Universidad de las Ciencias Informáticas
Facultad 2

Título: Herramienta informática de Minería de Uso de la Web sobre los registros de navegación por Internet. Implementación del módulo para realizar la tarea descriptiva Reglas de Asociación.

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autores: Julio Antonio Hernández Pérez

Husseyn Despaigne Reyes

Tutor: Lic. Darian Horacio Grass Boada

Co-tutores: Yoanni Ordoñes Leyva

Ernesto Avilés Vázquez

La Habana, Junio de 2011. “Año 53 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Facultad 2 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Firma de los Autores:

Julio Antonio Hernández Pérez

_____

Husseyn Despaigne Reyes

_____

Firma del Tutor:

Darian Horacio Grass Boada

_____

DATOS DE CONTACTO

Síntesis del Tutor:

Graduado en el año 2005 como Lic. Ciencias de la Computación en la Universidad de Oriente. Ha impartido durante 5 cursos las asignaturas de Programación-2 e Inteligencia Artificial. Posee la categoría docente de profesor Asistente desde el año 2008. Se encuentra matriculado en la maestría de Ciencias de la Computación en la Universidad de La Habana. Investiga las líneas relacionadas con: Minería de Datos, Sistemas Inteligentes y Algoritmos Meta heurísticos.

Correo: dgrass@uci.cu

Provincia: Santiago de Cuba

RESUMEN

La Minería de Datos se encarga de encontrar variaciones de comportamiento en un alto volumen de información. En este trabajo se presenta una herramienta informática que permite descubrir patrones de comportamiento en la navegación por Internet de los usuarios de la Universidad de las Ciencias Informáticas (UCI) a partir de los registros de navegación (*logs*), generados por el servidor *proxy*. Esta herramienta es de utilidad para la Dirección de Redes y Seguridad Informática (DRSI), pues le brinda información necesaria para la toma de decisiones. Desarrolla un proceso de Descubrimiento de Conocimientos en Base de Datos (KDD[1]) en el cual se aplican la tarea de Agrupamiento, con el fin de encontrar grupos de usuarios en el uso de la navegación por Internet y la tarea Reglas de Asociación para encontrar relaciones entre los atributos de los usuarios. También integra dos tecnologías con características diferentes para la ejecución del algoritmo de Agrupamiento y el de Reglas de Asociación (*JAVA* y *Python*), con un diseño extensible a otras bibliotecas de algoritmos de varias tecnologías. Se realiza un procesamiento distribuido de los registros de navegación reduciendo considerablemente el tiempo de la preparación de los datos. El diseño en el almacenamiento de la información procesada agiliza las consultas de datos. Además logra integrar a los datos de la navegación la información personal de los usuarios.

PALABRAS CLAVE

Integración de tecnologías; Minería de datos; Programación distribuida; Toma de decisiones

[1] del inglés: *Knowledge Discovery in Databases*

TABLA DE CONTENIDOS

INTRODUCCIÓN

El desarrollo de todas las ramas de la sociedad moderna se ha acelerado considerablemente con la llegada de las computadoras y la informatización. La inmensa mayoría de los sistemas informáticos, además de la información que gestionan continuamente, almacenan en forma de registros, *logs* o bitácoras, una traza de los sucesos que ocurren en ellos. La información que contienen estos registros representa parte de la historia de una organización, empresa u otra entidad, por lo que sería interesante procesarla.

Con el auge de las herramientas informáticas, el acceso a los servicios telemáticos (1) tales como: correo electrónico, servidor HTTP[2], DNS[3], FTP[4], etc., se vio inundado de clientes, incrementando considerablemente la producción de registros de sucesos. Dada la importancia que poseen los datos recopilados, se comenzó su estudio para poder describir a modo de números o gráficas, el comportamiento de los usuarios con acceso a estos servicios. No obstante, si bien el incremento de los datos generados eleva la utilidad de la información disponible, sitúa en una posición limitada la condición humana para evaluar esas grandes cantidades de registros, quedando en un lugar imperceptible diferentes variables y observaciones.

En este sentido, se utiliza la Minería de Datos como técnica para la obtención de patrones ocultos en los datos. La Minería de Datos es una tecnología novedosa que integra diferentes técnicas de análisis de datos y extracción de modelos. Está basada en varias disciplinas que se distinguen por su funcionamiento pero tienen un propósito común, permitiendo que se complementen entre sí. La posibilidad de extraer patrones, describir tendencias y regularidades, predecir comportamientos y en general, aprovechar la utilidad de la información digitalizada que nos rodea hoy en día, comúnmente heterogénea y en grandes cantidades, permite a individuos y organizaciones analizar, entender y modelar de una manera más eficiente y precisa el contexto en el que deben actuar y tomar decisiones (2).

Una de las grandes fuentes de *logs,* es el servidor *proxy.* Un *proxy* no es más que un intermediario entre una red privada de computadoras e *Internet* (3), que contribuye a mitigar riesgos de ataques a la red, debido a que la conexión a *Internet* se realiza por un solo punto. Además, permite llevar un control centralizado de las peticiones que realizan los usuarios y posibilita almacenar la información accedida

[2] del inglés: Hypertext Transfer Protocol
[3] del inglés: Domain Name System
[4] del inglés: File Transfer Protocol

recientemente, la cual puede ser utilizada por otros usuarios que la necesiten, minimizando el tiempo de respuesta a las solicitudes.

Uno de los ejemplos de uso del servidor *proxy* lo representa la UCI, donde *Internet* constituye una de las principales fuentes de consulta de conocimientos para la docencia, la producción, la preparación profesional y la investigación por parte de los estudiantes y profesores. El alto número de usuarios que usan Internet en la UCI unido al insuficiente ancho de banda con el que se cuenta, hacen necesario pensar en un sistema para gestionar las solicitudes de acceso a *Internet*. La DRSI de la UCI en su misión de tener un mejor control del uso del canal de navegación, ha creado un sistema de asignación de cuotas de navegación. Este sistema, en su primera versión sólo se limitaba a descontar de la cuota asignada a los usuarios, el monto en cuanto a capacidad de almacenamiento que ocupan los recursos accedidos; en su segunda versión, después de un estudio realizado y aplicando la experiencia de los miembros de la DRSI, el sistema tiene en cuenta una clasificación realizada a los dominios y sitios de *Internet* y el horario del día en que se realizan las peticiones. Con esta última versión, el control del acceso a *Internet* se hace un poco más flexible. Aun así, el monitoreo de toda esa información se realiza utilizando métodos estadísticos sin explotar las diferentes variables contenidas en los datos, decisivas para extraer conocimiento (4). Es aquí donde la Minería de Datos contribuye a desenmascarar el conocimiento implícito en los datos, mediante su análisis con algoritmos encaminados a este propósito.

En el curso 2009-2010, como parte de una plataforma para la gestión de servicios telemáticos, perteneciente al Centro de Telemática de la Facultad 2 (TLM), se desarrolló una **Herramienta informática de Minería de Uso de la Web aplicada a los registros de navegación por Internet** (HERMINWEB), con el propósito de analizar los registros del servidor *proxy* para extraer patrones que describan el comportamiento de los usuarios en el uso de sus cuotas de navegación por *Internet*, ayudando de esta forma a la DRSI de la UCI en la toma de decisiones.

Se realizó un estudio diagnóstico de esta herramienta para conocer las características de la misma, siendo necesario probarla, entrevistar al equipo de desarrollo, revisar el código fuente. A partir de este estudio se constató que HERMINWEB brinda la posibilidad de obtener patrones en términos de la tarea descriptiva Agrupamiento o Segmentación[5], la cual consiste en obtener grupos “naturales” a partir de los datos, de forma tal que los objetos de un mismo grupo son muy similares entre sí y, al mismo tiempo, son muy diferentes a los objetos de otro grupo (5). Como resultado de aplicar esta tarea de Minería de Datos

[5] del inglés: *Clustering*

en el análisis de los *logs* generados por el servidor *proxy* de la UCI, es posible agrupar a los usuarios de acuerdo al comportamiento que reflejan en el uso de las cuotas de navegación por *Internet*. Esta fragmentación es de notable interés para la DRSI, dadas las importantes ventajas que aporta al permitir el tratamiento de una gran cantidad de usuarios de forma personalizada, en el más idóneo punto de equilibrio entre el tratamiento individualizado y aquel totalmente masificado, lo que se puede traducir en una mejora considerable en el sistema de cuotas de navegación.

No obstante, resulta interesante también para la DRSI conocer otro tipo de información como por ejemplo, que si los estudiantes son masculinos de 4to año de la facultad 6 y navegan desde la beca en el horario de la tarde, entonces visitan el sitio de redes sociales Facebook, con una confianza del 80 por ciento y soporte del 75 por ciento. Esta información se puede conocer aplicando la tarea descriptiva Reglas de Asociación en el análisis de los *logs* generados por el servidor *proxy*, lo que resulta imposible en el estado actual de HERMINWEB, ya que la misma no contiene las funcionalidades necesarias para desarrollar esta tarea de Minería de Datos.

La tarea Reglas de Asociación tiene como objetivo identificar relaciones no explícitas entre atributos categóricos o nominales y se emplea frecuentemente para reconocer cómo la ocurrencia de un suceso o acción puede inducir o generar la aparición de otros. Es utilizada cuando el objetivo es realizar análisis exploratorios, buscando relaciones dentro del conjunto de datos, las cuales pueden ser de muchas formas, aunque la formulación más común es del estilo "si el atributo 'X' toma el valor 'a' entonces el atributo 'Y' toma el valor 'b'". Estas relaciones no son causa-efecto, es decir, puede no existir una causa para que los datos estén asociados (2).

La **situación problemática** actual se centra en la incapacidad de HERMINWEB para extraer patrones en términos de la tarea descriptiva Reglas de Asociación, que enriquezcan la descripción del comportamiento de los usuarios en el uso de las cuotas de navegación por Internet, apoyando de forma más precisa a la DRSI de la UCI en la toma de decisiones.

Como **propuesta de solución** al problema, se plantea el desarrollo de un Proceso de Extracción de Conocimiento en Bases de Datos, que consta de varias fases. Una de estas fases es la Minería de Datos, en la cual se utilizará Reglas de Asociación como tarea descriptiva para la extracción de patrones.

Tomando en cuenta lo anteriormente expuesto se plantea como **problema científico**: ¿Cómo extraer patrones de comportamiento que se manifiestan en el uso de las cuotas de navegación por *Internet* de los

usuarios de la UCI en términos de la tarea descriptiva Reglas de Asociación? El **objeto de estudio** son los patrones de comportamiento que se manifiestan en el uso de las cuotas de navegación por *Internet*. Se toma como **campo de acción** la extensión de las funcionalidades de HERMINWEB en términos de la tarea descriptiva Reglas de Asociación en la Minería de Uso de la Web.

La investigación tiene como **objetivo general** implementar nuevas funcionalidades a HERMINWEB, que permitan realizar la tarea descriptiva Reglas de Asociación en el estudio del uso de las cuotas de navegación en *Internet* por parte de los usuarios de la UCI.

Los **objetivos específicos** que se persiguen son:

- Investigar las tendencias actuales sobre Minería de Datos, Minería Web, Minería de uso Web, la tarea descriptiva Reglas de Asociación y las soluciones informáticas más usadas en la Minería de uso Web.
- Identificar limitaciones de la primera iteración de HERMINWEB en la implementación de los procesos de Preparación, Minería, Evaluación y Difusión de Resultados.
- Elaborar el Modelo de Sistema para la nueva iteración de la herramienta HERMINWEB.
- Elaborar el Modelo de Diseño para la nueva iteración de la herramienta HERMINWEB.
- Implementar las nuevas funcionalidades para la realización de la tarea Reglas de Asociación y el mejoramiento de los procesos de Preparación, Minería, Evaluación y Difusión de Resultados.
- Probar las nuevas funcionalidades añadidas a HERMINWEB para verificar el correcto funcionamiento de las mismas.

Como **idea a defender** se plantea lo siguiente: con la implementación de las nuevas funcionalidades a HERMINWEB se obtendrán patrones en términos de Reglas de Asociación que describan el comportamiento de los usuarios en el uso de las cuotas de navegación por *Internet* y sirvan de apoyo en la toma de decisiones desde el punto de vista administrativo a la DRSI de la UCI.

Para cumplir el objetivo de la investigación se trazaron las siguientes **tareas de investigación**:

- Investigación de las tendencias actuales de Minería de Datos, Minería Web, Minería de uso Web, la tarea descriptiva Reglas de Asociación y de las soluciones informáticas libres y privativas más usadas en la Minería de uso Web.
- Identificación de las limitaciones de la primera iteración de HERMINWEB en la implementación de los procesos de Preparación, Minería, Evaluación y Difusión de Resultados.

- Desarrollo del Modelo de Sistema para la nueva iteración de la herramienta HERMINWEB.
- Desarrollo del Modelo de Diseño para la nueva iteración de la herramienta HERMINWEB.
- Implementación de las nuevas funcionalidades para la realización de la tarea Reglas de Asociación y el mejoramiento de los procesos de Preparación, Minería, Evaluación y Difusión de Resultados.
- Realización de pruebas a las nuevas funcionalidades añadidas a HERMINWEB para verificar el correcto funcionamiento de las mismas.

Se utilizaron **métodos científicos** (6) que permitieron el correcto desarrollo de la investigación a partir de una caracterización del objeto de estudio y el estudio de las tendencias actuales sobre el contexto en el que se enmarca la problemática.

Métodos teóricos:

Análisis-síntesis (6): Este método fue utilizado en todo el proceso investigativo, ya que permitió descomponer todo el problema en varias partes que posibilitaron una mejor comprensión del mismo y luego buscar la relación entre esas partes.

Inductivo-deductivo (6): Para obtener patrones de comportamiento de los usuarios es necesario estudiar una muestra representativa de ellos, lo que posibilita realizar una generalización del uso que estos le dan a su cuota de navegación por *Internet*.

Histórico-lógico (6): Fue de gran importancia para elaborar la fundamentación teórica de la investigación, ya que permitió estudiar lo más relevante en el plano teórico acerca de la Minería de Datos, las metodologías, herramientas y tecnologías que se utilizarán para el desarrollo del trabajo.

Modelación (6): Fue el método más importante para la investigación debido a que brindó la posibilidad de representar las propiedades y funcionalidades del sistema desarrollado.

Métodos empíricos:

Observación (6): Se puso de manifiesto durante todo el transcurso de la investigación. Los autores del presente trabajo pertenecen a la UCI y pueden observar, de manera real y oportuna, cómo un determinado grupo de usuarios utilizan su cuenta de navegación, pudiendo de esta forma ubicarse en todo momento como observadores de la realidad que los circunda y captar las mejores observaciones.

El contenido del presente documento está estructurado en cuatro capítulos: **Fundamentación Teórica**: incluye conceptos, definiciones y valoraciones que conforman la base teórica del tema tratado;

Características del sistema: aborda la aplicación de los procesos de negocio de un proceso KDD a la problemática planteada para conformar una propuesta de solución; **Diseño del Sistema**: presenta de manera general la estructura del sistema a partir de las funcionalidades previstas; e **Implementación y Pruebas**: documenta el desarrollo de la arquitectura como un todo, basándose en los resultados obtenidos en el **Diseño del Sistema**; además, refleja la estrategia de pruebas desarrollada y .los resultados obtenidos con la realización de las mismas.

Capítulo 1

FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se abordan varios aspectos teóricos, conceptos y definiciones relacionados con las tendencias actuales en el contexto en el cual se desarrolla este trabajo, con el propósito de tener una base teórica para el desarrollo de la herramienta. Se argumenta sobre el concepto de Minería de Datos, se describe el ciclo completo de un proceso de extracción de conocimientos en base de datos, se establece la relación entre un proyecto de Minería de Uso de la Web y el presente trabajo y se puntualiza sobre la tarea y algoritmo empleados para obtener modelos o patrones a partir de los datos. Además incluye un estudio sobre las metodologías utilizadas tanto para el desarrollo de software como para guiar el proceso KDD, así como las tecnologías, herramientas, lenguajes y notaciones empleadas para tales propósitos.

1.2 Minería de Datos

Según lo planteado por J. Han (7), una definición simple para el término Minería de Datos, sería que se refiere a: extraer conocimiento a partir de grandes volúmenes de datos. Tal como se aprecia en lo expresado por Hernández Orallo (2), con la Minería de Datos el resultado histórico de los sistemas de información deja de ser "producto", pasando a ser la "materia prima" necesaria para extraer el verdadero "producto elaborado": el conocimiento, para lo cual es imprescindible que sea realizada como un proceso automático o semiautomático (asistido).

Autores como J.Han (7) y Hernández Orallo (2), han aportado varias definiciones para la Minería de Datos, pero en su esencia, todas coinciden en que su tarea fundamental es descubrir conocimiento (reglas, patrones) a partir de grandes volúmenes de datos, apoyados en técnicas o herramientas (automáticas o asistidas), de tal manera que su uso ayude a tomar decisiones más seguras que reporten algún tipo de beneficio a las organizaciones (8).

Para tomar decisiones sobre el sistema de cuotas de navegación por *Internet* de la UCI a partir del comportamiento de los usuarios con respecto al uso de su cuota personal, es necesario analizar un gran

cúmulo de información en distintos formatos proveniente de diversas fuentes, tales como: bases de datos, servicios web, clasificaciones de sitios web y registros de sucesos del servidor *proxy*, lo que sitúa en una posición limitada la condición humana para evaluar esas grandes cantidades de datos. Por tanto, se definió el problema como un proyecto de Minería de Datos, que permitiera encontrar patrones a partir de los datos recopilados.

1.3 La Minería de Datos y el proceso de descubrimiento de conocimiento

En el desarrollo de la Minería de Datos, existe un concepto llamado Extracción de Conocimiento en Bases de Datos, definido como el proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y en última instancia, comprensibles a partir de los datos (2).

KDD define un conjunto de fases o etapas para guiar el desarrollo del proceso. De forma general contiene cinco fases como muestra la Figura 1:

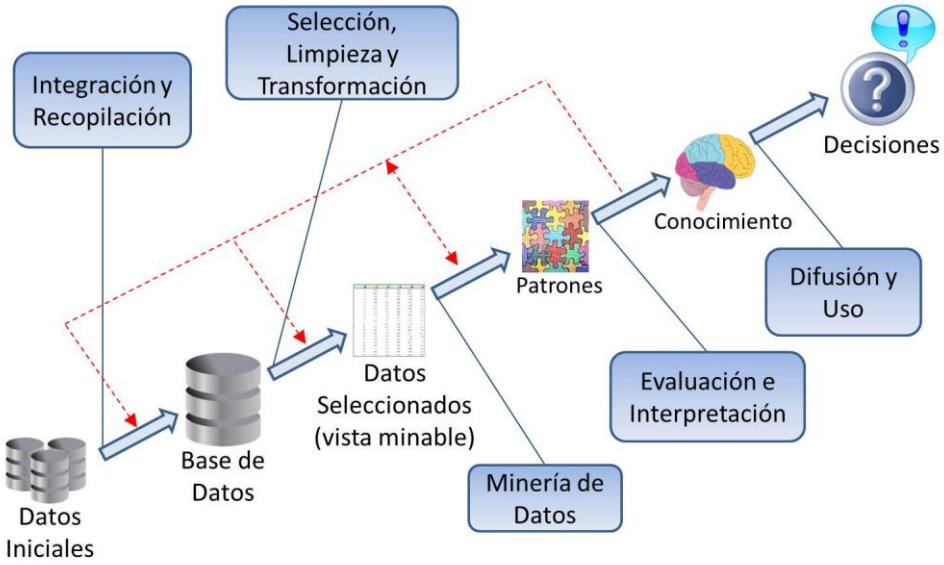


Figura 1: Fases de KDD.

La Figura 1 revela que la salida de una fase constituye la entrada de la siguiente, no obstante, esta salida puede hacer que se vuelva a pasos anteriores. Además, el usuario o experto en el dominio del problema en cuestión debe interpretar, validar, valorar y evaluar el conocimiento obtenido a partir de los patrones extraídos, para luego difundirlo y utilizarlo de forma que se puedan tomar decisiones, lo que también

puede hacer retroceder el proceso a un estado anterior. Este carácter iterativo e interactivo del proceso KDD es muy importante para extraer conocimiento de alta calidad.

Debe señalarse que las dos primeras fases suelen englobarse bajo el nombre de **preparación de datos***;* y, por otro lado, en muchas ocasiones se incluye previo a las fases descritas, una etapa de **entendimiento del dominio** para el análisis de las necesidades de la organización, o sea, para definir y priorizar los objetivos del negocio (2). Además, aproximadamente el 80 por ciento del esfuerzo total para realizar un proceso de KDD, se emplea durante las etapas de entendimiento del dominio y la preparación de los datos (9).

Si bien el término KDD se emplea para describir el proceso completo, y dentro de este existe una fase denominada Minería de Datos, que engloba la aplicación de técnicas y herramientas, ambas terminologías son utilizadas indistintamente con frecuencia para referirse a todo el proceso (8).

Esta investigación se encuentra en concordancia con las cinco fases propuestas por KDD, tal como se podrá apreciar más adelante en el acápite referente a la metodología para enfrentar procesos de KDD: *CRoss-Industry Standard Process for Data Mining* (CRISP-DM) (10).

1.4 Minería Web

Minería Web[6] se refiere al proceso global de descubrir información o conocimiento potencialmente útil y previamente desconocido a partir de datos de la web. Descubrir los patrones interesantes en la estructura, contenido y la utilización de los sitios web constituye el objetivo fundamental de la minería web (11).
El proceso de minería web se divide en cuatro fases fundamentales (2):

- Descubrimiento de las fuentes (recursos): localizar los documentos y servicios en la web.
- Selección y pre-procesamiento de la información: extraer automáticamente información específica desde las fuentes web descubiertas.
- Generalización: descubrir patrones generales desde los sitios web individuales así como desde múltiples sitios.
- Análisis: validación e interpretación de los patrones descubiertos.

[6] del inglés: *Web Mining*

La minería web suele clasificarse en tres dominios de extracción de conocimiento de acuerdo con la naturaleza de los datos, esto se conoce como: “Taxonomía de la minería de web”. Los dominios son los siguientes:

- Minería de contenido web[7].
- Minería de estructura web[8].
- Minería de uso web[9].

1.4.1 Minería de contenido web

Muy heterogéneos son los documentos que podemos encontrar en la web, ya sea de hipertexto, documentos de texto, imágenes, videos, entre otros. Esta diversidad dificulta la organización de los mismos. La minería de contenido web busca información relevante sobre los contenidos de la web que ayude en su clasificación, aumentando la organización de estos y mejorando así el acceso y recuperación de ellos (11).

1.4.2 Minería de estructura web

Se basa en la búsqueda de modelos subyacentes de la estructura de enlaces de la web y analiza la topología de los hipervínculos. Este modelo puede usarse para categorizar páginas web y es útil para generar información como la similitud y relación entre diferentes sitios web (11).

1.4.3 Minería de uso web

Es el proceso de analizar la información de acceso a la web (disponible en los servidores web fundamentalmente) para la extracción de modelos interesantes de uso de la web por parte de los usuarios. A diferencia de la minería de contenido y de estructura, que usan datos reales sobre la web, la minería de uso mina datos secundarios derivados de la interacción de los usuarios con la web. Estos datos incluyen los archivos de *logs* de acceso al servidor, *logs* del navegador, *logs* de los servidores *proxy* y en general cualquier otro dato de la interacción (2).

La minería de uso web posee dos objetivos fundamentales: el primero es encontrar patrones de comportamiento generales de uso de un sitio web de manera que se pueda reestructurar el mismo para facilitar su uso y mejorar el acceso a la información por parte de los usuarios, y el segundo, obtener

[7] del inglés: *Web content mining*
[8] del inglés: *Web structure mining*
[9] del inglés: *Web usage mining*

perfiles de los distintos tipos de usuarios a través de su comportamiento y navegación para poder atender sus exigencias de forma personalizada. En la presente investigación se utilizó la minería de uso web para obtener perfiles de usuarios basados en patrones de comportamiento que permitan mejorar la asignación de cuotas de navegación de acuerdo al acceso a los sitios web por parte de estos, a diferencia de cómo regularmente se trata este dominio de conocimiento, pues solo se buscan estos perfiles en aras de mejorar el diseño de los repositorios de recursos (13).

Para analizar el proceso de minería de uso web es necesario distinguir sus tres fases (14):

- Pre-procesamiento de datos: antes de desarrollar el algoritmo de minería de datos, se lleva a cabo una preparación de los datos en bruto a otros que tengan una abstracción para futuros procesos. Los datos pueden ser recolectados en el lado del servidor, lado del cliente, servidores *proxy* u obtenidos de una base de datos.
- Descubrimiento de patrones: son el componente fundamental en la minería de uso web, entre los algoritmos y las técnicas para el descubrimiento de patrones se encuentran: análisis estadístico, reglas de asociación, agrupamiento, clasificación, entre otras.
- Análisis de patrones: es el paso final de la minería de uso web y tiene como objetivo extraer los patrones interesantes de la salida del paso de descubrimiento de patrones. La salida de los algoritmos de minería web regularmente no se ajustan de forma directa a la concepción humana, por lo que se hace necesario transformarlo a un formato que sea fácil de comprender.

1.5 Modelos, Tareas y Algoritmos de Minería de Datos

El objetivo fundamental de la minería de datos es analizar los datos para la extracción de conocimiento, el cual se puede representar a través de patrones o reglas inferidos a partir de este análisis o en forma más compacta a través de resúmenes. Estos resúmenes o relaciones constituyen el modelo de los datos analizados (8).

1.5.1 Modelos de Minería de Datos

Los modelos constituyen la forma de representar el conocimiento obtenido a partir de los datos analizados, y su construcción está determinada por la tarea de minería de datos escogida y el algoritmo seleccionado para realizarlo (2).

Dependiendo de las características de cada modelo de minería estos pueden clasificarse en **predictivos** o **descriptivos**.

Los modelos predictivos pretenden estimar valores futuros o desconocidos de variables de interés, que denominamos variables objetivo o dependientes, usando otras variables o campos, a las que nos referiremos como variables independientes o predictivas. Por ejemplo, un modelo predictivo sería aquel que permite estimar la demanda de un nuevo producto en función del gasto en publicidad (2).

Los modelos descriptivos, en cambio, identifican patrones que explican o resumen los datos, es decir, sirven para explorar las propiedades de los datos examinados, no para predecir nuevos datos. Por ejemplo, una agencia de viaje desea identificar grupos de personas con los mismos gustos, con el objeto de organizar diferentes ofertas para cada grupo y poder así remitirles esta información; para ello analiza los viajes que han realizado sus clientes e infiere un modelo descriptivo que caracteriza estos grupos (2).

1.5.2 Tareas de Minería de Datos

El proceso de minería requiere de establecer previamente los objetivos para el análisis de los datos que se disponen, de ahí en virtud de ellos existan las llamadas tareas de minería, entre las que se encuentran:

- Agrupamiento.
- Reglas de Asociación.

En dependencia del tipo de búsqueda empleado para obtener conocimiento, las tareas mencionadas se pueden clasificar en directas o indirectas. Con las tareas directas se conoce claramente lo que se busca. El agrupamiento y la asociación son indirectas, y se emplean para descubrir patrones que describan los datos sin un objetivo concreto definido (8).

Otras clasificaciones de las tareas a tono con el modelo que pueden generar son: predictivas y descriptivas. Entre estas últimas encontramos el agrupamiento y las reglas de asociación

1.5.2.1 Agrupamiento

El agrupamiento es la tarea descriptiva por excelencia y consiste en obtener grupos "naturales" a partir de los datos. Hablamos de grupos y no de clases, porque, a diferencia de la clasificación, en lugar de analizar datos etiquetados con una clase, los analiza para generar esta etiqueta (2).

1.5.2.2 Reglas de Asociación

Reglas de Asociación es una tarea descriptiva, la cual es el eje fundamental de esta investigación. Tiene como objetivo encontrar relaciones no explícitas entre atributos nominales y es empleada para reconocer

cómo la ocurrencia de determinados sucesos puede provocar la aparición de otros. Las relaciones pueden ser de muchas formas, aunque la formulación más común es del estilo "si el atributo 'X' toma el valor 'a' entonces el atributo 'Y' toma el valor 'b'". Estas relaciones no son causa-efecto, es decir, puede no existir una causa para que los datos estén asociados. Este tipo de tarea se utiliza frecuentemente en el análisis de la cesta de la compra, para identificar productos que son frecuentemente comprados juntos, información esta que puede usarse para ajustar los inventarios, para la organización física del almacén o en campañas publicitarias. Las reglas se evalúan usando dos parámetros: precisión (confianza) y soporte (cobertura) (8).

Las reglas de asociación juegan un papel muy importante en el contexto de la nueva visión de la web con el auge de las técnicas de comercio que se manejan de forma electrónica, que permiten el desarrollo de estrategias de marketing (11).

Normalmente esta técnica está relacionada con el uso de Bases de Datos transaccionales para descubrir todas las asociaciones entre los pares atributo-valor donde la presencia de un conjunto de ellos en una transacción implica la presencia de otros. Generalmente está asociada con en el número de ocurrencias de estos pares dentro de los *logs* de transacciones, pudiendo así, por ejemplo, identificar la cantidad de usuarios que acceden a determinadas páginas y mejorar considerablemente la organización del sitio web (11).

1.5.3 Algoritmos de Minería de Datos

La forma de implementar cada una de las tareas de minería la constituyen los algoritmos. De la correcta elección de ellos dependerá la representación del conocimiento obtenido del análisis realizado a los datos ya sea en forma de regla, patrones o resumen. Entre los principales algoritmos de minería se encuentran:

- K-medias[10] (Agrupamiento).
- Apriori (Asociación).

1.5.3.1 K-medias

Este algoritmo es uno de los más utilizados para tareas de agrupamiento. Para su implementación primero se determina la cantidad de clústeres o grupos que se quiere obtener y se seleccionan los ‘n’ elementos o centroides de cada clúster aleatoriamente. Luego cada instancia se asigna al grupo más cercano teniendo

[10] del inglés: KMeans

en cuenta una medida de similitud dada. A continuación para cada clúster se obtienen los centroides de sus instancias, repitiéndose este procedimiento hasta que los centroides de los clúster se hayan estabilizado. K-medias es un algoritmo ávido cuyo objetivo es minimizar el error cuadrado entre la media del grupo y sus elementos, en otras palabras, encontrar grupos donde en cada uno de ellos estén elementos semejantes y que los elementos de grupos diferentes no lo sean (4).

K-medias fue el algoritmo utilizado en el desarrollo de la primera iteración de HERMINWEB.

1.5.3.2 FP-Growth

En el lado derecho de una regla de asociación puede aparecer cualquier par o pares atributo-valor, los que se conocen como ítem, mientras que a un conjunto de ellos se les llama ítem-set. Para la evaluación de la regla se utilizan dos medidas: la cobertura o soporte, que indica el número de casos o instancias que cubre la regla, y la confianza, que indica el número de casos que predice correctamente la regla y que está dada por el cociente entre el número de casos que cumplen la regla y el número de instancias que se le aplican porque cumplen las premisas. Las reglas interesantes son aquellas que tengan un valor de soporte elevado.

La generación de reglas de asociación comprende dos pasos fundamentales:

- Encontrar los conjuntos de ítems frecuentes en el conjunto de datos iniciales que igualen o superen el nivel de soporte propuesto.
- Generar las reglas de asociación que igualen o superen los niveles de confianza propuestos a partir de los ítems frecuentes encontrados.

Para la selección del algoritmo a aplicar se realizaron varias comparaciones entre una serie de ellos dentro de los que se encuentran: Apriori, Partition, Eclat y FP-Growth. Para la selección se tomaron como base los siguientes criterios:

- Cantidad de consultas al conjunto de datos iniciales.
- Rendimiento.
- Implementación en RapidMiner.

En el caso del algoritmo Apriori realiza un número elevado de consultas al conjunto inicial de datos lo que representa una desventaja para la solución por la ralentización del proceso. Por otro lado los algoritmos Partition y Eclat presentan un costo computacional elevado y no son implementados por RapidMiner. Por

último tenemos el FP-Growth que es implementado por RapidMiner, realiza menos consultas al conjunto de datos iniciales que el resto de los algoritmos analizados y para generar los ítems frecuentes utiliza la estructura FP-Tree, que para su creación no necesita de un costo computacional elevado.

Luego de realizadas las comparaciones se determinó utilizar el algoritmo FP-Growth, pues según las valoraciones resultantes fue el que más se ajustó a las necesidades de la solución.

FP-Growth en un primer paso borra todos los ítems del conjunto de datos iniciales que no son frecuentes individualmente o no sobrepasan en el mínimo soporte. Luego de este paso se genera el árbol FP-Tree. Un árbol FP-Tree es básicamente una estructura formada por los prefijos para las transacciones, cada rama del árbol representa el grupo de transacciones que comparten el mismo prefijo y cada nodo corresponde a un ítem. Todos los nodos que referencian al mismo ítem son referenciados juntos en una lista, de modo que todas las transacciones que contienen un ítem específico pueden encontrarse fácilmente y contarse al atravesar la lista. Esta lista puede ser acezada a través de la cabeza, lo cual también expone el número total de ocurrencias del ítem en la base de datos (7).

1.6 Herramientas para la Minería del Uso de la Web

Con el auge de *Internet* y con ello del comercio y del negocio electrónico se han desarrollado varias herramientas para la búsqueda de patrones de comportamiento en la navegación de los usuarios de disímiles sitios web, entre las que se encuentran:

Privativas:

- *Lyris HQ* (antes *ClickTracks*) (15).
- *Amadea Web Mining* (16).
- *123LogAnalyzer* (17).
- *WebTrends* (18).

Libres:

- *AlterWind* (19).
- *Analog* (20).
- *Htminer* (21).

- *Visitator* (22).

De forma general todas estas herramientas se centran en la búsqueda de patrones de comportamiento de los usuarios en los sitios web para la personalización de los mismos y no para la mejora de la asignación de las cuotas de navegación de los usuarios.

1.7 Metodologías

Metodología es una palabra compuesta por tres vocablos griegos: *metà (“más allá”), odòs (“camino”) y logos (“estudio”).* Se define como metodología al conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal (23).

1.7.1 Metodologías para enfrentar un proceso de KDD

Como se vio anteriormente, en un proyecto de KDD, la mayor parte del esfuerzo se produce en la comprensión del negocio y la preparación de los datos, lo que unido al hecho de que no se puede arribar a conclusiones por adelantado, normalmente provoca retrasos y desviaciones en la planificación inicial. De aquí surge la necesidad, en muchos casos, de utilizar alguna metodología que guíe la ejecución del proyecto y facilite la planificación, dirección y seguimiento del mismo.

Entre las más conocidas y empleadas, pueden mencionarse: CRISP-DM, SEMMA[11] (24), la metodología de las 5 A’s[12] (25), CRITIKAL[13] (26).

Se decidió utilizar la metodología CRISP-DM, debido a que ha sido diseñada de forma genérica, sin importar las herramientas que se utilicen para el desarrollo del proyecto; su distribución es libre, encontrándose en constante desarrollo por la comunidad internacional; concibe el proyecto de KDD de forma global y estrechamente relacionado al negocio en cuestión; propone un preciso y sólido repertorio de tareas de propósitos generales, por lo que goza de una importante popularidad, siendo, por tanto, frecuentemente empleada; además fue utilizada en el desarrollo de HERMINWEB, arrojando buenos resultados.

[11] del inglés: Sample, Explore, Modify, Model, Assess. En español: Muestreo, Exploración, Manipulación, Modelado, Valoración.
[12] del inglés: Assess, Access, Analyze, Act, Automate. En español: Evaluar, Acceder, Analizar, Actuar, Automatizar.
[13] del inglés: Client-Server Rule Induction Technology for Industrial Knowledge Acquisition from Large Databases.

1.7.1.1 CRISP-DM

CRISP-DM fue propuesta inicialmente por un consorcio de empresas encabezadas por *Statistical Product and Service Solutions (SPSS)* en 1996, y luego liberada para su empleo y desarrollo por parte de la comunidad internacional. Toma en cuenta seis etapas o fases fundamentales como guía durante el ciclo de vida de un proyecto que implemente un proceso de KDD, tal como se muestra en la Figura 2 (4).

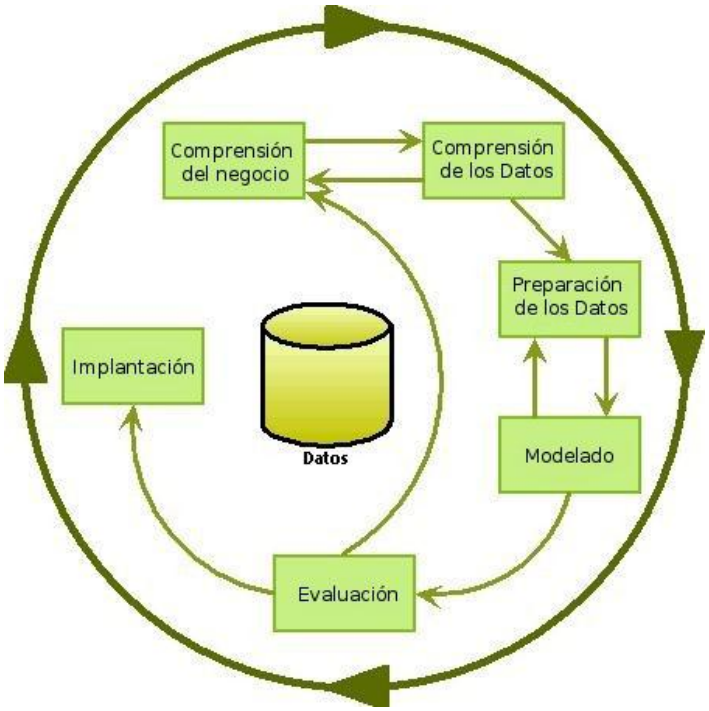


Figura 2: Fases de CRISP-DM (4)

Según se explica en (10), las fases son detalladas brevemente a continuación:

1. **Comprensión del Negocio**: Se determinan los objetivos y requerimientos del negocio, se evalúa la situación para definir el problema como un proyecto de Minería de Datos y se elabora el plan del proyecto.
2. **Comprensión de los Datos**: Se recopilan los datos iniciales, se describen, se verifica la calidad de los mismos y se exploran para detectar los primeros puntos de vista o subconjuntos de carácter interesante que permitan crear hipótesis sobre la información oculta.

3. **Preparación de los Datos**: Se seleccionan los datos, se construyen, integran y estructuran para ser minados. Estas tareas incluyen tablas, registros, selección de atributos, así como la transformación y limpieza de los datos para que sean utilizados en las herramientas de modelado.
4. **Modelado**: Se seleccionan y aplican las técnicas de modelado, se genera el diseño del experimento y se construyen y evalúan los modelos.
5. **Evaluación**: Se evalúa a fondo el modelo revisando el proceso para construirlo, se asegura que el modelo cumple apropiadamente con los objetivos del negocio, se determinan temas relevantes en el negocio que no hayan sido suficientemente considerados y se valoran los resultados.
6. **Despliegue**: Se traza la estrategia de empleo de los resultados, se planifica el mantenimiento del proceso, se organiza y presenta el conocimiento obtenido como resultado de manera que el cliente pueda usarlo, y se documentan las experiencias.

Como se puede apreciar en la Figura 8, CRISP-DM propone un orden lógico a sus fases, aunque permite retrocesos en algunas de ellas, lo que contrasta con el hecho de que durante el desarrollo de un proyecto, en numerosas ocasiones es necesario volver atrás para reanalizar los resultados obtenidos. A la vez, el proceso se torna cíclico, debido a que no se termina con el despliegue de la solución, al contrario, el conocimiento obtenido puede provocar nuevas preguntas enfocadas en el negocio, donde los procesos de minería subsecuentes se beneficiarán de las experiencias anteriores.

Las seis fases propuestas por la metodología CRISP-DM tuvieron una marcada influencia en esta investigación. Durante la primera etapa, se realizaron entrevistas a los administradores de redes de la UCI y el Instituto Superior Politécnico José Antonio Echeverría (CUJAE) y se llevó a cabo un estudio diagnóstico sobre HERMINWEB con el apoyo del equipo que desarrolló la herramienta. Con toda la información recopilada se aplicó la técnica de tormenta de ideas, lo que permitió una clara comprensión del negocio. Posteriormente, mediante consultas a la base de datos de Akademos se recopilaron los datos iniciales de los estudiantes, y consultando los servicios web ASSETS y Registro Personal se obtuvieron los datos de los trabajadores internos y externos respectivamente. Los registros del servidor *proxy* y las *Blacklists* fueron proporcionados por la DRSI, siendo necesaria una nueva entrevista a los administradores de redes de la UCI para tener una mejor comprensión de los datos recogidos en estos archivos. Se realizó una exploración de estos datos y se comprobó la calidad de los mismos, lo que permitió tener una primera vista sobre cuáles de ellos podrían aportar información relevante para la investigación.

Concluidas las dos primeras fases, se seleccionaron los datos con los cuales se desarrolló la investigación y se trabajó posteriormente en la transformación y mezcla de la información contenida en los *logs* del proxy con los datos personales de los usuarios, siendo de especial importancia en todo este proceso la discretización de los datos para poder crear finalmente las sesiones de usuario, debido a que las reglas de asociación solo tienen en cuenta los atributos nominales.

Posteriormente se procedió a ejecutar el algoritmo *FP-Growth* para extraer los patrones a partir de los datos y poder evaluar los modelos inteligentes obtenidos mediante la utilización de gráficas, lo que permitió realizar un análisis empírico de los resultados.

Finalmente, en la fase de despliegue se presentará el conocimiento obtenido a la DRSI, de manera que sus directivos puedan utilizarlo para tomar decisiones. Además, se analizarán las mejoras que se puedan realizar a la herramienta y la posibilidad de incorporarle nuevas funcionalidades y tareas de Minería de Datos, para que formen parte del proceso de mantenimiento y crecimiento del proyecto.

1.7.2 Metodologías de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos informáticos. Habitualmente se utiliza el término “método” para referirse a técnicas, notaciones y guías asociadas, que son aplicables a actividades del proceso de desarrollo (27).

Un proceso define *quién* está haciendo *qué*, *cuándo* y *cómo* alcanzar un determinado objetivo. En la Ingeniería del Software, el objetivo es construir un producto de software o mejorar uno existente. Un proceso efectivo proporciona normas para el desarrollo eficiente de software de calidad (28).

No existe una metodología de software universal. Las características de cada proyecto exigen que el proceso sea configurable, por lo que actualmente existen un conjunto de metodologías de desarrollo de software con características específicas (27). Entre las más representativas se encuentran: *Proceso Unificado de Rational (RUP[14]) y Extreme Programming (XP)*, seleccionando en este caso RUP.

[14] del inglés: Rational Unified Process

1.7.2.1 Proceso Unificado de Rational

El Proceso Unificado de Rational es más que un simple proceso; es un marco de trabajo genérico que puede especificarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto (28).

Para llevar a cabo esta investigación, se implementó un gran número de funcionalidades con alta complejidad, por lo que fue necesario agrupar las funcionalidades similares de tal manera que el desarrollo de la aplicación fuera un proceso comprensible y quedara bien documentado. Al ser RUP **dirigido por casos de uso (CU)** (28), su utilización garantizó que se satisficiese esta necesidad, ya que los CU guían el proceso de desarrollo, pues los modelos que se obtienen como resultado de los diferentes flujos de trabajo, representan su realización.

Igualmente se necesitaba tener una visión del sistema completo, realizando primeramente los CU arquitectónicamente significativos, de forma que constituyeran los cimientos del sistema, que fueron necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP es **centrado en la arquitectura** (28), característica que se encuentra en total concordancia con esta necesidad.

La herramienta se desarrolló en varias iteraciones desde las mínimas funcionalidades hasta que se completó el ciclo de vida (28), reduciendo la posibilidad de que aparecieran riesgos que provocaran una compleja solución cuando se terminara el producto. Además, se especificaron las posibles funcionalidades futuras para otros ciclos de desarrollo. Para ello RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto, lo que lo convierte en un proceso **iterativo e incremental** (28).

Fue necesario también que la aplicación se realizara en un corto tiempo, eliminándose algunas actividades durante la ejecución de algunos flujos de trabajo, lo que está en correspondencia con que RUP utilice componentes, sea adaptable y lleve a cabo constantemente la ejecución de pruebas.

Se espera que el producto final contenga manuales de usuario y buena documentación que asegure una eficiente transferencia tecnológica. RUP es ideal para esto debido a que genera la documentación necesaria para validar los artefactos.

Se considera que la utilización de RUP fue de suma importancia para el desarrollo de la herramienta de forma eficiente y con alta productividad debido a que define qué se tiene que hacer, cómo y quién lo hace en cada momento del proceso de desarrollo.

1.8 Notación de Modelado de Procesos de Negocio (BPMN[15])

BPMN provee una notación que es comprendida tanto por usuarios, analistas, grupo de desarrollo que implementan las tecnologías que van a ser usadas en los procesos, como por las personas del negocio que se encargarán de administrar y controlar los procesos. Define un estándar que sirve de puente entre el diseño de los procesos del negocio y los procesos de implementación. Además brinda la capacidad de comprender los procesos internos de negocio mediante una notación gráfica, dándole a las organizaciones la posibilidad de comunicar esos procesos en un estándar (30).

En este proyecto, los procesos del negocio se comportan como un flujo de actividades o procesos al ser especificadas claramente en la metodología CRISP-DM las fases que son necesarias para la extracción de conocimiento en base de datos. Entre estas fases existen dependencias, las salidas de unas constituyen entradas de otras.

Durante el desarrollo de HERMINWEB también se utilizó la metodología CRISP-DM y se analizaron varias notaciones para modelar los procesos del negocio, teniéndose en cuenta UML (29), IDEF09[16] (31) y por último BPMN (30). UML e IDEF0 no describían con suficiente claridad estos procesos de negocio. Se seleccionó en ese momento la notación BPMN, la cual refleja de una forma clara y sencilla (mediante una notación gráfica) cómo se comportan los procesos del negocio de dicho trabajo, brindando mayor documentación y descripción de los mismos (4). Por tal motivo se decidió utilizar BPMN en el actual trabajo, manteniendo la documentación del negocio en concordancia con la que ya se contaba producto del desarrollo de HERMINWEB.

1.9 Lenguajes de Programación

1.10.1 Python

Es un lenguaje interpretado, orientado a objetos, de alto nivel que permite escribir código con una alta claridad y legibilidad, permitiendo así un rápido aprendizaje del mismo. La legibilidad permitirá en futuros

[15] del inglés: Business Process Modeling Notation
[16] Notación para modelar procesos o funciones del negocio, sus siglas significan en inglés: Integration Definition for Funcion Modeling

accesos al código una clara comprensión de lo antes implementado, haciendo más fácil el mantenimiento de las aplicaciones (32).

Su biblioteca estándar es muy amplia, conteniendo funcionalidades de gran ayuda desde el más bajo nivel hasta el más alto, facilitándole al programador la implementación de aplicaciones sin la necesidad de recurrir continuamente a bibliotecas externas. Además dispone de una extensa colección de bibliotecas libres disponibles en la mayoría de los repositorios de los sistemas GNU\Linux (32).

Las potencialidades brindadas por el lenguaje *Python* fueron de mucha ayuda en el proceso de desarrollo de HERMINWEB, al ser multiplataforma ayudó en la satisfactoria creación de un *parser* distribuido para procesar los registros del servidor *proxy*. Su amplia colección de funcionalidades de la biblioteca estándar posibilitó que se utilizaran la menor cantidad de bibliotecas externas y la existencia de una amplia colección de bibliotecas libres no requirió el uso de alguna biblioteca privativa (4).

Como aspecto importante, *Python* ofrece una librería que implementa una interfaz para el *framework* de comunicaciones *ICE*[17], el cual proporciona un conjunto de funcionalidades que permiten publicar y ejecutar funciones en un servidor remoto (33), (34), haciendo posible el procesamiento distribuido de los registros generados por el *proxy*.

Teniendo en cuenta que para el desarrollo de la presente investigación era necesario programar nuevas funcionalidades para HERMINWEB, así como modificar otras existentes y que *Python* fue decisivo en el desarrollo de esa herramienta, se decidió utilizar este lenguaje de programación. Además, esta investigación pertenece a un proyecto productivo de la UCI en el cual se está desarrollando una plataforma en dicho lenguaje, por lo que en un futuro la herramienta en desarrollo pudiera integrarse a la misma.

1.10 Herramienta CASE

Dentro de las herramientas claves en el desarrollo de aplicaciones informáticas se encuentran las herramientas de Ingeniería de Software Asistida por Ordenador *(CASE*[18]*)*, las cuales son las encargadas de ayudar en el ciclo de desarrollo, con el fin de aumentar la productividad y reduciendo el coste en términos de tiempo y dinero. En el ciclo de desarrollo pueden ayudar en el proceso de diseño del proyecto,

[17] del inglés: Internet Communication Engine
[18] del inglés: Computer Aided Software Engineering

en el cálculo de costes, pueden implementar una parte del código, compilación automática y documentación (4). En el presente trabajo se utilizó *Visual Paradigm* para UML.

1.10.1 Visual Paradigm para UML

Una de las herramientas CASE más usadas es la *suite* creada por *Visual Paradigm International (VPI)*. VPI es un proveedor de soluciones informáticas que incluye organizaciones para desarrollar aplicaciones de calidad, rápidas y baratas. Está compuesta por productos que facilitan a las organizaciones la visualización y diseño de diagramas. Sus soluciones se enfocan en eliminar la complejidad, aumentando así la productividad y disminuyendo el tiempo de desarrollo de las aplicaciones informáticas (36).

La herramienta CASE *Visual Paradigm* fue escogida para desarrollo de la aplicación debido a que utiliza UML como lenguaje de modelado, agiliza la creación de los diferentes diagramas definidos en la metodología RUP, se integra con *Eclipse, Python* y *PostgreSQL*, se puede utilizar en sistemas operativos *GNU/Linux*, soporta el modelamiento de procesos de negocio con BPMN y genera una excelente documentación en varios formatos (*jpg*, *html*, *pdf*, etc.).

1.11 Entorno de Desarrollo Integrado

Una de las herramientas que juegan un papel importante en el desarrollo de soluciones informáticas son los Entornos de Desarrollo Integrado (*IDE*[19]). Estos ofrecen facilidades al equipo de desarrollo cuando se implementan las aplicaciones debido a que permite corrección de errores comunes que se comenten a diario (4).

1.12.1 Eclipse

La arquitectura de *Eclipse* basada en *plugins* extiende las funcionalidades de la herramienta, por ejemplo, con la adición del soporte para varios lenguajes, como pueden ser: *Java, Ruby, PHP, Python* o *Prolog*. Fue creado en el 2003 por un grupo de desarrollo de software llamado *nexB* dedicados al análisis de código abierto (37).

Para desarrollar la investigación se decidió utilizar *Eclipse* debido a que actualmente se sitúa a la vanguardia de los entornos de desarrollo para *Python* existentes por su alto nivel de integración con este lenguaje, completamiento de código, resaltado de sintaxis, depuración de la ejecución, funcionamiento en

[19] del inglés: Integrated Development Enviroment

varias plataformas, así como diferentes funcionalidades que le facilitan el trabajo al programador como la refactorización del código. Además, posee una excelente integración con *SubVersion*.

1.12 Interfaz gráfica de usuario

1.13.1 PyQt: Qt para Python

Las bibliotecas gráficas *Qt*, constituyen actualmente una de las mejores en su tipo en el mundo del software libre. Realizadas en el lenguaje de programación *C++*, están hechas con el fin de realizar más con menos código. Su carácter multiplataforma hace fácil su despliegue. *PyQt* es la combinación de *Python* y *Qt*, estableciendo una interfaz transparente para acceder desde *Python* a dichas bibliotecas gráficas; así con la facilidad de *Python* sumada a la excelencia de *Qt*, se hace más agradable la programación visual. Cuenta con una amplia documentación proveniente de *Qt* y de *Python* a la vez (4).

Utilizando la biblioteca *PyQt* en el desarrollo del sistema se puede crear una interfaz visual sencilla y sin muchos contratiempos, ya que *PyQt* posee los componentes visuales necesarios para el desarrollo de la herramienta, así como una abundante documentación y ejemplos. Al funcionar en varias plataformas y tener una alta calidad hacen de la biblioteca uno de los productos con alto uso en la actualidad.

1.13.2 Qt Designer

Qt Designer es la herramienta por excelencia para la el diseño gráfico del *Framework Qt*. Se creó con el mismo fin para el que fue confeccionado *Qt*: agilizar el desarrollo y funcionar en una amplia variedad de plataformas. *Qt Designer* está disponible para las mismas plataformas sobre las que está sustentado el *framework* para el que confecciona las interfaces. Con la ayuda de una herramienta perteneciente a la mencionada biblioteca *PyQt*, se puede crear el código *Python* correspondiente a las interfaces confeccionadas en *Qt Designer* (4).

Qt Designer es muy útil en el desarrollo de las interfaces visuales de la aplicación, brindando la posibilidad de crearlas sencilla y cómodamente, debido a la variedad de componentes visuales que posee la biblioteca y la fácil manipulación de las variables de configuración de cada uno de ellos.

1.14 PostgreSQL

PostgreSQL es un Sistema Gestor de Bases de Datos (SGBD) relacional, ampliamente considerado como una de las alternativas entre los SGBD de código abierto. Soporta gran parte del *SQL* estándar y muchas

funcionalidades como son: consultas complejas, *triggers*, vistas, integridad transaccional, así como el control de versiones concurrentes que es una estrategia de almacenamiento que permite trabajar con grandes volúmenes de datos; ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que otros productos, conservando todas las características de estabilidad y rendimiento (38).

Una de las necesidades principales de la investigación fue la utilización de un SGBD potente para el almacenamiento de un alto volumen de información. En la actualidad existen varios que cumplen con esta característica, como son: *Oracle, Microsoft SQL Server y PostgreSQL*, dentro de los cuales fue seleccionado *PostgreSQL* por ser el único software libre de todos, lo que está en correspondencia con el resto de las aplicaciones utilizadas en el desarrollo de la herramienta. Además posee sencillas interfaces para *Python* y *Java* que ayudan en el almacenamiento y consulta de los datos procesados y fue utilizado en el desarrollo de HERMINWEB con buenos resultados.

1.15 pgAdmin

Cuando se trabaja con bases de datos, es necesario tener una herramienta para la gestión de las mismas. *pgAdmin* es una herramienta para la administración de bases de datos sobre el SGBD *PostgreSQL,* creada en el año 2002 por una comunidad de voluntarios de varias partes del mundo. Posee licencia *BSD* que proporciona permisos de uso, copia, modificación y distribución del *software*, así como su documentación. Es compatible con las plataformas *Windows*, *GNU\Linux*, *FreeBSD*, *Mac OS X* y *Solaris*. Su acceso al SGBD es por medio de *PostgreSQL* nativo, posee una potente herramienta de consultas y un rápido componente para la entrada y salida de datos (40).

Una de las herramientas claves en el éxito de una investigación que cuente con una base de datos son las herramientas de administración de base de datos. *pgAdmin* es una de ellas y fue de gran importancia para el sistema desarrollado, porque brinda la posibilidad de administrar la base de datos, realizar consultas, crear funciones, *triggers*, tablas, etc.

1.16 MongoDB

MongoDB es un sistema de base de datos NoSQL multiplataforma orientado a documentos, de esquema libre. Está escrito en C++, lo que le confiere cierta cercanía a los recursos de hardware de la máquina, de modo que es bastante rápido a la hora de ejecutar sus tareas. Las características más destacables de MongoDB son su velocidad y su rico pero sencillo sistema de consulta de los contenidos de la base de

datos. En MongoDB, cada registro o conjunto de datos se denomina documento, los cuales se almacenan en formato BSON, o Binary JSON, que es una versión modificada de JSON, permitiendo búsquedas rápidas de datos (41).

Los sistemas NoSQL como MongoDB proponen una estructura de almacenamiento más versátil que los relacionales. Entre las características más importantes de estos sistemas se encuentran: ausencia de esquema en los registros de datos, lo que posibilita aumentar la claridad y el rendimiento; escalabilidad horizontal sencilla, posibilitando aumentar el rendimiento del sistema añadiendo más nodos, sin necesidad en muchos casos de realizar ninguna otra operación más que indicar al sistema cuáles son los nodos disponibles; mayor velocidad pues estos sistemas realizan operaciones directamente en memoria, y sólo vuelcan los datos a disco cada cierto tiempo. Esto permite que las operaciones de escritura y lectura sean realmente rápidas (41).

Debido a todas estas características y a la necesidad de almacenar y recuperar de forma rápida los resultados obtenidos en el proceso de minería se utilizó este sistema de base de datos para el almacenamiento de los resultados.

1.17 RapidMiner

RapidMiner (42) es una herramienta creada en la Universidad de Dortmund para el descubrimiento del conocimiento y la minería de datos. Es un entorno con muchos algoritmos de aprendizaje y otras utilidades añadidas, está desarrollada sobre el lenguaje *Java* y funciona en los sistemas operativos más conocidos, constituyendo un software de código abierto y de libre distribución. Se retroalimenta de las librerías de funciones de WEKA (43) en su entorno de aprendizaje, posee alrededor de 400 operadores que pueden ser combinados, usa el lenguaje de *scripting* XML[20] para describir los operadores y su configuración y puede ejecutarse por línea de comandos.

En este trabajo fue de gran utilidad para aplicar los algoritmos de minería sin necesidad de implementarlos, obteniéndose resultados con calidad.

[20] del inglés: *Extensible Markup Language*

1.18 Pentaho

La compañía Pentaho es una alternativa de código abierto para la Inteligencia de Negocio (*BI*[21]). Desarrolla varias herramientas; una de ellas engloba a las demás, la cual se denomina *Pentaho BI Suite Enterprise Edition* que provee reportes, *OLAP*[22], integración de datos, minería de datos y una plataforma de *BI* (44).

Pentaho representa una solución completa de herramientas para la integración de datos. Para el desarrollo de HERMINWEB fue probada la *ETL*[23] *Kettle* de la *suite* en varias circunstancias de trabajo, comportándose correctamente salvo en la ejecución por líneas de comando, por lo que se hizo necesario implementar varias herramientas para la integración de datos, centradas en el ámbito y características de la UCI (4).

En la actual investigación, se probó la versión más reciente de *ETL Kettle*, la 4.0.1, la cual no presentó problemas para ejecutarla por líneas de comandos.

1.19 Conclusiones del capítulo

Se puede concluir que para dar solución a la problemática planteada fue necesario definirla como un proyecto de Minería de Datos, específicamente de Minería de uso web, donde se aplica un proceso de KDD que arroja resultados interesantes, dotando a la DRSI de la UCI de mayores elementos para tomar decisiones con respecto al sistema de asignación de cuotas de navegación por *Internet* y se usa un modelo descriptivo que implementa el algoritmo *FP-Growth* para la tarea Reglas de Asociación. Para ello, la selección de un lenguaje de programación como *Python*, multiplataforma, con una colección de librerías que lo hacen muy potente y una sintaxis sencilla, contribuyó en gran medida al desarrollo de la herramienta. El uso de potentes SGBD como *PostgreSQL* y MongoDB, permite el tratamiento del alto cúmulo de datos a analizar. La herramienta CASE *Visual Paradigm*, el IDE *Eclipse*, el uso de RapidMiner y Pentaho, así como el administrador de base de datos *pgAdmin,* permitieron agilizar el desarrollo de la investigación. CRISP-DM y RUP fueron de suma importancia para guiar el proceso de descubrimiento de conocimiento en base de datos y desarrollo de software respectivamente.

[21] del inglés: *Business Intelligence*
[22] acrónimo en inglés de procesamiento analítico en línea (*On-Line Analytical Processing*)
[23] del inglés: *Extract, Transform and Load*

Capítulo 2

CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En el presente capítulo se abordan los procesos de negocio, diagramas y descripciones asociados a la extracción de patrones en términos de reglas de asociación sobre los registros de navegación de los usuarios de la UCI. Además, son relacionados los requisitos funcionales, no funcionales y el modelo de casos de uso del sistema de la herramienta HERMINWEB para realizar la tarea Reglas de Asociación y mejorar algunas de las funcionalidades ya existentes en la misma.

2.2 Objeto de estudio

El objeto de estudio de la presente investigación radica en los patrones de comportamiento que se manifiestan en el uso de las cuotas de navegación por Internet de los usuarios de la UCI.

2.3 Problema y Situación Problemática

En la UCI, *Internet* constituye una de las principales fuentes de consulta para estudiantes, profesores y trabajadores, en disimiles ramas como: la investigación, docencia, producción y la superación profesional. El alto número de usuarios y el insuficiente ancho de banda que existe en la universidad hacen necesario buscar un sistema que realice una mejor gestión de las cuentas de navegación por Internet.

La DRSI ha creado varios sistemas informáticos para mantener el control sobre el comportamiento del uso de las cuotas de navegación por *Internet*, basados fundamentalmente en herramientas estadísticas. Estas herramientas muestran el comportamiento de ciertas variables presente en la navegación de los usuarios, aunque no son explotadas otras que se generan durante este proceso, que resultan interesantes en la búsqueda los objetivos planteados. Con el uso de una herramienta que permita extraer patrones de navegación explotando más el gran cúmulo de datos generado de este proceso a través de la aplicación

de algoritmos de minería de datos, se vería beneficiada la misión de la DRSI teniendo un mejor control sobre la asignación de cuotas de navegación por *Internet* y la toma de decisiones al respecto.

2.4 Objeto de automatización

Con las nuevas funcionalidades añadidas a la herramienta HERMINWEB se tiene como objetivo informatizar el proceso KDD sobre los registros de navegación por *Internet* en términos de la tarea descriptiva Reglas de Asociación, como apoyo a la toma de decisiones de la DRSI. Como procesos a automatizar se definen:

- Preparar Datos.
- Aplicar Minería.
- Evaluar Comportamiento.
- Difundir Conocimiento.

2.5 Propuesta de solución

Se decidió añadir nuevas funcionalidades a la herramienta HERMINWEB que posibilitaran resolver la problemática existente de forma automática. Dicha herramienta es una aplicación de escritorio que integra tecnologías libres sobre el sistema operativo GNU/Linux, donde se realiza las tarea descriptiva Agrupamiento y se le añadió Reglas de Asociación. Entre las características que tiene se encuentran: selección, integración, limpieza, transformación de los datos; la aplicación de las tareas descriptivas de Minería de Datos: Agrupamiento y Reglas de Asociación; la visualización y difusión de los resultados.

2.6 Modelo de negocio

A continuación se describen los procesos de negocio utilizando la notación BPMN. Por su importancia, sólo se muestran las descripciones correspondientes a los procesos: General, Preparar Datos, Aplicar Minería y Evaluar Comportamiento. Los diagramas y descripciones textuales del resto de los procesos pueden ser consultados en el expediente del proyecto en el cual se desarrolló esta solución. En (4) se detallan los principales estereotipos de esta notación utilizados para graficar y describir los procesos implicados en el negocio.

2.6.1 Proceso General

Ficha de Proceso

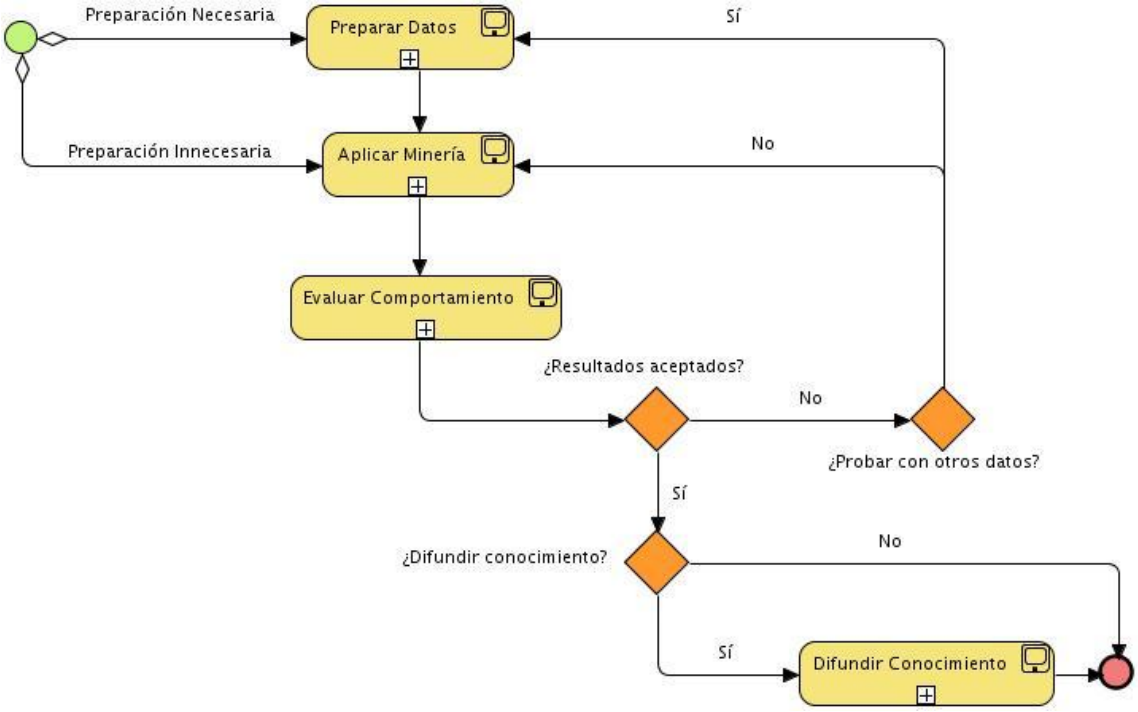
Proceso	Proceso General
Entradas	Fuentes de donde proceden los datos (información personal de los usuarios, datos de la navegación y clasificaciones de dominios) a preparar o datos ya preparados y atributos seleccionados para filtrar y crear las vistas minables.
Salidas	Patrones de comportamiento de los usuarios en forma de conocimiento
Descripción del Proceso	
La ejecución de la herramienta se ha dividido en 4 subprocesos que coinciden con fases del flujo de KDD. Si es necesaria la preparación, esta se ejecuta y luego se aplica el algoritmo de Minería de Datos para obtener las reglas de asociación. Posteriormente se extraen los patrones, que de ser aceptables se difunden en forma de conocimiento mediante gráficas. De no estar de acuerdo con los patrones obtenidos, el interesado puede volver a preparar los datos o cambiar los parámetros del algoritmo de minería.	
Diagrama de Proceso	
	

Tabla 1: Descripción del Proceso General

2.6.2 Preparar Datos

Ficha de Proceso	
Proceso	Preparar Datos

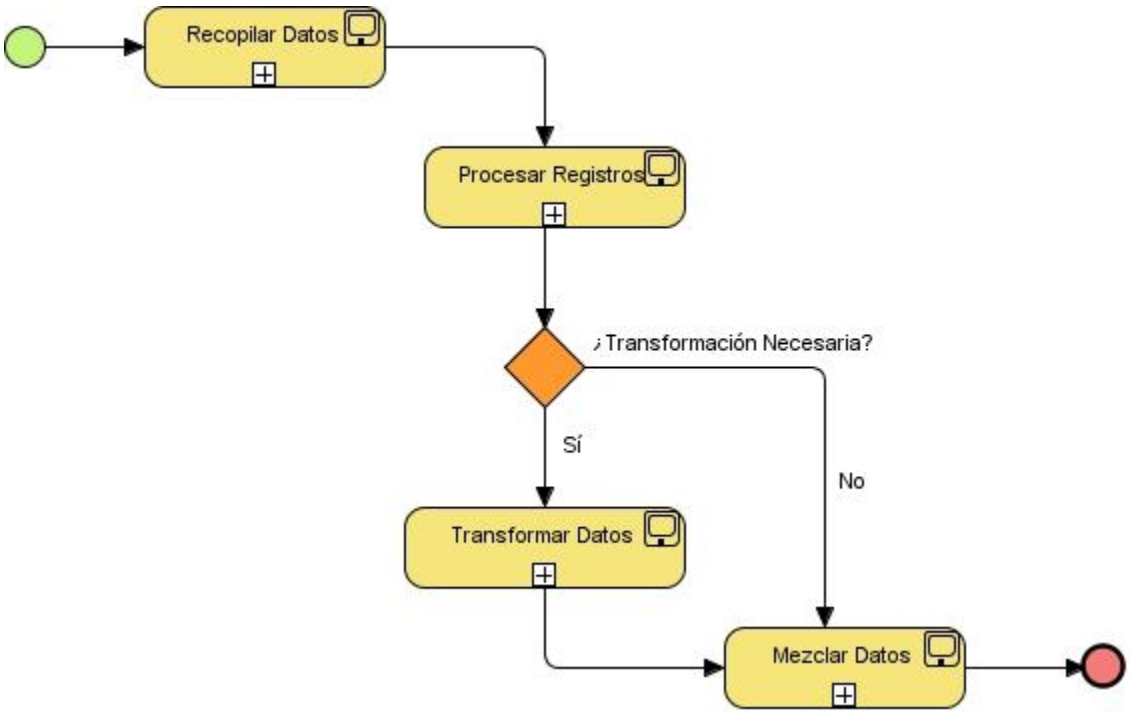
Entradas	Fuentes de donde proceden los datos a preparar y atributos seleccionados para filtrar y crear las vistas minables.
Salidas	Sesiones de usuarios
Descripción del Proceso	
La preparación de los datos se ha dividido en 4 subprocesos acordes con las tareas que se realizan en esta fase de KDD. Primero se recopilan los datos provenientes de las fuentes teniendo en cuenta los y atributos seleccionados para filtrar y crear las vistas minables, transformándolos de ser necesario; luego se procesan los registros del *proxy*. Finalmente se mezclan las clasificaciones de dominio con los datos de la navegación y los usuarios para obtener las sesiones de usuarios.	
Diagrama de Proceso	
	

Tabla 2: Descripción del proceso Preparar Datos

2.6.3 Aplicar Minería

Ficha de Proceso	
Proceso	Aplicar Minería
Entradas	Parámetros del algoritmo y credenciales de conexión a la base de datos
Salidas	Reglas de Asociación

Descripción del Proceso
Los parámetros del algoritmo y las credenciales de conexión a la base de datos se envían a la biblioteca de algoritmos de Minería de Datos, la cual ejecuta el algoritmo que permite obtener las reglas de asociación, guardándose estas en base de datos.
Diagrama de Proceso

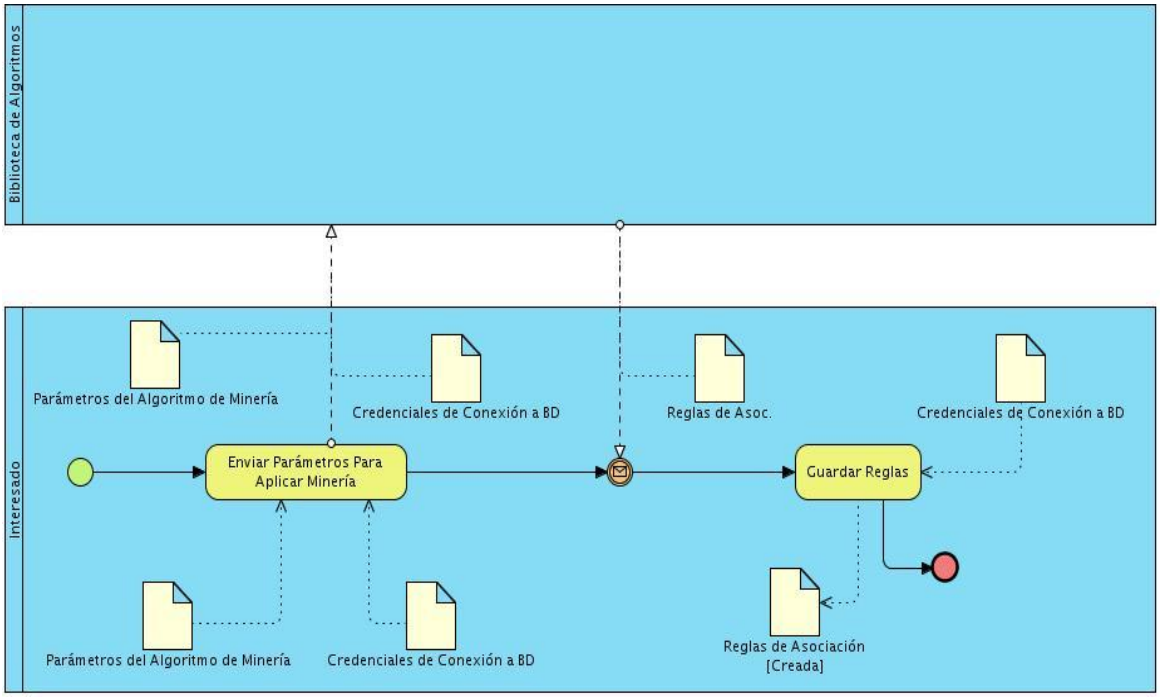


Tabla 3: Descripción del proceso Aplicar Minería

2.6.4 Evaluar Comportamiento

Ficha de Proceso	
Proceso	Evaluar Comportamiento
Entradas	Credenciales de conexión a la base de datos
Salidas	Patrones de comportamiento de los usuarios en forma de conocimiento
Descripción del Proceso	
Se obtienen las reglas de asociación previamente guardadas en base de datos y se crean gráficas que expresen los patrones de comportamiento de los usuarios de manera comprensible (conocimiento).	
Diagrama de Proceso	

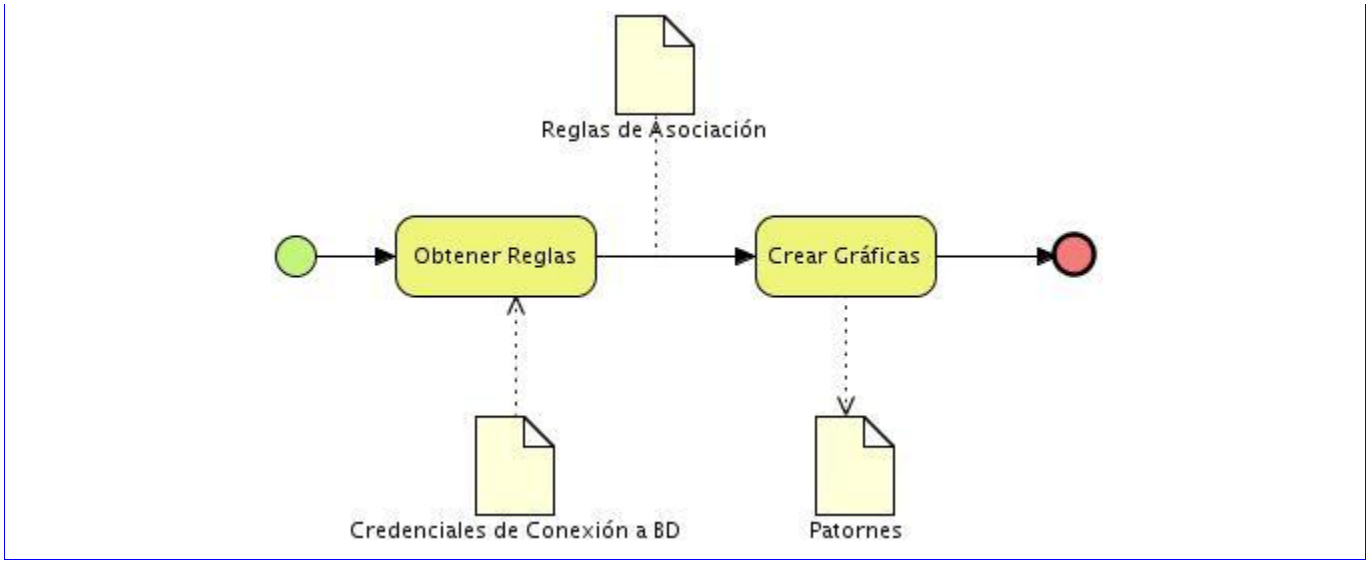


Tabla 4: Descripción del proceso Evaluar Comportamiento

2.7 Especificación de los requisitos del software

2.7.1 Requerimientos funcionales

Los requerimientos funcionales son las capacidades o condiciones que el sistema debe cumplir (28). A continuación se muestran los requisitos funcionales con los que debe cumplir el sistema, teniendo en cuenta también los que fueron definidos e implementados previamente en HERMINWEB (4). Los requisitos resaltados en negrita son los que se definieron para la presente investigación debido a que son nuevos o a que fueron ya implementados en HERMINWEB pero sufrieron cambios significativos, considerándose igualmente nuevos.

R1: Autenticar Usuario: Identifica a los usuarios y los autoriza a utilizar las funcionalidades de la aplicación.

R2: Gestionar las configuraciones: Permite almacenar o cargar las configuraciones desde ficheros.

R2.1: Almacenar las configuraciones: Almacena las configuraciones en un fichero de texto.

R2.2: Cargar las configuraciones: Obtiene las configuraciones desde un fichero de texto.

R2.3: Modificar Configuración: Modifica las configuraciones en el fichero de texto.

R3: Preparar los datos: Recopila, transforma y limpia los datos de los usuarios y las clasificaciones de dominios y los mezcla con los datos de la navegación procedentes de los *logs* del proxy.

R3.1: Autenticar en el host del servidor web: Identifica a un usuario en el host del servidor web y lo autoriza a utilizar los registros del *proxy* compartidos en él.

R3.2: Obtener usuarios: Realiza un recorrido por el árbol de directorios donde están los registros de navegación compartidos en el servidor web y obtiene los nombres de usuario a partir del nombre que tiene cada registro.

R3.3: Preparar expedientes usuarios: Obtiene los datos de los usuarios consultando el Servicio de Directorio para crear los expedientes.

R3.4: Preparar datos de estudiantes: Se obtienen de la base de datos de Akademos los datos necesarios de los estudiantes, se filtran y se almacenan, con apoyo de la *ETL Kettle* de *Pentaho.*

R3.5: Preparar datos de trabajadores: Se obtienen de los servicios web ASSETS y Registro Personal los datos necesarios de los trabajadores, se filtran y se almacenan, con apoyo de la *ETL Kettle* de *Pentaho*.

R3.6: Procesar *blacklists*: Obtiene las clasificaciones de las páginas contenidas en los ficheros de *blacklists* y se almacenan.

R3.7: Ejecutar transformación: Crea el XML de la transformación a ejecutar y manda a ejecutar la misma en la herramienta *Pentaho*.

R3.8: Reorganizar registros del proxy: Reorganiza la estructura de carpetas que tienen los *logs* del *proxy,* compartidos en el servidor web, utilizando para ello un acceso directo.

R3.9: Procesar registros del proxy: Se procesan los *logs* de los usuarios para obtener los datos de la navegación y se almacenan estos datos.

R3.10: Publicar recursos en servidor remoto: Publica en los nodos de procesamientos las funcionalidades necesarias para crear las sesiones de usuario desde la aplicación.

R3.11: Distribuir usuarios: Envía a los nodos de procesamiento los usuarios que se van a procesar para obtener los datos de su navegación.

R3.12: Mezclar datos: Obtiene los datos de los estudiantes, trabajadores, de la navegación y las clasificaciones y los mezcla para crear las sesiones de usuarios con apoyo de la *ETL Kettle* de *Pentaho*.

R4: Aplicar minería: Aplica el algoritmo de minería de datos a la vista minable y almacena los resultados en base de datos, apoyado en una biblioteca de algoritmos de minería.

R5: Evaluar comportamiento: Muestra las gráficas creadas a partir de las reglas de asociación.

R5.1: Crear gráficas: Obtiene las reglas de asociación generadas y crea gráficas con la información que el interesado necesita conocer.

R6: Difundir los resultados: Crea reportes en formato *pdf* y hojas de cálculo, permitiendo enviarlos de forma opcional en un mensaje de correo electrónico a los interesados.

R6.1: Enviar correo a interesados: Envía un mensaje de correo electrónico a los interesados.

R6.2: Almacenar datos en *pdf*: Almacena el reporte de los resultados en un fichero *pdf*.

R6.3: Almacenar modelos en hojas de cálculo: Almacena en hojas de cálculo los modelos obtenidos como resultado de la extracción de patrones.

2.7.2 Requerimientos de comportamiento

2.7.2.1 Requerimientos de Usabilidad

Se debe garantizar un acceso fácil y rápido a los usuarios autorizados de la DRSI a la aplicación. El sistema podrá ser usado solo por personas autorizadas de la DRSI.

2.7.2.2 Requerimientos de Soporte

Una vez desplegado el software se continuará asesorando a los clientes durante 2 semanas, además de que se realizará un mantenimiento total del producto en cada semestre, adaptándolo a nuevas necesidades en caso de que se solicite.

2.7.2.3 Requerimientos de Software

- En la estación de trabajo donde se utilice la herramienta deberá estar instalada la distribución de GNU\Linux: Ubuntu (4), en sus versiones 10.04 o 10.10.
- Es necesaria la previa instalación del intérprete para el lenguaje Python (4).

- Es necesaria la previa instalación de la JVM (4).
- Se requiere un servidor de base de datos PostgreSQL previamente instalado y configurado (4).
- Se requiere un servidor de base de datos NoSQL MongoDB previamente instalado y configurado.
- Se requiere un servidor web Apache previamente instalado y configurado en un ordenador que permita la realización de conexiones utilizando el protocolo SSH[24], en el cual se compartirán los registros del proxy que serán procesados (4).

2.7.2.4 Restricciones en el diseño y la implementación

Aquí se especifican las restricciones que deben tenerse en la construcción y codificación del sistema y que deben ser cumplidas estrictamente.

- La versión del lenguaje Python 2.6.
- La codificación será la determinada en la guía de estilo para el lenguaje *Python* propuesta por Guido van Rossum (el creador de este lenguaje).
- Para la codificación se utilizará *Eclipse*, facilitando considerablemente la Programación con *Python* en el mismo.
- Han de estar instaladas las bibliotecas de *Python*:
    - python-dev
    - pyRXP-1.13
    - pyro 3.10
    - python-paramiko
    - python-pexpect
    - python-ldap
    - python-dnspython
    - python-pyparsing 1.5.2
    - python-impacket
    - python-reportlab
    - python-numpy
    - python-soappy

[24] del inglés: Secure SHell

- python-crypto 2.3
- python-psycog2
- python-sqlalchemy
- python-qt4
- pyqt4-dev-tools
- python-qt4-dev
- python-pymssql
- python-zeroc-ice
- python-xlwt
- python-pycha
- python-cairo
- python-pymongo
- libnspr4-dev
- libnss3-dev

2.7.2.5 Requerimientos de Seguridad

Disponibilidad: El sistema debe de estar disponible siempre que se necesite ejecutarlo. En caso de no encontrarse disponible alguna de las fuentes de datos el sistema debe informar al usuario esta situación.

Confidencialidad: La aplicación asegura que cada usuario sólo pueda entrar al sistema autenticándose y con previa autorización de la DRSI de la UCI. Las carpetas donde se reorganizan y comparten los registros de navegación de los usuarios por defecto deben estar protegidas con contraseña. Para reorganizar los registros se debe utilizar un protocolo seguro para la comunicación entre HERMINWEB y el servidor web.

Integridad: La información manejada por el sistema será objeto de cuidadosa protección.

2.7.2.6 Requerimientos legales

Se prohíbe vender, reproducir o comercializar esta aplicación sin el debido permiso de los autores de la misma y los directivos de la facultad 2 de la UCI.

2.8 Modelo de Casos de Uso del Sistema

2.8.1 Actores del Sistema

Actor	Descripción
Interesado	Persona que necesita conocer los patrones de navegación de los usuarios, buscando una ayuda para la toma de decisiones: administradores de redes y directivos de la DRSI.
Pentaho	Representa la *ETL Kettle* de la *suite Pentaho*, donde se realizan las transformaciones de los datos.
Biblioteca de Algoritmo	Representa la biblioteca de algoritmos de minería de datos donde se realiza el proceso de minería.
Intermediario ICE	Aplicación utilizada en el procesamiento distribuido de los registros de navegación.

Tabla 5: Descripción de los actores del sistema

2.8.2 Casos de Uso del Sistema

Los CU son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Por tanto, establece un acuerdo entre clientes y el grupo de desarrollo sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema (28).

A continuación se muestran los CU teniendo en cuenta también los que fueron definidos e implementados previamente en HERMINWEB (4). Al igual que en la definición de los requisitos del sistema, los CU resaltados en negrita son los que se definieron para la presente investigación debido a que son nuevos o a que fueron ya implementados en HERMINWEB pero sufrieron cambios significativos, considerándose igualmente nuevos. Los que se resaltan en rojo constituyen los CU críticos del sistema, que serán de aquí en adelante a los que se les dará seguimiento en el presente documento. Para consultar la descripción detallada de todos los CU se puede revisar el documento Modelo de Casos de Uso del Sistema del proyecto productivo al cual pertenece este trabajo.

Casos de uso del sistema agrupados por paquetes:

Configuración:

CU-1 **Autenticar usuario.**

CU-2 Gestionar Configuración.

CU-3 Gestionar Configuración Visual.

CU-4 Gestionar Configuración desde Fichero.

Preparación, Minería y Evaluación:

CU-5 **Preparar Datos.**

CU-6 Preparar Expedientes de Usuarios.

CU-7 **Procesar Usuarios.**

CU-8 **Procesar Estudiantes.**

CU-9 **Procesar Trabajadores.**

CU-10 **Ejecutar Transformación.**

CU-11 **Procesar Registro.**

CU-12 **Reorganizar Registro.**

CU-13 Procesar *blacklists*.

CU-14 **Aplicar Minería.**

CU-15 **Evaluar comportamiento.**

Difusión:

CU-16 **Almacenar PDF.**

CU-17 **Almacenar hojas de cálculo**

CU-18 Enviar correo.

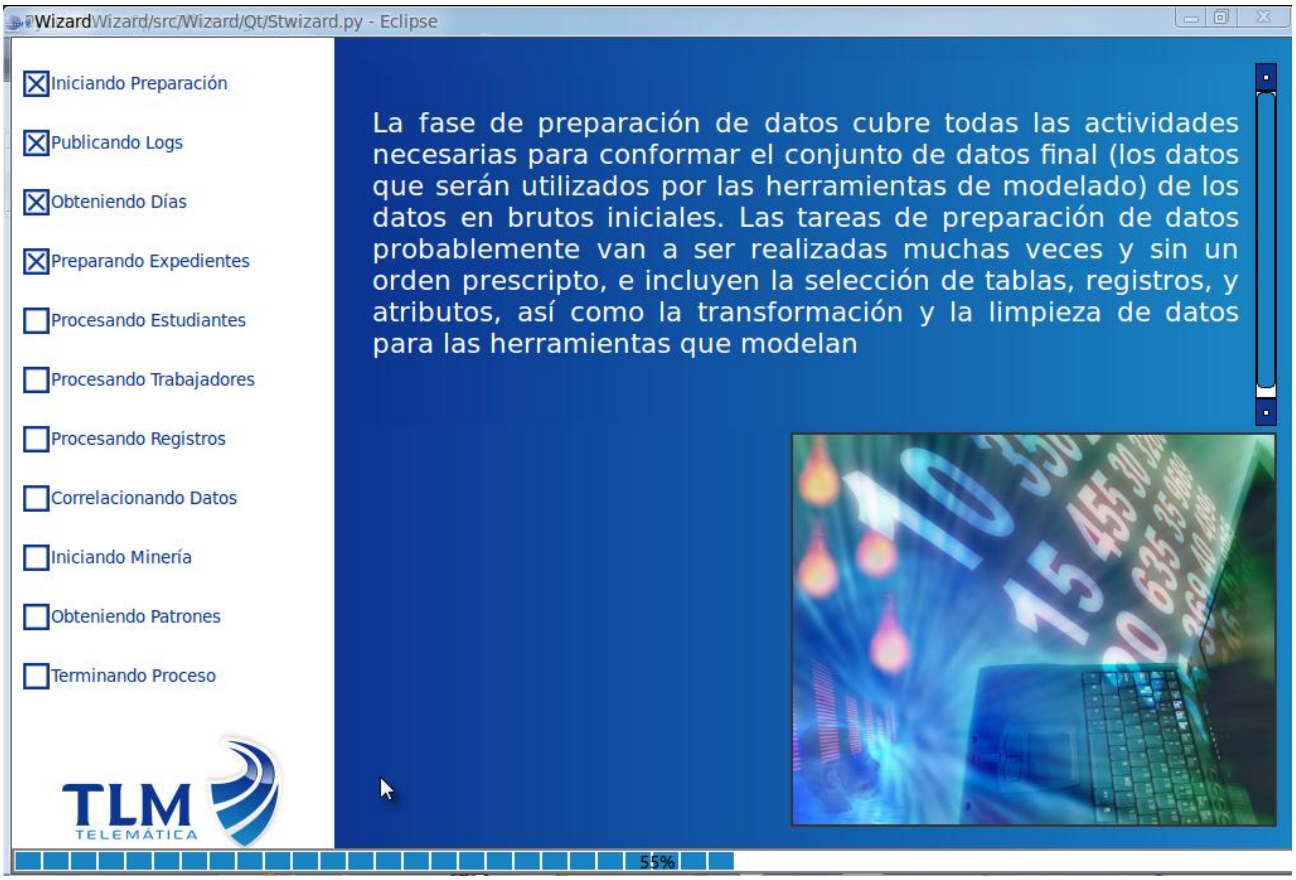
2.8.1.1 Descripción detallada del caso de uso Preparar Datos

Caso de Uso	
CU-5	Preparar Datos
Actores	Interesado

Resumen	En el presente caso de uso se obtienen, limpian, transforman y filtran los datos personales y los datos de la navegación para cada tipo de usuario seleccionado (Estudiantes, Trabajadores Internos y Trabajadores Externos), luego se mezclan y se almacenan en base de datos, construyéndose así las vistas minables.
Precondiciones	Configuración completada
Referencias	R-3, R-3.1, R-3.2, CU-6(inclusión), CU-7(inclusión), CU-10(inclusión), CU-11(inclusión), CU-13(extensión).
Prioridad	Crítico
Flujo Normal de Eventos	
Sección "Preparar Datos"	
Acción del Actor	**Respuesta del Sistema**
	1- El sistema se conecta al servidor donde están publicados los registros del *proxy* y obtiene los usuarios a analizar.
	2- El sistema prepara los expedientes de todos los usuarios a analizar (Ver CU-6).
	3- El sistema procesa los usuarios, obtiene sus datos personales, los filtra teniendo en cuenta los atributos seleccionados y los almacena en base de datos (Ver CU-7 y CU-8).
	4- El sistema de manera distribuida procesa los registros, obtiene los datos de navegación para cada usuario, los filtra y almacena el resultado en base de datos (Ver CU-11).
	5- El sistema apoyado en la herramienta *ETL Kettle* de *Pentaho* aplica una transformación y mezcla los datos de navegación y

	personales de cada usuario para crear las vistas minables (Ver CU-10).

Prototipo de Interfaz



Flujos Alternos	
Acción del Actor	**Respuesta del Sistema**
	1.1 Error en la conexión al servidor: el sistema muestra un mensaje de error y muestra la interfaz para gestionar la configuración.
	2.1 Error en la conexión al Servicio de Directorio: el sistema muestra un mensaje de error y muestra la interfaz para gestionar la configuración.

	2.2 No se obtuvieron expedientes: El sistema muestra un mensaje con la información de esta situación y muestra la interfaz para gestionar la configuración.
	3.1 No se obtuvieron datos de ningún tipo de usuario al aplicar el filtro: El sistema muestra un mensaje con la información de esta situación y muestra la interfaz para gestionar la configuración.
	3.2 Se obtuvieron datos sólo para algunos tipos de usuario al aplicar el filtro: El sistema muestra un mensaje con la información de esta situación y brinda la posibilidad al usuario de continuar o volver a la gestión de la configuración.
3.2-A El Interesado selecciona la opción de continuar.	3.2-A.1 El sistema continúa con el proceso de preparación de los datos.
3.2-B El Interesado selecciona la opción de volver a la gestión de la configuración.	3.2-B.1 El sistema muestra la interfaz para gestionar la configuración.
	3.3 Error en la conexión a la base de datos: el sistema muestra un mensaje de error y muestra la interfaz para gestionar la configuración.
	4.1 Error en la conexión a servidor ICE: el sistema muestra un mensaje de error y muestra la interfaz para gestionar la configuración.
	4.2 Error en la conexión al servidor web: el sistema muestra un mensaje de error y muestra la interfaz para gestionar la configuración.
	4.3 Error en la conexión a la base de datos: el sistema muestra

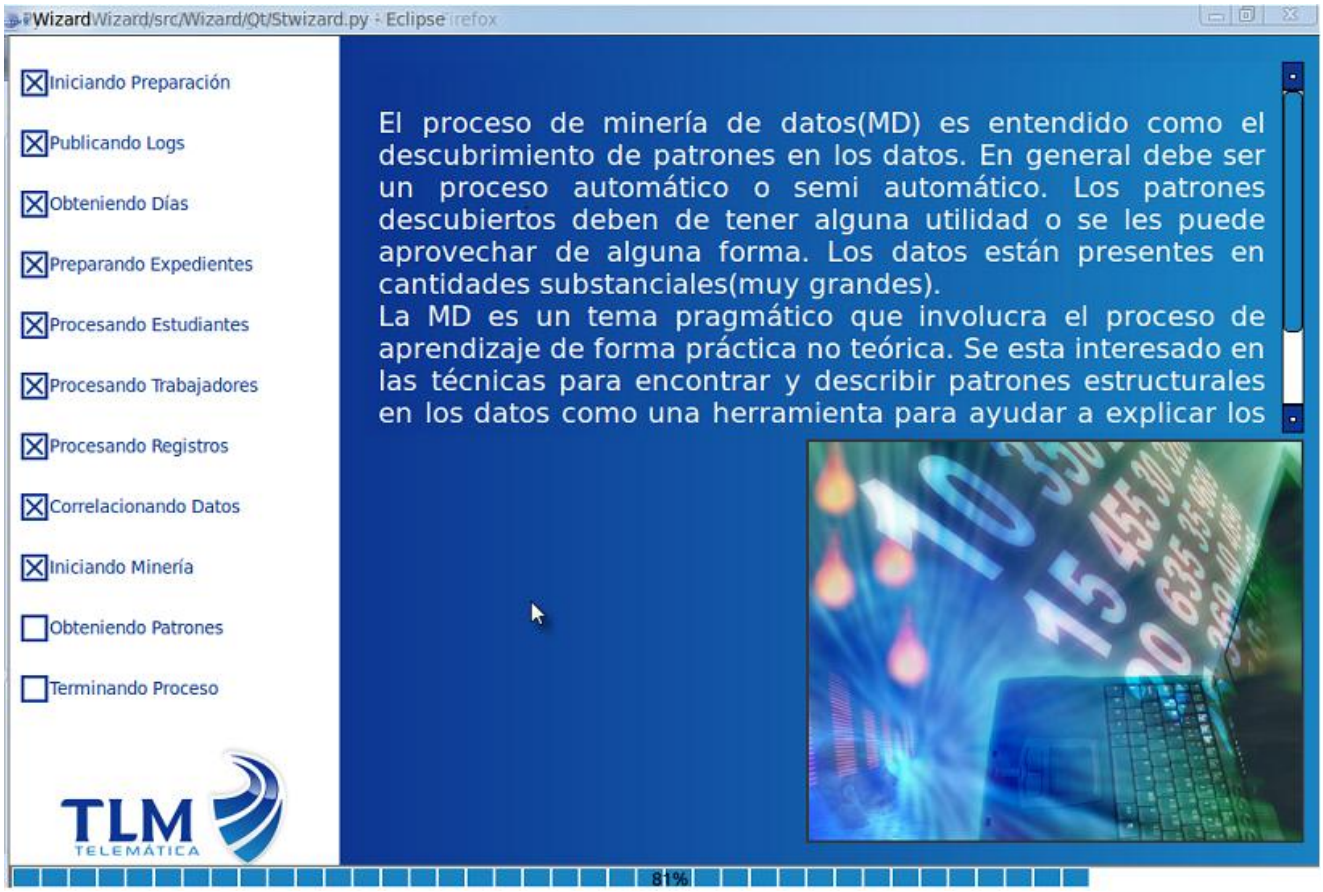
	un mensaje de error y muestra la interfaz para gestionar la configuración.
	4.4 No se obtuvieron datos de ninguna vista minable al aplicar el filtro: El sistema muestra un mensaje con la información de esta situación y muestra la interfaz para gestionar la configuración.
	4.5 Se obtuvieron datos sólo de algunas de las vistas minables al aplicar el filtro: El sistema muestra un mensaje con la información de esta situación y brinda la posibilidad al usuario de continuar o volver a la gestión de la configuración.
4.5-A El Interesado selecciona la opción de continuar.	4.5-A.1 El sistema continúa con el proceso de preparación de los datos.
4.5-B El Interesado selecciona la opción de volver a la gestión de la configuración.	4.5-B.1 El sistema muestra la interfaz para gestionar la configuración.
	5.1 Error durante la ejecución de la transformación: el sistema muestra un mensaje de error y muestra la interfaz para gestionar la configuración.
Poscondiciones	Vistas minables creadas y almacenadas en base de datos

Tabla 6: Descripción detallada del CU Preparar Datos

2.8.1.2 Descripción detallada del caso de uso Aplicar Minería

Caso de Uso	
CU-14	Aplicar Minería
Actores	Interesado (inicia), Biblioteca de algoritmos.
Resumen	En este caso de uso se realiza la extracción de patrones

<table>
<tr><td></td><td colspan="2">utilizando el algoritmo para la tarea seleccionada apoyado de la biblioteca de algoritmos de minería. Luego se guardan en ficheros de texto los patrones obtenidos.</td></tr>
<tr><td>Precondiciones</td><td colspan="2">El Interesado debe estar autenticado en el sistema y las vistas minables listas.</td></tr>
<tr><td>Referencias</td><td colspan="2">R-4,CU-5(extensión)</td></tr>
<tr><td>Prioridad</td><td colspan="2">Crítico</td></tr>
<tr><td colspan="3">Flujo Normal de Eventos</td></tr>
<tr><td colspan="3">Sección “Aplicar Minería”</td></tr>
<tr><td>Acción del Actor</td><td colspan="2">Respuesta del Sistema</td></tr>
<tr><td rowspan="3">1. El Interesado selecciona la opción de Iniciar el proceso KDD.</td><td colspan="2">2. El sistema crea el XML que describe el proceso a ejecutar en la biblioteca de algoritmos.</td></tr>
<tr><td colspan="2">3. El sistema envía la orden de ejecutar el proceso.</td></tr>
<tr><td colspan="2">4. La biblioteca aplica el algoritmo de minería a los datos de las vistas minables y almacena los patrones obtenidos en ficheros de texto.</td></tr>
<tr><td colspan="3">Prototipo de Interfaz</td></tr>
</table>

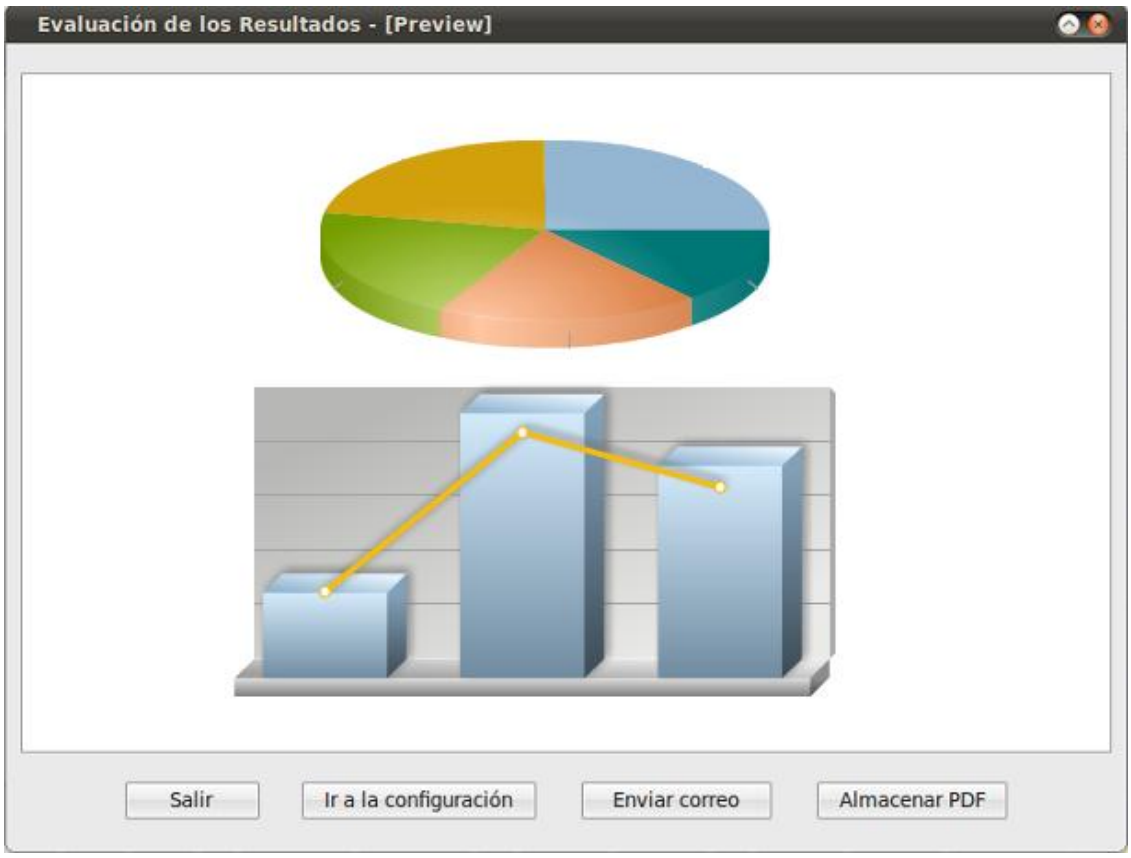


Flujos Alternos	
Acción del Actor	**Respuesta del Sistema**
	2.1 De ser necesario el sistema realiza la preparación de los datos. (Ver CU-5).
	4.1 Error durante la extracción de patrones: La aplicación muestra un mensaje de error y muestra la interfaz para gestionar la configuración.
Poscondiciones	Patrones obtenidos y almacenados en ficheros de texto

Tabla 7: Descripción detallada del CU Aplicar Minería

2.8.1.3 Descripción detallada del caso de uso Evaluar Comportamiento

Caso de Uso:	
CU-15	Evaluar Comportamiento
Actores:	
Resumen:	En este caso de uso, son mostrados al Interesado mediante gráficos y tablas los resultados de los patrones encontrados después de **Aplicar Minería**.
Precondiciones:	Patrones obtenidos
Referencias	R-5, R-5.1, CU-16 (extensión).
Prioridad	Crítico
Flujo Normal de Eventos	
Sección "Evaluar Comportamiento."	
Acción del Actor	**Respuesta del Sistema**
	1. El sistema procesa los ficheros de texto con los patrones obtenidos. (Ver CU-14)
	2. El sistema crea gráficas y tablas para reflejar los patrones obtenidos de manera comprensible.
	3. El sistema muestra una interfaz con las gráficas y tablas creadas.
Prototipo de Interfaz	



Poscondiciones	

Tabla 8: Descripción detallada del CU Evaluar Comportamiento

2.8.3 Diagramas de casos de uso del sistema agrupados por paquetes

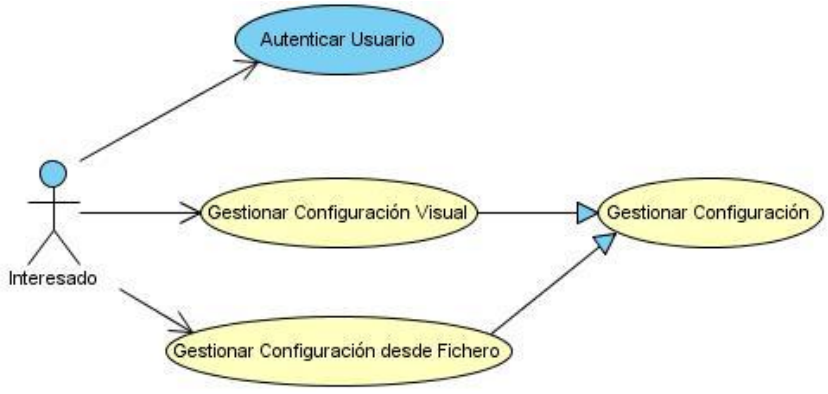


Figura 3: Diagrama de CU del paquete Configuración

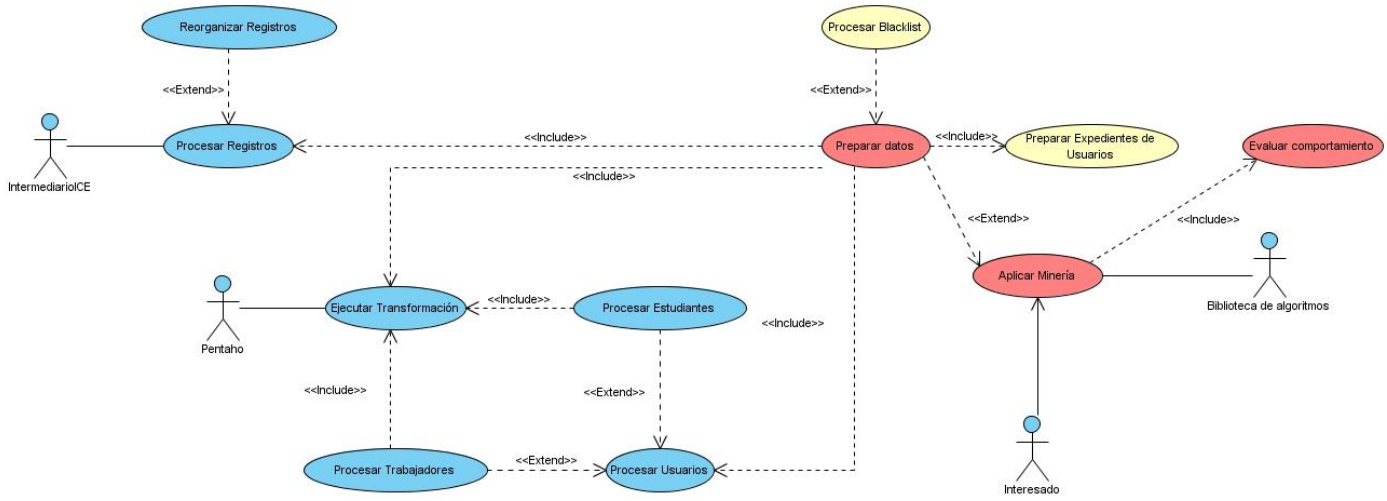


Figura 4: Diagrama de CU del paquete Preparación, Minería y Evaluación.

2.9 Conclusiones del capítulo

En el presente capítulo se realizó un análisis de los procesos de negocio como base fundamental para la captura de requisitos. Dichos procesos fueron modelados con la notación BPMN, permitiendo una clara comprensión de los mismos. Fueron reflejados los requisitos funcionales y no funcionales, así como los casos de uso del sistema, haciendo énfasis en los críticos. Fue estructurado el modelo de casos de uso del sistema que constituye una entrada importante para desarrollar el siguiente flujo de trabajo.

Capítulo 3

DISEÑO DEL SISTEMA

3.1 Introducción

En el presente capítulo se dará una vista general de la arquitectura y patrones utilizados en el diseño de la aplicación. Se profundiza en las mejoras de las que fue objeto HERMINWEB y los cambios que se realizaron en el sistema de configuraciones y los procesos de preparación de los datos, minería, evaluación y difusión de los resultados. Será mostrado el modelo físico de datos, el diagrama de paquetes y las clases del diseño con sus relaciones, sirviendo de base para la implementación del sistema. Se presenta de manera general la estructura del sistema a partir de las funcionalidades previstas.

3.2 Arquitectura y Patrones

3.2.1 Arquitectura

Para el desarrollo de este trabajo se mantendrá la arquitectura sobre la cual fue implementada HERMINWEB. En la Figura 13, se presenta una vista general de la distribución de los componentes y capas que constituyen el sistema, siendo la clase **Principal** quien comienza la ejecución del mismo. En la capa **Dominio**, se encuentran todas las entidades que serán utilizadas durante la ejecución por más de una clase de las diferentes capas. En la capa **GUI** se implementan interfaces gráficas de usuario que serán mostradas por el sistema. La capa **Entornos de Ejecución** representa la abstracción entre la capa **GUI** y la de **Servicios**; esta última contiene toda la lógica del negocio. El **acceso y la modificación de los datos** se realizan a través de la capa del mismo nombre. El sistema se apoya sobre el **Framework** (implementado por el equipo de desarrollo del proyecto Servicios Telemáticos) para la utilización de variadas funcionalidades como: conexión a bases de datos, control de procesos externos, envío de correos, ejecución de funciones en servidores remotos, etc. Todas las configuraciones de las interfaces e implementaciones del sistema son almacenadas en archivos XML, los cuales son utilizados por el **Framework** para el conocimiento de las clases empleadas, así como las interfaces que las mismas implementan (4).

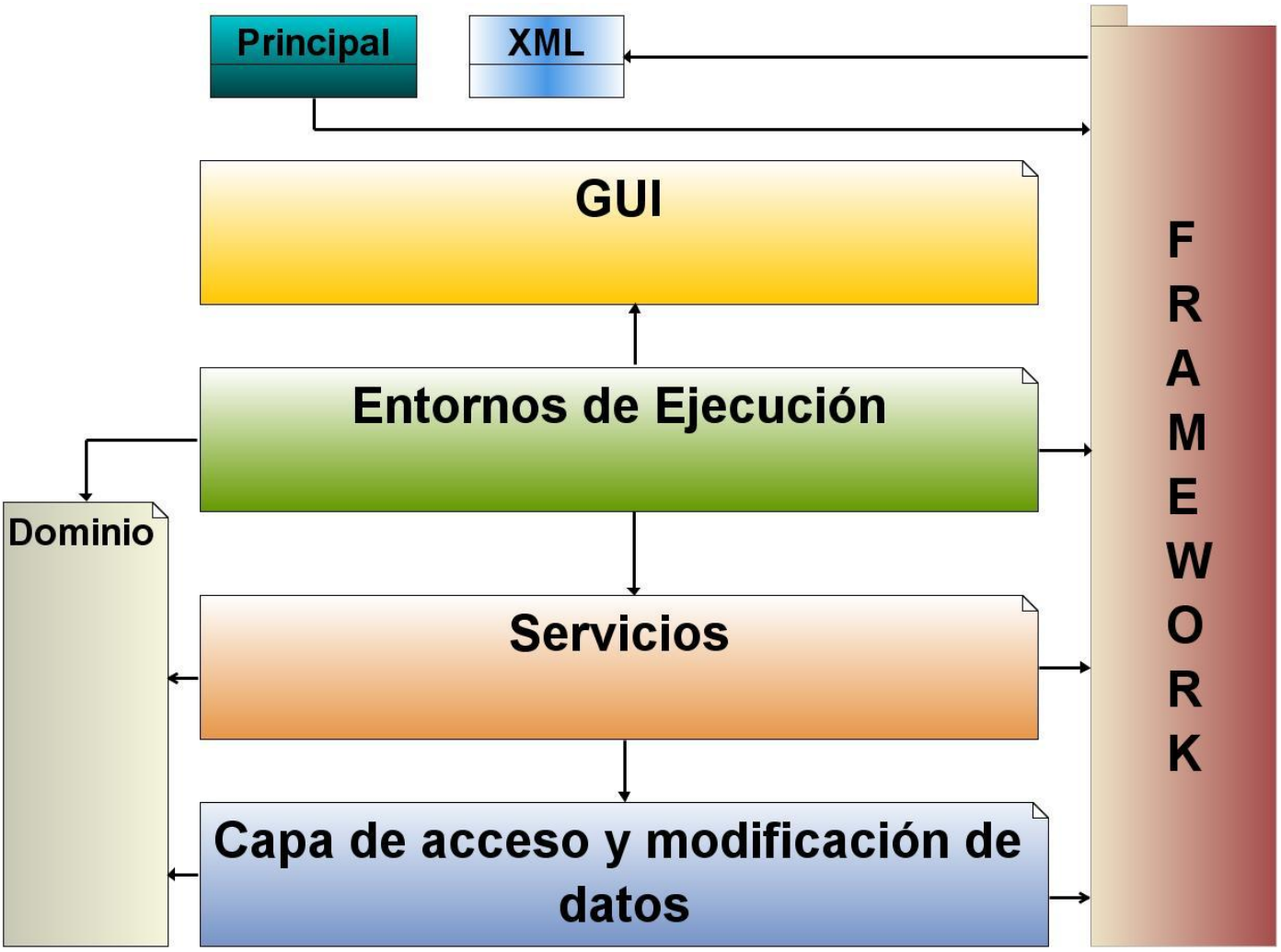


Figura 13: Arquitectura del sistema

3.2.2 Patrones de diseño

Un patrón es un par problema/solución con nombre que se puede aplicar en nuevos contextos, con consejos acerca de cómo aplicarlo en nuevas situaciones y discusiones sobre sus compromisos (45). En este trabajo no fue necesario utilizar nuevos patrones diferentes a los que fueron utilizados en HERMINWEB, por tanto, a continuación se muestran los distintos patrones aplicados en el diseño del sistema, tal como se explican en (4):

3.2.2.1 Proxy y Abstract Factory

El patrón **Proxy** proporciona un representante o delegado que se encargue de controlar el acceso a un objeto, generalmente con motivos de eficiencia. Posee la ventaja de permitir el acceso a objetos que residen en espacios distintos de memoria, abstrayendo al programador de la ubicación del objeto solicitado.

El patrón **Abstract Factory**, brinda una interfaz para la creación de familias de objetos interdependientes o interrelacionados, sin la necesidad de especificar sus clases concretas. Potencia el encapsulamiento e incrementa la flexibilidad del diseño. Esto ayuda a modificar las clases que son encapsuladas sin la necesidad de realizar cambios que puedan representar trastornos en la implementación que las ha usado.

Ambos fueron utilizados en la creación de los consultores para las conexiones a las distintas bases de datos. Se hizo necesario su uso pues el acceso a base de datos de la herramienta se ha de realizar a distintos gestores de bases de datos, los cuales no tienen las mismas interfaces de acceso.

3.2.2.2 Facade

Proporciona una interfaz de alto nivel unificada para facilitar el acceso a los métodos. Se utiliza para simplificar los métodos expuestos de una clase u objeto facilitando considerablemente su uso.

Se utiliza en la creación de las clases que administran las distintas fases del proceso completo que realiza la herramienta, se diseña una interfaz sencilla de cada una para simplificar y facilitar el uso de las mismas por las clases de las capas superiores.

3.2.2.3 Observer y Chain of Responsibility

Observer define una dependencia uno a muchos entre objetos, de modo que cuando el estado de un objeto cambia, se notifica el cambio a todos los que dependen de él y se actualizan de forma automática.

El patrón **Chain of Responsibility** proporciona a un objeto la capacidad de atender una petición para así evitar el acoplamiento con el objeto que realiza la petición. Se forma con estos objetos una cadena en la cual cada uno satisface la petición o la envía al siguiente; reduciéndose el acoplamiento pues se libera a un objeto que realiza la petición de conocer quién la atiende. Proporciona una mayor flexibilidad pues se puede añadir o modificar la capacidad de atender una solicitud, simplemente haciendo los cambios en la cadena de responsabilidades dinámicamente.

Fue necesario utilizar estos dos patrones en conjunto, pues en el sistema se ejecutarán en determinadas circunstancias procesos de manera concurrente, sobre los cuales en todo momento se necesitará conocer su estado, además de poder comunicarse entre sí. Gracias a su uso, el diseño para poder establecer la comunicación entre los procesos fue más sencillo y consistente.

3.2.2.4 Decorator

Decorator brinda la posibilidad de añadir dinámicamente funcionalidades a un objeto. Esto ofrece un poco más de flexibilidad que la herencia a la hora de extender funcionalidades evitando que las clases más altas de la jerarquía se carguen de funcionalidades.

En determinados escenarios, se hace necesario brindarle a una función la posibilidad de ser ejecutada en varios hilos, mediante este patrón se hace posible aplicarle esta funcionalidad en tiempo de ejecución sin la necesidad de modificar el código de la función o crear una clase con esta nueva funcionalidad multihilos.

3.2.2.5 Iterator

Este patrón define una interfaz que declara los métodos necesarios para poder acceder de forma secuencial a una colección de objetos. Las clases que utilizan esta interfaz para acceder a los objetos lo hacen de forma independiente de la que implementa la interfaz. Permite que la estructura interna a la que se accede para iterar permanezca oculta.

Se aplica dicho patrón pues en numerosas ocasiones es necesario crear objetos que se comporten como secuencias, permitiendo además una personalización del acceso al contenedor interno sin necesidad de cargar todos los datos en memoria.

3.2.2.6 Bridge

Tiene como propósito separar la abstracción de la implementación, evitando la unión permanente entre estas. Esto ayuda a aplicar cambios futuros sin la necesidad de que un cambio en uno pueda afectar el otro.

En este caso fue necesaria la utilización del patrón para poder luego desacoplar las implementaciones de las herramientas para la extracción de patrones, las cuales pueden ser modificadas o sustituidas sin causar considerables modificaciones en la implementación de HERMINWEB.

3.3 Mejoras que reducen el tiempo de duración del proceso de preparación de los datos

3.3.1 Integración con la *ETL Kettle* de *Pentaho*

La utilización de esta herramienta significó un cambio sustancial en el diseño de la aplicación, específicamente en la preparación de los datos, ya que no son necesarias varias funcionalidades implementadas en HERMINWEB para este propósito.

HERMINWEB ahora es la encargada de solicitar al usuario las clasificaciones que desee realizar para cada uno de los atributos (edad, sexo, horario de navegación, etc.) y cuáles de esos atributos serán incluidos en la vista minable. Estas clasificaciones son procesadas para que se creen dinámicamente las transformaciones a ejecutar en *Pentaho* a través de un *parser* que construya el XML de dichas transformaciones, que se ejecutan por líneas de comandos, aumentando el rendimiento de la aplicación, debido a que *Pentaho* ejecuta las transformaciones con mayor rapidez. La mezcla de los datos personales con los datos de la navegación también es realizada con *Pentaho.*

3.3.2 Flexibilidad en el filtrado y selección de los atributos a utilizar en la vista minable

HERMINWEB obligaba al interesado a realizar el estudio sobre el comportamiento de los usuarios con respecto al uso de su cuota de navegación sólo teniendo en cuenta ciertos atributos predeterminados en la herramienta para los estudiantes, trabajadores internos y externos y para los registros del *proxy.* Esto hacía a la aplicación poco flexible y escalable en cuanto al análisis de nuevos atributos que pudieran resultar interesantes o no, por ejemplo, no se podía descartar el sexo de los estudiantes entre los datos de interés o incluir el nivel escolar de los padres entre los que sí lo son. Con este trabajo se flexibilizó la selección de estos atributos para arrojar mejores resultados y permitirle a HERMINWEB adaptarse a las necesidades del especialista que la utilice en el contexto de la UCI. Ahora el interesado puede aplicar el filtro que considere necesario e incluir en la vista minable los atributos que le resulten interesantes. De esta forma se evita procesar todos los datos de los usuarios y realizar consultas para recopilar datos innecesarios a partir de las fuentes.

3.3.3 Reorganización de los *logs* del *proxy*

Una de las tareas fundamentales que afecta el rendimiento en cuanto a tiempo de HERMINWEB, es el procesamiento de los *logs* del *proxy* para obtener los datos de la navegación y crear las sesiones de usuarios (la media de *logs* generados en un mes es 30 GB). En este proceso es de suma importancia

acceder al contenido de cada uno de los *logs* correspondientes a los usuarios que se encuentran en la vista minable creada. Los *logs* se encuentran compartidos en un servidor web mediante el protocolo HTTP. HERMINWEB realizaba esta operación tratando de leer el contenido de cada uno de estos ficheros, ignorando el error que es lanzado en caso de que no existiera. Evidentemente esto no era eficiente ya que se estaba tratando de acceder a recursos que en la práctica podían no existir, sobrecargando al servidor de peticiones que no cumplían ningún objetivo e incrementando el tiempo para realizar el proceso.

Se analizó la alternativa de reorganizar los *logs* en el servidor web, usando para ello un acceso directo a los mismos sin cambiar la estructura de carpetas que poseen al ser almacenados. Se realizaron pruebas, que consistieron en calcular el tiempo que demoraban en leerse los *logs* pertenecientes a un número considerable de usuarios, compartidos con el servidor web Apache. Para ello se utilizaron 3 variantes, de las cuales la tercera fue la mejor.

1. Se publicó un enlace a los *logs* reales y un fichero para cada usuario donde se encontraban las rutas hasta los *logs* de los días en que habían navegado partiendo del enlace. Luego se creaba un diccionario para cada usuario, teniendo como clave el nombre de usuario y como valor las mencionadas rutas.
2. Se utilizó la que estaba anteriormente que consistía en tratar de leer los ficheros de un usuario, teniendo en cuenta que hubiese utilizado su cuenta los 31 días que puede tener un mes, existieran o no los ficheros.
3. Se publicó un enlace a los *logs* reales y un solo fichero para todos los usuarios donde se encontraban las rutas hasta los *logs* de los días en que habían navegado partiendo del enlace. Luego se creaba un solo diccionario, teniendo como claves los nombres de usuario y como valor las mencionadas rutas.

Como aspecto interesante, cabe destacar que se realizaron estas mismas pruebas con el servidor FTP *ProFTP*, pero las mismas se extendieron por más de una hora, bloqueándose incluso el proceso. Fue totalmente inefectivo.

3.3.4 Utilización de ICE

Una aplicación ICE se puede usar en entornos heterogéneos: los clientes y los servidores pueden escribirse en diferentes lenguajes de programación, pueden ejecutarse en distintos sistemas operativos y

en distintas arquitecturas, y pueden comunicarse empleando diferentes tecnologías de red. Además, el código fuente de estas aplicaciones puede portarse de manera independiente al entorno de desarrollo (33).

Como se ha mencionado anteriormente, el procesamiento de los *logs* consume una considerable cantidad de tiempo y recursos, por lo que en HERMINWEB se creó un *parser* distribuido en varios clientes para realizar esta tarea en el menor tiempo posible y consumiendo la cantidad mínima de recursos. En la iteración anterior se utilizó la librería *pp*[25] de *Python* con este propósito, pero la misma solo distribuía el trabajo cuando tenía una carga elevada; mientras no ocurriera esto, el nodo principal sería el encargado de realizar todo el procesamiento de los datos. Por tanto, en la actual investigación se decidió utilizar ICE, ya que proporciona una implementación eficiente en ancho de banda, en uso de memoria, y en carga de CPU, siendo una buena solución para este problema; además de que con su uso se garantiza que se pueda manejar cuándo y de qué forma distribuir el procesamiento de los registros en dependencia de la cantidad de nodos con las que se cuente.

Primeramente se definió la función que realiza el procesamiento de un registro y que debe ser publicada por cada nodo de procesamiento para que sea ejecutada desde el nodo central. Los módulos necesarios para esta tarea son: *urllib2, cStringIO, re, time, psycopg2, os, base64, itertools*. A continuación se almacenaron en una cola las tareas con los usuarios a procesar y se le enviaron a los nodos de procesamiento.

Los nodos de procesamiento obtienen del servidor de registros los *logs* pertenecientes al usuario solicitado, y procesan las peticiones realizadas por el usuario, creando las sesiones, que son formadas mediante el módulo *itertools*, agrupando las peticiones según los datos de la navegación que se hayan seleccionado para crear las vistas minables.

En el centro TLM se está implementando un sistema genérico basado en la arquitectura cliente-servidor que utiliza ICE para la publicación de funciones en un servidor remoto independientemente del lenguaje de programación y plataforma en que hayan sido implementadas (46). Esto posibilitaría que todo el proceso de gestión de los servidores y clientes ICE quede a cargo del mencionado sistema una vez que HERMINWEB se integre a la plataforma que se está desarrollando.

[25] del inglés: Parallel Pyhton. Librería utilizada para la programación paralela con Python.

3.4 Integración con RapidMiner

HERMINWEB utilizaba la biblioteca de algoritmos WEKA para la obtención de los modelos descriptivos de agrupamiento. Esta librería si bien es rápida en el procesamiento de la información, al no estar orientada a componentes, no permite modificar los distintos pasos para la manipulación de la vista minable, limitándose a solo variar algunos parámetros básicos de los algoritmos de minería.

En la iteración actual, para la aplicación de los algoritmos de minería de datos se empleó el entorno de algoritmos de aprendizaje Rapidminer 5.0.Con este objetivo se realizó el diseño de varias clases que se encargan de la construcción de los XML que describen los operadores y configuraciones del proceso rapidminer para la obtención del modelo a partir de la vista minable resultante después de preparar los datos. Luego de construido este XML teniendo como base además las configuraciones de la aplicación dadas por el usuario, es ejecutado el proceso por línea de comandos obteniéndose los modelos esperados.

Si bien Rapidminer 5.0 en lo que respecta a la rapidez del procesamiento de la información de la vista minable de manera general es menos rápido que WEKA; si posee mayor flexibilidad debido al uso de varios componentes para medir la calidad y mejorar el proceso de obtención de los modelos descriptivos. Además puede consultar diferentes tipos de fuentes de datos. De esta manera se eleva la calidad de los resultados, compensando el gasto en tiempo que implica el uso de esta herramienta.

3.5 Mejoras en los reportes

Dentro de las limitaciones encontradas en el reporte generado por la aplicación obtenida durante la primera iteración de HERMINWEB, se encontró que este sólo se limitaba a mostrar gráficos de barra y de sector con los modelos de agrupamiento obtenidos. Para el enriquecimiento de este reporte se agregaron al mismo otras variables como: el número de vistas minables creadas y procedas así como la fecha de creación de las mismas, los valores de los parámetros de los algoritmos de minería aplicados, el tiempo aproximado del proceso de minería y el significado de las transformaciones realizadas a los datos durante la fase de preparación. Por otro lado se incorporaron otros reportes en hojas de cálculo con los modelos obtenidos, lo que posibilita un mejor entendimiento de los resultados.

3.6 Diagrama de paquetes del diseño

El diagrama de paquetes del diseño se estructuró en concordancia con las capas propuestas en la arquitectura del sistema, por lo que el color de cada paquete está relacionado con la capa a en la que se encuentra.

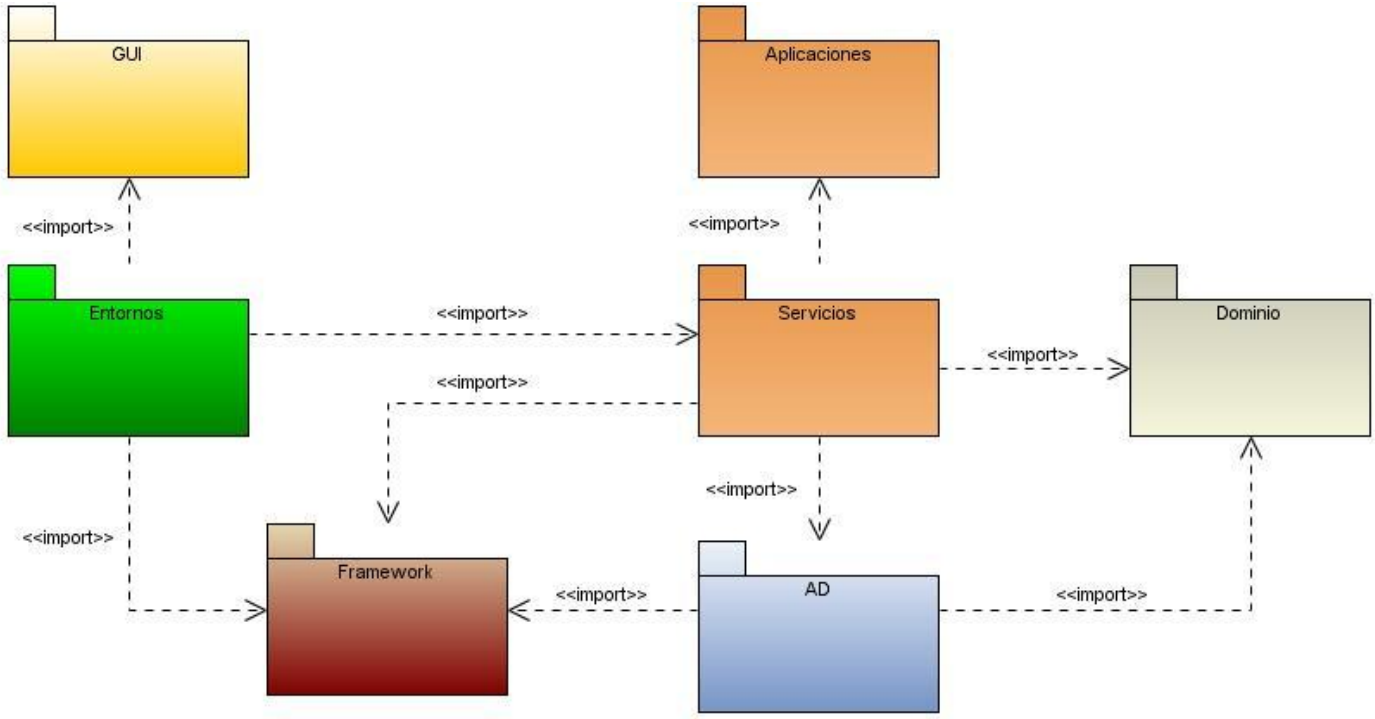


Figura 5: Diagrama de Paquetes del Diseño.

El diagrama de clases del diseño se realiza para describir las especificaciones de las clases y las interfaces de una aplicación (4). Los diagramas de clases del diseño para cada caso de uso se pueden consultar en el expediente del proyecto productivo al cual pertenece este trabajo.

3.7 Modelo Físico de Datos

Describe las representaciones físicas de los datos persistentes utilizados en la herramienta y que serán almacenados en base de datos.

3.8.1 Modelo relacional

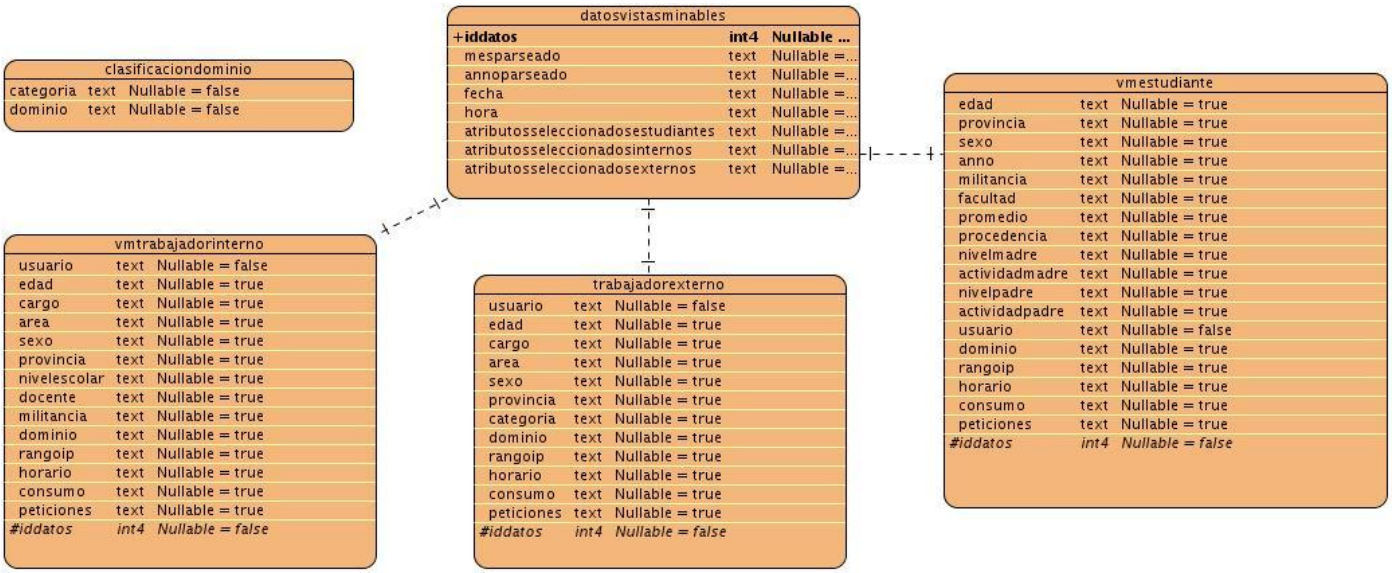


Figura 6: Modelo Físico de Datos

3.8.2 Modelo NoSQL

Las bases de datos MongoDB están constituidas por colecciones y estas a su vez por documentos en formato JSON. Una base de datos en MongoDB es un contenedor independiente de los datos, siendo este el análogo a un esquema en una base de datos relacional.

Una colección es un conjunto de documentos. Una de las diferencias clave entre MongoDB y bases de datos relacionales es que todos los documentos en una colección, no necesariamente tienen la misma estructura. Esto contrasta con una base de datos relacional donde todas las filas de una tabla deben tener la misma estructura. Un documento es un registro único de una colección que tienes varios atributos.

El modelo para MongoDB de HERMINWEB está constituido por la base de datos del mismo nombre, que a su vez contiene la colección Resultados donde se almacenan los documentos con los modelos obtenidos en cada proceso realizado.

Figura 7: Modelo de Datos NoSQL

3.8 Conclusiones del capítulo

En este capítulo fueron mostrados los diferentes diagramas de clases del diseño para los casos de uso críticos. Fueron explicados los diferentes patrones de diseño empleados que dan una mayor robustez a la arquitectura N-Capas utilizada en la herramienta, la cual brinda la posibilidad de desacoplar funcionalidades, permitiendo una mayor reutilización al separar el negocio, el acceso a datos y las interfaces de usuario, teniendo además un intermediario entre el negocio y la interfaces visuales. Esta arquitectura garantiza que no haya mezcla de código en las diferentes capas. Los patrones de diseño empleados son buenas prácticas que a lo largo de la historia han demostrado capacidades de reutilización, extensibilidad y escalabilidad. Cada uno de los patrones utilizados está bien justificados (4) y con la actual investigación se ha podido constatar que su uso ha sido efectivo, garantizando al grupo de desarrollo y a los ingenieros de software de forma general la posibilidad de extender la herramienta sin muchas contradicciones y de manera sencilla.

Se propusieron además mejoras significativas que permitirán reducir de manera considerable el tiempo de duración del proceso de preparación de los datos en HERMINWEB. Los artefactos generados en este capítulo constituyen una entrada indispensable para la implementación del sistema.

Capítulo 4

IMPLEMENTACIÓN Y PRUEBAS

4.1 Introducción

En el flujo de implementación se comienza el desarrollo del sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares; a partir de los resultados del diseño. Además se distribuye el sistema asignando componentes ejecutables a nodos en el diagrama de despliegue (28).

4.2 Diagrama de despliegue

En el diagrama de despliegue se muestra la situación física de los componentes lógicos desarrollados; en otras palabras se sitúa al software en el hardware que lo contiene (28). Típicamente se utiliza para el modelado del hardware utilizado, así como las distintas relaciones entre los componentes en forma de grafo.

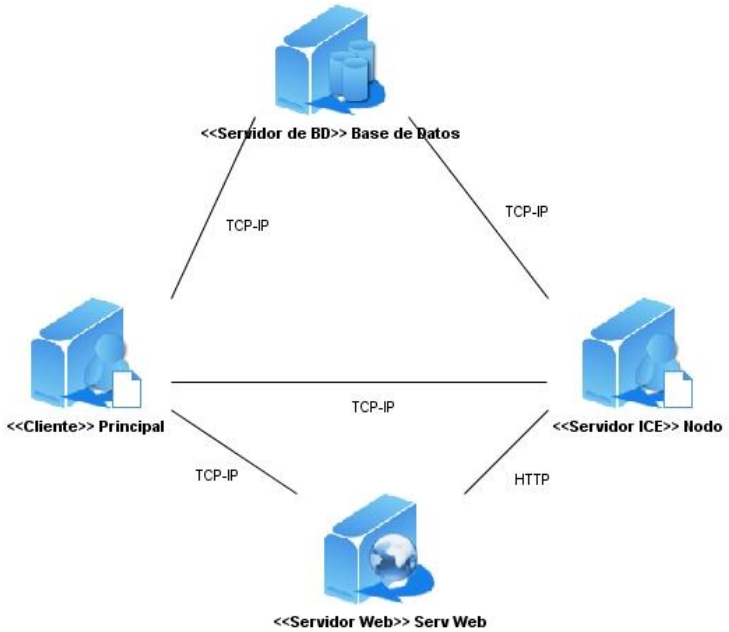


Figura 8: Diagrama de despliegue.

Nombre del Nodo	Descripción
Principal	Computadora desde la que será ejecutada la herramienta.
Base de Datos	Servidor de Base de Datos donde se alojará toda la información recopilada y creada por la herramienta.
Nodo	Nodo de procesamiento que participará en el *parseo* de los registros del *proxy.* Se utilizarán tantos nodos como se tengan disponibles.
Serv. Web	Servidor web donde serán compartidos los registros del *proxy*.

Tabla 9: Descripción de los nodos del diagrama de despliegue.

4.3 Diagramas de componentes

A continuación se muestran los diagramas de componentes asociados a los CU críticos del sistema:

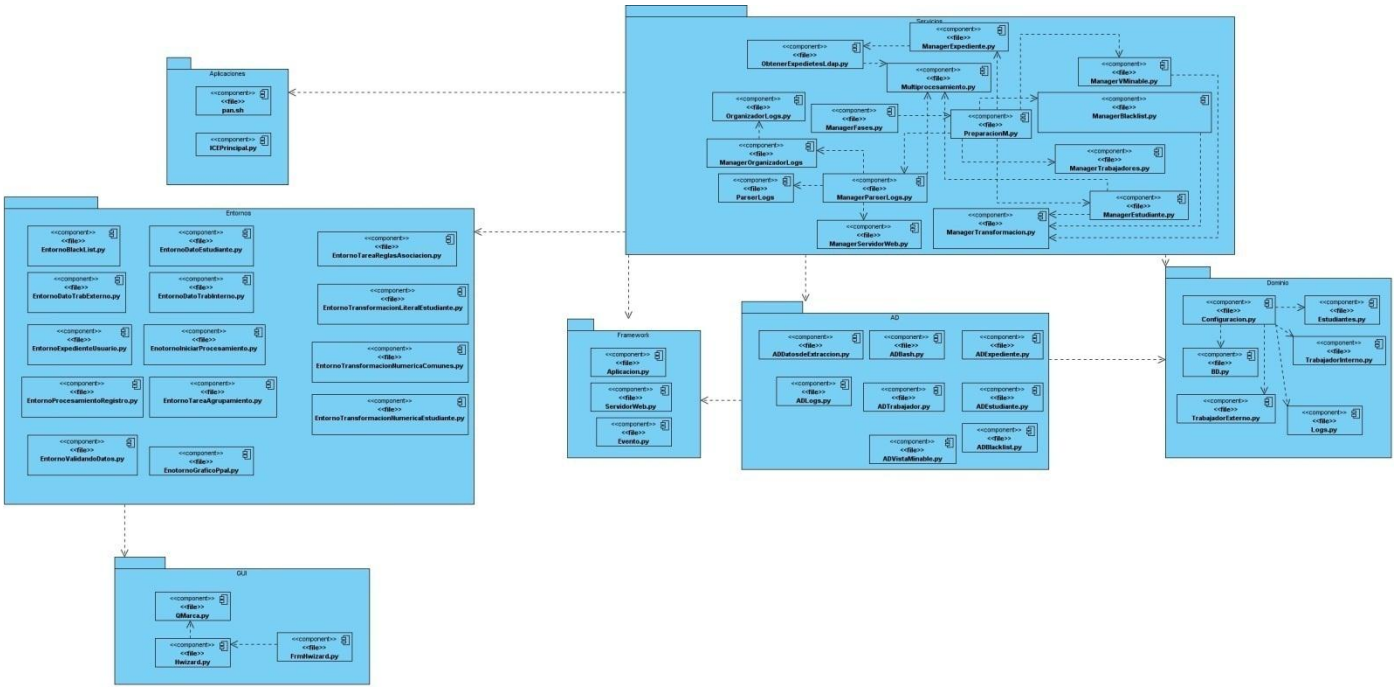


Figura 9: Diagrama de componentes del CU Preparar Datos.

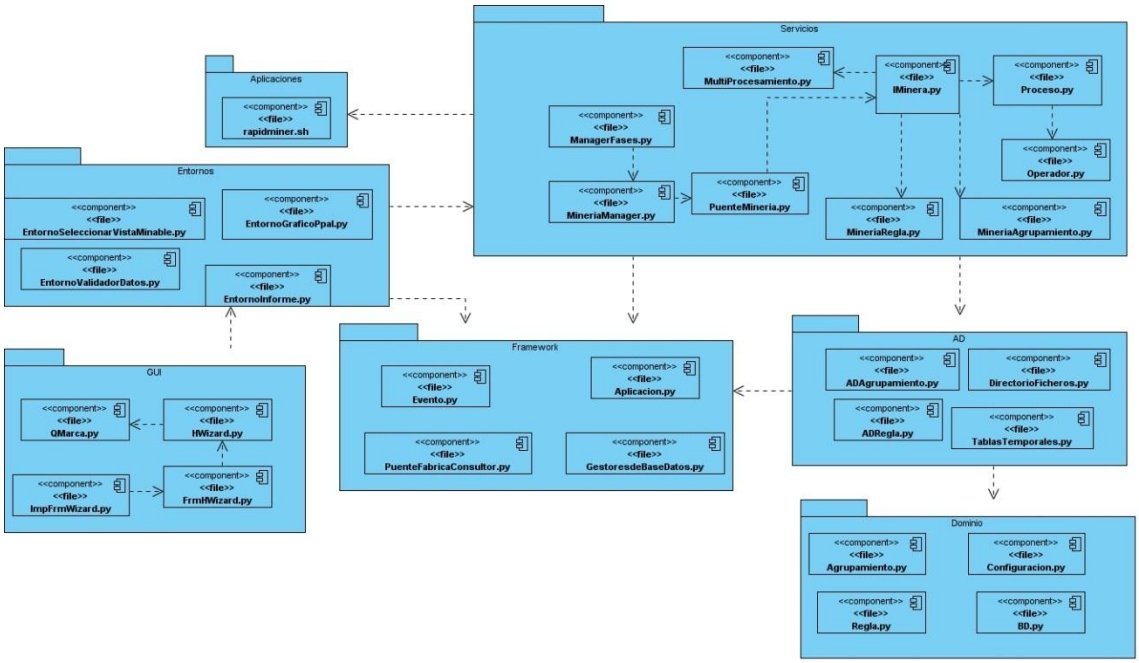


Figura 10: Diagrama de componentes del CU Aplicar Minería.

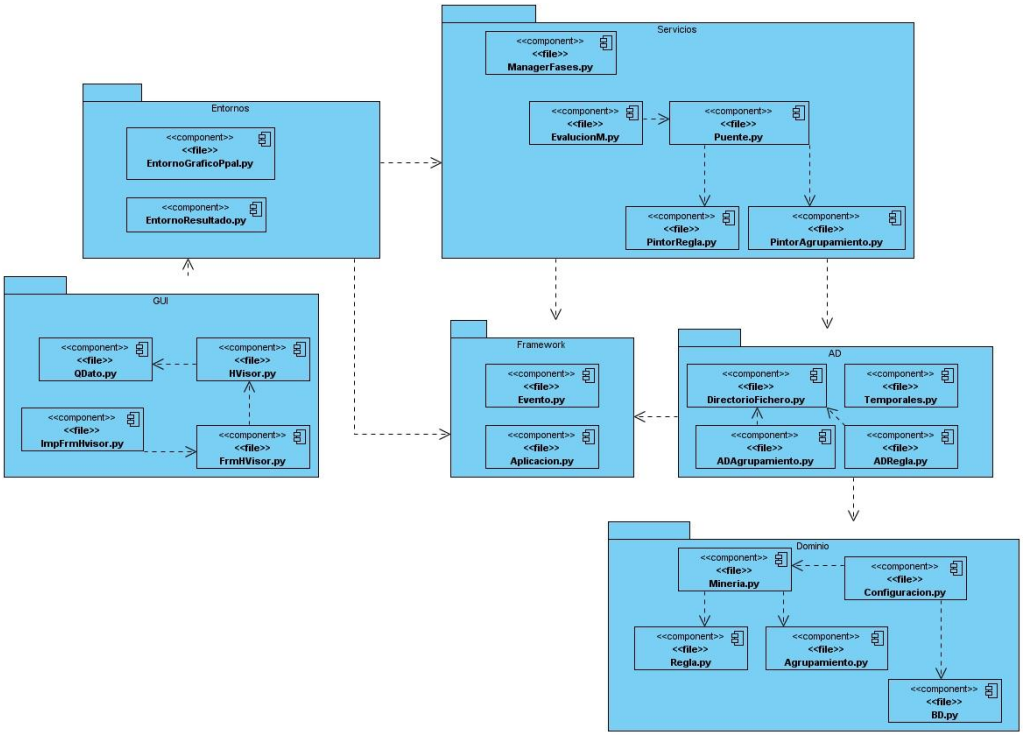


Figura 11: Diagrama de componentes del CU Evaluar Comportamiento.

4.4 Estrategia de Pruebas

La creciente inclusión del software como un elemento más de muchos sistemas y la importancia de los costos asociados a un fallo del mismo, han motivado la creación de pruebas cada vez más minuciosas y bien planificadas. Luego de la implementación de la solución y para la verificación del funcionamiento de la misma fueron realizadas las pruebas. Se definió una estrategia con niveles, tipos de pruebas, métodos y técnicas correspondientes, que permitieron solucionar errores que presentaba la aplicación y perfeccionar la solución implementada.

La estrategia elaborada hace énfasis en los niveles Integración y Sistema. Relacionados a cada nivel de pruebas están los tipos de pruebas; estos a su vez se aplican a través de métodos, para los cuales existen técnicas.

Niveles de Pruebas	Tipos de Pruebas	Métodos	Técnicas
Integración	• Funcionalidad Función	Caja Negra	Partición de equivalencia.
Sistema	• Funcionalidad Seguridad • Funcionalidad Volumen • Rendimiento Carga	Caja Negra	Partición de equivalencia.

Tabla 10: Estrategia de pruebas desarrollada

4.4.1 Caja negra

La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, o sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

Caso de Prueba Aplicar Minería: a partir del establecimiento de los valores de confianza y soporte se obtendrán los patrones en términos de reglas de asociación.

Escenario	Descripción	Soporte	Confianza	Respuesta del sistema	Flujo central
EC 1.1 – Aplicar Minería	Permite obtener reglas de asociación con valores de confianza y soporte superiores a 0.5.	V	V	Se obtuvieron reglas de asociación	En el panel de configuración se selecciona la opción de configurar los parámetros para la tarea reglas de asociación
EC 1.2 Error en los parámetros de minería	Notifica errores en los parámetros básicos para la obtención de reglas.	I	V	El sistema muestra un mensaje de error.	En el panel de configuración se selecciona la opción de configurar los parámetros para la tarea reglas de asociación
		V	I		
		I	I		

Tabla 11: Poner el nombre

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Soporte	Campo texto	No	Número de 0.01 a 1.00
2	Confianza	Campo texto	No	Número de 0.01 a 1.00

Tabla 12: Descripción de las variables

No	Cantidad de instancias analizadas	Tiempo demorado en minutos
1	513	0.5
2	1203	1.40
3	322	0.22

Tabla 13: Resultados de las Pruebas de rendimiento.

4.4.2 Entorno de pruebas

Conforme a los requisitos no funcionales que se definieron para el desarrollo de esta solución el entorno de pruebas cuenta con las características siguientes:

Hardware	**Datos**
Procesador	Dual Core CPU 2.1GHz
Memoria	1GB
Tarjeta Madre	INTEL-965
Ancho de Banda	100 Mbps
Sistema Operativo	Ubuntu 10.04
Cantidad de Servidores ICE	1

Tabla 14: Características del Entorno de pruebas

4.4.3 Análisis de los resultados

Luego de las pruebas realizadas se concluye que las mismas fueron satisfactorias. Fueron detectadas un total de 14 no conformidades clasificadas de la siguiente forma:

Total	**Alta**	**Media**	**Baja**
14	3	4	7

Tabla 15: Resultados de las pruebas.

Entre las principales no conformidades encontradas podemos relacionar:

- Problemas con el tamaño de la fuente (muy pequeña) en algunas interfaces.
- Problemas con el tamaño de fuente en los reportes generados.
- Problemas con el orden lógico de las gráficas en los reportes generados.
- Mala validación del límite inferior de los campos soporte y confianza en la interfaz de configuración de las tareas de minería.
- Problemas de ortografía (tildes) en el panel de configuración.

En cuanto a las pruebas de rendimiento realizadas los resultados obtenidos se evalúan como buenos, teniendo en cuenta el elevado volumen de datos que maneja la aplicación. Las no conformidades encontradas fueron resueltas.

4.5 Conclusiones del capítulo

En el presente capítulo se mostró el modelo de implementación de la solución basados en sus diagramas de componentes fundamentales, así como el diagrama de despliegue. Se hizo especial énfasis en los subsistemas asociados a los CU críticos del sistema.

Por otro lado se expuso la estrategia de pruebas definida para verificar las funcionalidades de la solución. Durante la realización de pruebas de funcionalidad función se detectaron 14 no conformidades que fueron resueltas. Además fueron evaluados como buenos los resultados en las pruebas de carga teniendo en cuenta el elevado número de datos que maneja la aplicación.

CONCLUSIONES

Luego del proceso desarrollado se implementaron nuevas funcionalidades a HERMINWEB, que permiten realizar la tarea descriptiva Reglas de Asociación en el estudio del uso de las cuotas de navegación en *Internet* por parte de los usuarios de la UCI.

El estudio de las tendencias actuales sobre Minería de Datos, Minería Web, Minería de uso Web y la tarea descriptiva Reglas de Asociación contribuyó de manera decisiva al enriquecimiento de la herramienta. Entre las aplicaciones más utilizadas en la Minería de uso Web se encontraron como limitaciones que en la mayoría de los casos estas herramientas están orientadas a la minería sobre registros de servidores web y no integran el análisis sobre la mezcla obtenida de la información de la navegación y la información personal de los usuarios.

Entre las limitaciones encontradas en la primera iteración de HERMINWEB figuran:

- Poca escalabilidad y flexibilidad en la fase Preparación de Datos.
- Pobre rendimiento en cuanto a velocidad durante la fase de Preparación de Datos.
- Consulta innecesaria a las fuentes de datos.
- Poca flexibilidad en los pasos del proceso de Minería de Datos.
- Poca información en el reporte a difundir del proceso realizado.

En aras de dar una solución a las limitaciones detectadas en HERMINWEB y cumplir el objetivo de la presente investigación se elaboraron los Modelos de Sistema y Diseño para la actual iteración de la aplicación. En busca de imprimir mayor escalabilidad y flexibilidad en la fase de Preparación de Datos hoy se le brinda al usuario la posibilidad establecer sus propias discretizaciones creándose otras opciones de filtrado para los atributos de la vista minable. Para aumentar la velocidad de la extracción, transformación y carga de los datos para la confección de la vista minable se integró a HERMINWEB la *ETL Kettle* de la *Suite Pentaho BI*. Para disminuir las consultas innecesarias a las fuentes de datos se ubicó como primer proceso en la implementación de KDD, la selección de los datos. Por otro lado para evitar las excesivas peticiones fallidas al servidor web donde están publicados los registros de navegación, estos fueron reorganizados sin alterar la estructura de carpetas que poseen al ser almacenados.

Buscando mayor flexibilidad en los pasos del proceso de Minería de Datos se sustituyó la librería WEKA por el entorno de algoritmos de aprendizaje RapidMiner.

Para el enriquecimiento de los reportes dados por la herramienta se añadieron otros datos al mismo como el valor de los parámetros de los algoritmos de minería utilizados en el proceso, significado de las transformaciones realizadas a los datos durante la fase de preparación y el tiempo que demoró la ejecución del proceso de minería. Además se elaboraron reportes con los modelos obtenidos en hojas de cálculo, para un mejor entendimiento de los resultados.

Hoy HERMINWEB es una herramienta que automatiza un proceso de KDD, aplicado a los registros de navegación por Internet almacenados por el servidor proxy, utilizando tecnologías libres, apoyado en la Minería de Uso de la Web, donde se obtienen modelos en términos de las tareas Agrupamiento y Reglas de Asociación.

RECOMENDACIONES

Diseñar e implementar un *Datawarehouse* para el almacenamiento de los datos durante la fase de preparación de datos. Esto permitirá una mejor organización de los datos y facilitará la construcción de la vista minable agilizando el tiempo que demora la realización de esta fase.

Implementar un módulo para gestión de resultados. Con la implementación de un módulo de gestión resultados en la herramienta se podrán visualizar los resultados obtenidos de todos los procesos ejecutados sin que se haga necesaria la ejecución de las fases de minería y preparación de datos.

Añadir la tarea predictiva clasificación en aras de enriquecer la información dada por la herramienta a la DRSI de la UCI.

Implementar un módulo para la gestión de las clasificaciones de los dominios, buscando la mejora este proceso.

Para aumentar las vías de difusión de los resultados se recomienda el estudio de la integración de la herramienta con plataformas como Asterisk, permitiendo el envío de avisos a través de mensajes telefónicos.

Estudiar el lanzamiento de nuevas versiones de las herramientas que se integran en HERMINWEB tales como: MongoDB, PostgreSQL, RapidMiner, y Pentaho en aras de incorporar nuevas funcionalidades de las mismas que puedan mejorar el funcionamiento de la herramienta.

BIBLIOGRAFÍA

1. **Miralles Aguiñiga, Marcel y Laporta, Jorge Lázaro.** *Fundamentos de Telemática.* Valencia : Editorial de la Universidad Politécnica de Valencia, 2005.
2. **Hernández Orallo, José, Ramírez Quintana, María José y Ferri Ramírez, César.** *Introducción a la Minería de Datos.* Madrid : Pearson Prentice Hall, 2004.
3. **Jorba Esteve, Josep y Suppi Boldrito, Remo.** *Administración Avanzada de GNU/Linux.* Cataluña : s.n., 2004.
4. **Ordoñes Leyva, Yoanni y Avilés Vázquez, Ernesto.** *Herramienta informática de Minería de Uso de la Web sobre los registros de navegación por Internet.* Universidad de las Ciencias Informáticas. Ciudad de La Habana : s.n., 2010.
5. **Xu, Rui y Wunsch, Donald C.** *Clustering.* New Jersey : IEEE Press, 2009.
6. **Pérez Martinto, Pedro Carlos.** *El diseño metodológico de la investigación científica. Teoría de muestreo: población y muestra. Diseño experimental y métodos.* Entorno Virtual de Aprendizaje, Universidad de las Ciencias Informáticas. 2010.
7. **Han, Jiawei y Kamber, Micheline.** *Data Mining. Concepts and Techniques.* San Francisco : Morgan Kaufmann Publishers, 2006.
8. **Brito Sarasa, Raycos.** *Minería de Datos aplicada a la Gestión Docente del Instituto Superior Politécnico José Antonio Echeverría.* Instituto Superior Politécnico José Antonio Echeverría. Ciudad de La Habana : s.n., 2008.
9. **Molina López, José Manuel y García Herrero, Jesús.** *Técnicas de Análisis de Datos. Aplicaciones Prácticas utilizando Microsoft Excel y WEKA.* Universidad Carlos III de Madrid. Madrid : s.n., 2006.
10. **Chapman, Pete, y otros, y otros.** *CRISP-DM 1.0: Step-by-step data mining guide.* s.l. : SPSS Inc., 2000.
11. **De Gyves Camacho, Francisco Manuel.** *Web Mining: Fundamentos Básicos.* Universidad de Salamanca. Salamanca : s.n.
12. **Medina Pagola, José E.** *Estado del Arte del Web Mining.* Centro de Aplicaciones de Tecnología de Avanzada. Ciudad de La Habana : s.n., 2005.
13. *Escavando la web.* **Baeza-Yates, Ricardo.** 1, 2004, El profesional de la información, Vol. 13.
14. Sitio Oficial de Lyris HQ. [En línea] 2011. [Citado el: 27 de octubre de 2010.] [Disponible en: http://www.lyris.com].
15. I-Soft. [En línea] 2011. [Citado el: 27 de octubre de 2010.] [Disponible en: http://alice-soft.com/html/prod_amadea_en.htm].
16. Sitio Oficial de 123LogAnalyser. [En línea] 2011. [Citado el: 27 de octubre de 2010.]
17. Sitio Oficial de WebTrends . [En línea] 2011. [Citado el: 27 de octubre de 2010.] [Disponible en http://www.webtrends.com].
18. Sitio Oficial de AlterWind. [En línea] 2011. [Citado el: 27 de octubre de 2010.] [Disponible en http://www.alterwind.com].
19. Sitio Oficial de Analog. [En línea] 2011. [Citado el: 27 de octubre de 2010.] [Disponible en http://www.analog.com].
20. Sitio oficial de Htminer. [En línea] 2011. [Citado el: 27 de octubre de 2010.] [Dsiponible en: http://www.htminer.org].

21. **Bazhenova, Natalia.** VISITaTOR. [En línea] 2005. [Citado el: 27 de octubre de 2010.] [Disponible en: http://www.fh54.de/visitator/home/1About/index.php].
22. **RAE.** Real Academia Española. [En línea] 2011. [Citado el: 16 de diciembre de 2010.]
23. **SAS Institute Inc.** SAS. [En línea] 2011. [Citado el: 16 de diciembre de 2010.] [Disponible en: http://www.sas.com/offices/europe/uk/technologies/analytics/datamining/miner/semma.html].
24. **Martínez de Pisón Ascacibar, F. J.** *Optimización mediante técnicas de minería de datos del ciclo de recocido de una línea de galvanizado.* Universidad de La Rioja. La Rioja : s.n., 2003.
25. XpertRule. [En línea] [Citado el: 16 de diciembre de 2010.] http://www.xpertrule.com/pages/critikal.htm.
26. **Castillo Suárez, Mayrilis y Sablón Marrero, Daymi.** *Teleidentificador Personal:Diseño de la arquitectura de la plataforma manejadora de peticiones.* Universidad de las Ciencias Informáticas. Ciudad de La Habana : s.n., 2009.
27. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software.* s.l. : Pearson Adisson-Wesley, 2000.
28. **Object Management Group.** OMG, Business Process Model and Notation (BPMN). [En línea] 2011. [Citado el: 8 de diciembre de 2010.] [Disponible en: http://www.omg.org].
29. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Lenguaje Unificado de Modelado. Manual de Referencia.* s.l. : Pearson Adisson-Wesley, 2000.
30. **Alonso Riverón, Yisel, Cruz Navarro, Yaneisy y Tornés Medina, Yordanis.** *IDEF. Una alternativa para modeamiento de negocio con RUP.* Universidad de las Ciencias Informáticas. Ciudad de La Habana : s.n., 2008.
31. **Python Software Foundation.** Sitio Oficial de Python. [En línea] 2011. [Citado el: 18 de noviembre de 2010.] [Disponible en: http://www.python.org/].
32. **Vallejo Fernández, David.** *Documentación de ZeroC ICE.* Universidad de Castilla-La Mancha. Castilla : s.n., 2006.
33. PyTables. [En línea] 2011. [Citado el: 18 de noviembre de 2010.] [Disponible en: http://www.pytables.org].
34. **Visual Paradigm.** Sitio Oficial de Visual Paradigm. [En línea] 2011. [Citado el: 9 de noviembre de 2010.] [Disponible en: http://www.visual-paradigm.com].
35. **Eclipse.** Sitio Oficial de Eclipse. [En línea] 2011. [Citado el: 9 de noviembre de 2010.] [Disponible en: http://www.eclipse.org].
36. **PosgreSQL-es.** Portal en español de PostgreSQL. [En línea] 2011. [Citado el: 9 de noviembre de 2010.] [Disponible en: http://www.postgresql-es.org].
37. **pgAdmin.** Sitio Oficial de pgAdmin. [En línea] 2011. [Citado el: 9 de noviembre de 2010.] [Disponible en: http://www.pgadmin.org].
38. Sitio Oficial de MongoDB. [En línea] 2011. [Citado el: 20 de mayo de 2011.] [Dsiponible en: http://www.mongodb.org].
39. Sitio Oficial de RapidMiner. [En línea] 2011. [Citado el: 20 de diciembre de 2010.] [Disponible en: http://www.rapidminer.com].
40. **Witten, Ian H. y Frank, Eibe.** *Data Mining: Practical Machine Learning Tools and Techniques.* San Diego : Morgan Kaufmann Publishers, 2005.
41. **Pentaho Company.** Sitio Oficial Pentaho. [En línea] 2011. [Citado el: 15 de Febrero de 2011.]

[Disponible en http://www.pentaho.com/].
42. **Larman, Craig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* 1999.
43. **Carbonell Marcé, Alberto Carlos y Morciego González, Carlos Manuel.** *Sistema Distribuido Cliente-Servidor.* Universidad de las Ciencias Informáticas. Ciudad de La Habana : s.n., 2011.
44. *Educational Data Mining: A Survey from 1995 to 2005.* **Romero, Cristóbal y Ventura Soto, Sebastián.** 1, 2007, Expert Systems with Applications, Vol. 33.
45. **Nodal González, Ricelda y Ortiz Nodal, Dunia.** *Paquete de instalación de réplicas en servidores PostgreSQL.* Universidad de las Ciencias Informáticas. Ciudad de La Habana : s.n., 2009.
46. **Mobasher, Bamshad y Liu, Bing.** *Web Data Mining: Exploring Hyperlinks, Contents, and Usage.Data (Data-Centric Systems and Applications) (Hardcover).* Chicago : Springer, 2007.
47. **Jain, Anil K.** *Data Clustering: 50 Years Beyond K-Means.* Universidad del Estado de Michigan. Michigan : s.n., 2009.
48. **Herrero Núñez, Julio Alberto.** Minería de Textos Web. Recuperación y organización de la información. [En línea] 2007. [Citado el: 8 de enero de 2011.] [Disponible en: http://mineria-textos-web.awardspace.com].
49. **Berkhin, Pavel.** *A Survey of Clustering Data Mining Techniques.* s.l. : Accrue Software Inc., 2002.
50. **Arias Londoño, Alexander y Ovalle Carranza, Demetrio A.** *Web Usage Mining: Revisión del Estado del Arte.* Universidad Nacional de Colombia. Medellín : s.n.
51. **Oracle.** Sitio Oficial de Java. [En línea] 2011. [Citado el: 8 de noviembre de 2010.] [Disponible en: http://www.java.com].