recommended Practice for Software Requirements Specifications. New York.  
ISBN 0-7381-0332-2

(ISO/IEC 9126-1 2001) INTERNATIONAL STANDARDS ORGANIZATION (2001-2004). ISO/IEC Standard 9126-1: Software Engineering Product Quality Part 1: Quality Model.

(Henttonen 2007) Henttonen, K., Matinlassi, M., Niemela, E., & Kanstren, T. (2007). Integrability and Extensibility Evaluation from Software Architectural Models- A Case Study. *The Open Software Engineering Journal*, 1(1), 1-20.

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



## **ANEXO A COMPARACIÓN DE HERRAMIENTAS**

---

En este anexo se describe la metodología y los resultados de la comparación de las herramientas candidatas a integrar. Se inicia con la descripción de la metodología QSOS usada para evaluar y seleccionar las herramientas a integrar, posteriormente se definen los criterios a evaluar y los resultados obtenidos.



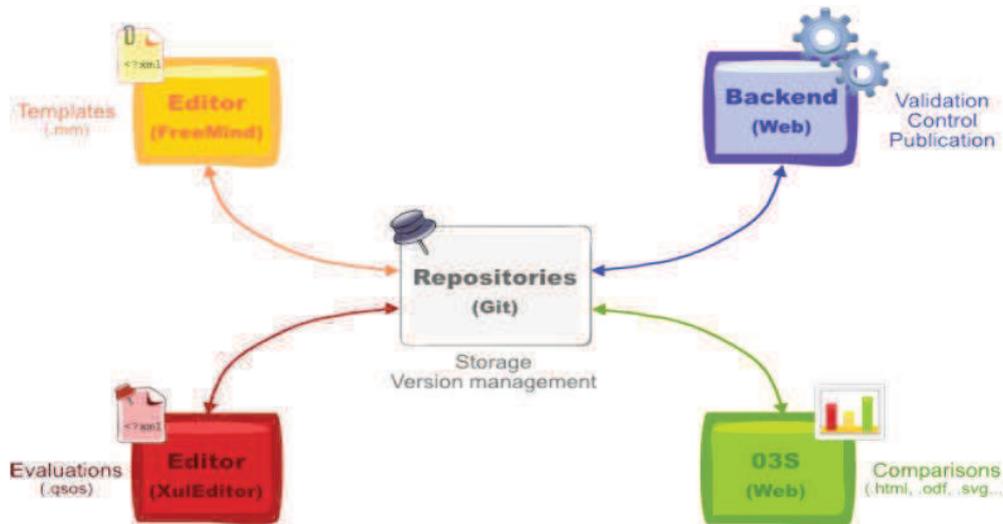
## 1. Metodología QSOS

Para hacer la comparación y evaluación de las herramientas FLOSS a integrar en este ambiente de soporte a minería de datos de repositorios de software se utilizó la metodología QSOS (del inglés *Qualification and Selection of Open Source software*) que su objetivo es evaluar herramientas FLOSS (del inglés Free/Libre Open Source Software). (QSOS, 2014)

La metodología QSOS consiste en una lista de cuatro pasos dentro de un proceso iterativo:

- Definir - y organizar lo que será evaluado (criterios)
- Evaluar - el software contra los criterios definidos
- Calificar - establecer pesos relacionados al contexto
- Seleccionar - la herramienta apropiada

Para la implementación de esta metodología sus creadores proponen el uso de herramientas para llevar a cabo este proceso (Figura A.1).



**Figura A.1** Herramientas propuestas para implementar QSOS

## 2. Mapa mental de criterios a evaluar

Como parte del primer paso se elaboró un mapa mental sobre los criterios a evaluar en las diferentes herramientas FLOSS. En la Figura A.2 se muestran los criterios relacionados a la madurez y algunas características de cada herramienta.

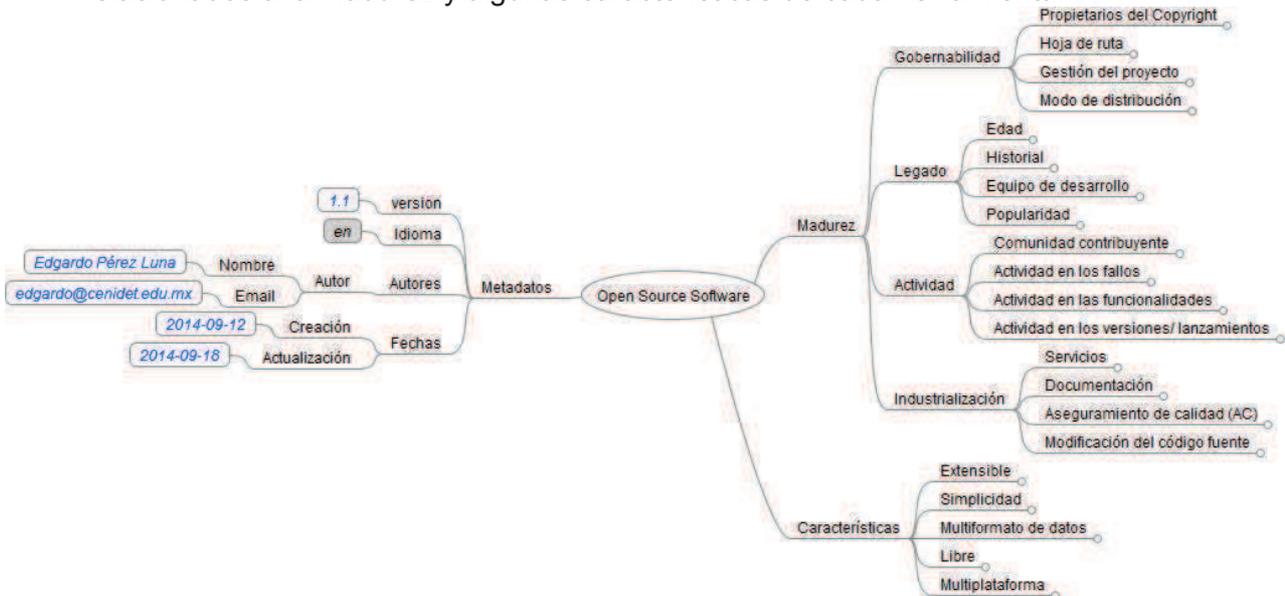


Figura A.2 Mapa mental de los criterios a evaluar

Se estableció el peso de cada criterio a evaluar (Figura A.3, A.4, A.5 Y A.6).

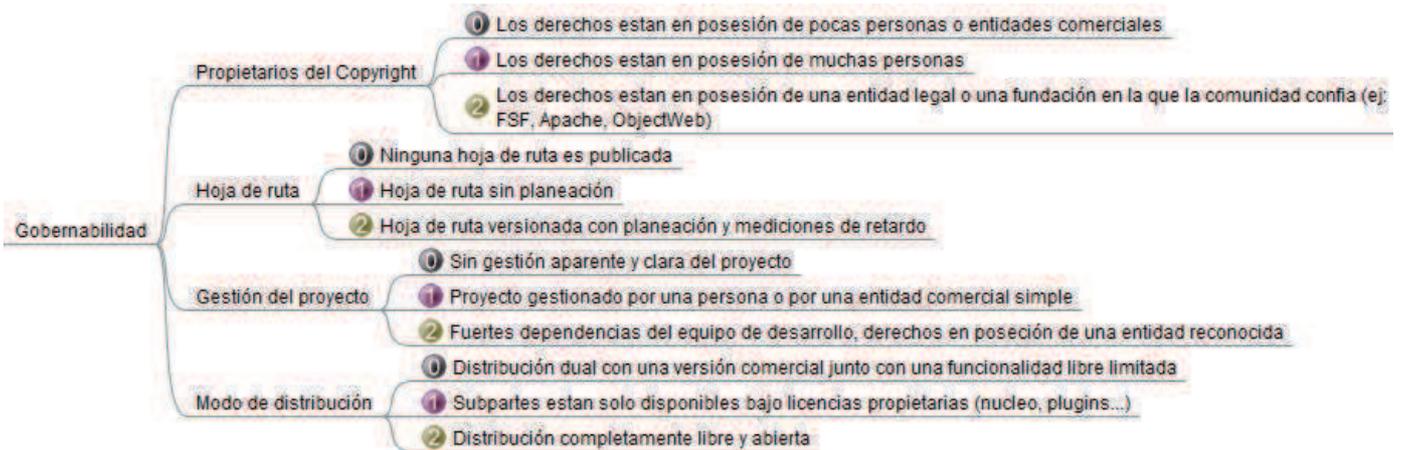


Figura A.3 Pesos para los criterios relacionados a la gobernabilidad

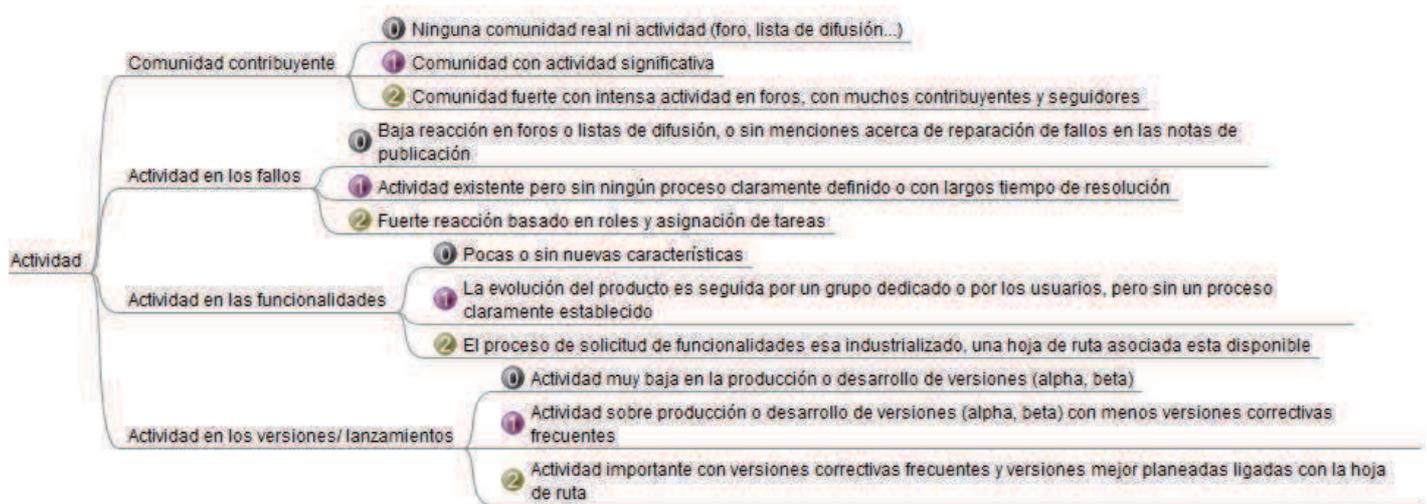


Figura A.4 Pesos para los criterios relacionados a la actividad

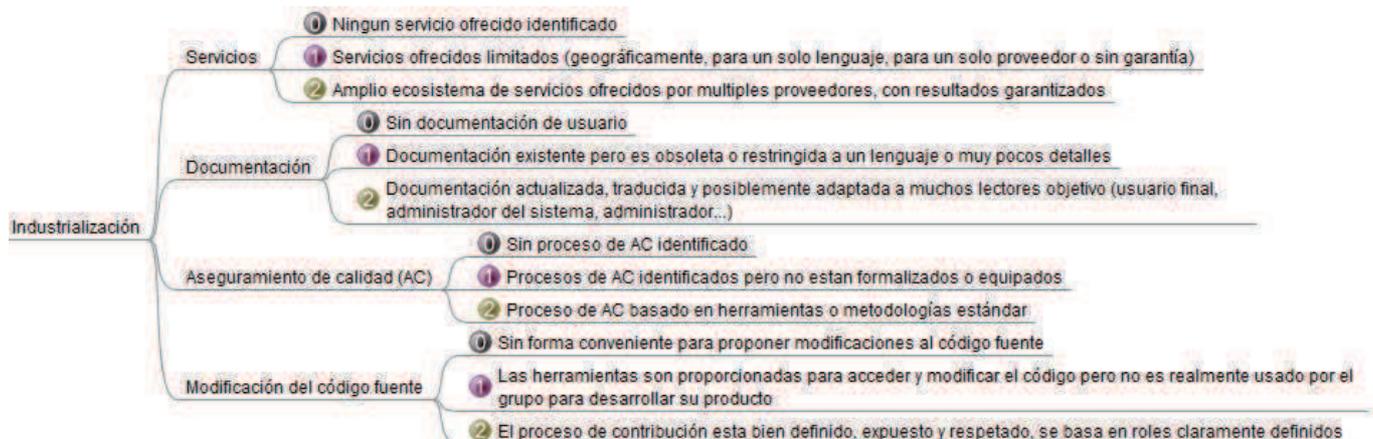


Figura A.5 Pesos para los criterios relacionados a la industrialización

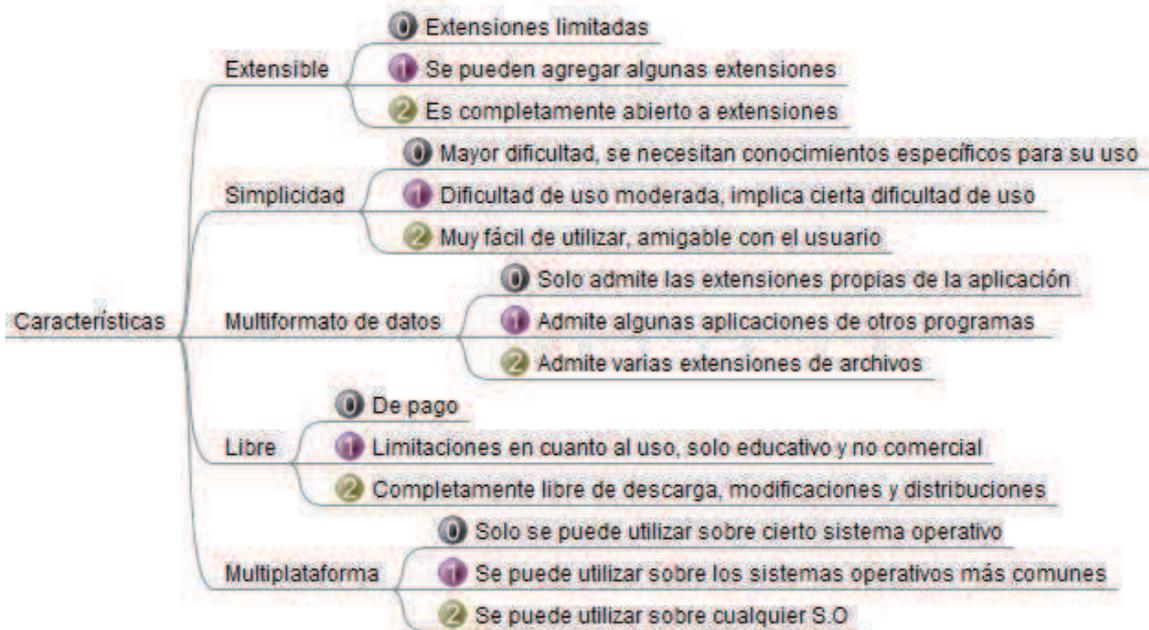


Figura A.6 Pesos para los criterios de algunas características de las herramientas

### 3. Uso de plantilla

A continuación se carga la plantilla en el editor XuEditor, en este caso se usó el *plugin* para Firefox (Figura A.7).

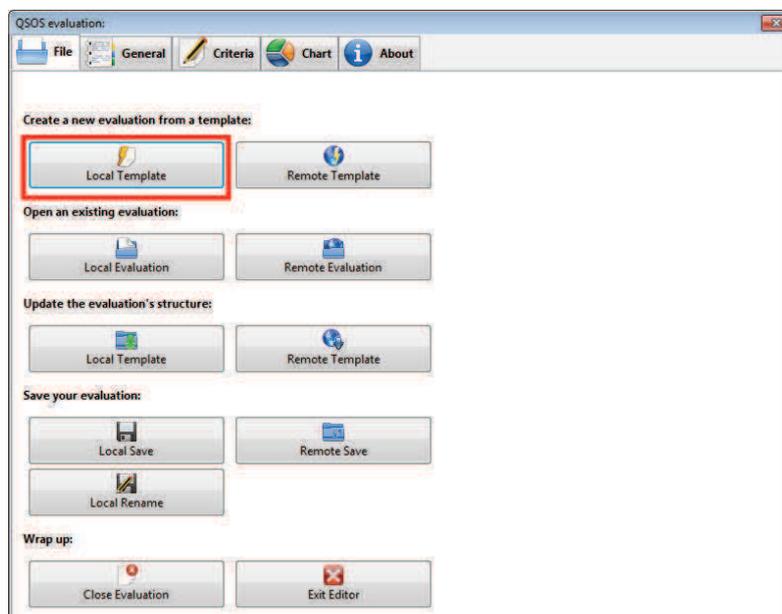


Figura A.7 Interfaz XuEditor para cargar la plantilla

Como siguiente paso se llenó la plantilla con los datos de cada herramienta (Figura A.8).

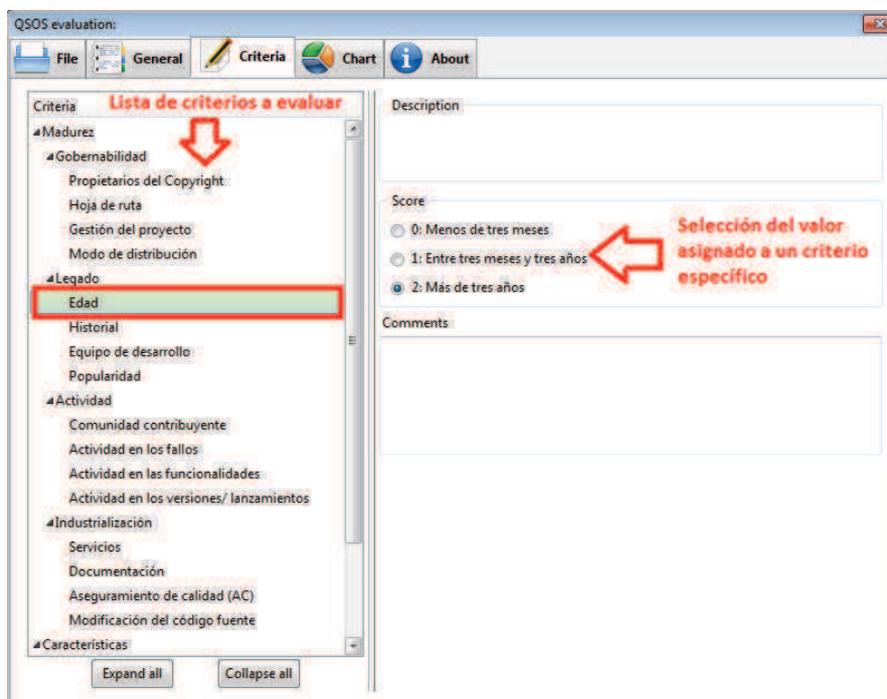


Figura A.8 Lista de criterios para seleccionar valores

#### 4. Resultados

Una vez terminadas las evaluaciones se debe hacer la comparación utilizando la aplicación Web O3S, aquí se mostrará la comparación de forma gráfica.

**Nota:** La aplicación O3S en este momento no está disponible, he enviado un correo electrónico a los responsables al cual obtuve respuesta, ellos tienen un problema en uno de sus servidores y están trabajando para solucionarlo y me mantendrán al tanto de lo sucedido. Por esta razón se ha hecho la comparación manual de las herramientas, obteniendo un puntaje promedio con las evaluaciones elaboradas anteriormente.

Tabla de comparaciones de las herramientas, se muestra sombreada la herramienta mejor puntuada (Tabla A.1, A.2, A.3 y A.4):

**Tabla A.1** Comparación de herramienta

Criterio\Herramienta	Knime	Orange	Pentaho	RapidMiner	Weka	SEE	R
Gobernabilidad	2	1.75	1.5	1.5	1.75	1.75	1.75
Legado	1.75	1.25	1.75	1.75	1.75	1.5	1.75
Actividad	1.5	1	2	2	1.25	0.75	2
Industrialización	1.75	1	1.75	1.75	1	1.25	1.75
Características	1.8	1.8	2	2	1.6	1.4	1.6
Promedio	1.76	1.36	1.8	1.8	1.47	1.33	1.77

**Tabla A.2** Comparación de herramientas

Criterio\Herramienta	EditPadLite	Collabtive	GanttPV	ClockingIT	TaskJuggler	dotProject
Gobernabilidad	0.25	1	1	2	1.5	2
Legado	1.5	1.5	1.5	1.75	1.5	2
Actividad	1	1.25	1	1.5	1	2
Industrialización	1.5	1.5	1.25	1.5	1.5	1.75
Características	1.4	1.8	1.8	1.8	1.8	1.8
Promedio	1.13	1.41	1.31	1.71	1.46	1.91

**Tabla A.3** Comparación de herramientas

Criterio\Herramienta	OpenRefine	MS Excel
Gobernabilidad	1.5	0.5
Legado	1.5	2
Actividad	1.25	1.5
Industrialización	1.25	1.5
Características	2	1.6
Promedio	1.5	1.42

**Tabla A.4** Comparación de herramientas

Criterio\Herramienta	MS Access	MySQL	PostgreSQL
Gobernabilidad	0.5	1.75	2
Legado	2	2	2
Actividad	1.75	2	2
Industrialización	1.5	1.5	2
Características	1.6	1.8	2
Promedio	1.47	1.81	2

Según los resultados anteriores las herramientas a integrar son:

- RapidMiner
- dotProject
- OpenRefine
- MySQL

RapidMiner y Pentaho obtuvieron el mismo puntaje así que se basó en las encuestas realizadas por KDNuggets (KDNuggets, 2014) sobre las herramientas más utilizadas en proyectos reales de minería de datos donde RapidMiner desde el año 2010 ocupa el primer lugar, se decidió elegir RapidMiner.

Se eligió MySQL ya que dotProject necesita este gestor de base de datos para trabajar, aunque PostgreSQL haya salido mejor puntuado en la evaluación.

Además se integra una herramienta más, la cual no fue objeto de comparación y es de utilidad para registrar incidencias surgidas durante el proceso de minería de datos, esta herramienta es MantisBT.



## **ANEXO B PRUEBAS**

---

En este anexo se describen las pruebas realizadas al ambiente de soporte a minería de datos de desarrollo de software. El documento se estructura de la siguiente manera: plan de pruebas, diseño de pruebas, casos de pruebas, procedimiento de pruebas, registro de pruebas y reporte de pruebas.



## 1. Plan de pruebas

### 1.1. Introducción

En este documento se expone el plan de pruebas a cubrir para asegurar el funcionamiento adecuado del producto de software obtenido de la investigación “Componentes de acoplamiento para la infraestructura de soporte a minería de datos de desarrollo de software”. Se define el alcance del plan de pruebas, los criterios tanto de aceptación, como de suspensión y/o reanudación. Además se nombran cuáles son los elementos que se toman en cuenta para este plan de pruebas.

### 1.2. Identificador del documento

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

**Tabla B.1** Nomenclatura de documentos

Identificador del documento		
Nomenclatura	Identificador	Descripción
CA	Nombre del proyecto	<b>CA</b> = Componentes de Acoplamiento para la infraestructura de soporte a minería de datos de desarrollo de software
AB	Tipo de documento	<b>TP</b> - Plan de Pruebas (Test Plan) <b>TD</b> - Diseño de Pruebas (Test Design) <b>TC</b> - Casos de Prueba (Test Case) <b>TPr</b> - Procedimientos de Pruebas (Test Procedure) <b>TL</b> - Registro de Pruebas (Test Log) <b>TR</b> - Reporte de Pruebas (Test Report)
XY	Numeración	Representa el número correspondiente al documento.

### 1.3. Alcance

El alcance del plan de pruebas es verificar la funcionalidad, comportamiento y cumplimiento de los requerimientos funcionales que fueron especificados para la implementación de los componentes de acoplamiento los cuales son: CAProyectos, CAUsuarios, CATareas, CAArchivos, CANucleo, CAI y CASTarter.

### 1.4. Referencias

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

## **1.5. Detalles del Plan de Pruebas de Sistema**

A continuación se provee una **matriz de trazabilidad de pruebas** con el fin de conocer que requerimiento queda cubierto por una prueba. Se definen también los criterios de aceptación y de suspensión o reanudación.

### **1.5.1. Requerimientos funcionales a probar**

El plan de pruebas consiste en probar que se han cumplido los requerimientos de software definidos en el SRS y son los siguientes:

#### **1.5.1.1. Integrar Usuarios**

Este requerimiento es ejecutado por el componente de acoplamiento CAUsuarios. Si se registra una nueva cuenta de usuario en la herramienta dotProject, se deben crear la cuenta correspondiente en la herramienta MantisBT

**Crear cuenta de usuario en dotProject:** Se crea una cuenta de usuario en dotProject y se extraen los datos necesarios para la creación de una nueva cuenta de usuario en la herramienta MantisBT

**Crear cuenta de usuario en MantisBT:** Crear una nueva cuenta de usuario con los datos extraídos de la herramienta dotProject

**Registrar la relación entre las cuentas de usuario:** Registrar en la base de datos CABD la relación entre las cuentas de usuario, se obtiene el id de cada nueva cuenta y se almacena en la BD como un nuevo registro

#### **1.5.1.2. Integrar proyectos**

Este requerimiento es ejecutado por el componente de acoplamiento CAProyectos. El usuario registra los datos de un nuevo proyecto en dotProject (herramienta para la administración de proyectos), automáticamente se debe crear un proyecto en MantisBT para el seguimiento de defectos, así como un proyecto en OpenRefine para el tratamiento de datos, además de un proyecto en RapidMiner que es la herramienta para la aplicación de algún modelo de minería de datos, y por ultimo registrar la relación entre los proyectos con el fin de vincularlos.

**Crear proyecto en dotProject:** Se registra un nuevo proyecto en dotProject y se extraen los datos necesarios para crear el proyecto en las demás herramientas.

**Crear proyecto en MantisBT:** Crear un proyecto en la herramienta MantisBT con los datos del proyecto extraído de la herramienta dotProject.

**Crear proyecto en OpenRefine:** Crear un proyecto en la herramienta OpenRefine con los datos del proyecto extraído de la herramienta dotProject.

**Crear proceso en RapidMiner:** Crear un proceso en la herramienta RapidMiner con los datos del proyecto extraído de la herramienta dotProject.

**Registrar la relación entre los proyectos:** Registrar en la base de datos CABD la relación entre los proyectos, se obtiene el id de cada nuevo proyecto y se almacena en la BD como un nuevo registro.

#### 1.5.1.3. Integrar Tareas

Este requerimiento es ejecutado por el componente de acoplamiento CATareas. Si se registra un nuevo defecto en la herramienta MantisBT, el componente de acoplamiento encargado de las tareas (CATareas) debe registrar una nueva tarea en dotProject relacionado el defecto insertado y al proyecto correspondiente.

**Crear incidencia en MantisBT:** Se crea una nueva incidencia en MantisBT, relacionado a un proyecto y un usuario.

**Crear tarea en dotProject:** Se registra una nueva tarea, relacionada con la incidencia creada y asignada al proyecto y usuario relacionado.

**Registrar la relación entre defecto-tarea:** Registrar en la tabla tareas de la base de datos CABD la relación entre el defecto- tarea, se obtiene el id de cada nueva defecto-tarea y se almacena en la BD como un nuevo registro.

#### 1.5.1.4. Integrar Archivos

Este requerimiento es ejecutado por el componente de acoplamiento CAArchivos. Si se registra un archivo tanto en dotProject como en MantisBT, este archivo se debe

adjuntar al registro relacionado de la otra herramienta, por lo que debe existir este registro relacionado (incidencia-tarea o viceversa)

**Adjuntar archivo a incidencia de MantisBT o tarea de dotProject:** Se registra (adjunta) un archivo en cualquiera de estas herramientas.

**Adjuntar archivo a incidencia de MantisBT o tarea de dotProject:** Se adjunta el archivo, en la herramienta que corresponda, si el archivo se adjuntó inicialmente a una incidencia ahora debe existir ese archivo en la tarea de dotProject relacionada o viceversa.

**Registrar la relación entre los archivos:** Registrar en la tabla archivos de la base de datos CABD la relación entre los archivos, se obtiene el id de cada nuevo archivo y se almacena en la BD como un nuevo registro.

La ejecución de los componentes anteriormente mencionados está a cargo de los componentes CANucleo, CAI y CASarter.

CANucleo colecta los componentes CAUsuarios, CAProyectos, CATareas y CAArchivos a través de la interfaz CAI y CASarter es el encargado de iniciar, detener o reanudar el sistema.

### 1.5.2. Matriz de Trazabilidad de Pruebas

Se provee la lista (tabla B.2) de requerimientos que debe cumplir el sistema y el componente de acoplamiento correspondiente a ese requerimiento, así como el identificador de su caso de prueba.

**Tabla B.2** Matriz de trazabilidad de pruebas

Requerimiento	Componente de Acoplamiento	Caso de Prueba
Integrar Usuarios	CAUsuarios	CA-TC-01
Integrar Proyectos	CAProyectos	CA-TC-02
Integrar Tareas	CATareas	CA-TC-03
Integrar Archivos	CAArchivos	CA-TC-04
-	CANucleo	CA-TC-05
-	CAI	CA-TC-06
-	CASarter	CA-TC-07

Otro elemento a probar es la interfaz de usuarios de AMDADS (Tabla B.3).

**Tabla B.3** Características a probar Interfaz

Elemento a probar	Características	Caso de prueba
Interfaz	<ul style="list-style-type: none"> <li>• Autenticación de usuarios</li> <li>• Listado de proyectos asociados</li> </ul>	CA-TC-08

### 1.5.3. Criterio de Aceptación

Los casos de prueba son aprobados si cumplen con cada una de las funcionalidades especificadas en los requerimientos sin ninguna clase de contratiempo.

### 1.5.4. Criterio para Suspensión y Reanudación de Pruebas

La suspensión de las pruebas está regida por la presencia de fallas en los componentes que impidan el cumplimiento de sus funcionalidades, al registrarse una falla se suspende la prueba y se reanuda al ser corregida.

### 1.5.5. Entregables de Pruebas

Como resultado de las actividades correspondientes a las pruebas se obtienen los siguientes entregables:

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

## 1.6. Administración de Pruebas

### 1.6.1. Actividades Planificadas

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

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

- Ejecución de casos de pruebas.
- Generación de reporte de pruebas.

Las actividades son ejecutadas por el autor del proyecto.

### **1.6.2. Infraestructura**

La infraestructura para las pruebas se especifica a continuación:

- Software
  - Oracle VirtualBox 4.3.12 r93733
  - Sistema Operativo: GNU/Linux Ubuntu 14.04
  - Servidor Web: Apache 2.2.22
  - Tomcat 7
  - JVM: Oracle Java 1.7.0\_80
  - Gestor de Base de Datos: MySQL 5.5.43
  - Eclipse for RCP and RAP Developers Version: Luna Service Release 2 (4.4.2)
  - Servidor OSGi Equinox 3.9.1 (KeplerSR2)
  - dotProject 2.1.8
  - MantisBT 1.2.19
  - OpenRefine 2.5 [r2407]
  - RapidMiner 6.3
  - Mozilla Firefox 37.0.2
- Hardware
  - Procesador: AMD E2-1800 1.7 GHz.
  - Memoria RAM en anfitrión: 6 GB
  - Memoria RAM en huésped: 2.5 GB

## 2. Diseño de pruebas

### 2.1. Introducción

En este documento se refina el alcance de las pruebas y se identifican las características a ser cubiertas por el diseño de pruebas asociadas.

#### 2.1.1. Identificador de Documento

A continuación se especifica el identificador asignado al documento según la nomenclatura definida en el Plan de Pruebas: CA-TD-01.

#### 2.1.2. Alcance

Este diseño de pruebas se enfoca en las características que permiten el cumplimiento de los requerimientos funcionales definidos en el documento SRS. Las pruebas serán usando la inspección visual para corroborar la existencia de los registros esperados.

#### 2.1.3. Referencias

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

### 2.2. Detalles del Diseño de la Prueba

Dentro de esta sección están contenidas las características a probar, los casos de prueba así como los criterios de aceptación y entregables.

#### 2.2.1. Características a Probar

El diseño de pruebas abarca las características referentes a los siguientes requerimientos: integrar usuarios, integrar proyectos, integrar tareas e integrar archivos. A continuación se muestra una matriz de trazabilidad de pruebas (Tabla B.4), en donde se muestran las características que van a ser robadas, así como su caso y procedimiento de prueba correspondiente.

**Tabla B.4** Matriz de trazabilidad de pruebas

Requerimiento	Componente de Acoplamiento	Característica	Caso de Prueba	Procedimiento de Prueba
Integrar Usuarios	CAUsuarios	Crear cuenta de usuario dotProject	CA-TC-01	CA-TPr-01

		Crear cuenta de usuario en MantisBT Registrar la relación entre cuentas de usuario		
Integrar Proyectos	CAProyectos	Crear un proyecto en dotProject Crear proyecto en MantisBT Crear proyecto en openRefine Crear proceso en RapidMiner Registrar la relación entre proyectos	CA-TC-02	CA-TPr-02
Integrar Tareas	CATareas	Crear defecto en MantisBT Crear tarea en dotProject Registrar la relación entre defecto-tarea	CA-TC-03	CA-TPr-03
Integrar Archivos	CAArchivos	Adjuntar archivo a defecto de MantisBT o tarea de dotProject Adjuntar archivo a defecto de MantisBT o tarea de dotProject Registrar la relación entre archivos	CA-TC-04	CA-TPr-04
-	CANucleo	Este componente es el encargado de coleccionar los componentes que implementen la interfaz del componente CAI.	CA-TC-05	CA-TPr-05
-	CAI	Proporciona la interfaz que enlazará a los componentes que implementan dicha interfaz con el componente CANucleo.	CA-TC-06	CA-TPr-06
-	CASstarter	Iniciar, detener o reanudar el sistema.	CA-TC-07	CA-TPr-07

También es necesario hacer pruebas a la interfaz de usuario, en la tabla B.5 se muestran las características a probar.

**Tabla B.5** Características a probar Interfaz

<b>Componente a probar</b>	<b>Característica a probar</b>	<b>Caso de Prueba</b>	<b>Procedimiento de Prueba</b>
Interfaz de usuario	<ul style="list-style-type: none"> <li>• Autenticación de usuarios</li> <li>• Listado de proyectos asociados</li> </ul>	CA-TC-08	CA-TPr-08

### 2.2.2. Estrategia de Pruebas

La realización de las pruebas será de la siguiente manera: se utilizarán datos de proyectos/usuarios ficticios y se registrarán en la herramienta dotProject, al igual se usarán datos de defectos ficticios y se registrarán en MantisBT, para probar el componente CAArchivos se usarán archivos en diferentes formatos. El resultado esperado es el cumplimiento de cada uno de los requerimientos establecidos anteriormente.

### 2.2.3. Criterio de Aceptación del Caso de Prueba

Para que las pruebas sean aprobadas deben cumplir con cada una de las funcionalidades especificadas en los requerimientos sin ninguna clase de contratiempo.

### 2.2.4. Entregables de Pruebas

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

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

### 3. Casos de prueba

#### 3.1. Introducción

En este documento se especifican las entradas necesarias, las salidas anticipadas y el conjunto de condiciones de ejecución para cada prueba.

##### 3.1.1. Alcance

El alcance de los casos de prueba se enfoca en las características que permiten el cumplimiento de los requerimientos funcionales: Integrar Proyectos, Integrar Usuarios, Integrar Tareas e Integrar Archivos, definidos en el documento SRS. Así como la interfaz de usuario.

##### 3.1.2. Referencias

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

#### 3.2. Caso de Prueba CA-TC-01

A continuación se describen el objetivo, entradas y salidas del caso de prueba CA-TC-01.

##### 3.2.1. Identificador del Caso de Prueba

Se describe el requerimiento funcional (Tabla B.6), el componente de acoplamiento al que corresponde y el caso de prueba relacionado.

Tabla B.6 Caso de prueba 01

Requerimiento funcional	Componente de Acoplamiento	Caso de Prueba
Integrar Usuarios	CAUsuarios	CA-TC-01

##### 3.2.2. Objetivo

El objetivo de este caso de prueba es validar el cumplimiento del requerimiento **integrar usuarios** que corresponde al componente de acoplamiento **CAUsuarios**.

##### 3.2.3. Entradas

La entrada son los datos ingresados en la herramienta dotProject (Tabla B.7), para posteriormente extraerla y almacenarla en las demás herramientas.

**Tabla B.7** Entradas a dotProject CA-TC-01

Nombre del campo	Tipo
Nombre para login	Cadena
Rol de usuario	Cadena
Contraseña	Cadena
Nombre	Cadena
E-mail	Cadena

### 3.2.4. Salidas

La salida de CAUsuarios es el registro automático de una cuenta de usuario en MantisBT además de registrar la relación entre las cuentas de usuario en la base de datos CABD

En seguida se describen las salidas esperadas en cada herramienta y en la base de datos CABD.

#### 3.2.4.1. MantisBT

La salida esperada (Tabla B.8), en MantisBT es el registro de una nueva cuenta de usuario con los datos del registrado en dotProject.

**Tabla B.8** Salida MantisBT CA-TC-01

Nombre del campo	Tipo
Nombre de usuario	Varchar
Nombre real	Varchar
email	Varchar
password	Varchar

#### 3.2.4.2. Tabla de relaciones

La salida esperada en la base de datos de relaciones (CABD) y específicamente en la tabla usuarios es el registro de los valores del id de cada nueva cuenta de usuario creado (Tabla B.9).

**Tabla B.9** Salida en tabla relaciones CA-TC-01

Nombre del campo	Tipo de valor
id	Entero

dotproject	Entero
bugtracker	Entero

### 3.3. Caso de Prueba CA-TC-02

A continuación se describen el objetivo, entradas y salidas del caso de prueba CA-TC-02.

#### 3.3.1. Identificador del Caso de Prueba

En la tabla B.10 se describe el requerimiento funcional, el componente de acoplamiento al que corresponde y el caso de prueba relacionado.

**Tabla B.10** Caso de prueba 02

Requerimiento funcional	Componente de Acoplamiento	Caso de Prueba
Integrar Proyectos	CAProyectos	CA-TC-02

#### 3.3.2. Objetivo

El objetivo de este caso de prueba es validar el cumplimiento del requerimiento integrar proyectos que corresponde al componente de acoplamiento CAProyectos.

#### 3.3.3. Entradas

La entrada son los datos ingresados en la herramienta dotProject (Tabla B.11) para posteriormente extraerla y almacenarla en las demás herramientas.

**Tabla B.11** Entrada en dotProject CA-TC-02

Nombre del campo	Tipo
Nombre	Cadena
Propietario	Cadena
Empresa/compañía	Cadena
Fecha de inicio	Fecha
Fecha de fin objetivo	Fecha
Prioridad	Cadena
Nombre corto	Cadena
Color de identificación	Cadena
Tipo de proyecto	Cadena

Estado del proyecto	Cadena
Descripción del proyecto	Texto

### 3.3.4. Salidas

La salida de CAProyectos es el registro automático de un proyecto en cada una de las demás herramientas MantisBT, OpenRefine y un proceso en RapidMiner, además de registrar la relación de proyectos en la base de datos CABD

En seguida se describen las salidas esperadas en cada herramienta y en la base de datos CABD.

#### 3.3.4.1. MantisBT

La salida esperada en MantisBT es el registro de un nuevo proyecto (Tabla B.12) con los datos del registrado en dotProject.

**Tabla B.12** Salidas en MantisBT CA-TC-02

Nombre del campo	Tipo
Nombre	Cadena
Descripción	Texto
Estado	Cadena

#### 3.3.4.2. OpenRefine

La salida esperada en OpenRefine es el registro de un nuevo proyecto (Tabla B.13) con los datos del registrado en dotProject.

**Tabla B.13** Salidas en OpenRefine CA-TC-02

Nombre del campo	Tipo
Nombre	Cadena

#### 3.3.4.3. RapidMiner

La salida esperada en RapidMiner es el registro de un nuevo proceso (Tabla B.14) con los datos del registrado en dotProject.

**Tabla B.14** Salida en RapidMiner CA-TC-02

Nombre	Tipo
--------	------

Directorio	Cadena
Nombre	Cadena

#### 3.3.4.4. Tabla de relaciones

La salida esperada en la base de datos de relaciones (CABD) y específicamente en la tabla proyectos es el registro de los valores del id de cada nuevo proyecto creado (Tabla B.15).

**Tabla B.15** Salida en tabla relaciones CA-TC-02

Nombre	Tipo
id	Entero
dotproject	Entero
rapidminer	Entero
openrefine	Long
bugtracker	Entero

#### 3.4. Caso de Prueba CA-TC-03

A continuación se describen el objetivo, entradas y salidas del caso de prueba CA-TC-03.

##### 3.4.1. Identificador del Caso de Prueba

En la tabla B.16 se describe el requerimiento funcional, el componente de acoplamiento al que corresponde y el caso de prueba relacionado.

**Tabla B.16** Caso de prueba 03

Requerimiento funcional	Componente de Acoplamiento	Caso de Prueba
Integrar tareas	CATareas	CA-TC-03

##### 3.4.2. Objetivo

El objetivo de este caso de prueba es validar el cumplimiento del requerimiento **integrar tareas** que corresponde al componente de acoplamiento **CATareas**.

##### 3.4.3. Entradas

La entrada son los datos ingresados en la herramienta MantisBT (Tabla B.17) para posteriormente extraerla y almacenarla en la herramienta dotProject.

**Tabla B.17** Entrada en dotProject CA-TC-03

<b>Nombre del campo</b>	<b>Tipo</b>
Categoría	Cadena
Reproducibilidad	Cadena
Severidad	Cadena
Prioridad	Cadena
Asignado a	Entero
Resumen	Texto
Descripción	Texto
Pasos para reproducir	Texto
Información adicional	Texto
Archivo	Archivo

#### 3.4.4. Salidas

La salida de CAProyectos es el registro automático de un proyecto en cada una de las demás herramientas MantisBT, OpenRefine y un proceso en RapidMiner, además de registrar la relación de proyectos en la base de datos CABD

En seguida se describen las salidas esperadas en cada herramienta y en la base de datos CABD.

##### 3.4.4.1. dotProject

La salida esperada en dotProject es el registro de una nueva tarea con los datos del registrado en MantisBT (Tabla B.18).

**Tabla B.18** Salida en dotProject CA-TC-03

<b>Nombre del campo</b>	<b>Tipo</b>
Nombre	Cadena
Descripción	Texto
Propietario	Entero
Asignado a	Entero
Archivo	Archivo

##### 3.4.4.2. Tabla de relaciones

La salida esperada en la base de datos de relaciones (CABD) y específicamente en la tabla tareas es el registro de los valores del id de la nueva incidencia y la nueva tarea (Tabla B.19).

**Tabla B.19** Salida en tabla relaciones CA-TC-03

Nombre	Tipo
id	Entero
dotproject	Entero
bugtracker	Entero

### 3.5. Caso de Prueba CA-TC-04

A continuación se describen el objetivo, entradas y salidas del caso de prueba CA-TC-04.

#### 3.5.1. Identificador del Caso de Prueba

En la (Tabla B.20) se describe el requerimiento funcional, el componente de acoplamiento al que corresponde y el caso de prueba relacionado.

**Tabla B.20** Caso de prueba 04

Requerimiento funcional	Componente de Acoplamiento	Caso de Prueba
Integrar archivos	CAArchivos	CA-TC-04

#### 3.5.2. Objetivo

El objetivo de este caso de prueba es validar el cumplimiento del requerimiento integrar archivos que corresponde al componente de acoplamiento CAArchivos.

#### 3.5.3. Entradas

La entrada son los datos ingresados (Tabla B.21) en la herramienta MantisBT o dotProject para posteriormente extraerla y almacenarla en la herramienta dotProject o MantisBT según sea el caso.

**Tabla B.21** Entrada en MantisBT o dotproject CA-TC-04

Nombre del campo	Tipo
Archivo	Archivo

### 3.5.4. Salidas

La salida de CAArchivos es el registro automático de un archivo la herramienta MantisBT o dotProject según sea el caso (si la entrada del archivo es en dotProject, la copia se muestra en MantisBT y viceversa), además de registrar la relación de archivos en la base de datos CABD

#### 3.5.4.1. dotProject o MantisBT

La salida esperada en dotProject o MantisBT según sea el caso es el registro de un nuevo archivo en la herramienta complementaria (Tabla B.22).

**Tabla B.22** Salida en dotProject o MantisBT CA-TC-04

Nombre del campo	Tipo
Archivo	Archivo

#### 3.5.4.2. Tabla de relaciones

La salida esperada en la base de datos de relaciones (CABD) y específicamente en la tabla tareas es el registro de los valores del id de la nueva incidencia y la nueva tarea (Tabla B.23).

**Tabla B.23** Salida en tabla relaciones CA-TC-04

Nombre	Tipo
id	Entero
dotproject	Entero
bugtracker	Entero

### 3.6. Caso de Prueba CA-TC-05/06/07

A continuación se describen el objetivo, entradas y salidas del caso de prueba CA-TC-05/06/07.

#### 3.6.1. Identificador del Caso de Prueba

En la (Tabla B.24) se describe el requerimiento funcional, el componente de acoplamiento al que corresponde y el caso de prueba relacionado.

**Tabla B.24** Caso de prueba 05-06-07

<b>Componente de Acoplamiento</b>	<b>Característica</b>	<b>Caso de Prueba</b>
CANucleo	Colectar los componentes que implementan la interfaz del componente CAI y ejecuta los componentes	CA-TC-05
CAI	Sirve de interfaz entre el componente CANucleo y los componentes que requerían el servicio	CA-TC-06
CASarter	Inicia el sistema, lo detiene o lo reanuda	CA-TC-07

### 3.6.2. Objetivo

El objetivo de estos casos de prueba es el de verificar el funcionamiento de la de los componentes CANucleo, CAI y CASarter.

### 3.6.3. Entradas y Salidas

En este caso la prueba se realiza con la ejecución de los componentes a través de las acciones start, stop, restart y sleep

### 3.7. Caso de Prueba CA-TC-08

A continuación se describen el objetivo, entradas y salidas del caso de prueba CA-TC-08.

#### 3.7.1. Identificador del Caso de Prueba

En la (Tabla B.25) se describe el requerimiento funcional, el componente de acoplamiento al que corresponde y el caso de prueba relacionado.

**Tabla B.25** Caso de prueba 08

<b>Componente</b>	<b>Caso de Prueba</b>
Interfaz de usuarios	CA-TC-08

### 3.7.2. Objetivo

El objetivo de este caso de prueba es validar la funcionalidad de la interfaz de usuario, en lo correspondiente a la autenticación y el listado de proyectos asociados al usuario.

### 3.7.3. Entradas

La entrada para esta prueba son las credenciales de un usuario (Tabla B.26).

**Tabla B.26** Entradas en interfaz

Nombre del campo	Tipo
Usuario	Cadena
Contraseña	Cadena

### 3.7.4. Salidas

La salida de esta prueba es la autenticación exitosa del usuario, por lo que se muestra un mensaje de bienvenida e información general del usuario. Además de la lista de proyectos que el usuario tiene asociados en ese momento.

## **4. Procedimientos de prueba**

### **4.1. Introducción**

En este documento se especifica la secuencia de acciones para la ejecución del conjunto de pruebas usadas para analizar los componentes de software con el fin de evaluar el conjunto de características.

#### **4.1.1. Alcance**

En el alcance de los procedimientos de prueba se encuentran los casos de prueba CA-TC-01, CA-TC-02, CA-TC-03, CA-TC-04, CA-TC-05, CA-TC-06, CA-TC-07, CA-TC-08 que se definieron en el documento de casos de prueba.

#### **4.1.2. Referencias**

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

### **4.2. Procedimiento de Prueba CA-TPr-01**

A continuación se describe la secuencia de pasos requeridos para la ejecución del caso de prueba CA-TC-01, así también las entradas y salidas.

#### **4.2.1. Entradas, Salidas, y Requerimientos Especiales**

Para la ejecución adecuada del caso de prueba CA-TC-01 son necesarios los siguientes requerimientos.

#### **Documentos:**

- Plan de Prueba
- Diseño de Prueba
- Procedimiento de Prueba
- Reporte de Prueba

#### **Servidores:**

- Apache
- OSGi

**Aplicaciones:**

- Mozilla Firefox
- dotProject
- MantisBT
- Componente de acoplamiento CAUsuarios

**Datos:**

- Datos de un nuevo usuario

**4.2.2. Descripción del conjunto de pasos a seguir para ejecutar el caso de prueba CA-TC-01**

**Reportes:** La elaboración de los reportes es de forma manual al hacer una inspección visual de los eventos que ocurren a lo largo de la ejecución de la prueba y serán registrados en el documento de reporte de pruebas, de esta misma forma los errores, problemas o anomalías serán registrados en el mismo documento.

**Preparación:** Antes de iniciar la ejecución del procedimiento se deben iniciar las aplicaciones (herramientas) y servidores. Además se deben tener listos los datos de la nueva cuenta de usuario que se va a registrar.

**Inicio:** Usando el navegador Firefox se debe registrar una nueva cuenta de usuario en la herramienta dotProject.

**Proceso:** En este caso es el componente CAUsuarios el que participa, identifica el nuevo registro en la base de datos y recolecta los datos necesarios para registrar la cuenta de usuario en la herramienta MantisBT. Una vez hecho esto el componente registra la relación entre los proyectos en la base de datos CABD.

**Medición:** La herramienta MantisBT debe tener registrada la nueva cuenta de usuario permitiéndole al usuario ingresar con el mismo nombre y contraseña registrados en dotProject.

**Suspender:** EL componente CAUsuarios se ejecuta a cierto tiempo por lo que terminado ese plazo se puede detener la prueba sin problemas.

**Reiniciar:** La prueba se puede reiniciar en cualquier momento ejecutando el componente CAUsuarios teniendo en cuenta el paso de preparación de esta sección.

**Cierre:** La prueba termina con el registro en el documento de reporte de pruebas.

**Contingencias:** Al presentarse anomalías la prueba tiene que ser suspendida y el error o fallo corregidos. Para poder reiniciar la prueba de debe haber corregido la anomalía.

### **4.3. Procedimiento de prueba CA-TPr-02**

A continuación se describe la secuencia de pasos requeridos para la ejecución del caso de prueba CA-TC-02, así también las entradas y salidas.

#### **4.3.1. Entradas, Salidas, y requerimientos especiales**

Para le ejecución adecuada del caso de prueba CA-TC-02 son necesarios los siguientes requerimientos.

#### **Documentos:**

- Plan de Pruebas
- Diseño de Pruebas
- Procedimiento de Pruebas
- Reporte de Pruebas

#### **Servidores:**

- Apache
- OSGi

#### **Aplicaciones:**

- Firefox
- dotProject
- MantisBT
- OpenRefine

- RapidMiner
- Componente de acoplamiento CAProyectos

**Datos:**

- Datos sobre proyectos

**4.3.2. Descripción del conjunto de pasos a seguir para ejecutar el caso de prueba CA-TC-02**

**Reportes:** La elaboración de los reportes es de forma manual al hacer una inspección visual de los eventos que ocurren a lo largo de la ejecución de la prueba y serán registrados en el documento de reporte de pruebas, de esta misma forma los errores, problemas o anomalías serán registrados en el mismo documento.

**Preparación:** Antes de iniciar la ejecución del procedimiento se deben iniciar las aplicaciones (herramientas) y servidores. Además se deben tener listos los datos del nuevo proyecto que se va a registrar.

**Inicio:** Usando el navegador Firefox se debe registrar un nuevo proyecto en la herramienta dotProject.

**Proceso:** En este caso es el componente CAProyectos el que participa, identifica el nuevo registro en la base de datos y recolecta los datos necesarios para registrar el proyecto en las demás herramientas siguiendo esta secuencia: MantisBT, RapidMiner y OpenRefine. Una vez hecho esto el componente registra la relación entre los proyectos en la base de datos CABD.

**Medición:** Las herramientas MantisBT, RapidMiner y OpenRefine deben tener registrado el nuevo proyecto.

**Suspender:** Debido al uso de diferentes API's y librerías usadas en la creación de proyectos, la ejecución del componente CAProyectos tiende a tardar más de 30 segundos, la prueba puede ser suspendida después de este tiempo.

**Reiniciar:** La prueba se puede reiniciar en cualquier momento ejecutando el componente CAProyectos teniendo en cuenta el paso de preparación de esta sección.

**Cierre:** La prueba termina con el registro en el documento de reporte de pruebas.

**Contingencias:** Al presentarse anomalías la prueba tiene que ser suspendida y el error o fallo corregidos. Para poder reiniciar la prueba de debe haber corregido la anomalía.

#### **4.4. Procedimiento de Prueba CompAcop-TPR-03**

A continuación se describe la secuencia de pasos requeridos para la ejecución del caso de prueba CA-TC-03, así también las entradas y salidas.

##### **4.4.1. Entradas, Salidas, y Requerimientos Especiales**

Para le ejecución adecuada del caso de prueba CA-TC-03 son necesarios los siguientes requerimientos.

#### **Documentos:**

- Plan de Pruebas
- Diseño de Pruebas
- Procedimiento de Pruebas
- Reporte de Pruebas

#### **Servidores:**

- Apache
- OSGi

#### **Aplicaciones:**

- Firefox
- MantisBT
- dotProject
- Componentes de Acoplamiento

#### **Datos:**

- Datos sobre incidencias

#### 4.4.2. Descripción de los Pasos a Seguir Para Ejecutar El Caso de Prueba CompAcop-TC-03

**Reportes:** La elaboración de los reportes es de forma manual al hacer una inspección visual de los eventos que ocurren a lo largo de la ejecución de la prueba y serán registrados en el documento de reporte de pruebas, de esta misma forma los errores, problemas o anomalías serán registrados en el mismo documento.

**Preparación:** Antes de iniciar la ejecución del procedimiento se deben iniciar las aplicaciones (herramientas) y servidores. Además se deben tener listos los datos de la nueva incidencia que se va a registrar.

**Inicio:** Usando el navegador Firefox se debe registrar una nueva cuenta de usuario en la herramienta MantisBT.

**Proceso:** En este caso es el componente CATareas el que participa, identifica el nuevo registro en la base de datos y recolecta los datos necesarios para registrar la tarea en la herramienta dotProject, asignándola al mismo proyecto y usuario. Una vez hecho esto el componente registra la relación entre los proyectos en la base de datos CABD.

**Medición:** La herramienta dotProject debe tener registrada tarea asignada al mismo proyecto y usuario con el que se registró en MantisBT.

**Suspender:** EL componente CATareas se ejecuta a cierto tiempo por lo que terminado ese plazo se puede detener la prueba sin problemas.

**Reiniciar:** La prueba se puede reiniciar en cualquier momento ejecutando el componente CATareas teniendo en cuenta el paso de preparación de esta sección.

**Cierre:** La prueba termina con el registro en el documento de reporte de pruebas.

**Contingencias:** Al presentarse anomalías la prueba tiene que ser suspendida y el error o fallo corregidos. Para poder reiniciar la prueba de debe haber corregido la anomalía.

#### **4.5. Procedimiento de Prueba CompAcop-TPr-04**

A continuación se describe la secuencia de pasos requeridos para la ejecución del caso de prueba CA-TC-04, así también las entradas y salidas.

##### **4.5.1. Entradas, Salidas, y Requerimientos Especiales**

Para le ejecución adecuada del caso de prueba CA-TC-04 son necesarios los siguientes requerimientos.

##### **Documentos:**

- Plan de Pruebas
- Diseño de Pruebas
- Procedimiento de Pruebas
- Reporte de Pruebas

##### **Servidores:**

- Apache
- OSGi

##### **Aplicaciones:**

- Firefox
- MantisBT
- dotProject
- Componentes de Acoplamiento

##### **Archivo:**

- Archivo a adjuntar

##### **4.5.2. Descripción de los Pasos a Seguir Para Ejecutar El Caso de Prueba CompAcop-TC-04**

**Reportes:** La elaboración de los reportes es de forma manual al hacer una inspección visual de los eventos que ocurren a lo largo de la ejecución de la prueba y serán

registrados en el documento de reporte de pruebas, de esta misma forma los errores, problemas o anomalías serán registrados en el mismo documento.

**Preparación:** Antes de iniciar la ejecución del procedimiento se deben iniciar las aplicaciones (herramientas) y servidores. Además se deben tener listos los datos del nuevo archivo que se va a registrar.

**Inicio:** Usando el navegador Firefox se debe adjuntar el archivo a una incidencia de un proyecto en MantisBT o a alguna tarea de un proyecto en dotProject, siempre y cuando ya se encuentre su incidencia-tarea correspondiente.

**Proceso:** En este caso es el componente CAArchivos el que participa, identifica el nuevo registro en la base de datos y recolecta los datos necesarios para registrar el archivo en la herramienta dotProject o MantisBT, asignándola a la misma incidencia-tarea. Una vez hecho esto el componente registra la relación entre los proyectos en la base de datos CABD.

**Medición:** La herramienta dotProject o MantisBT debe tener registrada un archivo asignada al mismo proyecto e incidencia-tarea en el que se registró en la herramienta complementaria.

**Suspender:** EL componente CAArchivos se ejecuta a cierto tiempo por lo que terminado ese plazo se puede detener la prueba sin problemas.

**Reiniciar:** La prueba se puede reiniciar en cualquier momento ejecutando el componente CAArchivos teniendo en cuenta el paso de preparación de esta sección.

**Cierre:** La prueba termina con el registro en el documento de reporte de pruebas.

**Contingencias:** Al presentarse anomalías la prueba tiene que ser suspendida y el error o fallo corregidos. Para poder reiniciar la prueba de debe haber corregido la anomalía.

#### **4.6. Procedimiento de Prueba CompAcop-TPR-05/06/07**

A continuación se describe la secuencia de pasos requeridos para la ejecución del caso de prueba CA-TC-05/06/07, así también las entradas y salidas.

#### 4.6.1. Entradas, Salidas, y Requerimientos Especiales

Para la ejecución adecuada de los casos de prueba CA-TC-05/06/07 son necesarios los siguientes requerimientos.

##### Documentos:

- Documento de Plan de Pruebas
- Documento de Diseño de Pruebas
- Documento de Procedimiento de Pruebas
- Documento de Reporte de Pruebas

##### Servidores:

- OSGi

##### Aplicaciones:

- Componentes de Acoplamiento

#### 4.6.2. Descripción de los Pasos a Seguir Para Ejecutar El Caso de Prueba CompAcop-TC-05

**Reportes:** La elaboración de los reportes es de forma manual al hacer una inspección visual de los eventos que ocurren a lo largo de la ejecución de la prueba y serán registrados en el documento de reporte de pruebas, de esta misma forma los errores, problemas o anomalías serán registrados en el mismo documento.

**Preparación:** Verificar que el servidor osgi está en ejecución y los componentes están disponibles.

**Inicio:** Comando start para el componente CASstarter

**Proceso:** El componente CASstarter inicia el componente CANucleo que es el encargado de enlistar los componentes que implementan la interfaz CAI para ejecutar cada uno y se desactiva al finalizar.

**Medición:** Los componentes deben ser ejecutados correctamente.

**Suspender:** El componente CStarter se desactiva al terminar la ejecución de la lista de componentes que implementan la interfaz CAI, y procede a un periodo de inactividad de 5 segundos durante los cuales puede ser detenido mediante el comando stop.

**Reiniciar:** Ejecutar el componente CStarter y que los componentes estén disponibles para su ejecución.

**Cierre:** La prueba termina con el registro en el documento de reporte de pruebas.

**Contingencias:** Al presentarse anomalías la prueba tiene que ser suspendida y el error o fallo corregidos. Para poder reiniciar la prueba de debe haber corregido la anomalía.

#### **4.7. Procedimiento de Prueba CompAcop-TPr-08**

A continuación se describe la secuencia de pasos requeridos para la ejecución del caso de prueba CA-TC-08, así también las entradas y salidas.

##### **4.7.1. Entradas, Salidas, y Requerimientos Especiales**

Para le ejecución adecuada del caso de prueba CA-TC-08 son necesarios los siguientes requerimientos.

#### **Documentos:**

- Documento de Plan de Pruebas
- Documento de Diseño de Pruebas
- Documento de Procedimiento de Pruebas
- Documento de Reporte de Pruebas

#### **Servidores:**

- Tomcat 7

#### **Aplicaciones:**

- Interfaz de usuario

#### **4.7.2. Descripción de los Pasos a Seguir Para Ejecutar El Caso de Prueba CompAcop-TC-08**

**Reportes:** La elaboración de los reportes es de forma manual al hacer una inspección visual de los eventos que ocurren a lo largo de la ejecución de la prueba y serán registrados en el documento de reporte de pruebas, de esta misma forma los errores, problemas o anomalías serán registrados en el mismo documento.

**Preparación:** Verificar que el servidor tomcat está en ejecución y la interfaz está disponible, además debe estar registrado el usuario en la herramienta dotProject.

**Inicio:** Entrar a la interfaz de usuario usando el navegador Mozilla Firefox

**Proceso:** Ingresar las credenciales de un usuario previamente registrado en dotProject.

**Medición:** La interfaz debe estar disponible, al autenticar exitosamente mostrará un mensaje de bienvenida.

**Suspender:** En caso de encontrar fallas en la ejecución o la autenticación detener la prueba.

**Reiniciar:** Se puede reiniciar la prueba en cualquier momento teniendo en cuenta el apartado de preparación arriba mencionado.

**Cierre:** La prueba termina con el registro en el documento de reporte de pruebas.

**Contingencias:** Al presentarse anomalías la prueba tiene que ser suspendida y el error o fallo corregidos. Para poder reiniciar la prueba de debe haber corregido la anomalía.

## 5. Registro de pruebas

### 5.1. Introducción

En este documento tiene la función de proveer un registro cronológico de detalles relevantes sobre la ejecución de las pruebas.

#### 5.1.1. Alcance

El alcance de los registros de pruebas abarca los procedimientos de prueba CA-TPr-01, CA-TPr-02, CA-TPr-03, CA-TPr-04, CA-TPr-05, CA-TPr-06, CA-TPr-07, CA-TPr-08 definidos en el documento de Procedimientos de Pruebas.

#### 5.1.2. Referencias

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

### 5.2. Registro de prueba CA-TL-01

A continuación se describe el registro de prueba para el procedimiento de prueba CA-TPr-01, así como las actividades y eventos.

#### 5.2.1. Descripción

El registro de prueba CA-TL-01 cubre el procedimiento de prueba CA-TPr-01 correspondiente al caso de prueba CA-TC-01. Los datos usados en la ejecución de la prueba son datos ficticios.

#### 5.2.2. Actividades y eventos de la prueba

En la tabla B.27 se describe el registro de eventos y actividades al ejecutar el procedimiento de prueba CA-TPr-01.

**Tabla B.27** Registro de eventos y actividades CA-TPr-01

Paso	Actor	Descripción	Verificado
1	Encargado de pruebas	Prepara datos a ingresar y documentación de pruebas.	Si
2	Servidor Apache	Inicia servidor web	Si
3	MySQL	Inicia gestor de bases de datos	Si

4	OSGi	Inicia servidor OSGi y componente CAUsuarios	Si
5	Encargado de pruebas	Entra a dotProject a través de Mozilla Firefox	Si
6	Encargado de pruebas	Registrar una nueva cuenta de usuario a través del formulario proporcionado por dotProject	Si
7	CAUsuarios	Identifica el registro de una nueva cuenta de usuario en dotProject.	Si
8	CAUsuarios	Extrae los datos necesarios de la nueva cuenta de usuario	Si
9	CAUsuarios	Agrega una nueva cuenta de usuario en MantisBT	Si
10	CAUsuarios	Registra la relación entre las cuentas de usuario en la base de datos CABD	Si
11	CAUsuarios	Se pone en modo de espera	Si
12	Encargado de pruebas	Verifica la existencia de la nueva cuenta de usuario en MantisBT	Si
13	Encargado de pruebas	Documentar la prueba	Si

### 5.2.3. Anomalías

En la (Tabla B.28) se describen las anomalías encontradas en las pruebas. La columna Paso contiene el número del paso de la lista anterior en el que se originó la anomalía. Se presenta una breve descripción, así como la solución.

**Tabla B.28** Anomalías CA-TPr-01

Paso	Descripción	Solución
12	Se agregó la nueva cuenta de usuario en la herramienta MantisBT pero con contraseña diferente	Se encriptó la cadena antes de guardarla

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

A continuación se muestran los datos de un proyecto registrado con el fin de probar el requerimiento funcional *Integrar usuarios*. (Tabla B.29).

**Tabla B.29** Ejemplo de datos utilizados CA-TPr-01

Nombre del campo	Tipo	Valor
Nombre (login)	Cadena	daniel

Tipo de usuario	Cadena	Administrador
Rol de usuario	Cadena	Administrador
Contraseña	Cadena	daniel
Nombre	Cadena	Daniel
Apellido	Cadena	Aguilar
Empresa/compañía	Cadena	Cenidet
Correo electrónico	Cadena	daniel@cenidet.edu.mx
Firma de correo electrónico	Cadena	-Daniel Aguilar - Cenidet

### 5.3. Registro de Prueba CA-TL-02

A continuación se describe el registro de prueba para el procedimiento de prueba CA-TPr-02, así como las actividades y eventos.

#### 5.3.1. Descripción

El registro de prueba CA-TL-02 cubre el procedimiento de prueba CA-TPr-02 correspondiente al caso de prueba CA-TC-02. Los datos usados en la ejecución de la prueba son datos ficticios.

#### 5.3.2. Actividades y eventos de la prueba

A continuación se describe el registro de eventos y actividades al ejecutar el procedimiento de prueba CA-TPr-02. (Tabla B.30).

**Tabla B.30** Registro de eventos y actividades CA-TPr-02

Paso	Actor	Descripción	Verificado
1	Encargado de pruebas	Prepara datos a ingresar y documentación de pruebas.	Si
2	Servidor Apache	Inicia servidor web	Si
3	MySQL	Inicia gestor de bases de datos	Si
4	OSGi	Inicia servidor OSGi y componente CAProyectos	Si
5	OpenRefine	Iniciar el servidor OpenRefine	
6	Encargado de pruebas	Entra a dotProject a través de Mozilla Firefox	Si
7	Encargado de pruebas	Registrar un nuevo proyecto a través del formulario proporcionado por dotProject	Si
8	CAProyectos	Identifica el registro de un nuevo proyecto en dotProject.	Si

9	CAProyectos	Extrae los datos necesarios del nuevo proyecto	Si
10	CAProyectos	Agrega un nuevo proyecto en MantisBT con los datos extraídos	Si
11	CAProyectos	Agrega un nuevo proyecto en OpenRefine con los datos extraídos	Si
12	CAProyectos	Agrega un nuevo proceso en RapidMiner con los datos extraídos	Si
13	CAProyectos	Registra la relación entre los proyectos en la base de datos CABD.	Si
14	CAProyectos	Se pone en modo de espera	Si
15	Encargado de pruebas	Verifica la existencia del nuevo proyecto en MantisBT	Si
16	Encargado de pruebas	Verifica la existencia del nuevo proyecto en OpenRefine	Si
17	Encargado de pruebas	Verifica la existencia del nuevo proceso en RapidMiner	Si
18	Encargado de pruebas	Documentar la prueba	Si

### 5.3.3. Anomalías

A continuación se describen las anomalías encontradas en las pruebas. La columna Paso contiene el número del paso de la lista anterior en el que se originó la anomalía. Se presenta una breve descripción, así como la solución. (Tabla B.31).

**Tabla B.31** Anomalías CA-TPr-02

Paso	Descripción	Solución
13	Se agregó el nuevo registro con los valores de los id's correctos a excepción del id correspondiente al campo OpenRefine, aquí se agregó un -1	Al crear el proyecto en OpenRefine el método setID se llamaba dos veces, una pasando el id correcto y después se le asignaba un valor de -1, así que se eliminó la segunda llamada

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

En la tabla B.32 se muestran los datos de un proyecto registrado con el fin de probar el requerimiento funcional *Integrar Proyectos*.

**Tabla B.32** Ejemplo de datos utilizados CA-TPR-02

Nombre del campo	Tipo	Valor
Nombre	Cadena	Mining software repositories
Propietario	Cadena	Admin
Empresa/compañía	Cadena	CENIDET
Fecha de inicio	Fecha	03/09/2015
Fecha de fin objetivo	Fecha	08/10/2015
Prioridad	Cadena	normal
Nombre corto	Cadena	msr
Color de identificación	Cadena	990000
Tipo de proyecto	Cadena	Operative
Estado del proyecto	Cadena	In Planning
Importar tareas de	Cadena	1 Plantilla MIDADS (recomendado)
Descripción del proyecto	Texto	Test – Mining software repositories

#### 5.4. Registro de Prueba CA-TL-03

A continuación se describe el registro de prueba para el procedimiento de prueba CA-TPR-03, así como las actividades y eventos.

##### 5.4.1. Descripción

El registro de prueba CA-TL-03 cubre el procedimiento de prueba CA-TPR-03 correspondiente al caso de prueba CA-TC-03. Los datos usados en la ejecución de la prueba son datos ficticios.

##### 5.4.2. Actividades y Eventos de la Prueba

En la tabla B.33 se describe el registro de eventos y actividades correspondientes al procedimiento de prueba CA-TPR-03.

**Tabla B.33** Registro de eventos y actividades CA-TPR-03

Id	Actor	Descripción	Verificado
1	Encargado de pruebas	Preparar datos de la incidencia a registrar, así mismo la documentación de pruebas	Si
2	Servidor apache	Inicia servidor web	Si
3	MySQL	Inicia manejador de bases de datos	Si

4	OSGi	Inicia servidor y componente CATareas	Si
5	Encargado de pruebas	Entra a la aplicación MantisBT usando de Mozilla Firefox	Si
6	Encargado de pruebas	Registra, mediante el formato, la información respecto a una nueva incidencia	Si
7	CATareas	Identifica la nueva incidencia y extrae sus datos	Si
8	CATareas	Agrega una nueva tarea en dotProject asociándola con el proyecto y usuario equivalente	Si
9	CATareas	Se pone en modo de espera	Si
10	Encargado de pruebas	Verifica la tarea en dotProject	Si
11	Encargado de pruebas	Documentar la prueba	Si

#### 5.4.3. Anomalías

A continuación se describen las anomalías encontradas en las pruebas. La columna Paso contiene el número del paso de la lista anterior en el que se originó la anomalía. Se presenta una breve descripción, así como la solución. (Tabla B.34).

**Tabla B.34** Anomalías CA-TPR-03

Actividad	Descripción	Solución
14	La tarea se agrega pero el usuario correspondiente no	La variable que almacena el id del usuario tomaba el valor de 0 antes de insertar, se agregó la sentencia correcta

#### 5.4.4. Ejemplo de Datos Utilizados para la Prueba

A continuación se muestran los datos de un proyecto registrado con el fin de probar el requerimiento funcional *Integrar Tareas*. (Tabla B.35).

**Tabla B.35** Ejemplo de datos utilizados CA-TPR-03

Nombre	Tipo de Valor	Valor
Categoría	Cadena	General
Reproducibilidad	Cadena	Aun no intentada
Severidad	Cadena	Menor

Prioridad	Cadena	Normal
Asignado a	Cadena	Daniel
Resumen	Cadena	Falta campo
Descripción	Texto	Falta campo que se puede calcular
Pasos para reproducirlo	Texto	Vacío
Información adicional	Texto	El nuevo campo se calcula con el producto de los dos campos anteriores
Archivo	Archivo	Vacío

### 5.5. Registro de Prueba CA-TL-04

A continuación se describe el registro de prueba para el procedimiento de prueba CA-TPr-04, así como las actividades y eventos.

#### 5.5.1. Descripción

El registro de prueba CA-TL-04 cubre el procedimiento de prueba CA-TPr-04 correspondiente al caso de prueba CA-TC-04. Los datos usados en la ejecución de la prueba son datos ficticios.

#### 5.5.2. Actividades y Eventos de la Prueba

En la tabla B.36 se describe el registro de eventos y actividades correspondientes al procedimiento de prueba CA-TPr-04.

**Tabla B.36** Registro de eventos y actividades CA-TPr-04

Id	Actor	Descripción	Verificado
1	Encargado de pruebas	Prepara el documento a adjuntar y la documentación de pruebas	Si
2	Servidor apache	Inicia servidor web	Si
3	MySQL	Inicia manejador de bases de datos	Si
4	OSGi	Inicia servidor OSGi y componentes CAArchivos	Si
5	Tester	Entra a la aplicación dotProject o MantisBT usando Firefox	Si
6	Tester	Adjunta el archivo a un defecto-tarea que previamente ya está relacionado	Si
7	CAArchivos	Identifica el archivo y crea una copia en la herramienta complementaria	Si

8	CAArchivos	Se pone en modo de espera	Si
9	Encargado de pruebas	Se asegura que el archivo se encuentre agregado correctamente en dotProject o MantisBT según corresponda	Si
10	Encargado de pruebas	Documentar la prueba	Si

### 5.5.3. Anomalías

A continuación se describen las anomalías encontradas en las pruebas. La columna Paso contiene el número del paso de la lista anterior en el que se originó la anomalía. Se presenta una breve descripción, así como la solución. (Tabla B.37).

**Tabla B.37** Anomalías CA-TPr-04

Actividad	Descripción	Solución
15	El componente se detiene al encontrar un error	La variable is de tipo InputStream recibía un parámetro vacío

### 5.5.4. Ejemplo de Datos Utilizados para la Prueba

A continuación se muestran los datos de un proyecto registrado con el fin de probar el requerimiento funcional *Integrar Archivos*. (Tabla B.38).

**Tabla B.38** Ejemplo de datos utilizados CA-TPr-04

Columna	Tipo	Valor
Archivo	Archivo	alProjectsTopic2014-Mar.txt

## 5.6. Registro de Prueba CA-TL-05/06/07

A continuación se describe el registro de prueba para los procedimientos de prueba CA-TPr-05/06/07, así como las actividades y eventos.

### 5.6.1. Descripción

Los registros de prueba CA-TL-05/06/07 cubren los procedimientos de prueba CA-TPr-05/06/07 respectivamente, correspondientes a los casos de pruebas CA-TC-05/06/07 respectivamente. Las entradas utilizadas fueron los comandos con los parámetros *start*, *stop* y *restart*.

### 5.6.2. Actividades y Eventos de la Prueba

En la tabla B.39 se describen el registro de eventos y actividades correspondientes a los procedimientos de prueba CA-TPR-05/06/07.

**Tabla B.39** Registro de eventos y actividades CA-TPR-05-06-07

<b>Id</b>	<b>Actor</b>	<b>Descripción</b>	<b>Verificado</b>
1	Encargado de pruebas	Reunir los documentos de prueba	Si
2	OSGi	Inicia servidor OSGi	Si
3	Encargado de pruebas	Inicia al componente Starter con el comando start seguido del nombre del componente o su id	Si
4	OSGi	Inicia al componente Starter	Si
5	Starter	Inicia al componente CANucleo	Si
6	CANucleo	Revisa los componentes que implementa la interfaz CAI y los almacena en una lista	Si
7	CANucleo	Inicia al componente CAUsuarios	Si
8	CANucleo	Inicia al componente CAProyectos	Si
9	CANucleo	Inicia al componente CATareas	Si
10	CANucleo	Inicia al componente CAArchivos	Si
11	Starter	Se pone en modo de espera	Si
12a	Starter	No recibe ninguna acción y se repiten actividades 4 a 10	Si
12b	Encargado de pruebas	Ejecuta el comando restart seguido del nombre o id del componente y se repiten actividades 4 a 11	Si
12c	Encargado de pruebas	Ejecuta el comando stop seguido del nombre o id del componente	Si
13	Encargado de pruebas	Verifica los componentes se hayan ejecutado correctamente	Si
14	Encargado de pruebas	Se llena la documentación de prueba necesaria	Si

### 5.6.3. Anomalías

En la tabla B.40 se presentan las anomalías resultado de la prueba. La columna Actividad muestra a las actividades de la sección anterior a las que afecta la anomalía. La columna Descripción provee un resumen del problema presentado. La columna Solución muestra cómo se resolvió el problema.

**Tabla B.40** Anomalías CA-TPr-05-06-07

Actividad	Descripción	Solución
10 y 11	No se ejecutan los componente	Falta clausula <i>Bundle-ActivationPolicy</i> con el valor <i>lazy</i> en el archivo MANIFEST.MF de cada componente

#### 5.6.4. Ejemplo de Datos Utilizados para la Prueba

Para la ejecución de esta prueba se utilizaron los diferentes parámetros de ejecución del componente Starter. (Tabla B.41).

**Tabla B.41** Ejemplo de datos utilizados CA-TPr-05-06-07

Parámetro	Función
<b>start</b>	Inicia la ejecución del sistema
<b>stop</b>	Detiene la ejecución del sistema.
<b>restart</b>	Reanuda la ejecución del sistema.

#### 5.7. Registro de Prueba CA-TL-08

A continuación se describe el registro de prueba para el procedimiento de prueba CA-TPr-08, así como las actividades y eventos.

##### 5.7.1. Descripción

El registro de prueba CA-TL-08 cubre el procedimiento de prueba CA-TPr-08, correspondiente al caso de prueba CA-TC-08. Las entradas utilizadas fueron las credenciales de usuarios previamente registrados en dotProject.

##### 5.7.2. Actividades y Eventos de la Prueba

En la tabla B.42 se describen el registro de eventos y actividades correspondientes al procedimiento de prueba CA-TPr-08.

**Tabla B.42** Registro de eventos y actividades CA-TPr-08

Id	Actor	Descripción	Verificado
1	Encargado de pruebas	Preparar datos de la incidencia a registrar, así mismo la documentación de pruebas	Si
2	Servidor	Inicia servidor	Si

	tomcat		
3	MySQL	Inicia manejador de bases de datos	Si
4	Encargado de pruebas	Entra a la interfaz de usuario de AMDADS usando Mozilla Firefox	Si
5	Encargado de pruebas	Ingresa los datos del usuario	Si
6	Interfaz	Autentifica las credenciales	Si
7	Interfaz	Muestra mensaje de bienvenida	Si
8	Encargado de pruebas	Selecciona la opción <i>Proyectos</i> para ver la lista de proyectos asociados al momento.	Si
9	Interfaz	Lista los proyectos que el usuario tiene asociados	Si
10	Encargado de pruebas	Documentar la prueba	Si

### 5.7.3. Anomalías

En la tabla B.43 se presentan las anomalías resultado de la prueba. La columna Actividad muestra a las actividades de la sección anterior a las que afecta la anomalía. La columna Descripción provee un resumen del problema presentado. La columna Solución muestra cómo se resolvió el problema.

**Tabla B.43** Anomalías CA-TPr-08

Actividad	Descripción	Solución
9	Muestra la lista de proyectos correctamente pero el avance mostrado no corresponde al avance mostrado en la herramienta dotProject	Este avance se calcula cuando el usuario solicita la información de un proyecto, el cálculo presentaba un error y fue corregido

### 5.7.4. Ejemplo de Datos Utilizados para la Prueba

Para la ejecución de esta prueba se utilizaron credenciales de usuarios previamente registrados en la herramienta dotProject. (Tabla B.44).

**Tabla B.44** Ejemplo de datos utilizados CA-TPr-08

Columna	Tipo	Valor
Usuario	Cadena	daniel
Contraseña	Cadena	daniel

## 6. Reporte de pruebas

### 6.1. Introducción

En este documento se resumen las actividades de prueba y sus resultados, además contiene evaluaciones de las correspondientes pruebas.

#### 6.1.1. Alcance

El alcance de esta documentación se centra en los casos de prueba descritos en el documento de casos de prueba y son los siguientes: CA-TC-01, CA-TC-02, CA-TC-03, CA-TC-04, CA-TC-05/06/07 y CA-TC-08.

#### 6.1.2. Referencias

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

### 6.2. Detalles

A continuación se muestran los resultados de las pruebas, conclusiones y recomendaciones. (Tabla B.45).

#### 6.2.1. Resultados Generales

**Tabla B.45** Resultados generales

Requerimiento	Componente de Acoplamiento	Caso de Prueba	Resultado de la Prueba
1. Integrar Proyectos	CAProyectos	CA-TC-01	Exitoso
2. Integrar Usuarios	CAUsuarios	CA-TC-02	Exitoso
3. Integrar Tareas	CATareas	CA-TC-03	Exitoso
4. Integrar Archivos	CAArchivos	CA-TC-04	Exitoso
5. -	CANucleo	CA-TC-05	Exitoso
6. -	CAI	CA-TC-06	Exitoso
7. -	CASstarter	CA-TC-07	Exitoso
8. -	Interfaz de usuario	CA-TC-08	Exitoso

Después de corregir las anomalías internas fueron corregidas. Las pruebas fueron realizadas bajo condiciones normales y no se presentaron anomalías externas. Al final todas las pruebas fueron exitosas.

### 6.2.2. Resultados Detallados

A lo largo de las pruebas se presentaron anomalías que van desde pequeñas que permitían al sistema seguir su ejecución normal, hasta anomalías mayores que detenían por completo la ejecución del sistema, al final todas fueron corregidas. Todo esto resultó en el cumplimiento de los requerimientos por completo.

Los proyectos generados son complementarios para cumplir con el proceso de minería de datos de desarrollo de software, pero aún faltan herramientas que harían el sistema más completo.

El componente *CAProyectos* encargado del requerimiento *Integrar Proyectos* tarda más que los otros componentes debido a la implementación de API's, aun así no afecta en nada la ejecución del sistema.

La implementación de las especificaciones OSGI es una ventaja ya que facilita la inclusión de nuevos componentes sin modificar la ejecución de los demás, lo que hace al sistema escalable. Y en el caso de detener un componente los demás siguen trabajando normalmente a menos que explícitamente se le indique la dependencia.

Otra ventaja del uso de OSGI es el poder usar componentes versionados, así en el caso de que se requiera la ejecución de dos componentes pero diferentes versiones no haya problemas.

La interfaz de usuario es de gran ayuda, ya que proporciona información general del usuario y los proyectos que tiene asociados al momento además de la ayuda e información proporcionada como los son; los manuales de usuario y tutoriales tanto de las herramientas con del ambiente de soporte a minería de datos de desarrollo de software; información sobre las metodologías usadas para la elaboración de las plantillas; y acceso a las herramientas integradas.

### **6.2.3. Conclusiones y Recomendaciones**

El uso de un sistema integrado facilita el desarrollo de algún proceso en este caso el de minería de datos de desarrollo de software.

Se usaron dos tipos de integración y se mostró su éxito al aplicarlo.

Se recomienda abarcar las demás etapas del proceso de minería de datos ya que en este trabajo se hizo énfasis en la etapa de preparación de datos, lo que implicaría la integración de más herramientas y el uso de otros tipos de integración, además se recomienda ampliar la funcionalidad de los componentes para permitir la modificación de registro ya que en este caso solo se cubrió el registro de estos.