

HERMINWEB: Herramienta de Minería de Uso de la Web Aplicado a los Registros del Proxy

Yoanni Ordoñez Leyva¹ y Darian Horacio Grass Boada¹

¹Centro de Telemática, Universidad de las Ciencias Informáticas, Cuba. {yordones, dgrass}@uci.cu

RESUMEN

En este trabajo se presenta una herramienta capaz de descubrir parte de la información oculta en los registros de navegación (logs) generados por los servidores proxy que dan acceso a Internet a los usuarios de la Universidad de las Ciencias Informáticas (UCI). Esta herramienta es de utilidad para la Dirección de Redes y Seguridad Informática (DRSI) pues le brinda información necesaria para la toma de decisiones. Apoyado en la Minería de Datos que provee las técnicas para la obtención de patrones ocultos en un alto volumen de información. La herramienta automatiza un proceso de Descubrimiento de Conocimientos en Base de Datos (KDD). Aplica la tarea de Agrupamiento con el fin de encontrar clases de usuarios en el uso de las cuotas de navegación por Internet de la UCI. Se integran dos tecnologías con características diferentes (JAVA y Python). Se realiza un procesamiento distribuido de los registros de navegación mejorando considerablemente el tiempo de la preparación de los datos. Con un diseño jerárquico para almacenar la información procesada se logra un mejor rendimiento de las consultas a la base de datos. Se pudo integrar varias fuentes de información con los registros de un servidor en la UCI.

Palabras claves: Integración de Tecnologías, Minería de Datos, Programación Distribuida, Toma de Decisiones.

ABSTRACT

This paper shows software able to discover part of hidden information on Internet logs of proxy server. This server permits browsing on Internet to users of the University of Informatics Sciences (UCI). It gives useful information for making decisions to Networks and Information Security Management (DRSI). It is supported by Data Mining who gives the techniques to obtain useful patterns in a huge amount of information. It carries out a process of Knowledge Discovery in Databases (KDD). It was written in Python. In UCI each user has a quote to browse on Internet. This application supports clustering task to find clusters on use of browsing quotes. Furthermore, it was developed a distributed module to prepare logs. It was integrated two different technologies (JAVA and Python). With a hierarchical design of information it makes a better performance in database queries.

Keywords: Data Mining, Distributed Programming, Making Decisions, Technologies Integration.

1. INTRODUCCIÓN

El cursar de los años y el desarrollo de las tecnologías ha generado un crecimiento considerable de datos (Hernández Orallo, Ramírez Quintana, & Ferri Ramírez, 2004). El uso de esta información para el perfeccionamiento de las empresas e instituciones crea la necesidad del desarrollo de nuevas técnicas y herramientas para analizar esta enorme cantidad de datos (Olmos Pineda & González Bernal, 2007). El análisis de datos es una tarea que consiste en buscar o encontrar tendencias o variaciones de comportamiento en los mismos, de tal manera que esta información resulte de utilidad para los usuarios finales. A estas tendencias o variaciones se le conocen como patrón, los cuales si son de importancia y útiles para el dominio en cuestión se le denomina conocimiento (Olmos Pineda & González Bernal, 2007).

Una de las técnicas para el análisis de datos es la Minería de Datos. Larose plantea que la Minería de Datos: "... es el proceso de descubrir nuevas correlaciones significativas, patrones y tendencias ocultas a través de grandes cantidades de datos almacenados en los repositorios, utilizando tecnologías de reconocimiento de patrones, así como técnicas estadísticas y matemáticas" (Larose, 2005).

Uno de los escenarios donde se ha aplicado la Minería de Datos es la World Wide Web (WWW), su acelerado crecimiento y la competencia entre las organizaciones ha traído la necesidad de mejorar la calidad de los sitios web, utilizando como base el comportamiento de los usuarios que lo usan. Se le ha denominado Minería Web (MW) al descubrimiento de información útil en la WWW. La MW posee varias clasificaciones atendiendo al contenido que se analiza, una de ellas es la Minería de Uso de la Web (Web Usage Mining: WUM) centrada en el análisis de los logs o registros de navegación.

Al navegar por la Web se dejan rastros o registros de navegación en los servidores donde están hospedados los sitios y en los que permiten el acceso a Internet. Los servidores proxy encargados de gestionar el acceso a Internet de algunas instituciones generan un alto volumen de registros de navegación, que archivan la navegación de un grupo de usuarios de una determinada organización. La Universidad de las Ciencias Informáticas (UCI) cuenta con un servicio de navegación por Internet para miles de usuarios, la Dirección de Redes y Seguridad Informática (DRSI) no posee la capacidad de analizar el alto volumen de información presente en los logs del proxy en busca de patrones que identifiquen el uso de las cuotas de navegación por parte de los usuarios de la institución, dificultando la toma de decisiones.

El análisis de grandes volúmenes de datos es engorroso realizarlo manualmente, siendo necesaria su automatización. Existe un proceso definido para el análisis de datos con el fin de encontrar patrones en grandes cantidades de datos llamado Descubrimiento de Conocimiento en Bases de Datos (Knowledge Discovery in Databases: KDD). El presente trabajo muestra el desarrollo de un sistema, encargado de automatizar este proceso para el escenario de los registros de navegación por Internet de la UCI (Ordoñez Leyva & Avilés Vázquez, 2010). Se utilizan las metodologías CRISP-DM (Chapman, y otros, 2000) para guiar el proceso de Minería de Datos y RUP (Jacobson, Booch, & Rumbaugh, 2000) para el desarrollo de la aplicación, además se usan los algoritmos de MD de la biblioteca WEKA.

2. MATERIALES Y MÉTODOS

2.1 EL PROCESO DE DESCUBRIR CONOCIMIENTO EN BASE DE DATOS

El término Minería de Datos en muchas ocasiones se utiliza como sinónimo con el de Descubrimiento de Conocimiento en Bases de Datos, siendo en realidad la Minería de Datos una de las fases de las que está compuesto el proceso KDD, según la metodología es la llamada Modelado. La Figure 1 muestra las fases definidas por la metodología CRISP-DM para un proceso KDD (Chapman, y otros, 2000):

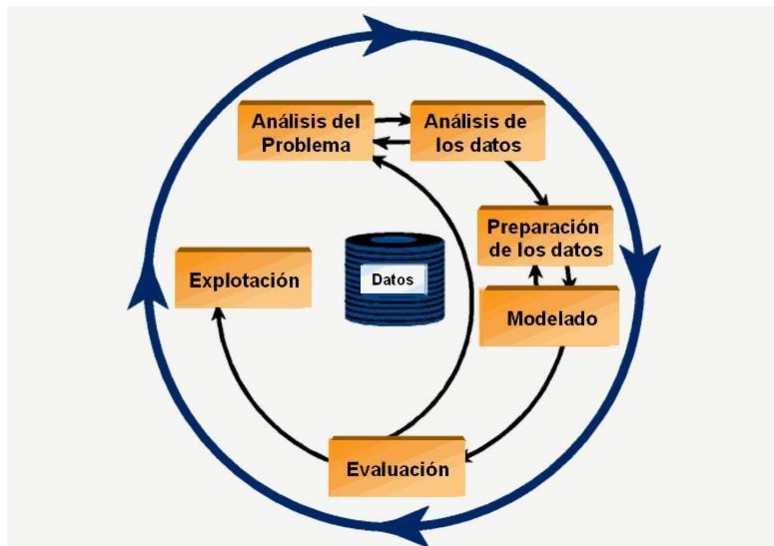


Figure 1: Fases del proceso KDD según la metodología CRISP-DM

Primeramente se debe estudiar el problema que se desea resolver, es imprescindible comprender a profundidad el dominio donde se desenvuelve la investigación centrando el análisis en las necesidades de la organización, así como para definir y priorizar los objetivos del negocio. En la *Preparación de Datos* se determinan las fuentes de información que pueden ser útiles; se transforman los datos a un formato común, además se detectan y se resuelven las inconsistencias presentes en los mismos. Posteriormente se eliminan o corrigen los datos incorrectos, decidiéndose la estrategia a seguir con los datos incompletos; además, se consideran únicamente aquellos atributos que van a ser relevantes (Hernández Orallo, Ramírez Quintana, & Ferri Ramírez, 2004).

En la fase *Modelado* se aplica el modelo, la tarea, la técnica y el algoritmo seleccionado para la obtención de reglas y patrones. Luego en la fase de *Evaluación* se comprueban los patrones y se analizan por expertos, se puede regresar a la fase *Análisis del Problema* en caso de querer perfeccionar los resultados. Finalmente, en la fase de *Explotación* se comparte el nuevo conocimiento con los interesados. Las fases que componen el KDD hacen que su desarrollo sea un proceso iterativo e interactivo con el usuario (Hernández Orallo, Ramírez Quintana, & Ferri Ramírez, 2004).

2.2 MINERÍA DE DATOS APLICADA A LOS REGISTROS DE NAVEGACIÓN EN LA UCI

La UCI cuenta con un servicio de navegación por Internet para miles de usuarios. Para ello cuenta con servidores proxy que gestionan todo el flujo de peticiones realizadas. Los sistemas actualmente instalados y en explotación (Martín Alvarez & García Martínez, 2007) no cubren todo el conocimiento implícito en estos logs, dificultando la toma de decisiones a la DRSI.

Por otra parte se cuenta con sistemas de gestión de información de los trabajadores y académica de los estudiantes. Esta información en conjunto con los registros de navegación es de mucha utilidad e interés para la DRSI, con ella se pueden encontrar patrones que describan el uso de las cuotas de navegación de los diferentes usuarios de la institución.

Para extraer esta información es necesario desarrollar una herramienta capaz de: mezclar los datos registrados por el servidor proxy con los contenidos en los sistemas de los trabajadores y estudiantes (Ordoñez Leyva & Avilés Vázquez, 2010); extraer patrones descriptivos presentes en los datos, enfocados en la tarea de agrupamiento (Hernández Orallo, Ramírez Quintana, & Ferri Ramírez, 2004) con el fin de encontrar clases de usuarios que se comporten de manera similar en el uso de las cuotas de navegación por Internet para ayudar a la toma de decisiones de la DRSI. La herramienta forma parte de una Plataforma de Gestión de Servicios Telemáticos

(PGST) desarrollada en el Centro de Telemática de la UCI (Pérez Hurtado & Padilla Moya, 2010) que es escrita en Python.

2.3 PROCESOS REALIZADOS POR LA APLICACIÓN

La aplicación posee una arquitectura de 4 Capas: la interfaz de usuario, *GUI*; *Servicios*, donde se realiza el negocio del sistema; la encargada de las acciones para comunicar el negocio con la interfaz de usuario llamada *Entorno de Ejecución*; y el *Acceso a Datos* para obtener y almacenar los datos.

La herramienta realiza cada una de las fases descritas en la metodología CRISP-DM. La aplicación es un flujo de procesos. Entre los procesos realizados por la herramienta se encuentran los correspondientes a las fases de: *Preparación de Datos*, *Modelado*, *Evaluación* y *Explotación* de los resultados. Los procesos fueron modelados con la notación BPMN (OM, 2009) por las facilidades de comprensión que brinda para flujos de procesos. Se muestran en la Figure 2 los procesos principales realizados por la herramienta.

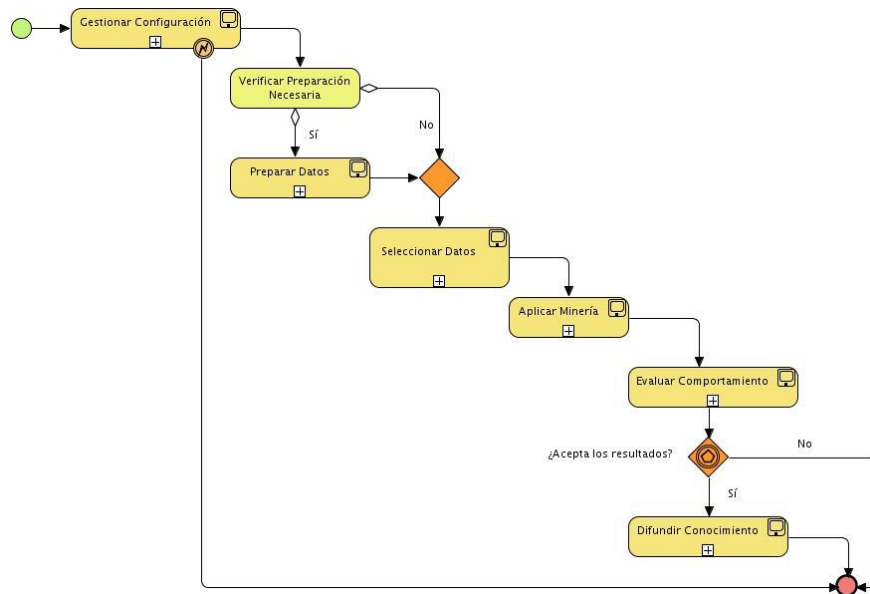


Figure 2: Procesos Generales de HERMINWEB

El proceso *Gestionar Configuración* es el encargado de obtener todas las configuraciones necesarias para la ejecución de la herramienta. Tiene dos modos de ejecución: consola y gráfica. El modo consola permite programar la ejecución de la herramienta con una configuración predefinida, mientras, en el modo gráfico es un modo interactivo con el usuario.

El proceso *Verificar Preparación Necesaria* consulta si se realizará la preparación de los datos, esta opción es necesaria cuando se desea estudiar varios grupos de usuarios en un mismo período de navegación, siendo necesario preparar los datos sólo una vez. *Preparar Datos* es una de las tareas que más procesamiento computacional necesita, pues analiza un volumen elevado de información (la media de logs generados en un mes es 30 GigaBytes). Por esta razón, no es óptimo realizar la preparación varias veces para los mismos datos. En esta fase se procesan los logs del servidor proxy y se integra con los datos de los usuarios.

Es necesario *Seleccionar Datos* pues las computadoras donde se realiza la investigación no poseen altas prestaciones y para manejar un alto volumen de información es necesario la selección de un subconjunto de los registros procesados, además puede ser de interés analizar un tipo de usuarios específicos. El corazón de un proceso KDD es la fase de Minería, desarrollada en la aplicación mediante *Aplicar Minería*, donde se aplican los

algoritmos a los datos seleccionados en el proceso anterior, dando como resultados los grupos o clústeres encontrados, los cuales serán almacenados para su posterior uso.

En ocasiones los resultados se querrán refinar, por tanto, el proceso *Evaluar Comportamiento* brinda la posibilidad de especificar si quiere persistir los resultados. En el proceso *Difundir Conocimiento*, es posible la creación de un reporte en formato portable (PDF) y el envío por correo electrónico a los interesados, terminando con este proceso la ejecución de la herramienta.

2.4 PREPARACIÓN DE DATOS

Para la realización de la preparación de datos se utilizaron varias fuentes con el objetivo de unir los datos de la navegación con los datos de los usuarios para así obtener patrones descriptivos del uso de las cuotas de navegación, dichas fuentes son las siguientes:

- Los registros del servidor proxy.
- Los sistemas de recursos humanos pertenecientes a los trabajadores (son dos servicios web uno para los trabajadores de la universidad y otro para los que brindan servicios en ella, en este documento se tratarán como trabajadores internos y externos respectivamente).
- El sistema de gestión académica estudiantil llamado *Akados*.

Los datos de los trabajadores son obtenidos mediante servicios web y los pertenecientes a los estudiantes accediendo directamente a la base de datos, en este caso Microsoft SQLServer. Al terminar este proceso de preparación de datos se tienen los datos listos para ser analizados por la fase de Minería.

El proxy identifica el propietario de las peticiones mediante el nombre del usuario, pero los sistemas de la Universidad usan un número de expediente. Lo que implica que sea necesario obtener el expediente consultando el servicio de directorio activo LDAP. Una vez obtenidos los expedientes de los usuarios a analizar, se consultan los sistemas de estudiantes y trabajadores.

Los registros de navegación son publicados mediante el servidor web Apache con el fin de que se pueda acceder desde cualquier lugar. Una de las necesidades para la preparación es la clasificación de las páginas web, en la Universidad existe una llamada FILPACON pero tiene insuficientes páginas registradas, decidiéndose utilizar *BlackList* que es la usada por la DRSI.

2.4.1 TRANSFORMACIONES REALIZADAS

En las investigaciones encontradas sobre Minería del Uso de la Web se define un concepto de sesión que no es aplicable a la presente investigación porque casi todos están centrados en el análisis a los registros de los servidores web. Siendo necesaria la definición de sesión que aportara mayor información en este entorno porque no es de interés conocer cada paso de los usuarios cuando navegan por un sitio determinado sino solamente que navegó por dicho sitio.

Se realizaron varias transformaciones a los datos seleccionados. Una de ellas fue etiquetar las direcciones de IP a un formato más agrupador. En este caso se convirtieron a una etiqueta que identifica el edificio desde el cual se realizó la petición, apoyándose en el diseño de la red de la Universidad.

A continuación se presenta una petición real realizada al servidor proxy, los campos que están marcados con X no se usarán en este documento.

El orden en que aparecen son: tiempo de la petición, dirección IP, bytes consumidos, recurso solicitado, nombre del usuario. El tiempo de la petición representa la hora 1/11/2010 12:59:11. Este campo se transforma de acuerdo a rangos de horario que se usan en los sistemas de redes de la Universidad los cuales son: "Madrugada" (2-7), "Mañana" (8-11), "Tarde" (12-19) y "Noche" (20-1). En el caso de esta petición sería transformado a "Tarde", esta transformación hace que el campo horario sea mucho más descriptivo para encontrar clases de usuarios.

La dirección IP sería transformada en este caso a "Docente 5". La lista de clasificaciones de páginas web *BlackList* clasifica sólo los dominios, por esta razón solo se deja el dominio de la petición, en el caso de ejemplo sería "www.uci.cu". Los bytes consumidos se transforman en el momento de crear las sesiones, siendo la suma de los consumos de cada una de las peticiones que formen parte de la sesión. Esta suma se convierte a "Poco", "Normal" o "Mucho".

La sesión fue definida como el grupo de peticiones de un usuario determinado a un sitio web desde un mismo edificio, en el mismo día y horario.

Si se tienen las siguientes peticiones:

- 1- "Tarde" "Docente 1" 5942 "www.uci.cu" yordones
- 2 - "Tarde" "Docente 1" 15628 "www.cuba.cu" yordones
- 3 - "Noche" "Docente 1" 2639 "www.uci.cu" yordones
- 4- "Mañana" "Docente 5" 569832 "www.uci.cu" jfcruz
- 5- "Mañana" "Docente 5" 593258 "www.uci.cu" jfcruz

Con estas peticiones se crearían 4 sesiones, las cuales serían: 1- (1), 2- (2), 3- (3), 4- (4,5).

En el momento de crear las sesiones se agrega un nuevo campo llamado "peticiones" que almacena la cantidad de peticiones que conforman la sesión. En este ejemplo la sesión 4 tendría 2 peticiones y las demás 1. Este campo es transformado a las mismas etiquetas que el campo consumo, los rangos de las etiquetas se definen mediante una exploración de los datos a analizar, decidiendo los valores para cada una de las etiquetas, esto también es aplicado al campo consumo.

2.4.2 DISEÑO DE LA BASE DE DATOS

HERMINWEB usa una base de datos relacional del gestor PostgreSQL en su versión 8.4. Esta investigación fue realizada usando computadoras con microprocesadores Intel-695 a 2.2 GHz, Core 2 Duo, con 1 GB de memoria RAM. Al contar con muchas peticiones (alrededor de 130 millones en un mes) las consultas realizadas a la base de datos demoraban mucho tiempo. Al crear las sesiones de los usuarios se redujeron considerablemente la cantidad de datos. Se definió una estructura jerárquica para el almacenamiento de las sesiones con el objetivo de mejorar el tiempo de respuesta del servidor de base de datos. Los estudiantes fueron divididos en 7 tablas, una tabla para cada Facultad de la Universidad. En cuanto los trabajadores internos se dividieron en trabajadores que dan clases y trabajadores que no. Los trabajadores externos se almacenan en una sola tabla por ser pocos.

2.4.3 PROCESAMIENTO DISTRIBUIDO

Una de las cuestiones importantes en el desarrollo de aplicaciones es el tiempo de ejecución en la realización de las tareas. Como HERMINWEB tiene que procesar una enorme cantidad de información fue descartado el procesamiento secuencial en la preparación de los datos. Una de las soluciones aplicadas fue el procesamiento basado en hilos, esta solución mejoró el tiempo de ejecución pero seguía siendo alto, para 30 GB de registros del proxy consumió 2 horas en realizar el procesamiento. Se tomó como alternativa la programación distribuida, siendo la definitiva a utilizar. Por lo tanto para el procesamiento de los logs del proxy se utilizó el módulo de Parallel Programming de Python. HERMINWEB es usado como cliente y las demás computadoras como servidores, la herramienta distribuye los usuarios a cada computadora para que esta procese los registros del mismo. El diseño se muestra en la Figure 3:

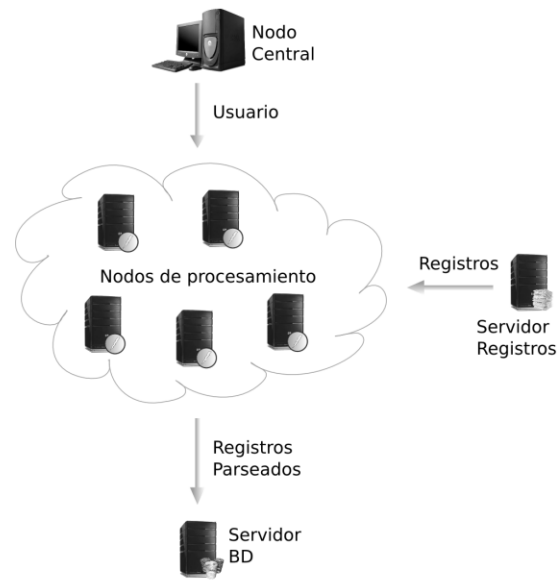


Figure 3: Distribución de los nodos

Utilizando esta solución se realizaron algunas pruebas arrojando los siguientes resultados:

Table 1: Tiempo requerido por cantidad de nodos

Número de nodos	Tiempo total aproximado(minutos)
1	86
2	43
3	30
4	15

Cada uno de estos nodos cuando recibe la solicitud de atender a un usuario determinado obtiene los registros pertenecientes al usuario en el tiempo de navegación a analizar. Posteriormente lee las peticiones realizadas y las almacena en una tabla en la base de datos. Se ejecuta un procedimiento en la base de datos que crea las sesiones del usuario. Una vez creadas las sesiones, se almacenan en la tabla correspondiente a las sesiones de dicho usuario y se elimina la tabla que contiene las peticiones realizadas por este usuario. Estas pruebas demuestran que se reduce considerablemente el tiempo de procesamiento de los registros del proxy con este diseño distribuido, además que mientras más computadoras existan para el procesamiento más rápido se realizará la preparación de los datos.

2.5 APLICACIÓN MINERÍA

El auge de la Minería de Datos y las posibilidades que brinda extraer patrones en grandes volúmenes de información ha permitido el perfeccionamiento de varios algoritmos o técnicas para extraer el conocimiento implícito en los datos. Se han desarrollado varias implementaciones para cada uno de ellos quedando algunos agrupados en bibliotecas.

WEKA es una de estas bibliotecas desarrollada en la Universidad de Waikato, Nueva Zelanda con varios años de uso en numerosas investigaciones con muy buenos resultados. Está escrita en la tecnología JAVA, brindando una alta portabilidad al poder ser usada en diferentes sistemas operativos. La herramienta desarrollada en esta investigación usa la implementación del algoritmo K-Medias (Jain, 2009) de la biblioteca WEKA.

Fue necesario integrar la herramienta y la biblioteca pues son de tecnologías diferentes. Los algoritmos pertenecientes a WEKA poseen varias opciones para su ejecución mediante líneas de comando, pero éstas son insuficientes para el problema en cuestión. Se decide desarrollar un módulo llamado *Agrupador* que funcionará como intermediario entre ambos. Este módulo fue escrito en JAVA. Para la ejecución del módulo se utilizó el patrón Bridge (Puente), este proporciona un desacople entre la implementación de herramienta y el módulo, impidiendo que un cambio en uno requiera cambio en el otro. El módulo es ejecutado por líneas de comandos por la herramienta.

Luego de tener los datos preparados, se seleccionan los usuarios a analizar de acuerdo a las características que se desean (ej. seleccionar los estudiantes de la Facultad 2 de quinto año). Este filtro es aplicado a los datos preparados para crear la vista minable (Hernández Orallo, Ramírez Quintana, & Ferri Ramírez, 2004). Esta vista será el conjunto de datos que utilizará el módulo *Agrupador* para su análisis. Las instancias encontradas en cada uno de los grupos o clústeres se almacenan en tablas en la base de datos (una tabla para cada grupo). Las medias y medianas de cada uno de los grupos son almacenadas en un fichero con formato CVS para no tener que calcularlo posteriormente.

3. RESULTADOS Y DISCUSIÓN

La investigación abordó el desarrollo de un proceso KDD aplicado a la navegación por Internet. Para obtener patrones más descriptivos se integraron a los logs del proxy con los datos de los trabajadores y estudiantes. Se diseñó con una arquitectura de 4 capas facilitando la reutilización por el desacople de las funcionalidades que implica, pudiéndose utilizar en otras investigaciones.

Se estudiaron varias clasificaciones de sitios web decidiendo utilizar BlackList agregando un campo bastante descriptivo de las páginas web. Se describieron las fases de KDD en un flujo de procesos específicos para el problema en cuestión. Se redujo el tiempo de procesamiento de los registros de navegación con un módulo distribuido demostrando que el uso de sistemas distribuidos para el procesamiento de grandes volúmenes de información mejora el rendimiento y reduce el tiempo de ejecución en computadoras de no muy altas prestaciones.

Se definió una nueva concepción de sesiones con el fin de encontrar mayor descripción en el escenario estudiado. Se integraron cuatro tipos de fuentes de datos: logs, base de datos, LDAP y servicios web. Se creó un diseño de base de datos relacional con una parte jerárquica para particionar los datos. Se desarrolló un módulo para la integración entre la biblioteca de algoritmos y la herramienta, permitiendo acoplar dos tecnologías diferentes. Se crean reportes con los resultados obtenidos y pueden difundirse mediante correo electrónico.

El diseño jerárquico de los datos demostró que agiliza considerablemente las consultas siendo una vía de solución cuando no se tienen equipos de altas prestaciones. Se demostró que es posible integrar la biblioteca de algoritmos WEKA con una plataforma escrita en Python. Con el uso de la biblioteca de algoritmos se ahorró tiempo de desarrollo de la herramienta. Se comprobó que el uso de patrones de arquitectura ayuda al desarrollo de aplicaciones, pues mejora la comprensión del diseño y la implementación del mismo. Esta herramienta ayuda en

la toma de decisiones a la DRSI, pues permite analizar un alto volumen de datos presente en los registros del proxy en busca de patrones que describan el uso de las cuotas de navegación.

REFERENCIAS

- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., y otros. (2000). *CRISP-DM 1.0*. SPSS.
- Hernández Orallo, J., Ramírez Quintana, J., & Ferri Ramírez, C. (2004). *Introducción a la Minería de Datos*. Madrid: Pearson Education.
- Jacobson, I., Booch, G., & Rumbaugh, J. (2000). *El Proceso Unificado de Desarrollo de Software*.
- Jain, A. K. (2009). Data Clustering: 50 Years Beyond K-Means. *Pattern Recognition Letters* .
- Larose, D. T. (2005). *Discovery Knowledge in Data: An Introduction to Data Mining*. New Jersey: Wiley.
- Martín Álvarez, L. O., & García Martínez, Y. (6 de 2007). Sistema de Reportes de Navegación por Internet. La Habana, Ciudad Habana, Cuba: UCI.
- Olmos Pineda, I., & González Bernal, J. A. (8-12 de 10 de 2007). Minería de Datos. Puebla, México.
- OM, G. (2009). *Business Process Model and Notation (BPMN)*.
- Ordoñez Leyva, Y., & Avilés Vázquez, E. (28 de 6 de 2010). Herramienta Informática de Minería de Uso de la Web sobre los registros de navegación por Internet. La Habana, Ciudad Habana, Cuba: UCI.
- Pérez Hurtado, A., & Padilla Moya, Á. L. (17 de 6 de 2010). Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Módulo DNS v2.0. La Habana, Ciudad Habana, Cuba: UCI.
- Hernández Orallo, J., Ramírez Quintana, J., & Ferri Ramírez, C. (2004). *Introducción a la Minería de Datos*. Madrid: Pearson Education.

Autorización y Renuncia

Los autores autorizan a LACCEI para publicar el escrito en las memorias de la conferencia. LACCEI o los editores no son responsables ni por el contenido ni por las implicaciones de lo que esta expresado en el escrito.