

## Componentes de acoplamiento para la infraestructura de soporte de minería de datos de desarrollo de software

PÉREZ-LUNA, Edgardo\*† y GONZÁLEZ-GARCÍA, Moisés.

*Centro Nacional de Investigación y Desarrollo Tecnológico, Interior Internadao Palmira S/N, Palmira, 62490 Cuernavaca, Mor., México*

Recibido Octubre 14, 2016; Aceptado Noviembre 7, 2016

### Resumen

Los datos de ingeniería de software (reportes de incidencias, documentación, código) contienen información potencialmente útil que al ser aplicada a futuros proyectos de desarrollo ayuda a mejorar la productividad y calidad del software. Las técnicas de minería de datos son una buena opción para analizar estos datos debido a los resultados exitosos que han mostrado estas técnicas en otros rubros (como seguridad o bancarios). Para soportar el proceso de minería de datos existen herramientas, las cuales cubren las necesidades de alguna de las etapas del proceso trabajando de manera aislada, ocasionando que algunas tareas tengan que realizarse más de una vez (ej. registrar un nuevo proyecto). En este artículo se presenta un ambiente que soporta el proceso de minería de datos, que permiten a herramientas que soportan la minería de datos trabajar de manera conjunta; en el cual se integran dichas herramientas usando componentes de acoplamiento desarrollados siguiendo las especificaciones de la *Open Services Gateway Initiative* (OSGI); con el fin de evitar la captura repetitiva de los mismos datos en diferentes herramientas y disminuir la carga de trabajo al usuario.

**Minería de datos, Desarrollo de software, Repositorios de software, OSGI**

### Abstract

Software engineering data (bug reports, documentation, and code) contains information potentially useful that being applied to future development projects help to improve software productivity and quality. Data mining techniques are a good option for analyzing this data due to their successful results in other fields (such as security or banking). To support data mining process exist tools which cover the needs of some of the process stages working in isolation, causing that some of the tasks have to perform more than once (e.g. recording a new project). In this article is presented an environment that supports the data mining process that allows data mining support tools to work together; wherein those tools are integrated using coupling components developed following the *Open Services Gateway Initiative* specifications; with the purpose of avoid repetitive typing of the same input data in different tools and reduce user workload.

**Data mining, Software development, Software repositories, OSGI**

**Citación:** PÉREZ-LUNA, Edgardo y GONZÁLEZ-GARCÍA, Moisés. Componentes de acoplamiento para la infraestructura de soporte de minería de datos de desarrollo de software. *Revista de Sistemas Computacionales y TIC'S* 2016, 2-6: 99-111

\* Correspondencia al Autor (Correo Electrónico: edgardo@cenidet.edu.mx)

† Investigador contribuyendo como primer autor.

## Introducción

La aplicación del proceso de minería de datos ha dado buenos resultados en muchos campos. Uno de los campos de creciente aplicación de la minería de datos es la ingeniería de software. Los datos generados durante el proceso de desarrollo de software se almacenan en repositorios y son una fuente potencial para la obtención de conocimiento, que aplicado en la toma de decisiones para proyectos futuros mejore la productividad y calidad del software.

Para soportar cada una de las fases del proceso de minería de datos existen herramientas de software 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 esta investigación no se encontró alguna integración de herramientas libres y de código abierto (FLOSS) que soporten el proceso de minería de datos de desarrollo de software.

Debido a esta carencia, en esta investigación se crearon *componentes de acoplamiento* desarrollados siguiendo las especificaciones de la *Open Services Gateway Initiative* (OSGI). Que integran herramientas FLOSS en un ambiente que soporta el proceso de minería de datos de desarrollo de software (que denominamos AMDADS). Con el fin de evitarla captura repetitiva de los mismos datos en diferentes herramientas y disminuir la carga de trabajo al usuario. AMDADS actualmente está enfocado principalmente en la etapa de preparación de datos, que es la etapa que implica mayor atención y el mayor esfuerzo.

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 registro en las demás herramientas, además se validó la creación de un repositorio que incluye el ID (identificación) 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.

Las secciones en las que está dividido este artículo se describen a continuación: primeramente se presenta explicada con más detalle la problemática que dio origen a esta investigación así como la solución propuesta. Posteriormente se presentan las herramientas integradas en AMDADS, la metodología empleada para la evaluación y selección de cada herramienta y los resultados obtenidos, continuando con los tipos de integración de herramientas empleadas en este trabajo. En la sección posterior se describen las funciones de los *componentes de acoplamiento* dentro de AMDADS para poder comprender la sección del modelo de integración empleado en este trabajo. Para poder entablar algún tipo de comunicación directa con el usuario final AMDADS cuenta con una interfaz de usuario, que proporciona información general del usuario como su perfil y los proyectos que tiene asociados al momento, esto se incluye en la sección Interfaz de usuario. Para validar la funcionalidad de AMDADS se presenta la sección Pruebas, donde se explica a detalle el proceso pruebas. Finalmente se presentan las conclusiones y trabajos futuros propuestos.

**Problemática**

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

La aplicación del proceso de minería de datos implica el uso de diferentes herramientas que en sus bases de datos usan datos con formatos que pueden ser incompatibles. Además, al requerirse varias herramientas se realizan actividades o tareas repetitivas que generan trabajo extra (Ej. Dar de alta usuarios y proyectos, o dar de alta automáticamente tareas o archivos que se requieren en varias herramientas).

Al inicio del proyecto, para mejorar el proceso se hace necesaria la planificación de proyectos y el seguimiento de incidencias (asuntos inesperados que requieren atención), con el fin de monitorearlas hasta que se le dé alguna solución a cada una.

**Solución**

En el contexto de esta problemática, 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 para proporcionar una forma de facilitar la gestión del proyecto y el seguimiento de las incidencias encontradas.

Se desarrollaron componentes (módulos) Java que monitorean cada cierto tiempo las bases de datos en espera de la inserción de un nuevo registro para posteriormente replicar dicho registro en las bases de datos de las demás herramientas.

Los módulos desarrollados reciben el nombre de *componentes de acoplamiento*, se desarrollaron usando la plataforma OSGI (*Open Services Gateway Initiative*, que define un sistema modular dinámico para Java, que da 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), para aprovechar las ventajas que proporciona su implementación.

Se desarrolló un componente encargado de cada uno de los tipos de registro que se manejan (usuarios, proyectos, incidencias/tareas o archivos). Cada uno de los *componentes de acoplamiento* extrae 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 réplica el componente extraerá el id del registro replicado, con el fin de almacenarlo en un *repositorio de relaciones* para mantener relacionados a los registros equivalentes.

Para validar la funcionalidad de los componentes se realizaron pruebas que verificaron que un nuevo registro se replicó en las demás herramientas y se mantuvieron sus relaciones correspondientes, estas pruebas se realizaron con registros sintéticos.

**Selección de herramientas a integrar**

Para hacer la comparación y evaluación de las herramientas FLOSS a integrar en AMDADS, 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*).

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 y establecer pesos relacionados al contexto
- Seleccionar las herramientas apropiadas

Como parte del primer paso se elaboró un mapa mental sobre los criterios a evaluar en las diferentes herramientas FLOSS (ver Fig. 1) y se estableció el peso de cada criterio a evaluar. Entre los criterios se encuentra la madurez del software, dentro de este criterio están otros sub-criterios entre los que están: gobernabilidad, legado, actividad e industrialización. También se establecen algunos criterios como son: la extensibilidad del software, su simplicidad, el manejo de diferentes formatos de datos, su libertad y su capacidad de ejecutarse en diferentes plataformas.

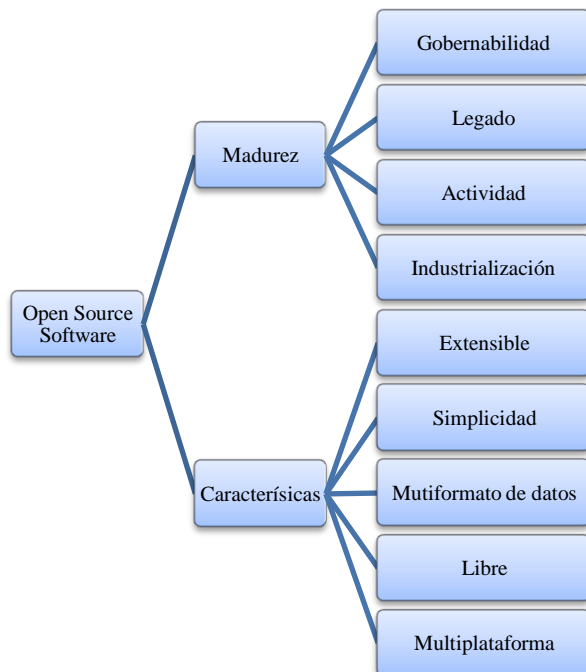


Figura 1 Criterios a evaluar.

Dentro del criterio de Madurez (entre los que están: Gobernabilidad, Legado, Actividad e Industrialización), existen subcriterios que se presentan en las figuras 2, 3,4 y 5.

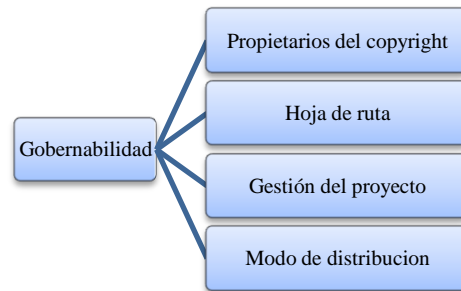


Figura 2 Criterios a evaluar. Gobernabilidad.

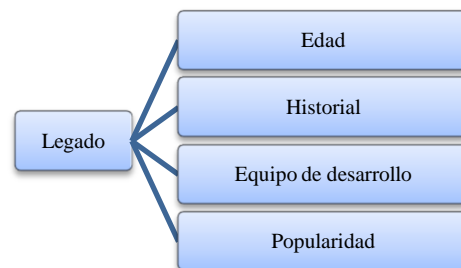


Figura 3 Criterios a evaluar. Legado.



Figura 4 Criterios a evaluar. Actividad.

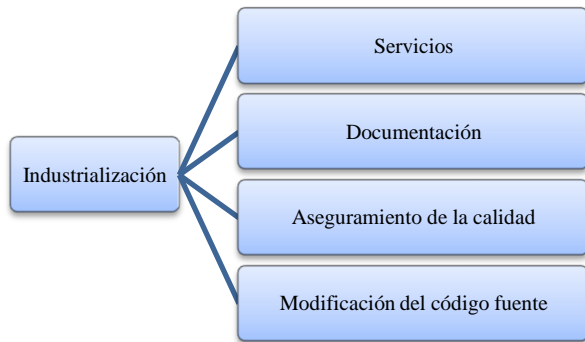


Figura 5 Criterios a evaluar. Industrialización.

Como siguiente paso se cargó la plantilla en el editor *XulEditor*, en este caso se usó el *plug-in* para Firefox. Y se llenó la plantilla con los datos de cada herramienta. Una vez terminadas las evaluaciones se debería haber hecho la comparación utilizando la aplicación Web O3S, para muestra la comparación de forma gráfica. Debido a que la aplicación O3S no se encuentra disponible, en esta investigación se realizó la comparación manual de las herramientas, y se obtuvo un puntaje promedio con las evaluaciones elaboradas anteriormente.

Se realizaron comparaciones entre cuatro grupos de herramientas, que se muestran de manera resumida a continuación (se muestra sombreada la herramienta mejor puntuada):

Herramienta	Knime	Orange	Pentaho	RapidMiner	Weka	SEE	R
Promedio	1.76	1.36	1.8	1.8	1.47	1.33	1.77

Tabla 1 Comparación de herramientas de soporte a técnicas de minería de datos.

Herramienta	EditPadLite	Collabtive	GanttPV	ClockingIT	TaskJugger	dotProject
Promedio	1.13	1.41	1.31	1.71	1.46	1.91

Tabla 2 Comparación de herramientas de soporte a planificación.

Criterio/Herramienta	OpenRefine	MS Excel
Promedio	1.5	1.42

Tabla 3 Comparación de herramientas de soporte a preparación de datos.

Criterio/Herramienta	MS Access	MySQL	PostgreSQL
Promedio	1.47	1.81	2

Tabla 4 Comparación de manejadores de base de datos.

Según los resultados anteriores las herramientas a integrar son:

- Rapidminer
- OpenProject
- OpenRefine
- MySQL

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

Se eligió MySQL ya que dotProject y MantisBT (esta herramienta no fue incluida en la evaluación ya que la idea de su inclusión surgió después de este paso) utilizan este gestor de base de datos para trabajar, aunque PostgreSQL haya salido mejor evaluado.

Con base en los resultados de la comparación y evaluación de las herramientas se eligieron las siguientes herramientas para el ambiente de soporte a minería de datos de desarrollo de software:

**MantisBT** es un software popular de código abierto para el seguimiento de incidencias basado en web que provee balance delicado entre simplicidad y poder.

**MySQL** es el sistema de gestión de base de datos de código abierto más popular, es desarrollado, distribuido y apoyado por Oracle Corporation.

**DotProject** es una aplicación para administración de proyectos soportada por voluntarios.

**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.

**RapidMiner** (Anteriormente llamado YALE) es una poderosa herramienta visual para minería de datos, máquinas de aprendizaje y análisis predictivo.

La suite de ofimática **LibreOffice** no se integra por algún *componente de acoplamiento* pero se agrega también al ambiente debido a su soporte potencial al proceso de minería de datos de desarrollo de software.

### Integración de herramientas

Existen diversas técnicas de integración de herramientas, para la implementación de AMDADS se utilizaron la integración por datos y por API's.

**Integración por Datos:** Se realiza por el 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.

**Integración por API's:** Se realiza por la interacción una herramienta con otras a través de sus API's (del inglés *Application Programming Interface*).

### Funciones de los componentes de acoplamiento

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
- Soportar principalmente las actividades de preparación de datos. Pero el ambiente está abierto a futuras integraciones de herramientas, que soporten el resto de las fases del proceso de MD
- Integrar proyectos, cuentas de usuario, incidencias/tareas y archivos, esto se lleva a cabo una vez que el usuario registra alguno de los elementos mencionados anteriormente, extrae los datos y los replica en las bases de datos de las demás herramientas, con esto evita que se capturen los mismos datos en todas las herramientas.
- Los usuarios tienen que registrarse para obtener un identificador de usuario y una contraseña, para poder hacer uso del ambiente
- Además el usuario podrá navegar por una sencilla interfaz para obtener ayuda con temas relacionados a AMDADS y a las herramientas integradas.

Las actividades y tareas del proceso de minería de datos del desarrollo de software, que se usan son las establecidas en la metodología híbrida MIDADS (Fig.6) propuesta por (Meza 2014), en donde se integran tres metodologías, y tareas adicionales a las establecidas en dichas metodologías. Las tres metodologías mencionadas son: 1) el modelo CRISP-DM (Chapman et al. 2000) que es la más usada y se usa como base; 2) la 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 (Meza 2014), la metodología MIDADS se definió para el dominio de datos de desarrollo de software. Los pasos 1 y 2 de MIDADS están ligados por lo que en la práctica se pueden realizar a la par, se determinan las tareas (programación, detección de errores, etc.) de las cuales se pueden obtener los datos (repositorios históricos, informes de error, etc.) y a cuales se aplicarán los resultados de la minería. El paso 3 es el entendimiento de los datos con el fin de saber que técnica de preparación de datos (paso 4) es la mejor. Dentro de la fase de preparación de datos la metodología MIDADS divide esta fase en sub-fases que pueden ser aplicadas de forma general (a todo tipo de proyecto) o de forma específica (depende del tipo de datos y del proyecto). El paso 5 consiste en la aplicación del algoritmo de minería de datos, una vez teniendo los resultados se procede a su evaluación (paso 6), desde aquí se pueden aplicar los resultados obtenidos (paso 7) o en caso de tener resultados convincentes regresar a los pasos iniciales (paso 1 y 2).

**Componente de acoplamiento**

La implementación de los componentes de acoplamiento se hizo siguiendo las especificaciones de la *Open Services Gateway Initiative (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.

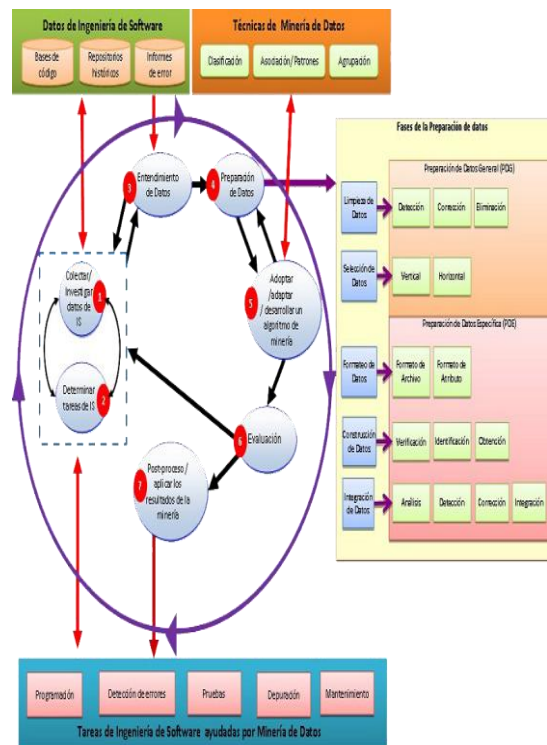


Figura 6 Metodología híbrida MIDADS

Los componentes de acoplamiento (llamados CAStarter, CANúcleo, CAI, CAUsuario, CAArchivos, CAProyectos, CATareas) se desarrollaron totalmente usando Java a excepción del componente encargado de la integración de proyectos (CAProyectos), 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. Es por esto que algunos usuarios han hecho aportaciones a la comunidad con APIs 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 integra 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 se replicaron en las herramientas destino y el elemento de la base de datos de la herramienta original.

### Modelo de integración

En la Fig.7 se observa el flujo que inicia con los datos que ingresa el usuario en la herramienta inicial y se almacenan en su base de datos. Periódicamente el componente de acoplamiento se ejecuta para detectar si hay un 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.

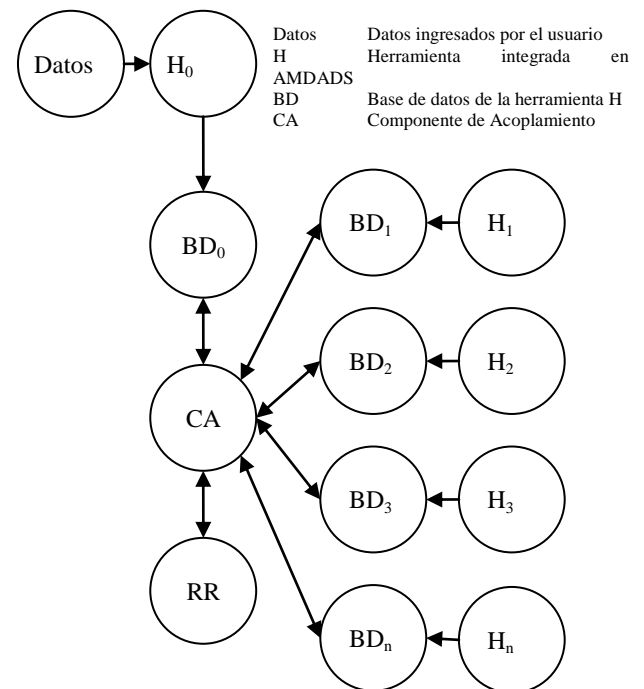


Figura 7 Modelo de integración AMDADS.

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 sus API's.

### Interfaz de usuario

Como parte del desarrollo de AMDADS se implementó una interfaz de usuario que sirve para dar información y ayuda sobre el uso del ambiente. La interfaz consiste en una primera ventana de acceso (para darle la identificación de usuario y su palabra clave), que está relacionada a la base de datos de usuarios de la herramienta dotProject. Esta interfaz está 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*).



Después de dar correctamente la identificación de usuario y la palabra clave, la segunda ventana de la interfaz de usuario es de gran ayuda, ya que proporciona información relevante a través de un menú con cinco opciones: Perfil, Proyectos, Metodologías de minería de datos (MD), Herramientas y Ayuda (ver figura 8). Cada una de estas opciones proporciona al usuario información acerca de la opción seleccionada; la opción Perfil muestra información general acerca del usuario; la opción Proyectos muestra una lista de proyectos asociados al usuario logueado al momento; la opción Metodologías MD muestra información acerca de las metodologías CRISP-DM (Metodología general) Y MIDADS (Metodología recomendada para AMDADS); la opción Herramientas lista las herramientas integradas disponibles para la etapa del proceso de minería de datos seleccionada por el usuario; y la opción Ayuda que brinda información de apoyo para el uso de AMDADS y cada una de las herramientas integradas.



Figura 8 Menú principal.

## Pruebas

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.

Estas pruebas verifican que un nuevo registro se replicó en las demás herramientas y se mantuvieron sus relaciones correspondientes, las pruebas se realizaron con registros sintéticos.

A continuación, en la tabla 5, se muestra la matriz de trazabilidad de pruebas del ambiente de soporte a minería de datos de desarrollo de software AMDADS. La primera columna incluye el requerimiento a ser cubierto por el componente de acoplamiento descrito en la segunda columna, por último en la tercera columna se incluye el identificador del caso de prueba que se aplicará al componente de acoplamiento y que cubrirá el requerimiento de la primera columna. Con esto los requerimientos quedarán completamente cubiertos por las pruebas realizadas. La columna *requerimiento* incluye el valor N/A lo que indica que la prueba realizada no cubre ningún requerimiento.

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
N/A	CANucleo	CA-TC-05
N/A	CAI	CA-TC-06
N/A	CAStarter	CA-TC-07

Tabla 5 Matriz de trazabilidad de pruebas.

Otro elemento a probar es la interfaz de usuario, en la tabla 6 se muestran las características a probar y el 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

Tabla 6 Matriz de trazabilidad para la interfaz de usuario.

A continuación se describe el caso de prueba CA-TC-02 cuyo objetivo es validar el cumplimiento del requerimiento Integrar Proyectos que corresponde al componente de acoplamiento CAProyectos, en la tabla 7 se muestra este caso de prueba.

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

**Tabla 7** Caso de prueba CA-TC-02.

En la tabla 8 se muestran las entradas solicitadas por la herramienta dotProject donde es el punto de partida para realizar la integración de proyectos, posteriormente el componente de acoplamiento CAProyectos extrae los datos ingresados y los replica en las bases de datos de las herramientas integradas.

dotProject	
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

**Tabla 8** Datos de entrada para la herramienta dotProject.

La salida o resultado del componente de acoplamiento CAProyectos es el registro automático de un proyecto en las herramientas MantisBT, OpenRefine y un proceso en RapidMiner con los datos previamente registrados en la herramienta dotProject. También se guarda la relación de todos estos nuevos registros usando el id de cada nuevo proyecto. En la tabla 9 se muestran los datos esperados en la herramienta MantisBT.

MantisBT	
Nombre del campo	Tipo
Nombre	Cadena
Descripción	Texto
Estado	Cadena

**Tabla 9** Salida en la herramienta MantisBT.

En la tabla 10 se muestran los datos esperados en la herramienta OpenRefine, en este caso solo es el nombre del proyecto.

OpenRefine	
Nombre del campo	Tipo
Nombre	Cadena

**Tabla 10** Salida en la herramienta OpenRefine.

En la tabla 11 se muestran los datos esperados en la herramienta Rapidminer.

Rapidminer	
Nombre	Tipo
Directorio	Cadena
Nombre	Cadena

**Tabla 11** Salida en la herramienta Rapidminer.

Otro elemento a probar fueron los módulos CANucleo, CAI y CAStarter necesarios para la ejecución de los componentes CAUsuario, CAProyectos, CATareas y CAArchivos, a continuación se muestra la matriz de trazabilidad, incluyendo las características de cada componente de acoplamiento a probar. Las pruebas se realizaron con las ejecución de los componentes a través de las acciones start, stop, restart y sleep.

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

**Tabla 12** Casos de prueba 05-06-07.

### Reporte de pruebas

Durante el proceso de pruebas se encontraron algunas anomalías las cuales fueron resueltas y al final todos los requerimientos fueron completamente cubiertos. En la tabla 13 se muestran los resultados de las pruebas, todas las pruebas fueron exitosas.

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
N/A	CANúcleo	CA-TC-05	Exitoso
N/A	CAI	CA-TC-06	Exitoso
N/A	CASarter	CA-TC-07	Exitoso
N/A	Interfaz de usuario	CA-TC-08	Exitoso

**Tabla 13** Resultados generales de las pruebas.

### 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 (como pueden ser proyectos, cuentas de usuario, tareas, etc.), pero se muestra el beneficio del uso de un ambiente integrado, pues facilita la realización de tareas por ejemplo: dar de alta una cuenta de usuario en una herramienta y poder usar esas credenciales para ingresar a una herramienta diferente. 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 y provee una clara definición de su modularidad.

Todo el proceso realizado en la investigación se desarrolló siguiendo tal es el caso de la metodología QSOS para evaluación y selección de herramientas libres y de código abierto 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) la criptografía que manejan las herramientas para almacenar contraseñas. 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 integrandola 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. Que 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 (ej. 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 era: 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: dotProject, RapidMiner, MantisBT y OpenRefine.
- El uso de la metodología de minería de datos de desarrollo de software (MIDADS), se facilitó mediante 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 proporciona 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.

### 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:

- Ampliar las funcionalidades necesarias para modificar o eliminar registros (usuarios, proyectos, tareas, archivos), con el fin de promover el uso de AMDADS.
- Determinar la manera de integrar herramientas adicionales, que cubran en su totalidad otras etapas del proceso de minería de datos de desarrollo de software.

### Referencias

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.

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.

Xie, T., Thummalapenta, S., Lo, D., & Liu, C. (2009). Data mining for software engineering. *Computer*, 42(8), 55-62.

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.

QSOS, Metodología de comparación y evaluación de herramientas FLOSS <http://www.qsos.org> Fecha de consulta: Septiembre 2014.

KDNUGGETS, Portal dedicado a Minería de datos, análisis de negocios y Big Data <http://www.kdnuggets.com>, Fecha de consulta: Septiembre 2014.

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.

Thomas, I., & Nejme, B. A. (1992). Definitions of tool integration for environments. *Software, IEEE*, 9(2), 29-35.

Hall, R. S., Pauls, K., McCulloch, S., & Savage, D. (2011). *OSGi in action. Creating Modular Applications in Java.*