
**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

"Sistema de análisis de patrones de navegación usando Minería Web"

**TESIS DE GRADO**

Previo a la obtención del título de:

**INGENIERO EN COMPUTACION ESPECIALIZACION EN SISTEMAS TECNOLOGICOS**
**INGENIERO EN COMPUTACION ESPECIALIZACION EN SISTEMAS TECNOLOGICOS**
**INGENIERO EN COMPUTACION ESPECIALIZACION EN SISTEMAS DE INFORMACION**

**Presentada por:**

Patricio Xavier Alcívar Zambrano
Fanny Elizabeth Idrovo Chiriboga
Víctor Hugo Macas Pizarro
GUAYAQUIL – ECUADOR
Año – 2007

# AGRADECIMIENTO

A **DIOS** el ser supremo que siempre nos llena de bendiciones, nos da salud y fuerzas para vivir.

A nuestros ***PADRES*** y hermanos que no dudaron en depositar toda su confianza en nosotros y nos brindaron su apoyo incondicional para lograr nuestros ideales.

# DEDICATORIA

Dedicamos este trabajo a Dios y a nuestros Padres ya que ellos son las personas más importantes en nuestras vidas.

A ***DIOS*** por darnos las fuerzas para seguir por el buen camino bendiciendo siempre todo lo que hacemos.

A nuestros ***PADRES*** que nos apoyan en la realización de nuestras metas y triunfos y siempre están a nuestro lado confiando en lo que nos proponemos a hacer.

# TRIBUNAL DE GRADUACIÓN

Ing. Holger Cevallos
SUBDECANO DE LA FIEC
PRESIDENTE

Ing. Fabricio Echeverría
DIRECTOR DE TÓPICO

Ing. Katherine Chiluiza
MIEMBRO DEL TRIBUNAL

Ing. Galo Valverde
MIEMBRO DEL TRIBUNAL

# DECLARACION EXPRESA

"La responsabilidad del contenido de esta Tesis de Grado, me corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL".

(Reglamento de Graduación de la ESPOL).

Patricio Alcívar Zambrano

Víctor Macas Pizarro

Fanny Idrovo Chiriboga

# RESUMEN

Nuestra tesis se enfoca en el campo de Minería Web, por tanto, el presente documento describe la utilización de técnicas de minería de datos sobre el contenido de archivos log generados en un servidor Web. En estos archivos se registran los miles de accesos o visitas a páginas que realizan los navegantes en un sitio Web.

Específicamente usaremos algoritmos que ejemplifican el uso de técnicas como Reglas de Asociación, Secuencia de Patrones y Clusterización. Además mostraremos estadísticas de uso de varios parámetros del servidor Web.

Estos algoritmos de minería analizan datos y generan conocimiento útil sobre patrones de navegación de los usuarios cuando interactúan por un sitio Web. Desarrollamos una aplicación que muestra el proceso necesario para obtener este conocimiento, el cual comienza con el ingreso del archivo log del servidor, luego se realiza tareas de procesamiento en su contenido para que estos datos sean tomados por los algoritmos de minería y finalmente presenta los resultados obtenidos que ayudan a entender como usan un sitio Web los navegantes.

La descripción del proceso que analiza los datos de la Web, junto con los patrones encontrados son los principales enfoques de la presente tesis y de la aplicación desarrollada.

# INDICE GENERAL

# INDICE DE FIGURAS

# INDICE DE TABLAS

# INTRODUCCION Y JUSTIFICACION

*Introducción*

La World Wide Web es el repositorio más grande y ampliamente conocido de páginas Web. Estos documentos o páginas Web están escritos en una gran diversidad de idiomas y abarcan todos los tópicos del conocimiento humano. La utilización de la Web ha experimentado un crecimiento exponencial desde su aparición y resulta notorio debido al gran número de sitios que ofrecen variados servicios en línea, por ejemplo en campos como: la educación, la investigación, las consultas de servicios, el entretenimiento, el comercio electrónico, entre otros.

Los millones de usuarios que navegan por la Web van dejando registrados sus accesos o visitas en archivos log del servidor Web. Cada acción de un usuario en un sitio Web genera datos, no sólo cuando realiza acciones de alto nivel como comprar algo, sino que también en acciones tan simples como la búsqueda de algún contenido o simplemente al navegar a través del sitio. Ésta desproporcionada cantidad de datos contiene información escondida que puede ser analizada y así tener resultados que ayuden a la efectividad de un sitio en Internet.

El problema con que nos encontramos es que los administradores o incluso empresas u organizaciones que están detrás de estos sitios Web no aprovechan la valiosa información que se puede obtener analizando los logs con técnicas de minería de datos. Es decir, pierden el conocimiento sobre la forma en que los navegantes se mueven a través de las páginas de su sitio en Internet.

Esta tesis busca resolver este problema permitiendo que los datos registrados por un servidor Web sean analizados para descubrir los patrones de navegación de los usuarios del

sitio. Para fines académicos relacionados con el desarrollo de esta tesis, usaremos como caso real el sistema manejador de contenidos, que usa la ESPOL, SIDWEB, (www.sidweb.espol.edu.ec) el cual es administrado por el Centro de Tecnologías de Información de la ESPOL.

El presente trabajo se divide en 6 capítulos, cuyos contenidos se describen a continuación:

Fundamento teórico, destacamos conceptos y ejemplos relacionados a la Minería Web y sus áreas de enfoque. Damos especial énfasis en información teórica sobre el área donde desarrollamos nuestro trabajo.

Circulo virtuoso, describimos la metodología general para realizar minería de datos desde la recolección de datos hasta su difusión y uso. Dentro de esta metodología general mencionamos las tareas necesarias que empleamos para realizar minería en ambientes Web.

Análisis del sistema, describimos los requerimientos funcionales y no funcionales de nuestra aplicación que analiza los patrones de navegación de los usuarios, así como el alcance de la misma.

Diseño e implementación, describimos principalmente como están formados los componentes del sistema y la estructura de los algoritmos de minería que usamos.

Pruebas, mostramos las pruebas realizadas con los algoritmos de minería, los resultados producidos y la aceptación del usuario principal de la aplicación.

Definición del producto, se describe el segmento del mercado hacia donde va nuestra aplicación, además de análisis de costos e ingresos para nuestra aplicación.

*Justificación*

El crecimiento explosivo del uso de sitios Web, ha hecho necesario que los administradores de estos sitios mejoren los servicios que ofrece. Una forma de realizar esta mejora es extrayendo el conocimiento sobre los patrones de navegación de los usuarios. Además, con la transformación de la Web en una herramienta primaria para los campos de uso mencionados, se hace imperativo indispensable poder rastrear y analizar modelos de acceso de los usuarios con el fin de cumplir objetivos de negocios y mejoras del sitio.

# OBJETIVOS

El objetivo principal de nuestra tesis es desarrollar una aplicación que permita analizar los archivos log de un servidor Web y así encontrar patrones de navegación de los visitantes de un sitio Web.

Para encontrar estos patrones usaremos diferentes técnicas de minería de datos, cada una de ellas genera un tipo de conocimiento distinto. Por tanto, además del objetivo principal descrito nos planteamos otros según la técnica empleada:

- Encontrar las asociaciones entre enlaces o páginas que se producen cuando los visitantes navegan en un sitio Web determinado, utilizando reglas de asociación

- Encontrar y predecir el comportamiento de los visitantes de un sitio Web con respecto al tiempo, utilizando secuencia de patrones.

- Encontrar entre los distintos visitantes grupos con características similares de navegación Web, utilizando clusterización.

# 1. FUNDAMENTO TEORICO

Actualmente, Internet es un medio popular para difundir información. Esta situación hace que los usuarios tengan una sobrecarga de información y algunos problemas al interactuar con los sitios de servicio Web, tales como [1]:

- Encontrar información relevante: cuando un usuario utiliza servicios de búsqueda para encontrar información específica en la Web, normalmente introduce una pregunta con las palabras clave y obtiene como respuesta una lista de páginas ordenadas según su similitud con la pregunta. Sin embargo, estas herramientas de búsqueda tienen por lo general, una precisión bastante baja debido a la irrelevancia de muchos de los resultados de la búsqueda. A esto se une su limitada memoria que las hace incapaces de indexar toda la información disponible en la Web, por lo que hace incluso más necesario encontrar la información relevante a la pregunta.

- Crear nuevo conocimiento: la relevancia de la información obtenida en las consultas a la Web es un problema estrechamente relacionado con el de crear nuevo conocimiento a partir de la información disponible en la Web, es decir, una vez obtenidos los datos tras el proceso de búsqueda probablemente queramos extraer coincidencias, resúmenes, patrones, regularidades y, al fin y al cabo, conocimiento a partir de esos datos. Podemos decir, que si encontrar información en la Web es un proceso orientado a la recuperación, la obtención de conocimiento útil es un proceso orientado a la minería de datos.

- Personalización de la información: a menudo se asocia este problema con la presentación y el tipo de la información, ya que los diferentes usuarios suelen tener gustos distintos a la hora de preferir ciertos contenidos presentaciones cuando interactúan con la Web. Relacionado con este problema está el de aprender de los

usuarios, es decir, saber qué es lo que los usuarios hacen y quieren. Esto permite personalizar la información incluso para un usuario individual (diseño de portales Web, de herramientas software, filtros de correo, etc.).

La enorme cantidad de información disponible hace de la Web un área fértil para la minería de datos cuyas técnicas pueden resolver los problemas que se menciona. Un área creciente que esta aprovechando las ventajas del uso de estas técnicas es Minería Web.

## 1.1. Minería Web

### 1.1.1. Definición

Minería Web puede definirse como el descubrimiento y análisis de información útil en la World Wide Web [2].

Minería de datos es la tecnología usada para descubrir información no-obvia, potencialmente útil y previamente desconocida, a partir de fuentes de datos. El potencial de la Minería Web radica en la aplicación de algoritmos de minería a los datos generados en Internet, los cuales incluyen principalmente archivos log del servidor Web.

### 1.1.2. Áreas de enfoque

La minería Web se clasifica en función de la parte de la Web que se mina, por tanto existen tres áreas o enfoques: Minería del contenido de la Web, Minería de la estructura de la Web y Minería del uso de la Web [3].

Según el área donde se realice Minería Web podemos mencionar beneficios como [4]:

*Personalización*

Minería Web permite a los vendedores influenciar a sus clientes en línea mediante el entendimiento y predicción de su comportamiento: el minorista en línea ahora tiene acceso a inteligencia de mercadotecnia detallada acerca de los visitantes de su sitio Web.

Los beneficios comerciales de la minería Web permiten a los proveedores entregar servicios personalizados, dar filtros colaborativos, entregar apoyo reforzado al cliente, definir su estrategia de productos y servicios, detectar fraude, etc. En resumen, la habilidad de servir las necesidades de sus clientes y entregarles el mejor y el más apropiado servicio en un momento dado.

La minería Web juega un rol importante en el área del marketing uno a uno a través de la personalización (entregar contenido personalizado), enviar anuncios dirigidos y construcción de perfiles.

Minería Web entrega información sobre los mercados y sobre los datos colectados desde los sitios Web que tienen un enorme potencial para el mercadeo directo. Mediante este análisis puede entregarse mensajes personalizados a las personas individualmente. La oportunidad de identificar a los navegantes y dirigir a los compradores de productos y servicios a ofertas atractivas y promociones de venta es sin duda algo interesante.

*Entendimiento la conducta del consumidor*

Las empresas pueden perfeccionar sus sitios de negocios electrónicos para el máximo impacto comercial entendiendo la conducta dinámica de sus visitantes. Estos negocios pueden ahora obtener conocimiento de los gustos individuales y preferencias de los visitantes de su sitio. Con ello, pueden:

- Determinar la frecuencia de compra (o la probabilidad de que los clientes vuelvan a elegir su marca).
- Calcular la proporción de nueva adquisición del cliente.
- Descubrir y comparar modelos de clientes.
- Aprender quién está comprando en su sitio.

*Determinar la efectividad del sitio Web*

Con Minería Web, las compañías pueden descubrir las áreas de alto y bajo impacto de su sitio Web. Los administradores del sitio Web ya no tienen que confiar en la intuición al diseñar un esquema del sitio. Los proveedores en línea pueden desarrollar mejor su relación con sus clientes y personalizar el contenido online.

*Medición del éxito de los esfuerzos de Marketing*

En el mundo físico es difícil medir el éxito de las campañas de comercialización. Pero, en Internet usted puede obtener medidas reales del éxito de una campaña de mercadeo. Con Minería Web, las compañías pueden generar modelos para realizar campañas de marketing, y el sitio Web puede adaptarse a los clientes reconocidos por ella. Así, a cada segmento de clientes puede realizárseles campañas de marketing dirigido y ofertas especiales, y medir eficazmente el retorno a la inversión (ROI) en publicidad.

## 1.2. Contenido de la minería Web

Dada la enorme cantidad de información disponible en la Web y la gran diversidad de la misma, uno de sus principales usos es el de buscar información. La principal diferencia entre las técnicas de recuperación de la información y las técnicas de minería del contenido de la Web es que las primeras ayudan a los usuarios a encontrar documentos que satisfacen sus necesidades de información, mientras que

las segundas permiten: descubrir, reconocer o derivar información nueva a partir de uno, o generalmente, varios documentos.

### 1.2.1. Definición y objetivos

Minería del contenido Web es el descubrimiento de información útil desde los contenidos textuales y gráficos de los documentos Web, y tiene sus orígenes en el procesamiento del lenguaje natural y en la recuperación de la información. Es decir, analiza documentos más que los enlaces entre ellos [5].

### 1.2.2. Usos y aplicaciones

La minería de textos juega un papel importante en una amplia variedad de tareas de manipulación de la información más dinámicas y personalizadas, como en la ordenación en tiempo real de correo electrónico o archivos en jerarquías de carpetas, en el filtro del correo electrónico, búsqueda estructurada y/o en los navegadores Web, identificación de tópicos para soportar operaciones de procesamiento específicas a un tópico, catalogación de nuevos artículos y páginas Web y en los agentes de información personal.

## 1.3. Estructura de la minería Web

### 1.3.1. Definición y objetivos

Minería de la estructura de la Web es el proceso de descubrir el modelo subyacente a la estructura de enlaces de la Web y analiza, fundamentalmente, la topología de los hipervínculos (con o sin descripción de los enlaces). Este modelo puede usarse para categorizar páginas Web y es útil para generar información como la similitud y relación entre diferentes sitios Web [6]. El principal objetivo de la minería de la estructura de la Web es

inferir conocimiento a partir de la organización y las referencias o enlaces entre documentos de la Web.

Por ejemplo, muchos links que apuntan a un documento pueden indicar la popularidad de un documento, mientras los links que salen de un documento pueden indicar la riqueza o variedad de los temas que cubre el documento. De esta forma, se puede tomar ventaja de esta información para encontrar los documentos pertinentes en la Web y rastrear una estructura resumida de estos documentos.

Estas estructuras resumidas son comparables a palabras claves que se utilizan para referenciar citas bibliográficas. Existen métodos y software que toman ventaja de esta información, popularizando páginas y haciendo conteos de links para clasificar las páginas Web y facilitar su búsqueda.

### 1.3.2. Otras aplicaciones de la estructura de la minería Web

Minería de la estructura de la Web puede ser usado para categorizar páginas Web y es útil para generar información como la similitud y relación entre diferentes sitios Web, estudiar topologías, así como para detectar páginas autoridades y páginas concentradoras (que apuntan a páginas autoridades), siendo este su mayor campo de acción.

## 1.4. Uso de la minería Web

Según destaca Mariano Silva [7]: A medida que más empresas basan su negocio en Internet, las estrategias y técnicas tradicionales para el análisis del mercado deben ser vistas desde un nuevo contexto. Las organizaciones y compañías en Internet generan y almacenan grandes volúmenes de datos en sus funcionamientos diarios. La mayoría de esta información es generada automáticamente por los

servidores Web y se almacenan en archivos llamados archivos Log de acceso al servidor. Otras fuentes de información del usuario incluyen la referencia a otros sitios o páginas de la Web y registros de usuario en bases de datos vía formularios en línea. Analizar tales datos puede ayudar a las organizaciones a determinar que usuarios visitan su sitio, permitiendo generar estrategias de mercadeo de productos y aumentar la efectividad de sus campañas promocionales, entre otras cosas. El análisis del acceso al servidor y los datos del registro del usuario también puede proporcionar información valiosa de cómo mejorar la estructura del sitio, creando una presencia Internet más eficaz para las organizaciones.

### 1.4.1. Definición y objetivos

Minería del uso de la Web es el proceso de analizar la información sobre los accesos Web disponibles en los servidores Web [8]. El principal objetivo es entender y servir mejor las necesidades de los usuarios cuando navegan en aplicaciones basadas en la Web.

Las aplicaciones de este tipo de minería pueden clasificarse en:

- Aprendizaje de patrones de navegación
- Aprendizaje de perfiles de usuario para modelar interfaces adaptivas (personalización)

Esta tesis se enfoca en el desarrollo de los conceptos relacionados a la minería del uso de la Web y específicamente en el aprendizaje de patrones de navegación.

### 1.4.2. Etapas del uso de la minería Web

Minería del uso de la Web consiste de tres fases, las cuales son: preprocesamiento, descubrimiento de patrones y análisis de patrones [9].

## Preprocesamiento

Consiste en convertir: el uso, el contenido y la estructura de la información, contenida en varias fuentes disponibles de datos, en abstracciones de datos necesarios para el descubrimiento de patrones.

*Preprocesamiento de Uso*

Es posiblemente la tarea más difícil en el proceso de minería del uso de la Web debido al estado incompleto de los datos disponibles. A menos que del lado del cliente exista un mecanismo de rastreo, solo la dirección IP, agente y los clicks del lado del servidor son disponibles para identificar usuarios y sesiones de servidor. Algunos de los típicos problemas son:

- Dirección IP única / Múltiples sesiones de servidor.- Los proveedores de servicio de Internet (ISPs) típicamente tienen muchos servidores proxy que los usuarios acceden a través de la Web. Un único servidor proxy puede tener varios usuarios accediendo a un sitio Web, potencialmente sobre el mismo periodo de tiempo.

- Múltiples direcciones IP / Única sesión de servidor.- Algunos ISPs o herramientas privadas asignan aleatoriamente cada pedido de un usuario a una de las varias direcciones IP. En este caso, una única sesión de servidor puede tener múltiples direcciones IP.

- Múltiples direcciones IP / Único usuario.- Un usuario que accede a la Web desde diferentes maquinas tendría diferentes direcciones IP de sesión a sesión. Esto hace difícil el seguimiento de visitas repetidas del mismo usuario.

- Múltiples agentes / Usuario único.- De nuevo, un usuario que usa mas de un navegador, incluso en la misma máquina, aparecerá como múltiples usuarios.

Asumiendo que cada usuario ha sido ahora identificado (a través de cookies, log, o IP/agentes/ análisis de ruta), el flujo de clicks de cada usuario debe ser dividido en sesiones. Desde que los pedidos de las páginas de otros servidores no están típicamente disponibles, es dificultoso conocer cuando un usuario ha salido del sitio Web. Un tiempo de treinta minutos es frecuentemente usado como valor por defecto para interrumpir el flujo de clicks en las sesiones.

*Preprocesamiento del contenido*

Consiste de convertir el texto, las imágenes, los scripts, y otros archivos tipo multimedia en formas que son útiles para el proceso de minería del uso de la Web. Frecuentemente, esto consiste de realizar minería del contenido usando clasificación o clusterización. Mientras aplicamos minería de datos al contenido de los sitios Web es una interesante área de investigación en sí misma. En el contexto de minería del uso de la Web el contenido de un sitio puede ser usado para filtrar el ingreso, o salida del algoritmo de descubrimiento de patrones. Por ejemplo, podrían usarse los resultados de un algoritmo de clasificación para limitar los patrones descubiertos a aquéllos que contienen “vistas de página” o pedidos sobre un determinado asunto o clase de productos.

Además de clasificar o clusterizar dichas vistas de página basados en temas, éstas pueden ser clasificadas de acuerdo a su intento de uso.

Vistas de página puede usarse para transmitir información (a través de texto, gráficos u otros multimedia), obtener información desde el usuario, permitir navegación (a través de una lista de enlaces hipertexto), o alguna combinación de uso. El uso de vistas de página puede también filtrar sesiones antes o después del descubrimiento de patrones. Para ejecutar algoritmos de la minería del contenido en vistas de página, la información debe ser convertida en un formato cuantificable. Algunas versiones del modelo espacio vector es típicamente usado para cumplir esto. Archivos de texto pueden ser partidos en vectores de palabras. Palabras o descripciones de texto pueden ser sustituidos por gráficos o multimedia.

El contenido de estáticas vistas de página pueden ser fácilmente preprocesadas analizando el código HTML y reformateando la información o ejecutando algoritmos adicionales. Vistas de página dinámicas presentan más desafío. Servidores de contenido que emplean técnicas de personalización y/o utiliza base de datos para construir vistas de página pueden ser capaces de formar más vistas de páginas que pueden ser prácticamente preprocesadas. Un conjunto dado de sesiones de servidor puede ser solo acceder una fracción de las posibles vistas de páginas para un grande sitio dinámico. También el contenido puede ser revisado en una base regular. El contenido de cada pagina para ser procesado debe ser ensamblado, o por un pedido http desde un crawler, o una combinación de template, script y acceso de base de datos. Si sólo la porción de vista de pagina que se acceden son preprocesadas, la salida de algún algoritmo de clasificación o clusterización puede ser sesgadas.

*Procesamiento de la estructura*

La estructura de un sitio es creada por los enlaces hipertexto entre páginas. La estructura puede ser obtenida y procesada en la misma manera como el contenido de un sitio. De nuevo, el contenido dinámico (y por lo tanto los enlaces) plantean más problemas que paginas estáticas. Una diferente estructura del sitio puede tener que ser construida para cada sesión del servidor.

## Inferencia de patrones

El descubrimiento (inferencia) de patrones utiliza los métodos y algoritmos desarrollados de varios campos como la estadística, minería de datos, aprendizaje de máquina y reconocimiento de patrones.

Métodos desarrollados desde otros campos deben tomar en consideración los diferentes tipos de abstracciones de datos y el conocimiento previo disponible para Minería Web. Por ejemplo, en el descubrimiento de reglas de asociación, la noción de una transacción para el análisis de la cesta de mercado no toma en consideración el orden en el cual los ítems son seleccionados. Sin embargo, en Minería del uso de la Web, una sesión de servidor es una secuencia ordenada de páginas pedidas por un usuario. Además, debido a la dificultad en identificar sesiones únicas, previo conocimiento adicional es requerido (tal como seleccionar un periodo de tiempo por defecto, mencionado en nuestro posterior análisis de nuestra tesis)

## Análisis de patrones

Análisis de patrones es el último paso en el proceso total de Minería del Uso de la Web. La motivación detrás del análisis de patrones es filtrar reglas no interesantes o patrones de un conjunto encontrado en la fase de descubrimiento de patrones.

La metodología exacta de análisis es usualmente gobernada por la aplicación por la cual la Minería Web es hecha. La forma mas común de análisis de patrones consiste de un conocimiento de un mecanismo de consulta como SQL. Otro método es cargar los datos de uso en un cubo de datos para realizar operaciones OLAP. Técnicas de visualización tal como gráficos de patrones o asignación de colores a diferentes valores, pueden frecuentemente señalar el patrón total o tendencias en los datos. El contenido y la información de la estructura puede ser usado para filtrar patrones que contienen páginas de un cierto tipo de uso, tipo de contenido, o paginas que coinciden con una cierta estructura de enlaces.

### 1.4.3. Técnicas de aprendizaje aplicadas al uso de la minería [10]

Antes de aplicar cualquier técnica de minería de datos es necesario realizar una transformación de los datos para que éstos puedan ser operados eficientemente. A este proceso se lo conoce como el proceso de descubrimiento de conocimiento. En el marco de dicho proceso se filtrarán datos que no interesan y en general se transformará el log en una estructura más manipulable (por ejemplo una base de datos relacional). Es necesario el conocimiento de la estructura del servidor Web para poder determinar a partir de los accesos cual es la acción que quiere realizar el usuario.

Se han encontrado diferentes enfoques de cómo realizar este proceso que abarcan desde el filtrado o no de pedidos de recursos multimedia, la conversión de un conjunto de requerimientos en la supuesta acción requerida por el usuario llegando hasta la creación de un cubo n-dimensional.

Para poder aplicar las técnicas de minería de datos sobre los datos del log del servidor es necesario, además de aplicar las transformaciones en los datos típicas del proceso de KDD, realizar una adaptación en la definición de las transacciones y los ítems que las componen para los distintos algoritmos. Esto se debe a que en este caso no se tiene la noción de transacción como en una base de datos transaccional en donde existe un "identificador de transacción". Aquí para poder delimitar una transacción se debe utilizar por ejemplo una combinación entre el identificador del usuario que interactúa con el servidor y un período máximo de tiempo aceptado entre accesos. Por ejemplo si un usuario accede a una página del servidor a las 9:00 hs, y hasta las 9:15 hs navega dentro del site; y luego vuelve a acceder por la tarde, esto es considerado como dos transacciones distintas.

*Reglas de Asociación*

El descubrimiento de reglas de asociación es generalmente aplicado a bases de datos transaccionales, donde cada transacción consiste en un conjunto de ítems. En este modelo, el problema consiste en descubrir todas las asociaciones y correlaciones de ítems de datos donde la presencia de un conjunto de ítems en una transacción implica (con un grado de confianza) la presencia de otros ítems.

En el contexto de Minería Web este problema tiende a descubrir la correlación entre los accesos de los clientes a varios archivos disponibles en el servidor. Cada transacción está compuesta por un conjunto de URL accedidas por el cliente en una visita al servidor.

Utilizando reglas de asociación, se puede descubrir, por ejemplo, lo siguiente:

> 60% de los clientes que acceden a la página con URL /company/products/, también acceden a la página /company/products/product1.html.

Esta técnica, además, considera el soporte para las reglas encontradas. El soporte es una medida basada en el número de ocurrencias de los ítems dentro del log de transacciones.

En Minería Web existen otros factores que pueden ayudar a podar el espacio de búsqueda de las reglas. En general, los sitios están organizados jerárquicamente y la estructura de esta jerarquía es conocida con anticipación. Por ejemplo, si el soporte de /company/products/ es bajo, se puede inferir que la búsqueda de reglas de asociación en las páginas /company/products/product1.html y /company/products/product2.html no van a tener el soporte necesario.

El descubrimiento de estas reglas en el ámbito del comercio electrónico puede ayudar en el desarrollo de las estrategias de marketing.

Además las reglas de asociación pueden ayudar a mejorar la organización de la estructura del sitio. Por ejemplo, si descubrimos que el 80% de los clientes que acceden a /company/products y /company/products/file1.html también acceden a /company/products/file2.html, parece indicar que alguna información de file1.html lleva a los clientes a acceder a file2.html. Esta correlación podría sugerir que ésta información debería ser movida a /company/products para aumentar el acceso a file2.html.

*Secuencia de Patrones*

En general en las bases de datos transaccionales se tienen disponibles los datos en un período de tiempo y se cuenta con la fecha en que se realizó la transacción; la técnica de secuencia de patrones se basa en descubrir patrones en los cuales la presencia de un conjunto de ítems es seguido por otro ítem en orden temporal.

En el log de transacciones de los servidores de Web, se guarda la fecha y hora en la que un determinado usuario realizó los requerimientos. Analizando estos datos, se puede determinar el comportamiento de los usuarios con respecto al tiempo. Con esto, se puede determinar por ejemplo:

> 60% de los clientes que emitieron una orden en-línea en /company/products/product1.html, también emitieron una orden en-línea en /company/products/product4.html dentro de los siguientes 15 días.

El descubrimiento de secuencia de patrones en el log puede ser utilizado para predecir las futuras visitas y así poder organizar mejor los accesos y publicidades para determinados períodos. Por ejemplo, utilizando está técnica se podría descubrir que los días laborables entre las 9 y las 12 horas muchas de las personas que accedieron al servidor lo hicieron para ver las ofertas y en los siguientes días la mayoría compró productos. Entonces por la mañana debería facilitarse el acceso a las ofertas y brindar la publicidad más llamativa posible.

También puede ser utilizado para descubrir tendencias, comportamiento de usuarios, secuencias de eventos, etc. Esta información puede ser aprovechada tanto en el aspecto comercial (pensar una campaña de marketing) como en el aspecto técnico (mejorar los tiempos de acceso).

En general todas las herramientas que realizan minería sobre el log enfocan el análisis sobre secuencias de tiempo ya que los eventos que son almacenados están muy relacionados con el tiempo en que se producen.

*Clasificación y Clustering*

Las técnicas de clasificación permiten desarrollar un perfil para los ítems pertenecientes a un grupo particular de acuerdo con sus atributos comunes. Este perfil luego puede ser utilizado para clasificar nuevos ítems que se agreguen en la base de datos.

En el contexto de Minería Web, las técnicas de clasificación permiten desarrollar un perfil para clientes que acceden a páginas o archivos particulares, basado en información demográfica disponible de los mismos. Esta información puede ser obtenida analizando los requerimientos de los clientes y la información transmitida de los navegadores incluyendo el URL.

Utilizando técnicas de clasificación, se puede obtener, por ejemplo, lo siguiente:

> 50% de los clientes que emiten una orden en-línea en /company/products/product2.html, están entre 20 y 25 años y viven en la costa oeste.

La información acerca de los clientes puede ser obtenida del navegador del cliente automáticamente por el servidor; esto incluye los accesos históricos a páginas, el archivo de cookies, etc. Otra manera de obtener información es por medio de los registros y los formularios en-línea. La agrupación

automática de clientes o datos con características similares sin tener una clasificación predefinida es llamada “clustering” o agrupamiento.

La utilización de la técnica de agrupamiento sobre el log del servidor Web, puede ser utilizado para estrategias de marketing dirigido según las clases obtenidas.

# 2. CICLO VIRTUOSO

En la presente tesis nos referimos como ciclo virtuoso a la metodología empleada cuando se realiza la extracción del conocimiento. En este capítulo se presentará las fases del ciclo virtuoso y a su vez se hará relación con las etapas de minería del uso de la Web descritas en la sección 1.4.2. También dichas fases son reflejadas en la aplicación desarrollada.

La figura 2.1 muestra el proceso iterativo e interactivo. Es iterativo ya que la salida de alguna de las fases puede hacer volver a pasos anteriores y porque a menudo son necesarias varias iteraciones para extraer conocimiento de alta calidad. Es interactivo porque el usuario, o mas generalmente un experto en el domino del problema, debe ayudar en la preparación de los datos y validación del conocimiento extraído.

El ciclo virtuoso consta de cinco fases como se muestra en la figura 2.1.

1. **Fase de integración y recopilación** de datos, que determina las fuentes de información, en nuestro caso los archivos log del sitio web **www.sidweb.espol.edu.ec** .

2. **Fase de selección, limpieza y transformación de datos**, en la que se elimina datos que no son necesarios o se corrige los datos incorrectos. Además prepara y transforma los datos para que puedan ser utilizados por los algoritmos de minería.

3. **Fase de minería de datos**, en la que se ejecuta los algoritmos de minería de datos, en nuestra aplicación usaremos algoritmos para reglas de asociación, secuencia de patrones y clusterización.

4. **Fase de evaluación e interpretación** en la que se evalúan los patrones de navegación encontrados y la analizan los expertos, es decir los administradores del sitio Web que son las personas principales que conocen el contexto del sitio.

5. **Fase la de difusión y uso** en la que se hace uso del nuevo conocimiento sobre la manera en que los navegantes usan un sitio Web.

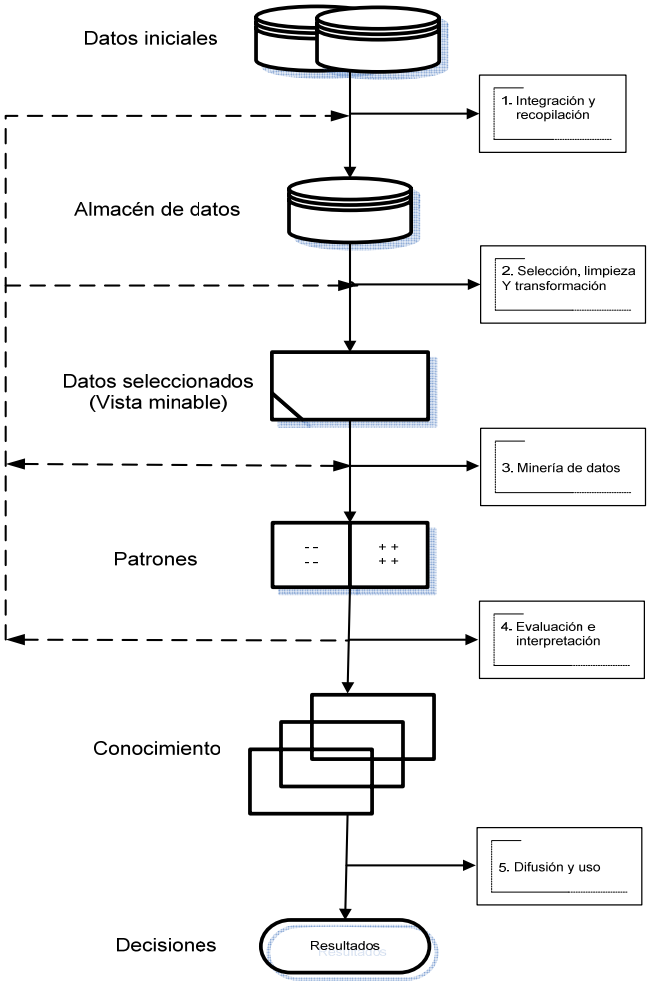



**Figura 2.1 Fases del ciclo virtuoso para la minería de datos**

## 2.1. Integración y recopilación

El primer paso del ciclo virtuoso consiste en recopilar los datos que serán analizados. En muchos trabajos sobre minería, la principal fuente de información es una base de datos. En nuestro contexto, donde vamos a minar el uso de un sitio Web necesitamos archivos log generados en un servidor Web.

En estos archivos planos se registran todos los eventos realizados por los visitantes, es decir, cada vez que alguien visita una página, se registra una serie de líneas por cada elemento que compone la página requerida. Los archivos log vienen en varios

formatos, pero nosotros usaremos los que vienen en formato CLF (Common Log File) y ELF (Extended Log File), la descripción detallada de sus estructuras se encuentra en el Anexo A.

Los archivos log que se recopilaron para probar los algoritmos de minería en nuestra aplicación, pertenecen al sitio Web www.sidweb.espol.edu.ec, que es un administrador de contenidos que usa los alumnos de la ESPOL.

## 2.2. Selección, limpieza y transformación

La calidad del conocimiento descubierto no solo depende del algoritmo de minería utilizado, sino también de la calidad de los datos minados. Por ello, después de la recopilación, el siguiente paso en este proceso es seleccionar y preparar los datos que se van a minar, esto se asemeja a la primera etapa preprocesamiento de Minería del uso de la Web descrita en la sección 1.4.2.

*Selección y limpieza*

Estas tareas no son demasiado complejas y son necesarias hacerlas ya que algunos datos del archivo log coleccionado en la etapa anterior son irrelevantes o innecesarios para descubrir patrones de navegación.

Cuando un visitante realiza la petición de una página al servidor Web, se inserta en el archivo log varias líneas correspondientes a las diferentes partes de las que está compuesta la página y esto es debido a que una página Web se compone no sólo de texto, sino también de imágenes, sonidos, hojas de estilo, archivos javascripts, entre otros objetos.

Por ejemplo, en la figura 2.2 si un usuario visita la dirección

http://www.sidweb.espol.edu.ec/index.html., en el log se incluirían líneas correspondientes no sólo a la petición de index.html, sino también a las peticiones de las imágenes gif y jpg que componga la página.

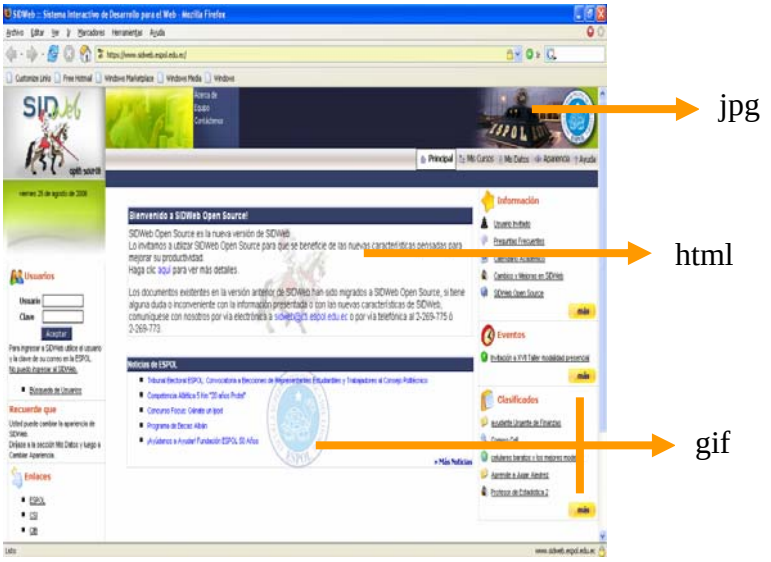



**Figura 2.2 Componentes de una página Web**

La tarea de selección y limpieza se la realiza eliminando del log todas las extensiones: jpg, bmp, gif, exe, js, css, pdf, doc, txt.

*Transformación*

Nosotros necesitamos identificar de alguna manera los usuarios del sitio, debido a que no es común que ellos se identifiquen mediante un usuario y contraseña esta tarea se vuelve un poco compleja. Por tanto, se ha tenido que realizar refinamientos para identificar una sesión de usuario según el formato del log ingresado.

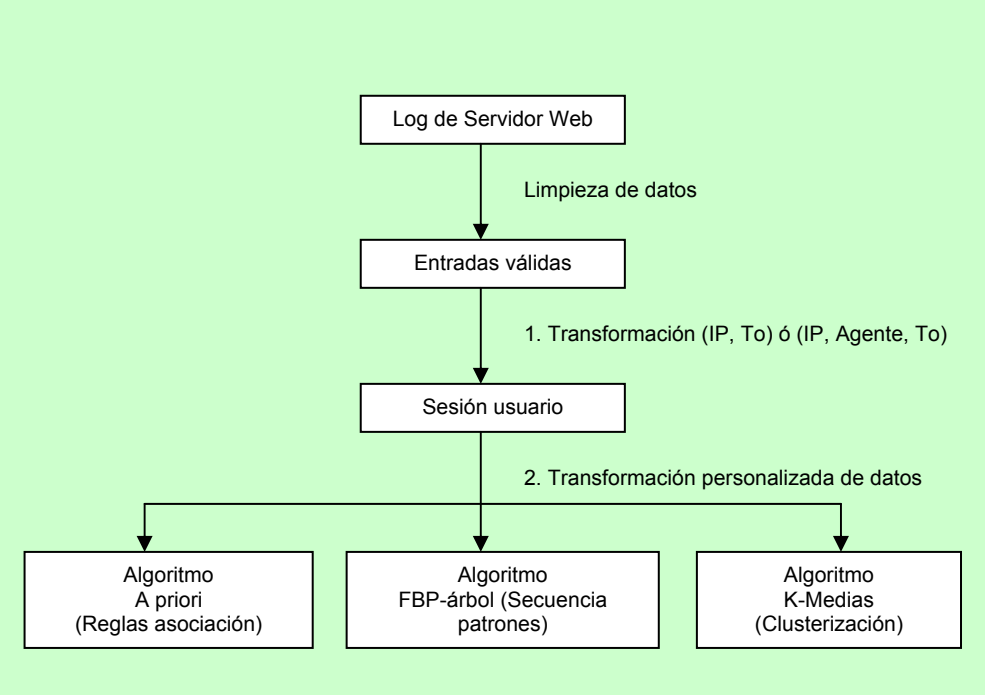
Si el formato es CLF:

1. Se separa las entradas del log por direcciones IP (el remotehost de un registro log). Esto, en un principio, identificaría máquinas diferentes y, por lo tanto, usuarios distintos.

2. Realizamos un segundo refinamiento que consiste en pensar que una misma persona puede realizar diferentes visitas a lo largo del día. Para ello debemos establecer un periodo de tiempo tal que dos peticiones consecutivas realizadas desde una misma dirección IP, separadas por un intervalo de tiempo superior a un cierto umbral, se consideren visitas distintas. El periodo estándar es de 30 minutos como tiempo de corte para diferenciar usuarios. Sin embargo en la aplicación desarrollada se puede seleccionar distintos tiempos de corte, esto es lo que llamamos sesionización. La identificación de una sesión de usuario sería (IP, To)

Si el formato del archivo log es ECLF:
Se realiza los 2 refinamientos anteriores y un tercero usando el campo Agente por lo que la identificación única de una sesión de usuario será por el esquema (IP, Agente, To).

Una vez identificados sesiones de usuarios únicos, se realiza una segunda transformación de datos personalizada según el algoritmo que usamos. Lo dicho anteriormente se resume en la figura 2.3




*Transformación de datos: Algoritmo de A priori (Reglas de Asociación)*

El algoritmo de reglas de Asociación A priori, cuya descripción se detalla en el Anexo B, acepta como datos de entrada una matriz numérica.

Para la realizar la minería Web nosotros formamos la matriz numérica de la siguiente forma: Cada fila representa las *sesiones de usuarios* identificados y cada columna las páginas que forman el sitio Web. Si un usuario visitó una página colocamos un 1 en la columna correspondiente en caso contrario colocamos un 0. Por ejemplo:

Si nuestro sitio Web tiene 4 páginas (a.html, b.html, c.html, d.html) y se identificó 3 sesiones de usuario (s1, s2, s3) cuyas visitas son:

s1(0,1,01) , s2(0,0,0,1), s3(1,1,0,0)

Entonces la matriz de entrada para el algoritmo k-medias quedaría de esta manera:

| | a.html | b.html | c.html | d.html |
|---|---|---|---|---|
| s1 | 0 | 1 | 0 | 1 |
| s2 | 0 | 0 | 0 | 1 |
| s3 | 1 | 1 | 0 | 0 |

**Figura 2.4 Matriz de datos - A priori**

*Transformación de datos: Algoritmo FBP-árbol (Secuencia de Patrones)*

El algoritmo FBP-árbol (árbol de caminos frecuentes de la conducta), cuya descripción se detalla en el Anexo B, acepta como datos de entrada una matriz numérica denomina de transiciones formada con la misma lógica que reglas de asociación.

*Transformación de datos: Algoritmo k-medias (Clusterización)*

El algoritmo de clusterización k-medias, cuya descripción se detalla en el Anexo B, acepta como datos de entrada una matriz numérica para luego obtener los clusters de datos. Para la realizar la minería Web nosotros formamos la matriz numérica siguiendo la misma lógica que para el algoritmo A priori de reglas de asociación., resultando una matriz con la misma estructura.

Las tres técnicas tienen en común que aceptan una matriz de datos numérica, sin embargo cada algoritmo la interpreta y procesa de manera distinta según el tipo de resultado o conocimiento que obtiene. Existe un nivel de detalle más amplio sobre el sentido de estas matrices en la sección 4.2.1

## 2.3. Minería de datos

En esta fase se obtiene ya el conocimiento sobre los patrones de navegación de los usuarios el mismo que luego será utilizado por el administrador Web para mejorar

los servicios de su sitio. Esta fase se asemeja a la segunda etapa de Minería del uso de la Web: descubrimiento de patrones, mostrada en la sección 1.4.2. Los resultados esperados por cada algoritmo son los que ya citamos en la sección de objetivos:

Reglas de asociación. Usando esta técnica se pretende encontrar las asociaciones de enlaces o páginas que se producen cuando los visitantes navegan en un sitio Web determinado.

Secuencia de Patrones. Usando esta técnica se pretende encontrar y predecir el comportamiento de los visitantes de un sitio Web con respecto al tiempo.

Clusterización. En esta técnica se pretende encontrar entre los distintos visitantes grupos con características similares de navegación Web.

## 2.4. Evaluación

Dependiendo del algoritmo de minería de datos existen diferentes medidas para la evaluación de los patrones encontrados.

*Evaluación Reglas de Asociación*

Para evaluar una regla se asociación entre páginas Web, se suele trabajar con dos medidas para conocer la calidad de la regla, estas medidas son:

1. Cobertura.
2. Confianza.

La *cobertura* a la cual también se la conoce como *soporte* de una regla, se define como el número de instancias en las que la regla se puede aplicar.

La *confianza* también llamada precisión mide el porcentaje de veces en que la regla se cumple cuando se puede aplicar.

*Evaluación Secuencia de Patrones*

La calidad de los resultados también son evaluados usando los conceptos de soporte y confianza.

*Evaluación Clusterización*

En este algoritmo hacemos una selección minuciosa de las páginas introduciendo el concepto de **soporte**, se usa dicho soporte para seleccionar de entre las posibles páginas visitadas aquellas que hayan sido visitadas con un mínimo soporte en todas las sesiones de usuario, despreciándose aquellas que no cumplan tal requerimiento. El valor del soporte en la aplicación esta dada en porcentaje (%).

## 2.5. Difusión y uso

Esta fase muestra los patrones de navegación encontrados en la ejecución de los algoritmos empleados. Para la difusión y uso es importante recalcar que los resultados a simple vista podrían no tener ningún sentido, por eso es importante la participación del administrador Web del sitio debido a que conoce la estructura del sitio Web y daría una mayor precisión del conocimiento obtenido de la manera de cómo los usuarios usan el sitio Web analizado. Esta fase es similar a la tercera etapa de Minería del uso de la Web: análisis de patrones, mostrada en la sección 1.4.2. Los resultados obtenidos se muestran en el anexo E.

Además de los resultados de los algoritmos también mostramos estadísticas obtenidas de la actividad del servidor como complemento para conocer como van ciertos parámetros. Los reportes que mostramos son:

- Páginas más visitadas
- URLs externos
- Tiempo de visita
- Tipo de navegador y Sistema Operativo

- Cantidad de usuarios que ingresan en el sitio Web
- Bytes de descarga de las paginas con las imágenes
- Número de visitas por página

# 3. ANALISIS DEL SISTEMA

## 3.1. Análisis de Requerimientos

La aplicación que se desarrolla tiene la estructura global mostrada en la figura 3.1. Los datos de visitas o accesos a páginas son generados por la navegación de sitios en Internet, registrados por los servidores Web en archivos planos llamados log y tomados luego para realizarle Minería Web y descubrir patrones que nos ayudarán a comprender mejor el comportamiento de los usuarios en la Web.

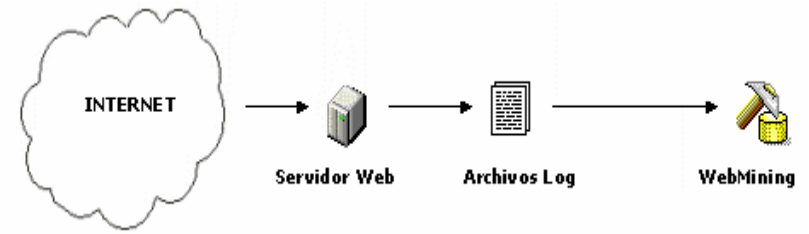



**Figura 3.1 Estructura global aplicación**

A lo largo de esta sección se especificarán todos los requisitos que debe cumplir nuestra aplicación de minería Web. Se especificará primero los requisitos funcionales, dividiendo los mismos en tres apartados: entrada del sistema, procesamiento y salida del sistema. Luego se especificarán los requisitos no funcionales.

### 3.1.1. Requerimientos funcionales

**Entrada del Sistema**

Los datos que alimentan nuestra aplicación vienen en un archivo log de servidor Web. Un archivo log contiene información recogida por el servidor de forma automática sin que el visitante del sitio Web intervenga en dicho proceso.

Los archivos log tienen diferentes formatos. Nuestra aplicación trabajará solo con logs que tengan el formato CLF (Common Log File) y ELF (Extended Log File), cuya descripción detallada de sus estructuras se encuentra en el Anexo A. El usuario tan sólo deberá indicar la ruta del archivo log dónde se encuentre.

**Procesamiento**

Luego de ser ingresado el archivo log la aplicación debe cumplir las siguientes tareas:

*Procesamiento del archivo de entrada*

Según el formato del archivo log (CLF, ECLF), encontramos campos. Para nuestro análisis consideraremos en general los siguientes:

- Remotehost: (host remoto)
- [fecha]
- Solicitud
- Estado
- Bytes
- Referrer
- Agente (Para formato ECLF)

Se realizará lo siguiente:

- Selección y limpieza de los datos

Se debe establecer de entre todos las entradas del archivo log cuáles son útiles y cuáles no (imágenes gif, jpg, etc) en el sentido de que no nos aporten conocimiento y por tanto eliminar los mismos. Esta tarea se describe en la sección 2.2.

- Transformación de los datos

  Los datos son transformados y consolidados de forma apropiada para los algoritmos de minería. La descripción de esta tarea se detalla en la sección 2.2.

*Análisis de los parámetros*

*Sesionización*

Una sesión de host será la secuencia de peticiones al servidor web que transcurren desde que un determinado host hace la primera petición al servidor hasta que realiza la última. Desde esta primera petición hasta la última, se habrán realizado peticiones secuenciales en espacios cortos de tiempo. El usuario podrá especificar el tiempo máximo (en minutos) que debe transcurrir entre una petición de un host y otra petición del mismo host para que se considere aún la misma visita.

*Especificación de visitas de Web*

La especificación del conjunto de archivos que se consideran forman del análisis para generar patrones de navegación (por ejemplo, podemos considerar que un archivo html es un archivo válido para su análisis). Esta especificación será hecha por el usuario que podrá elegir entre los siguientes archivos Web:

asa, asp, cdx, cer, cfm, dbm, htc, htm, html, htmls, htp, htpl, htr, htw, ida, idc, idq, jhtm, jhtml, jsp, mdl, php, php3, php4, sht, shtm, shtml, stm

*Especificación de la técnica elegida para el análisis*

Cuando se proceda al análisis de los ficheros log, el usuario tendrá la oportunidad de seleccionar diferentes técnicas para dicho proceso. En concreto, se podrán elegir las siguientes técnicas:

- Reglas de asociación.

  Si usa esta técnica se pedirá el *soporte y confianza*, El usuario podrá establecer los umbrales mínimos de soporte y confianza que deben satisfacer las reglas de asociación encontradas durante la ejecución del algoritmo. Los parámetros soporte y confianza sirven para evaluar la calidad de una regla.

- Secuencia de Patrones

  Si usa esta técnica también se pedirá los valores de *soporte y confianza* que permitirán evaluar la calidad de los resultados.

- Clusterización

  Si usa esta técnica deberá especificar el valor de *soporte* que servirá para evaluar las páginas que cumplen con este valor en un determinado clúster.

**Salida del Sistema**

La aplicación generará dos tipos de salidas: La primera dependiendo de las técnicas de análisis seleccionadas generará diferentes salidas y la segunda será reportes estadísticos del tráfico del log del servidor Web.

*Considerando la técnica seleccionada:*

- Reglas de asociación:

  Esta técnica proporciona una serie de relaciones entre las páginas visitadas que deberán superar unos umbrales de soporte y confianza mínimos. Estos valores representarán la utilidad y la certeza de las reglas descubiertas sobre las asociaciones entre páginas Web visitadas por los navegantes. La aplicación tratará de encontrar el conjunto de estas asociaciones a partir de los archivos log ingresados.

- Secuencia de Patrones:

  Esta técnica proporciona una serie de relaciones de visitas de páginas con respecto al tiempo. También utiliza los parámetros de soporte y confianza para refinar los patrones de navegación deseados.

- Clusterización: Algoritmo K-medias.

  Esta técnica proporciona como salida k conjuntos de patrones sobre las características similares de navegación (páginas visitadas) de los usuarios. siendo k el número de agrupamientos en los que se desean clasificar los datos.

*Considerando los reportes estadísticos:*

- Páginas más visitadas
- URLs externos
- Tiempo de visita
- Tipo de navegador y Sistema Operativo
- Cantidad de usuarios que ingresan en el sitio Web
- Bytes de descarga de las paginas con las imágenes
- Número de visitas por página

### 3.1.2. Requerimientos no funcionales

**Interfaz**

Nuestra aplicación tendrá una interfaz Web que permita navegar por sus opciones a través de enlaces. En cada paso u opción se dará información necesaria al usuario.

**Archivos de entrada**

La aplicación aceptara como ingresos solo archivos en formato texto**.**

**Representación de Datos**

Los archivos log soportados por la aplicación serán en formato CLF o ECLF**.**

**Tiempo de respuesta.**

El nivel de procesamiento para el ingreso del archivo log depende del tamaño que tenga. Para el procesamiento de los algoritmos depende de la cantidad de registros que consideremos en los análisis.

## 3.2. Alcance del sistema

**Funciones**

Las funciones globales que nuestra aplicación brindará a los usuarios serán:

- Ingreso de archivo log: el usuario seleccionará la ruta del archivo log que contiene los datos.
- Selección de parámetros para la ejecución del análisis: el usuario podrá especificar ciertos parámetros para el análisis de los archivos log del servidor Web
- Selección de técnica de Minería de Datos. El usuario debe poder especificar la técnica que se desee para analizar los datos de los archivos log.

- Salida de información. El usuario podrá ver los resultados de los análisis de datos de manera organizada y con gráficos demostrativos para características que se presten a ello.

**Rendimiento**

Nuestra aplicación pretende no consumir demasiado tiempo de respuesta, así sea el archivo log demasiado pesado

**Interfaces**

Se pretende contribuir a las buenas normas de Interacción Hombre Máquina, presentando una aplicación, intuitiva y con constante retroalimentación al usuario. Trabajaremos usando paginas Web con el beneficio de que muchos usuarios hoy en día no están ajenos al uso de este tipo de interfaces.

## 3.3. Componentes del sistema

La aplicación Web esta conformada por tres capas principales:

1. Interfaz de Usuario

   Esta formado por formularios que permitirán la selección e ingreso de un archivo log del servidor Web. Así como también por formularios donde muestra las salidas de los algoritmos usados para el descubrimiento de conocimiento correctamente formateados y con gráficos si los datos aplican.

2. Lógica del Sistema

   Esta formado por reglas y funciones que aceptan opciones recibidas desde la interfaz de usuario para ejecutar el análisis de los datos del archivo log.
   Además de la ejecución de las técnicas de Minería de Datos seleccionadas por el usuario.

3. Controlador de Acceso de Datos

Controla el acceso a la base de datos en donde se encuentra almacenada la información.

# 4. DISEÑO E IMPLEMENTACIÓN

Este capítulo incluye detalles del diseño que hemos utilizado para realizar este proyecto, así como también las particularidades al momento de realizar la implementación.

## 4.1. Diseño

En esta sección se presenta detalles del diseño como los componentes que forman al sistema, los casos de uso con sus funcionalidades, descripción del usuario que interviene, las clases, la interacción entre sus objetos, realizamos el diseño de la base de datos y el diseño de la interfaz del usuario. Este conjunto de diseño cubre todas las especificaciones del sistema debidamente detalladas en la sección 3.1.

### 4.1.1. Diseño de componentes del sistema

En esta sección describiremos la arquitectura y los componentes que forman el sistema. Además se ilustrará las principales funciones de cada uno de los componentes y la interacción entre ellos.

Para el desarrollo de esta tesis nos hemos propuesto emplear una arquitectura de múltiples capas. El sistema está conformado por tres capas principales:

Interfaz de Usuario: capa encargada de dotar la funcionalidad necesaria para la interacción entre el usuario y la aplicación. Esta capa a su vez esta formada por dos subcapas: Presentación y Proceso de Presentación.

Lógica del Sistema: Esta capa es la encargada de resolver las reglas del sistema especificadas para el dominio del problema. Esta capa se encuentra dividida en dos subcapas: la primera es el depurador y sesionizador y la segunda es el procesador de algoritmos de Minería Web.

Controlador de Acceso a Datos: esta capa es la parte del sistema encargada del dotarle la persistencia al proyecto. Controla el acceso a la base de datos en donde se encuentra almacenada la información.

Las tres capas son vitales y de gran importancia, sin embargo, el núcleo del sistema lo forma el procesador de algoritmos de Minería Web ya que nos permitirá generar los reportes para poder tomar la decisión más óptima del negocio.

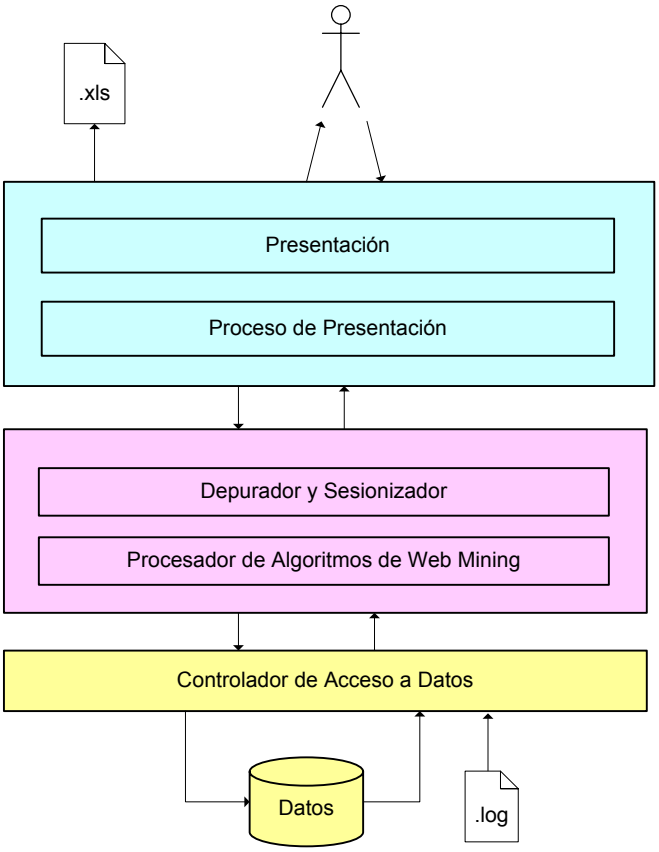



**Figura 4.1 Diagrama de Componentes del sistema**

## Acoplamiento de componentes

Como lo describimos anteriormente hemos dividido al sistema en tres capas y estas a su vez se encuentran divididas en subcapas, las que se encuentran representadas en la figura 4.1, dichas capas son los siguientes:

1. Interfaz de Usuario
2. Lógica del Sistema
3. Controlador de Acceso a Datos

*Interfaz de Usuario:*

Esta capa es la encargada de suministrar la funcionalidad necesaria para la interacción entre el usuario y la aplicación. A esta capa la conforman dos subcapas:

Presentación: Se encarga de dotar únicamente la funcionalidad de aspecto, y recibir las peticiones del usuario que en algunos casos serán validadas.

Entre las peticiones que recibirá se encuentra el ingreso del o los archivos log del servidor. Para poder ser almacenada la información, cada archivo log deberá tener formato CLF o formato ECLF. Además del o los archivos log, se deberá ingresar parámetros indispensables para el procesamiento de las diferentes técnicas de minería.

Proceso de Presentación: capa que se encarga de recibir las peticiones del usuario, gestionando la navegación del interfaz de usuario, y redireccionando las peticiones a la Lógica de Sistema, la cual procesará la petición y devolverá un resultado al Proceso de Presentación.

*Lógica del Sistema*

Esta capa resuelve las reglas del sistema especificadas para el dominio del problema. Se encarga básicamente de la preparación de la información que se ha almacenado en la base de datos para luego poder aplicar una técnica de minería. Como lo habíamos mencionado se encuentra dividida en dos subcapas:

El depurador y sesionizador: esta parte del sistema se encarga de atender las peticiones que se reciben del cliente, convirtiéndolas en procesos que se realizará de forma transaccional o no, de forma síncrona o asíncrona. Entre las peticiones que recibe se encarga de la limpieza y validación de los datos que se encuentran en el archivo log del servidor para luego ser almacenados en la base de datos

El procesador de algoritmos de Minería Web: esta subcapa es el núcleo del sistema, se encarga de ejecutar los algoritmos de minería de datos que se han diseñado para el sistema. Para cada técnica se deberá ingresar parámetros adicionales, que sirven exclusivamente para el proceso de cada técnica.

*Controlador de Acceso a Datos*

Esta capa controla la persistencia de los datos del proyecto. Es la encargada de la consistencia y mantenimiento de los datos. De los archivos log se extraerá la información necesaria para ser almacenada en la base para su posterior procesamiento. Una vez que el depurador y sesionizador valide la información, esta capa almacena los datos. Se encarga de proporcionar de información al procesador de algoritmos de Minería Web.

### 4.1.2. Diseño orientado a objetos

En esta sección se presenta los casos de uso del sistema y sus respectivas funcionalidades, las características del actor que interactúa con el sistema, además de los diagramas de interacción de objetos que plasman los principales procedimientos del proyecto.

#### Casos de Uso del Sistema

El sistema presenta 6 casos de uso en los que interviene un solo usuario definido con el rol de Administrador. A continuación se muestra el listado de casos de uso del sistema, los que a su vez se encuentran graficados en el diagrama de contexto de casos de uso de la figura 4.2:

**Lista de casos de uso:**

Caso de Uso 1: Ingresar archivo del log del servidor

Caso de Uso 2: Seleccionar parámetros para procesamiento

Caso de Uso 3: Presentar reporte

Caso de Uso 4: Reglas de asociación

Caso de Uso 5: Secuencia de patrones

Caso de Uso 6: Clustering

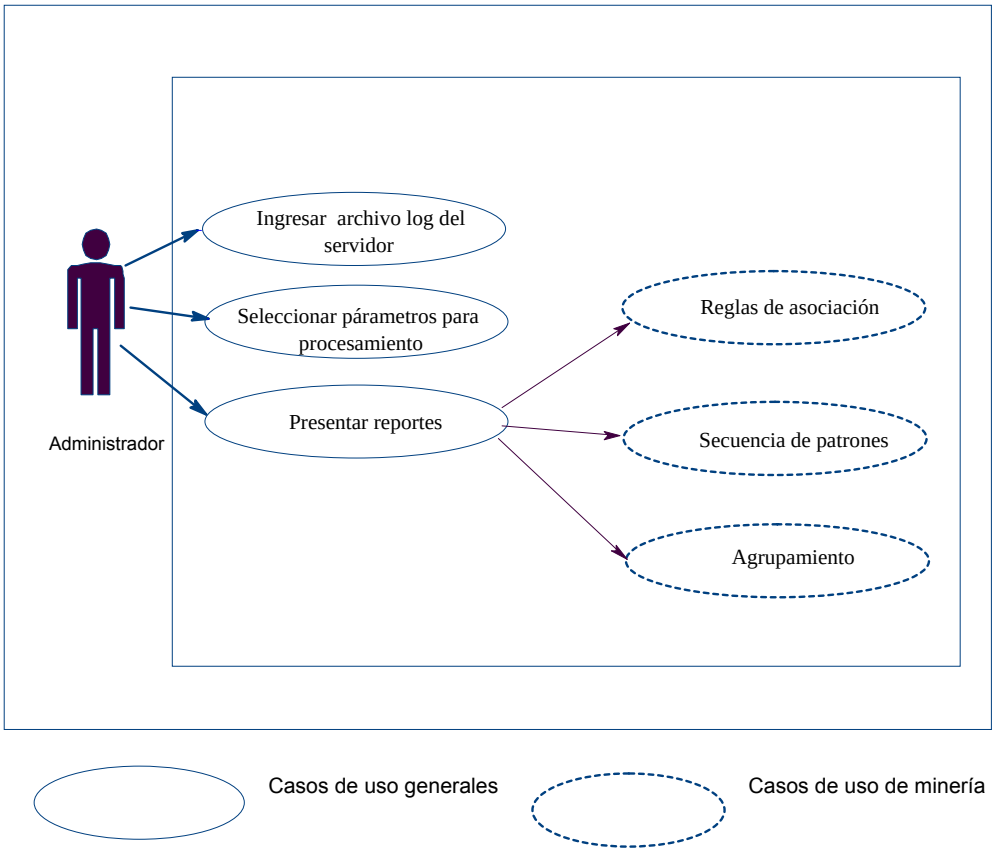



**Figura 4.2 Diagrama de contexto de casos de uso**

**Descripción de actores**

Como se mencionó anteriormente, en el sistema solo se ha definido un usuario para interactuar con el sistema, y tiene el rol de administrador, a continuación se detalla las características y obligaciones del mismo.

**Nombre**: Administrador

**Descripción**: Es el que se encarga de ingresar los archivos log del servidor, así como escoger los parámetros y la técnica de minería de datos para el procesamiento del log.

**Notas**: El administrador será el encargado de utilizar el sistema. Tomará las decisiones en base a los reportes que presentará el sistema.

Funcionalidad: Interviene como actor primario en los casos de uso 1, 2, 3, 4, 5 y 6.

**Descripción de los casos de usos**

Cada caso de uso tiene su funcionalidad e importancia definida, su debida documentación, y consideraciones especiales se muestra a continuación:

**Caso de uso 1:**

**Nombre**: Ingresar archivo del log del servidor

**Descripción**: Este caso de uso nos permite ingresar el archivo del log del servidor que será posteriormente analizado. El formato del archivo log del servidor podrá ser CLF y los ECLF.

**Notas**: El usuario deberá seleccionar la ruta del archivo log del servidor que contiene los datos.

**Caso de uso 2:**

**Nombre**: Seleccionar parámetros para procesamiento

**Descripción**: Este caso de uso nos permite escoger ciertos parámetros indispensables para el análisis de los archivos log del servidor.

**Notas**: Entre los parámetros que se podrá escoger tenemos:

Tiempo máximo de petición de un mismo host (tiempo de sesión).

Conjunto de archivos Web considerados.

**Caso de uso 3:**

**Nombre**: Presentar reportes

**Descripción**: Este caso de uso nos permite visualizar el reporte o resultados de los análisis de los datos de manera organizada.

**Notas**: El reporte será presentado de una forma organizada y con gráficos demostrativos para características que se presten a ello.

**Caso de uso 4:**

**Nombre**: Reglas de asociación

**Descripción**: Este caso de uso es una extensión del caso de uso 3, nos permite visualizar el reporte o resultados de los análisis de los datos de manera organizada. Utilizando la técnica de minería reglas de asociación

**Notas**: El reporte será presentado de una forma organizada y con gráficos demostrativos para características que se presten a ello.

**Caso de uso 5:**

**Nombre**: Secuencia de patrones

**Descripción**: Este caso de uso es una extensión del caso de uso 3, nos permite visualizar el reporte o resultados de los análisis de los datos de manera organizada. Utilizando la técnica de minería secuencia de patrones

**Notas**: El reporte será presentado de una forma organizada y con gráficos demostrativos para características que se presten a ello.

**Caso de uso 6:**

**Nombre**: Clustering

**Descripción**: Este caso de uso es una extensión del caso de uso 3, nos permite visualizar el reporte o resultados de los análisis de los datos de manera organizada. Utilizando la técnica de minería clustering

**Notas**: El reporte será presentado de una forma organizada y con gráficos demostrativos para características que se presten a ello.

Cada caso de uso presenta esczzenarios de vital importancia para el diseño del sistema, por lo que se encuentran documentados como un Anexo C en esta tesis.

## Clases

En esta sección detallamos las clases que se diseñaron para el sistema. Sus principales funciones, y la interacción entre ellas representada en el diagrama de interacción de clases.

Para el proyecto se han diseñado 13 clases, entre ellas 9 son para la ejecución de los algoritmos de minería, y las 4 restantes para manejar la información proveniente de la base.

Las clases según el algoritmo para las que fueron diseñadas se encuentran distribuidas de la siguiente forma:

Algoritmo clustering: clase KMEANS y clase CLUSTER

Algoritmo secuencia de patrones: clase ARBOL, clase RAIZ, clase PATRONES y clase NODOS.

Algoritmo reglas de asociación: clase APRIORI y clase APRIORI.

Las clases para manejar los datos provenientes de la base de datos se distribuyen de la siguiente forma:

REGISTRO_LOG: datos de la tabla Registro_Log

REGISTRO_SESION: datos de la tabla Registro_Sesion

REGISTRO_PAGINAS_SITE: datos de la tabla Registro_Paginas_Site

HTTP_INFO: datos de la tabla Http_Info

La interacción entre cada una de las clases descritas arriba se encuentran representada en el diagrama de interacción de clases de la figura 4.3 que se encuentra a continuación.

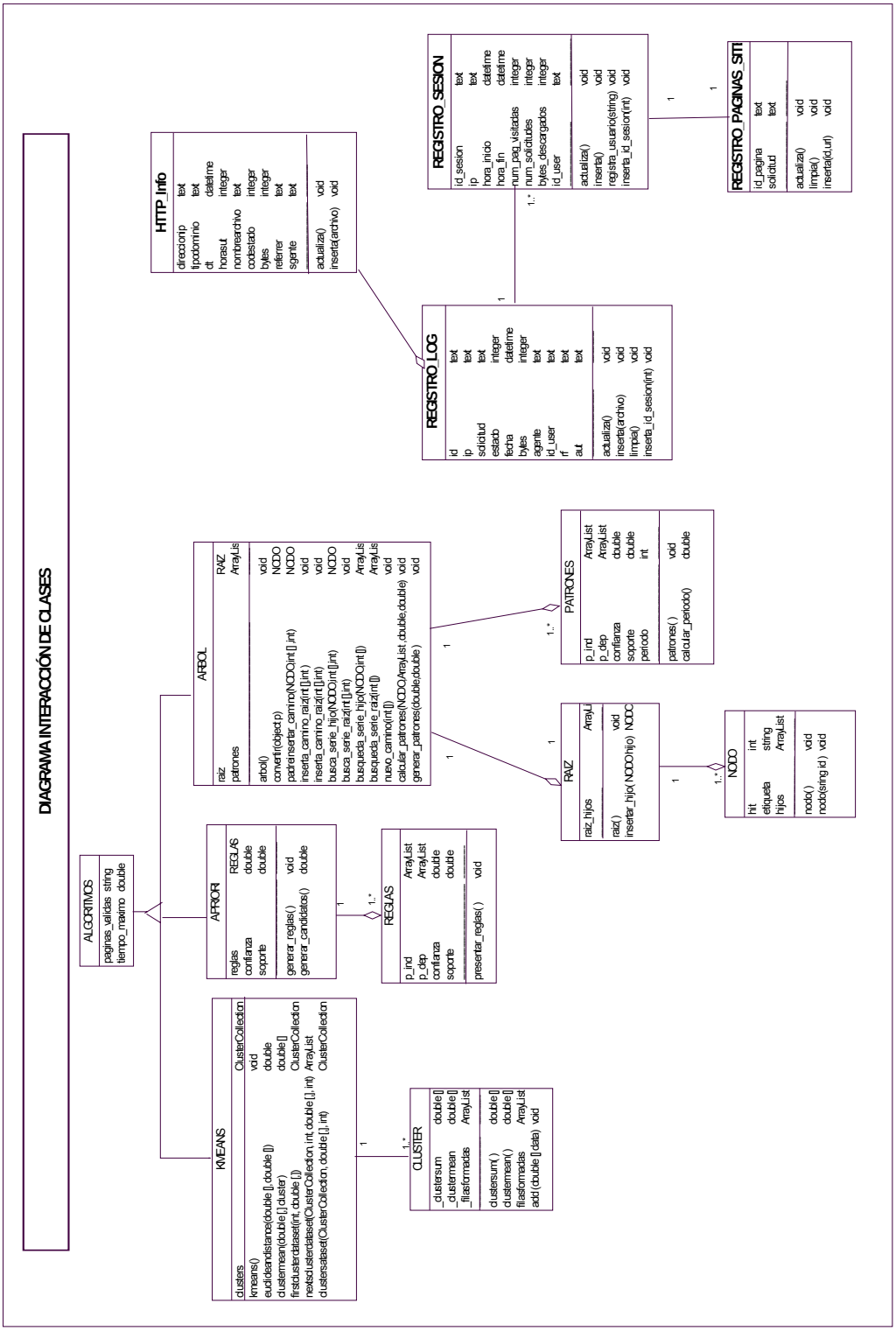



**Figura 4.3 Diagrama de interacción de clases**

### 4.1.3. Diseño de la base de datos

El sistema desarrollado para esta tesis necesita de un repositorio de datos, para lo cual se diseña una base de datos cuya estructura permita almacenar e identificar fácilmente a cada transacción, sesión y página, para poder emplearla según el algoritmo lo amerite.

**Modelamiento**

La base esta constituida por 4 tablas como: HTTP_Info, Registro_log, Registro_Sesion y Registro_Paginas_Site.

Como una breve descripción de cada tabla de la base de datos podemos decir que:

HTTP_Info: tabla que almacena todos las transacciones del archivo log.

Registro_log. Se genera a partir de limpieza y validación de los datos de la tabla HTTP_Info.

Registro_Sesion: almacena todas las sesiones que se encuentran en Registro_Log

Registro_Paginas_Site: almacena las páginas válidas del sitio Web.

El diseño de la base de datos está representado por medio del diagrama entidad-relación que mostramos a continuación en la figura 4.4:

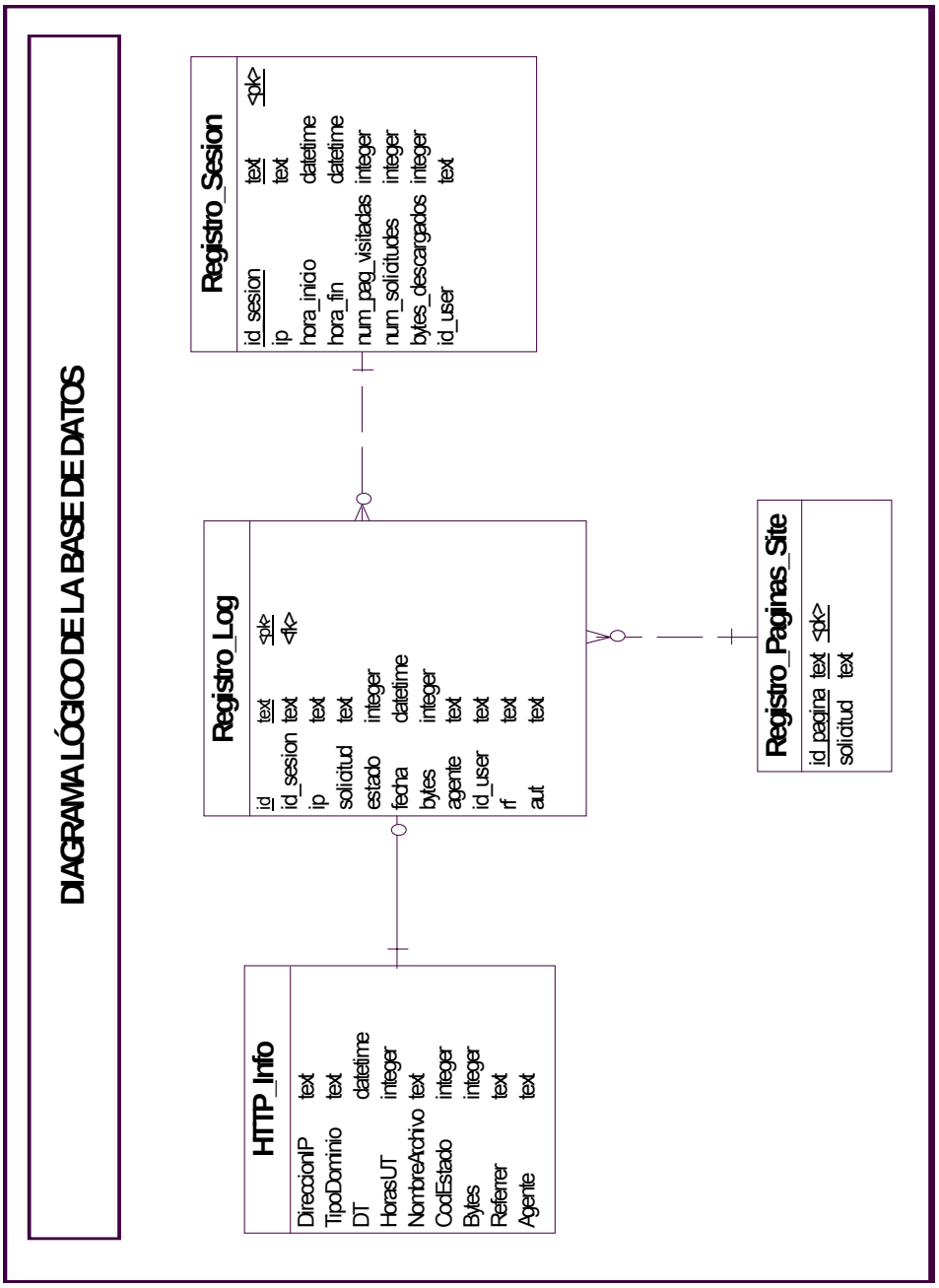



**Figura 4.4 Diagrama lógico de la base de datos**

### 4.1.4. Diseño de la aplicación e interfaz de usuario

Se ha diseñado un sistema en un ambiente Web con el propósito de generar reportes de forma rápida y efectiva, mediante el análisis de los archivos log del servidor del negocio vía Web. El sistema proporcionará varios tipos de

reporte. Entre los reportes generados tenemos: reporte para estadísticas de uso, reporte para agrupamiento, reporte para generación de reglas, reporte para secuencia de patrones. A continuación se muestra las distintas interfaces que forman parte del sistema.

El sistema tendrá una pantalla de inicio o bienvenida en donde se mostrará características y funcionamiento del sistema. En la figura 4.5 se muestra cual es la página de bienvenida desde donde inicia la aplicación.

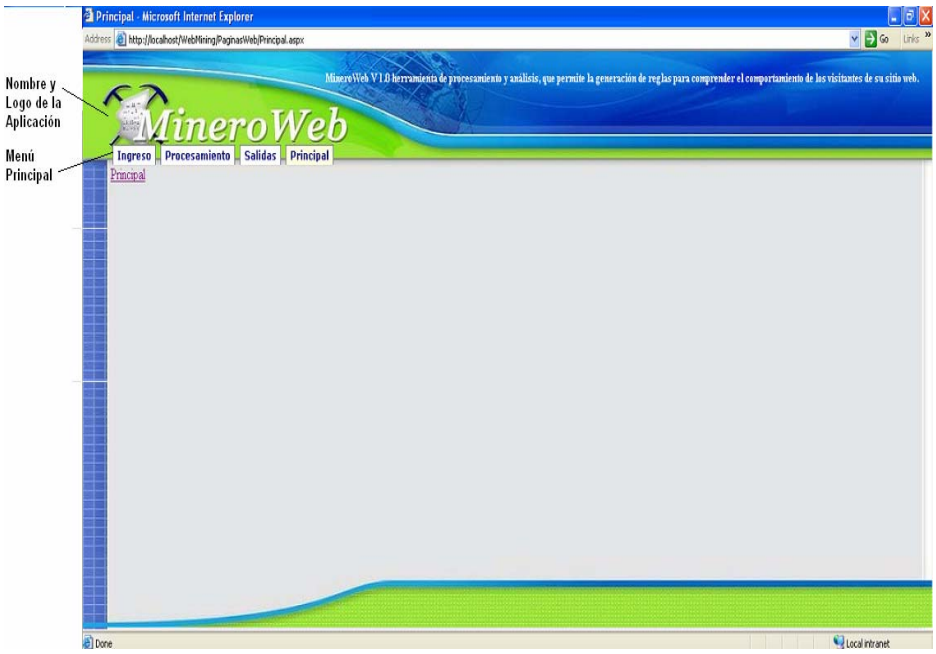



**Figura 4.5 Página de bienvenida**

Para empezar el proceso, y obedeciendo al diseño de componentes explicado anteriormente en la sección 4.1.1, hacemos el ingreso del o los archivo log del servidor. Seleccionando del menú principal la opción Ingreso de Archivo, se nos presenta la pantalla representada en la figura 4.6. El usuario deberá ingresar la dirección y el nombre del archivo log que se quiera analizar. Cuando tenga lista la dirección del archivo log se lo agrega y la aplicación mostrará en una tablas datos del archivo como:

- Número de archivo log agregado
- Nombre del archivo log.
- Formato del archivo log.
- Número de líneas que tiene el archivo log

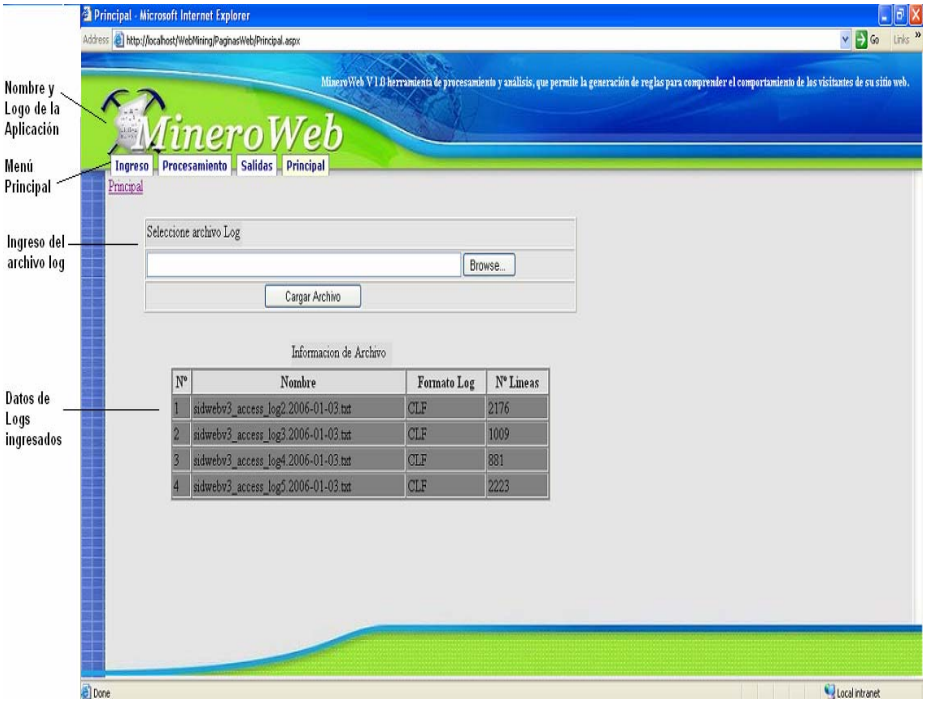



**Figura 4.6 Ingreso del archivo log del servidor**

A continuación se muestran las páginas que forman parte del procesamiento esencial para el análisis. Estás páginas representan al Procesador de limpieza de datos y sesionización pertenecientes al módulo de procesamiento de los componentes del sistema detallados en la sección 4.1.1. Sin este procesamiento, ningún reporte podrá ser presentado. Estas páginas son:

1. Página del proceso de limpieza del archivo log
2. Página de sesionización y selección de parámetros generales.

Una vez que hemos ingresado el o los archivo log, seleccionamos la opción Preprocesamiento del menú principal Procesamiento. La opción de preprocesamiento da inicio al proceso de limpieza.

El proceso de limpieza, es vital para la realización y optimización de los procesos de minería. Aunque resulte un poco abstracto, al realizar la limpieza estamos optimizando espacio y tiempo de respuesta, la reducción de la data a analizar puede disminuir notablemente. Se inicia la limpieza al activar el botón “Limpieza”. Cuando se ha concluido la limpieza, se presenta un gráfico estadístico que muestra el contraste de tamaños entre la data original del archivo log y la data resultante de la limpieza que es la que se va a analizar. Esta página se encuentra plasmada en la figura 4.7 que se encuentra a continuación.

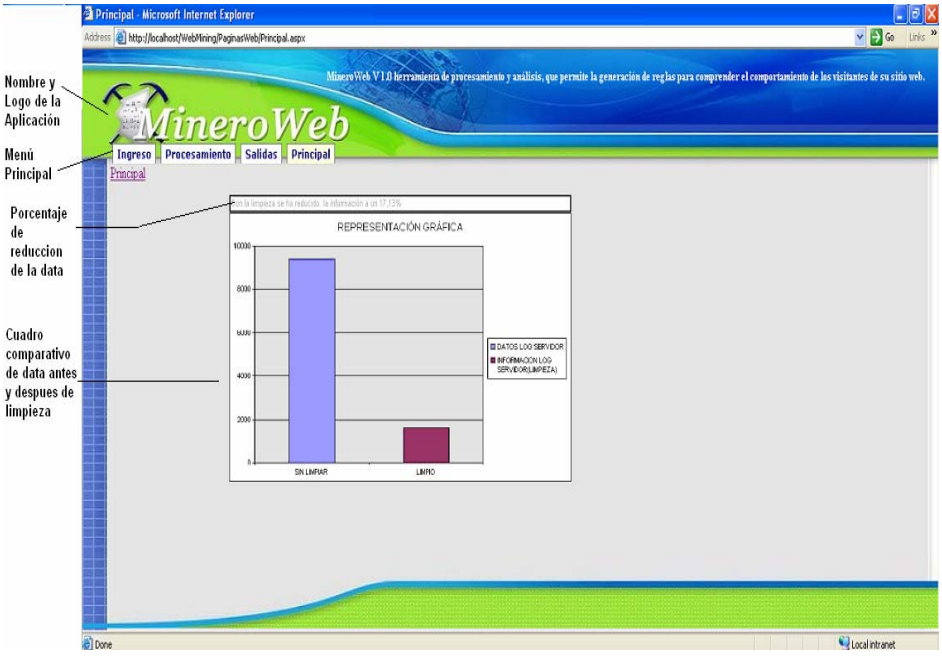



**Figura 4.7 Página del proceso de limpieza del archivo log**

Cuando se ha realizado la limpieza podemos continuar y seleccionar la opción de “Parámetros” del menú principal Procesamiento.

La página sesionización y de selección de parámetros generales (figura 4.8), permite establecer valores a variables que serán empleadas para la emisión de todos los reportes. Se debe escoger que páginas dentro del sitio Web serán consideradas como un nuevo requerimiento. Aquellas extensiones de páginas que no sean escogidas, serán descartadas al momento de emitir algún reporte. Además podrá escoger el tiempo máximo de sesión, lapso que validará si dos requerimientos realizados por el mismo usuario son o no parte de la misma sesión.

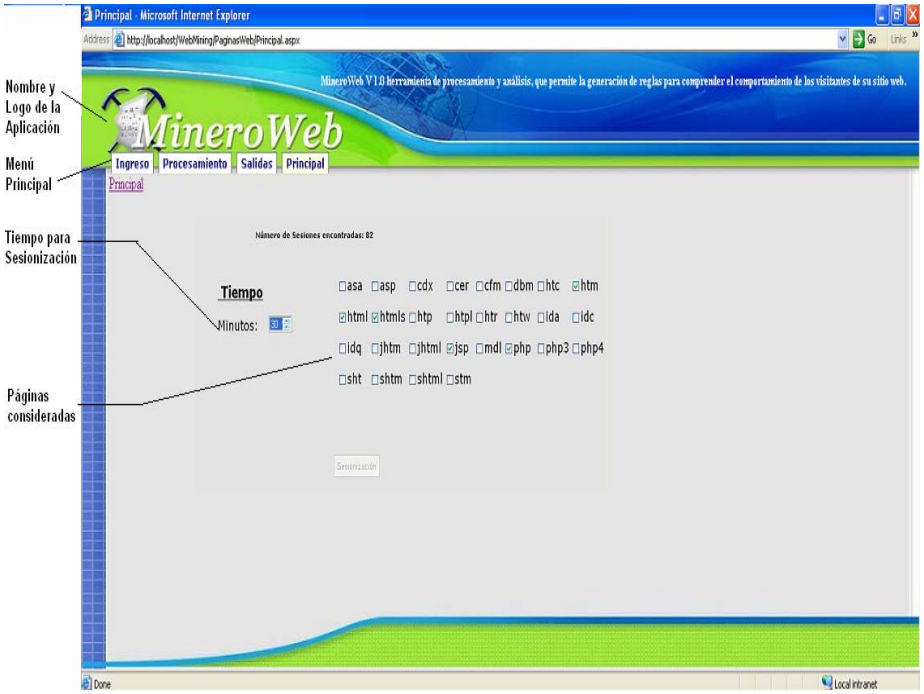



**Figura 4.8 Página de sesionización y selección de parámetros generales**

A continuación se detalla los tipos de reportes generados por el sistema una vez que se han realizado los procesos de limpieza y selección de parámetros:

*Reporte para generación de reglas de asociación*

En los reportes para generación de reglas de asociación, es necesario ingresar dos parámetros para poder realizar el análisis. Estos parámetros son: el soporte y la confianza. Estos parámetros serán los valores mínimos de soporte y confianza que puede tener cada regla generada. Es decir toda regla con una confianza y soporte menor a los establecidos por el usuario, serán depreciadas y no mostradas.

El número de reglas solo dependerá de la información existente y de los valores de soporte y confianza requeridos.

Cada reporte presentará el número de reglas generadas, el antecedente y precedente de cada regla, y la confianza y soporte real de cada regla.

La regla número uno que se presenta en la figura 4.9 nos dice, que el 85.71% de los usuarios que visitaron private/mycourses/website/email/index.jsp también visitaron private/mycourses/website/index.jsp.


**Figura 4.9 Reporte para generación de reglas de asociación**

*Reporte para secuencia de patrones.*

En los reporte para secuencia de patrones es necesario ingresar dos parámetros para poder realizar el análisis. Estos parámetros son: el soporte y la confianza. Estos parámetros serán los valores mínimos de soporte y confianza que puede tener cada patrón generado. Es decir todo patrón con una confianza y soporte menor a los establecidos por el usuario, serán depreciadas y no mostradas.

El número de patrones presentados solo dependerá de la información existente y de la confianza requerida.

El resultado que se mostrará son los distintos patrones que se generan al existir alguna secuencia entre las transacciones de un mismo usuario a lo largo del tiempo. En general, todos los algoritmos que se implementarán enfocan el análisis sobre secuencias de tiempo ya que los eventos que son almacenados están muy relacionados con el tiempo en que se producen.

Cada reporte presentará el número de patrones generados, el antecedente y precedente de cada patrón, y la confianza real de cada uno.

Las diferencias básicas entre el reporte de reglas de asociación de la figura 4.9 y el reporte de secuencia de patrones es que este último hace una aproximación de tiempo de ocurrencia y que muestra de forma secuencial las páginas por las que navegaron los usuarios para generar el patrón.

El patrón número uno que se presenta en la figura 4.10 nos dice, que el 88.88% de los usuarios que visitan la secuencia de páginas private/mycourses/website/folders/assignment/assignment.jsp, private/mycourses/website/folders/content.jsp, van a visitar private/mycourses/website/folders/groupView.jsp en aproximadamente 0 dias.

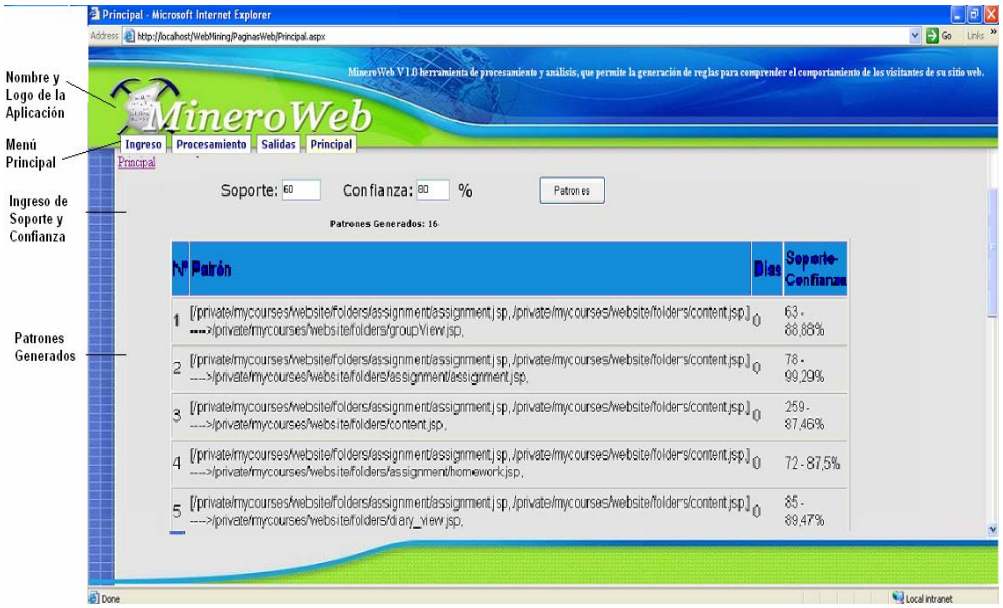



**Figura 4.10 Reporte para secuencia de patrones.**

*Reporte para clustering*

En la figura 4.11 podemos apreciar el reporte que se emite cuando aplicamos el método de clustering a la data almacenada por el sistema. Podemos realizar la clusterización siempre y cuando se haya realizado el ingreso y procesamiento del archivo log. Un requerimiento esencial es que el administrador haga el ingreso del soporte mínimo que debe presentar cada cluster en que divide el sistema a las páginas del sitio.

El reporte mostrará datos como: número de sesiones encontradas, número de iteraciones necesarias para determinar los clusters, número de clusters encontrados. Así como los clusters con cada una de las páginas que lo conforman.

El reporte nos indica que existe grupos con características similares de navegación, en este caso se han generado 4 grupos o clusters, en donde el cluster número uno define similitudes entre las páginas private/mycourses/website/folders/announcement_view.jsp, private/mycourses/website/ index.jsp y private/mycourses/website/scores/index.jsp.

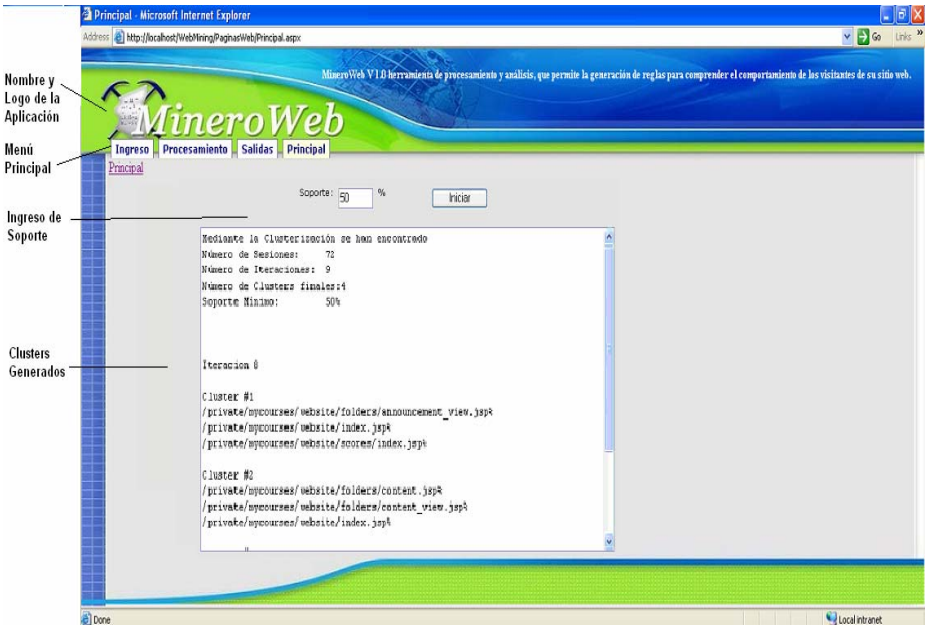



**Figura 4.11 Reporte para Clustering**

*Reporte para estadísticas de uso*

Se presentará estadísticas de tráfico de ciertos parámetros. Para poder acceder a estos reportes primero debe cumplir con realizar el ingreso del archivo, limpieza de la data y sesionización de las peticiones generadas en el archivo log. Luego seleccionamos la opción “Estadística de uso” del menú principal “Salidas”. Con esto podemos escoger entre los gráficos estadísticos que nos ofrece esta aplicación:

- Páginas más visitadas

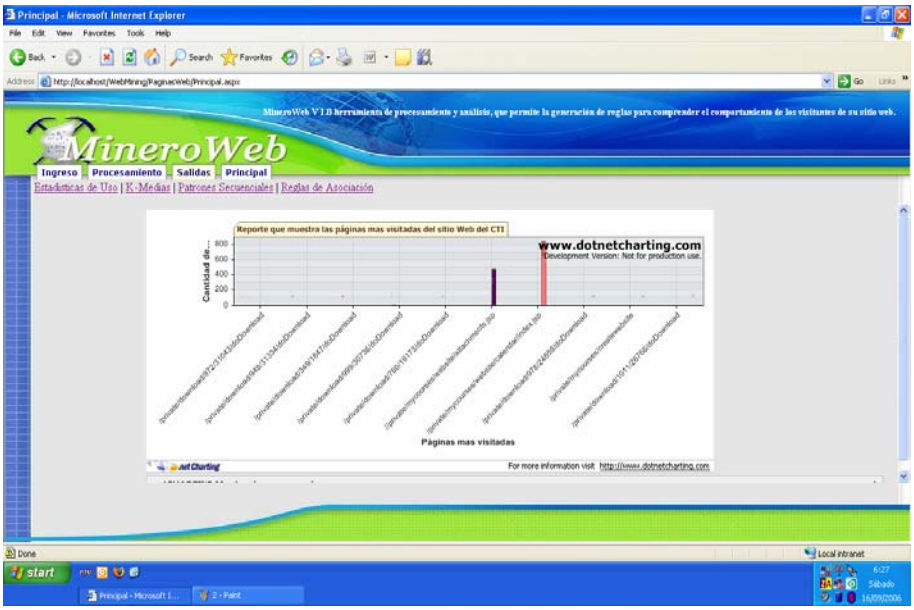



**Figura 4.12 Reporte páginas más visitadas**

- Tiempo de visita

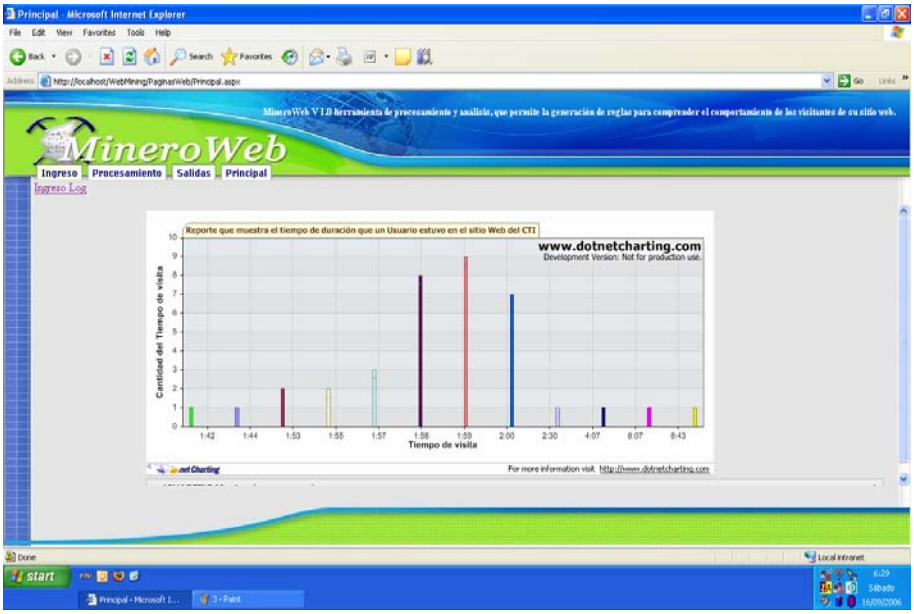



**Figura 4.13 Reporte tiempo de visita**

- Cantidad de usuarios que ingresan en el sitio Web

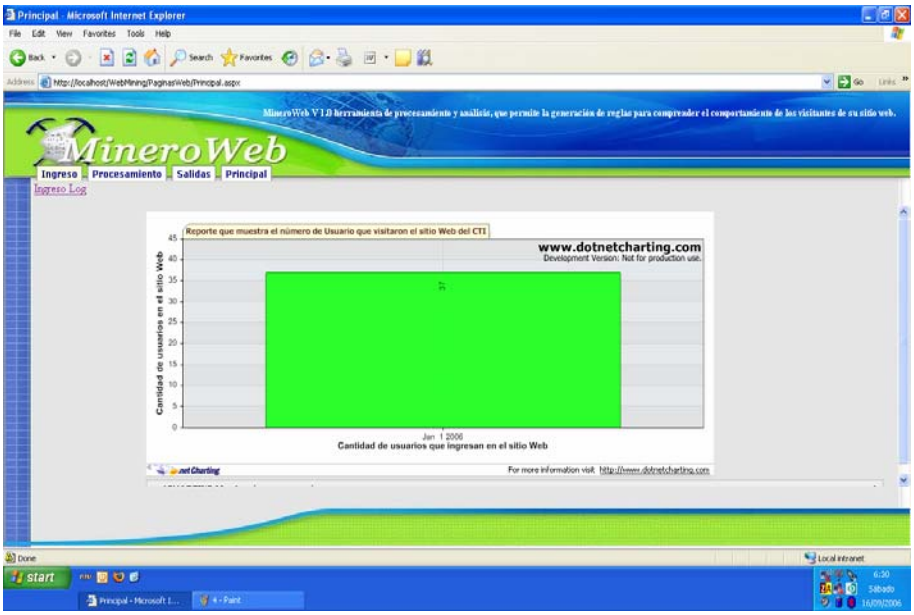



**Figura 4.14 Reporte cantidad de usuarios que ingresan en el sitio Web**

- Bytes de descarga de las paginas con las imágenes

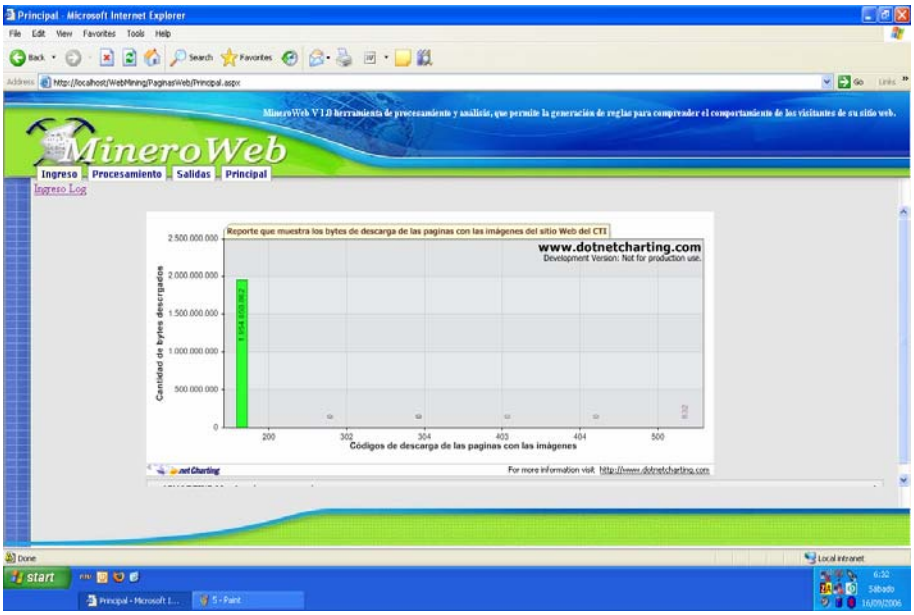



**Figura 4.14 Reporte cantidad Bytes de descarga de las paginas con las imágenes**

- Número de visitas por página

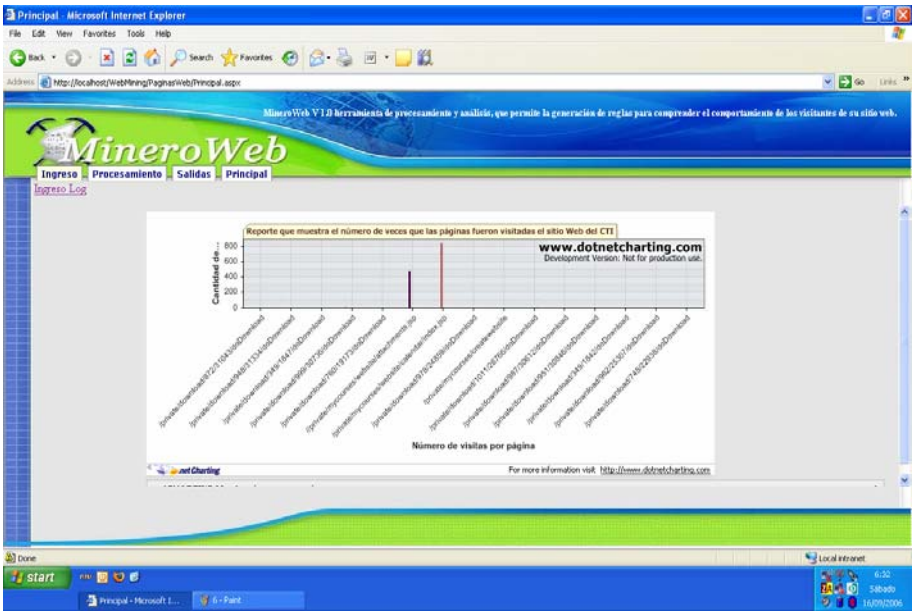



**Figura 4.15 Reporte números de visitas por página**

## 4.2. Implementación

En esta sección iniciaremos a explicar como se ha implementado y la lógica que hemos utilizado para procesar la información que proveen los archivos logs del servidor de un sito Web y la lógica para relacionar y emplear esa información con los algoritmos de minería de datos antes expuestos en la sección 2.3. La justificación del lenguaje de programación escogido para desarrollar la aplicación, así como de la base de datos.

**Lógica para procesar la información de los archivos logs del servidor**

Como se propuso anteriormente en el análisis del sistema (Capitulo 3), primero debemos realizar un procesamiento de la información inmersa en los archivos del servidor.

En forma general, el procedimiento planteado se detalla a continuación:

1. Ingreso y almacenamiento de los archivos log del servidor
2. Limpieza de los datos almacenados
3. Identificación de usuarios
4. Identificación de sesiones y reconocimiento de páginas consideradas como peticiones

1. *Ingreso y almacenamiento de los archivos log del servidor*

   Se proporcionará como entrada al sistema un conjunto de archivos log del servidor del sitio Web que se desea analizar.

   Para el ingreso y almacenamiento de los archivos log, se ha creado en la base de datos una tabla llamada http_info, en donde se almacena todos los campos que contienen los formatos de archivo log con los que trabaja nuestra aplicación descrito en la sección 3.1 y cuya estructura se encuentra en el anexo A.

   Una vez especificada la ruta del archivo, realiza la verificación de los mismos, para asegurarnos que el archivo tiene uno de los dos formatos permitidos.

   Identificado y validado el formato de archivo que se quiere ingresar, se procede a transformar y descomponer cada línea o transacción del archivo en datos útiles y entendibles para la base de datos. Se almacena esta información en la tabla Http_Info de la base de datos.

Si se ingresa más de un archivo log, esta tabla Http_Info no se encuentra ordenada cronológicamente entre archivos log, ya que a medida que se ingresan nuevos archivos se los añadirá a la tabla directamente.

2. *Limpieza de los datos almacenados*

Una vez que se ha llenado la tabla Http_Info (tabla que contiene la información sin depuración del archivo log), se debe realizar una limpieza sobre los datos anteriormente almacenados, solo los registros que son validados van a ser almacenados en una nueva tabla llamada Registro_Log. Por las características de la tabla Http_Info la limpieza se realiza sobre los datos de una vista ordenada por el campo DT (fecha de la transacción) de la tabla Http_Info.

Para que un registro sea considerado como válido debe cumplir los siguientes requisitos:

Todos los campos de cada registro dentro de la tabla http_info deben ser diferente al valor null.

El campo CodEstado debe tener alguno de los valores válidos para nuestro análisis. Es decir el campo CodEstado debe tener valores que comiencen con 2XX y 3XX, que son estados que emite el servidor cuando las peticiones o transacciones son exitosas.

El campo NombreArchivo debe almacenar una ruta de archivo diferente de los siguientes tipos de archivos: jpg, bmp, gif, exe, js, css, pdf, doc, txt.

Como mencionamos los registros son leídos de la vista ordenada por DT de la tabla http_info. Cada registro es analizado, si se considera como registro válido es almacenado en la tabla Registro_Log con su respectivo id.

Por tanto, la primera tarea consistirá en deshacernos de todos aquellos registros log que no cumplan algunos de los requisitos que acabamos de exponer.

3. *Identificación de usuarios*

Cuando la limpieza ya se ha realizado procedemos a la identificación de usuarios. Para poder identificar al usuario no basta con la IP que se registra dentro de los archivos log.

Debido al uso de proxy servers por parte de los proveedores del servicio de internet y de firewalls por parte de las corporaciones comerciales, la verdadera dirección IP del cliente no se encuentra disponible para el servidor de Web. En vez de tener varias direcciones IP distintas para varios clientes distintos, la misma dirección del proxy server o firewall es guardada en el log representando los requerimientos de diferentes usuarios que llegan al servidor desde el mismo proxy o firewall. Esto genera cierta ambigüedad en los datos del log y como el análisis para identificar el comportamiento de los usuarios debe basarse solo en las entradas del log, hemos decidido para poder identificar a los usuarios dentro del Registro_Log, seguir la heurística planteada en [11], que adaptada a nuestro proyecto es:

Registros dentro de tabla Registro_Log con mismo ip y con mismo agente (tipo de browser y sistema operativo de la máquina que hizo el requerimiento) se considerará como mismo usuario.

Como manejamos dos tipos de archivos log: ECLF y CLF, además que el campo agente solo esta presente en los archivos tipo ECLF, procedemos a setear el campo agente como "INDEFINIDO" cuando trabajemos con archivos CLF, manteniéndose así la heurística antes planteada.

La identificación de usuarios será notable al asignar un id_user a cada registro dentro de la tabla Registro_Log.

4. *Identificación de sesiones y reconocimiento de páginas consideradas como peticiones*

El siguiente paso es la identificación de sesiones y páginas consideradas como peticiones. Los datos necesitan una transformación que depende de las operaciones que se vallan a realizar sobre estos, esto se debe a que cada algoritmo requiere una entrada diferente. Por tanto, realizamos una transformación de los datos que nos proporciona una estructura fácil de manejar y adaptable en cada momento a cada uno del tipo de entrada de datos que requiera cada algoritmo.

La solución consiste en realizar una transformación de los datos de entrada a un formato general que hemos llamado Registro_Sesion. Luego, para cada una de las técnicas usadas, se realizará una nueva transformación que convierta el formato Registro_Sesion en el formato

adecuado para cada algoritmo. En términos de eficiencia es importante pues que el formato general Registro_Sesion posee una estructura que minimiza el esfuerzo realizado para transformar los datos para cada uno de los diferentes algoritmos.

A pesar de que más adelante se detalla con más claridad las tablas que vamos a usar dentro de la base de datos, para poder explicar la lógica de sesionización o identificación de sesiones es necesario detallar el formato de la tabla Registro_Sesión:

| | |
|---|---|
| id_sesion | identificador de la sesión. |
| id_user | identificador del usuario que realiza la sesión. |
| ip | ip del usuario que inicia la sesión |
| hora_inicio | fecha y hora que inicia la sesión. |
| hora_fin | fecha y hora que finaliza la sesión |
| num_pag_visitadas | número total de páginas accedidas en el sitio Web. |
| num_solicitudes | número total de peticiones realizadas en la sesión |
| bytes_descargados | total de bytes transferidos durante la sesión. |

Es importante recordar que debido a la estructura de los archivos log cada transacción o línea es realizada cronológicamente después de la línea o transacción anterior. Para realizar la sesionización seguimos el siguiente proceso:

Se recupera todos los registros del Registro_Log, en donde cada registro pertenece a un usuario identificado por id_user.

Seguimos la siguiente heurística: los registros que tengan el mismo id_user, y que la fecha en que realizó la transacción (hora_inicio) este dentro de un tiempo límite que establece el usuario (tiempo que va de 1 a 59 minutos), será considerada como parte de la misma sesión.

Cada sesión será almacenada en la tabla Registro_Sesion.
Cuando se crea una nueva sesión se almacena un nuevo registro y se le asigna su id_sesion, id_user, hora_inicio.

Si la sesión ya ha sido creada se actualizarán campos como: hora_fin, num_pag_visitadas, num_solicitudes, bytes_descargados.
El campo num_pag_visitadas se incrementa si la solicitud pertenece a una de las extensiones que el usuario ha validado.

Así como en este paso se requiere la acción del usuario ingresando el tiempo de sesionización, de igual forma el usuario ingresará las extensiones de archivos que desee se consideren como peticiones. Cuando el usuario hace la petición de una página del sitio, en el archivo se registra cada una de las partes de la página como gráficos, estilos, textos como transacciones diferentes. Es por esto que el usuario debe especificar que tipos de archivos dentro del sitio van a tener que ser considerados como peticiones distintas.

### 4.2.1. Técnicas de Minería de Datos

En esta sección explicaremos la lógica utilizada para implementar los algoritmos de minería de datos con los datos previamente procesados.

*Lógica para emplear la información con cada algoritmo de minería de datos elegido*

Algoritmo: Reglas de asociación

Para poder ejecutar el proceso de reglas de asociación a la data resultante de los archivo log, debemos realizar unos ajustes o parametrización sobre los datos.

Es importante definir sobre que datos vamos a trabajar, en primer lugar necesitamos los datos de las sesiones con todas cada una de las páginas accedidas, que se encuentran en las tablas Registro_log y Registro_Sesion.
De igual forma creamos dos clases que son:
APRIORI.- será la encargada de generar las reglas..
REGLAS.- tendrá el antecedente y el precedente del patrón, la confianza y soporte de la regla.

La lógica para realizar la parametrización de los datos es la siguiente:
Se construirá una matriz denominada historico de mxn donde m representa al número de sesiones creadas y n el número de páginas del sitio, es decir el elemento historico[i,j] representa la página j en la sesión i.

Si un usuario ha visitado la página i, historico [i,j] tendrá un valor de “1”, en donde indicará que la página fue visitada en la sesión i.
Si en la sesión no ha visitado la página i, el valor de historico [i,j] será igual a “0”.

Una vez parametrizados los datos, representados en la matriz historico, ejecutamos el algoritmo A priori [12], donde tomará como parámetros de entrada la matriz historico, además de la confianza y soporte estipulado por el administrador.

Apriori solo trabaja con datos cuantitativos. El algoritmo A priori se lo ha adjuntado como anexo pero básicamente sigue el proceso detallado a continuación:

Realiza varias pasadas sobre los datos.
En la primera, a partir de las URLs o páginas individuales, representados por los índices de las columnas de la matriz histórico, obtenemos los primeros conjuntos de grupos grandes de páginas y primeros antecedentes de las reglas ( si el soporte es 2, determinamos qué páginas aparecen en, al menos, 2 sesiones o filas de la matriz).

Con este grupo de páginas buscamos para cada uno en el resto de las columnas de la matriz histórico los consecuentes, es decir aquellas páginas que han sido visitadas al igual que el antecedente. Calculamos la confianza para cada regla, obteniendo así el primer grupo de reglas.

En cada paso subsiguiente, partimos del conjunto de antecedentes hallado en la pasada anterior.

Creamos un nuevo conjunto de grupos a partir de todas las combinaciones válidas de grupos de páginas iniciales (candidatos). Una combinación es válida en una sesión si todas las páginas pertenecen a la sesión.

Nos quedamos con los candidatos que cumplen el soporte mínimo y volvemos a empezar

El formato que presentamos las reglas de asociación es el siguiente:
El antecedente viene delimitado por unos corchetes ([ ]), y nos indica la página o páginas que fueron visitadas.

El consecuente se encuentra después de la flecha (→), y nos indica la página que será visitada si el antecedente se cumple. Además se muestra el soporte y confianza de cumplimiento de la regla encontrada.

| **Nº** | **Regla** | **Soporte-Confianza** |
|---|---|---|
| 1 | [a.html,b.htm,c.php] → d.html | 2 - 59% |
| 2 | [a.html,x.jsp]→ e.html | 6 -70% |
| 3 | [a.html,b.htm]→ c.php | 8 - 60% |

**Tabla 4.1 Formato de salida de reglas de asociación**

Con estos resultados podemos deducir que el 70% de usuarios que acceden a a.html y a x.jsp visitarán e.html.

Algoritmo: secuencia de patrones

Para encontrar las normas que generalicen el comportamiento de los visitantes del sitio Web, por medio de la secuencia de páginas en su historial de visitas en una sesión y para poder determinar una aproximación de día necesarios para que el patrón se cumpla, se decidió emplear secuencia de patrones, con un método implementado por Óscar Marbán Gallego [13], en donde plantea un algoritmo basado en estructuras auxiliares tipo árbol, la

estructura en la que está basado el algoritmo se ha denominado FBP-árbol (árbol de caminos frecuentes de la conducta ). Este algoritmo es una extensión del algoritmo A-priori, pero que considera la secuencia de páginas de visita. El FBP-árbol representa caminos dentro del sitio Web.

El algoritmo que seguimos se encuentra detallado en el anexo B pero los pasos básicos del algoritmo que seguimos son los siguientes:

Construimos una matriz de transición llamada TM. TM es una matriz nxn, donde n el número de páginas del sitio. TM[i,j] representa el número de veces que los usuarios han visitado la página j después de la página i.

Una vez construida la matriz TM, todas las celdas con valor menor a un umbral, el que esta estimado como un porcentaje de la data existente, toman el valor 0, por considerarse datos muy escasos que no generan resultados. La nueva matriz se llama FTM.

Con la matriz FTM, obtenemos los caminos frecuentes. La matriz de FTM se usa para la generación de los caminos y para realizar una poda de caminos (espacio de búsqueda). Esto se hace teniendo en cuanta que si un 2-camino(camino con 2 páginas) no es frecuente un 3-camino(camino con 3 páginas) del que sea subcamino el 2-camino tampoco lo será (propiedad Apriori). Los caminos se almacenan en una estructura multiárbol que se ha denominado FBP-árbol. En donde cada hoja o nodo del árbol representa a una página y tiene almacenada una etiqueta que identifica a la misma, además de datos como fechas en que se visitó por primera y última vez la página.

Usando el FBP-árbol se calcularán las reglas comportamiento-frecuentes de las que se desprenden los patrones. Para calcular los patrones se hace uso de los soportes de los caminos frecuentes (número de veces que el camino aparece en el conjunto de sesiones), se recorre la estructura FBP-árbol desde las hojas al nodo raíz y considerando el soporte de cada camino, para cada nodo: si el nodo no tiene más hermanos (esto significa que desde la página anterior sólo se alcanza esta página) entonces se sube al nodo anterior hasta encontrar un nodo que tenga algún hermano. Cuando un nodo tiene al menos un hermano, entonces hay más de un camino posible, por lo tanto se puede generar más de una regla. Estos nodos representan los puntos de ruptura y se marcarán como las páginas de punto de ruptura. En las páginas de punto de ruptura, el usuario puede seleccionar entre varios caminos frecuentes. Si la página no es una página de ruptura, solo hay un posible camino frecuente con lo que se puede conocer cual será probablemente la próxima página visitada.

Una vez se encuentran estos puntos de ruptura en el árbol, hay que obtener los patrones calculando la probabilidad condicional de cada patrón y el tiempo en días que se debe tomar desde que se da el antecedente del patrón hasta que se cumpla el precedente.

Como se detalló anteriormente los patrones están formadas por dos partes: el antecedente y el consecuente, por lo que los antecedentes serán tomados desde la raíz hasta el punto de ruptura, y el consecuente desde el punto de ruptura hasta la raíz.

El formato que presentamos los patrones generados es el siguiente:

El antecedente viene delimitado por unos corchetes ([ ]), y nos indica la página o secuencia de páginas que fueron visitadas.

El consecuente se encuentra después de la flecha (→), y nos indica la página o secuencia de páginas que serán visitadas si la secuencia antecedente se cumple. Además se muestra el soporte y confianza de cumplimiento del patrón encontrado.

| Nº | Patrón | Dias | Soporte-Confianza |
|---|---|---|---|
| 1 | [a.html,b.htm,c.php] → d.html | 0 | 2 - 59% |
| 2 | [a.html,x.jsp]→ e.html | 1 | 6 -70% |
| 3 | [a.html,b.htm]→ c.php | 0 | 8 - 60% |

**Tabla 4.2 Formato de salida de patrones secuenciales**

Con estos resultados podemos deducir que el 70% de usuarios que acceden a a.html y luego a x.jsp visitarán luego de 1 día a e.html, pudiendo transformarse esta información en una estrategia como mostrar propaganda de la página e.html directamente en la a.html

Algoritmo: clustering

Para poder ejecutar el proceso de clustering a la data resultante de los archivo log, debemos realizar unos ajustes o parametrización sobre los datos.

Es importante definir sobre que datos vamos a trabajar, en primer lugar necesitamos los datos de todos los usuarios con cada una de sus sesiones, que se encuentran en las tablas Registro_log y Registro_Sesion. Es importante recalcar que un usuario del sitio Web puede haber generado más

de una sesión y que cada sesión puede tener más de una solicitud o transacción.

La lógica en que se basa la parametrización de los datos es la siguiente:
Se construirá una matriz denominada matriz de mxn, donde m representa al número de usuarios del sitio y n el número de páginas del sitio, es decir el elemento matriz[i,j] representa al usuario i visitando la página j en algunas de las sesiones iniciadas por el usuario i.

Si el usuario j ha visitado la página i, el valor de la matriz[i,j] será igual a 1.
Si el usuario j no ha visitado la página i, el valor de la matriz[i,j] será igual a 0.

Con la parametrización de los datos, representados en la matriz [i,j], procedemos a ejecutar el algoritmo K medias, en donde el algoritmo tomará como parámetros de entrada la matriz y el soporte estipulado por el administrador.

Como ya se mencionó anteriormente, el algoritmo que hemos elegido para nuestro proyecto es el del K-medias de MacQueen [14], cuyo algoritmo se lo ha adjuntado como anexo. Este algoritmo solo trabaja con datos cuantitativos, es por eso que primero se ha realizado la parametrización de los datos. El sistema agrupará en forma aleatoria las filas de la matriz, inicialmente se crean 4 clusters a los que serán adjuntados cada una de las filas de la matriz. De igual forma en cada cluster será asignado de forma aleatoria un miembro del cluster como eje central o también llamado centroide del cluster. A cada miembro se le sacará la distancia euclideana

entre él y el centroide de cada cluster. El miembro será reasignado al cluster con quien tenga menor distancia euclideana.

Si la distancia euclideana es menor con el mismo cluster, entonces no será reasignado. Este proceso se lo realiza con todos los miembros de cada cluster.

Se reelige centroides o ejes centrales dentro de cada cluster y se vuelve a calcular la distancia euclideana entre clusters.

El proceso se detiene cuando los cluster generados no cambien de miembros.

El reporte que presentamos con el clustering presenta los siguientes datos: número de sesiones encontradas, número de iteraciones necesarias para determinar los clusters, número de clusters encontrados, soporte mínimo que desea el administrador, así como los clusters con cada una de las páginas que lo conforman en donde cada página tendrá su soporte real dentro de cada cluster.

La interpretación que se le da a estos resultados es que todas las páginas dentro del mismo cluster presentan una fuerte relación entre ellas, y tienen una diferencia con las páginas de otros clusters.

### 4.2.2. Lenguajes de programación

Los lenguajes de programación para el desarrollo de esta tesis, han sido escogidos entre otras cosas por la facilidad que brindan al momento de

traducir el diseño descrito anteriormente, la disponibilidad de las herramientas de desarrollo.

Como requerimiento propio del sistema, la aplicación tenía que ser desarrollada en un ambiente Web, el lenguaje de programación utilizado es ASP que incluye los lenguajes de scripts JavaScript y Visual C–Sharp, lenguajes necesarios para la generación de páginas de contenido dinámico.

Cabe indicar que ASP y Visual C-Sharp forman parte de una ambiente integrado de desarrollo, denominado Visual Studio, el cual permite el manejo de datos, su programación, compilación y depuración dentro del mismo ambiente.

Podríamos haber usado otras herramientas tales como Java pero nos decidimos por la primera opción debido al dominio de dicho lenguaje y la falta de experiencia en este tipo de proyectos en el lenguaje de Java.

Otros motivos por los cuales se optó por la opción de utilizar la herramienta de Visual Basic.net para el sistema de Minería Web fueron los siguientes:

1. Esta respaldado a través de una licencia de una marca reconocida (Microsoft).
2. Debido al previo conocimiento de la herramienta por las integrantes del grupo.
3. Por la funcionalidad fundamental de Visual Basic.net que sigue siendo familiar, flexible, sencilla e intuitiva
4. El soporte técnico es otorgado por Microsoft

5. Explota todas las características del Sistema Operativo Windows
6. El desarrollo de las interfaces graficas es rápida y sencilla debido a su conjunto de objetos gráficos.
7. Es una herramienta totalmente orientada a objetos.

# 5. PRUEBAS

En el siguiente capítulo describiremos los detalles de la fase de prueba que hemos desarrollado para el sistema. Encontraremos detalles como: tipos de prueba, objetivos de la misma, resultados obtenidos, tiempos de respuesta, aceptación del usuario.

La prueba del software es un elemento crítico para la garantía de calidad del mismo y representa una revisión final de las especificaciones, del diseño y de la codificación.

## 5.1. Objetivos

Los objetivos generales que nos hemos propuesto con las pruebas son los siguientes:

1. Descubrir mediante un proceso de ejecución algún error sea este de tipo lógico o técnico.
2. Corregir o implementar soluciones a los errores que se encuentren.
3. Asegurar la calidad del producto que se ha desarrollado.
4. Minimizar el riesgo de fallo en el producto.

Para esta fase hemos optado por realizar dos tipos de pruebas conocidas: las pruebas de caja blanca y las pruebas de caja negra.

1. Pruebas de caja negra: Puesto que conocemos la función específica para la que fue diseñado el producto, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa, y al mismo tiempo que encuentren posibles errores en cada función.

2. Pruebas de caja blanca: como conocemos el funcionamiento del producto, hemos desarrollado pruebas que aseguran que todas las piezas encajan, esto es que la

operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada.

## 5.2. Pruebas de caja negra

Hemos llevado a cabo una serie de pruebas sobre la interfaz del software, con ello hemos demostrado que las funciones de software son operativas, esto es, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, también hemos comprobado que la integridad de los datos externos al programa se mantiene.

Los objetivos que perseguíamos con este tipo de prueba son:

1. Encontrar funciones incorrectas o ausentes.
2. Encontrar errores de interfaz.
3. Encontrar errores en las estructuras de datos.
4. Encontrar errores de rendimiento.
5. Encontrar errores de terminación y de inicialización.

Para conseguir los citados objetivos hemos desarrollado dos tipos de pruebas en este sentido:

- Análisis de los valores límite
- Prueba de interfaces gráficas de usuario

### 5.2.1. Análisis de los valores límite

Por razones que no están del todo claras, los errores tienden a darse más en los límites del campo de entrada que en el "*centro*". Por ello, hemos desarrollado un análisis de valores límites. Para la realización y ejecución de estas pruebas hemos desarrollado la siguiente estrategia:

Hemos generado entradas con valores límite, esto es:

- Para el caso de ingreso del archivo log:
    - Hemos generado archivos log que no concuerdan con el formato de archivo para el que hemos diseñado la aplicación, esto es, archivos con entradas erróneas para comprobar que en esta fase la aplicación funciona como se esperaba. Se ha realizado la ejecución del ingreso del archivo con los archivos descritos anteriormente comprobando así el correcto funcionamiento de esta sección.

- Para el caso de la limpieza de datos:
    - Generamos archivos idóneos para la fase de ingreso pero que contenían valores límites para esta sección, es decir archivos en donde no existía información que desechar, archivos en donde toda la información era desechable. Después de sucesivas ejecuciones con distintos tipos de archivos con las restricciones que hemos descrito hemos concluido que esta parte de la ejecución es correcta.

- Para el caso de sesionización:
    - Entre los archivos que generamos, una parte de estos estuvo destinada a esta sección, archivos que luego de la fase de ingreso y limpieza obtendríamos información con los siguientes valores límites: todos los registros pertenecientes a un mismo usuario, todos los registros dentro del límite de tiempo de sesión pero con distinto usuario, todos los registros dentro del límite de tiempo de sesión y con mismo usuario. Tras someter a este tipo

de información a esta sección hemos comprobado su buen funcionamiento.

- Para el caso de las Reglas de Asociación:
  - Para este algoritmo hemos establecido los siguientes valores límite:
    1. Todos los registros tienen el mismo valor
    2. No tenemos registros
    3. Introducir valores erróneos de soporte y confianza, este caso no se puede dar puesto que se filtran los valores de entrada, estando estos comprendidos entre 1 y 100 para ambos casos, con lo que la respuesta del sistema es correcta.

    Para los dos primeros casos la respuesta del sistema sería, que no se ha obtenido ninguna regla útil.

- Para el caso de Clustering:
  - Para este algoritmo hemos establecido los siguientes valores límite:
    - Todos los registros tienen el mismo valor, para este caso la respuesta del sistema es que se agrupan en un solo grupo.
    - No tenemos registros, para este caso el sistema indica que no se ha encontrado ninguna agrupación.

Por tanto y después de todo lo observado, concluimos que la aplicación en lo referente a su modelo fundamental, sin tener en cuenta lo referente a la estructura lógica interna de la misma, es correcta.

### 5.2.2. Prueba de interfaces gráficas de usuario

Las interfaces gráficas de usuario presentan interesantes desafíos para los ingenieros de software. Debido a los componentes reutilizables provistos como parte de los entornos de desarrollo de las interfaces gráficas, la creación de la interfaz de usuario se ha convertido en menos costosa en tiempo y más exacta. Al mismo tiempo el desarrollo de este tipo de interfaces ha aumentado en complejidad, originando más dificultad en el diseño y en los casos de prueba.

Hemos tenido en cuenta los siguientes aspectos:

- En lo referente a las páginas:
    - Nuestras páginas son ajustables, móviles y se pueden desplegar.
    - Se regeneran adecuadamente
    - Están operativas todas sus funciones propias
    - Están disponibles y desplegados apropiadamente barras deslizantes, cuadros de diálogo y botones.
    - Se cierran adecuadamente

- En lo referente a la entrada de datos:
    - Los datos son introducidos adecuadamente en el sistema
    - Los modos gráficos de entrada de datos funcionan correctamente

- Los mensajes de entrada de datos son inteligibles y los de salida también.

## 5.3. Pruebas de caja blanca

En este sentido las pruebas que hemos realizado se han centra básicamente en la Prueba de bucles. Esta prueba se centra exclusivamente en la validez de las construcciones de bucles. Se pueden definir cuatro tipos de bucles:

1. Bucles simples
2. Bucles anidados
3. Bucles concatenados
4. Bucles no estructurados

En nuestro sistema encontramos principalmente de los tres primeros tipos, para exponer las pruebas realizadas a cada uno los explicamos independientemente:

**Bucles simples:**

Siendo n el número máximo de pasos permitidos por el bucle, hacemos m pasos por el mismo teniendo en cuenta que m<n, una vez realizada esta prueba a los bucles de este tipo encontrados, llegamos a la conclusión de que éstos eran correctos.

**Bucles anidados:**

Para realizar esta prueba hemos empezado realizando el análisis antes descrito para los bucles simples en los bucles más internos, mientras manteníamos los parámetros de iteración de los bucles externos en sus valores mínimos. Llegamos a la misma conclusión que en el apartado anterior.

**Bucles concatenados:**

Los bucles concatenados de los que consta nuestro programa, son del tipo en el que la salida del bucle primero se utiliza como entrada del segundo, con lo que el enfoque que hemos utilizado para probarlos es el descrito anteriormente para los bucles anidados, después de realizar dicha prueba hemos llegado a la conclusión de que todos los bucles de este tipo utilizados en nuestra implementación se comportan de manera correcta.

## 5.4. Resultados y Tiempos de respuesta

En la siguiente sección se detallaran los resultados obtenidos con el sistema Minería Web, se mencionará fuente de los datos obtenidos, características del computador en donde fueron realizadas las pruebas, así como los tiempos de respuesta de cada prueba.

***Fuente y Datos***

Las pruebas de la aplicación fueron realizadas con los archivos log del servidor del sitio Web **www.sidweb.espol.edu.ec**. Se nos proporcionó logs que almacenan información del mes de enero del 2006. Se ha predispuesto ingresar archivos log no mayor a 2MB cada uno.

***Características del Computador***

Las características de la máquina en donde se realizaron las pruebas son las siguientes:

Sistema Operativo: Microsoft Windows XP Professional

Procesador: Pentium IV 1.6 Ghz

Memoria Ram: 512 MB

Programas requeridos:

Internet Information Server (IIS) versión 5.1

Microsoft SQL Server 2000

## Prueba #1

Información perteneciente al 1 de enero del 2006 hasta el 5 de enero del 2006.

| Proceso | Tiempo de respuesta |
|---|---|
| Ingreso | 25 min |
| Limpieza | 30 seg |
| Sesionización | 22min |
| Reglas de Asociación | 40seg |
| Patrones Secuenciales | 25 min |
| Clusterización | 10seg |

**Tabla 5.1 Tiempos de respuesta para prueba 1**

**Resultados**

**Ingreso**: Para esta prueba se ingresaron archivos con información de 5 días (1 de enero del 2006 hasta el 5 de enero).

**Limpieza**: Con la limpieza se pudo reducir la información del 100% al 17,13%, con esta reducción se logra ahorrar tiempo de proceso para los siguientes algoritmos.

**Sesionización**: para el proceso de sesionización se ha seleccionado como tiempo máximo 30 minutos y como páginas válidas a las de extensión html, jsp, php y htm. Con esto se han encontrado 1058 sesiones.

Una vez que realizamos los procesos anteriores procedemos a realizar los algoritmos de minería.

**Reglas de asociación:** para poder realizar el algoritmo se ingresó los parámetros soporte= 8% y confianza=85%. Se han generado 78 reglas.

A continuación se muestra tres de las 78 reglas generadas:

| Nº | Regla | Soporte - Confianza |
|---|---|---|
| 1 | [/private/mycourses/website/calendar/index.jsp]---->/private/mycourses/website/email/index.js | 8,9% - 89,88% |
| 2 | [/public/about.jsp]----> /private/mycourses/website/email/index.js | 8,6% - 90,69% |
| 3 | [/public/team.jsp]----> /private/mycourses/website/email/index.js | 10,4% - 95,61% |

**Tabla 5.2 Reglas de asociación para prueba 1**

Con la primera regla podemos descubrir que el 89,88 % de los usuarios que visitaron [/private/mycourses/website/calendar/index.jsp] también visitaron /private/mycourses/website/email/index.js.

De manera similar la regla número 2 nos indica que el 90,60% que visitaron [/public/about.jsp] también visitaron /private/mycourses/website/email/index.js.

**Patrones secuenciales:** para poder realizar el algoritmo se ingresó los parámetros soporte =85 y confianza =90. Se han generado 30 patrones

A continuación se muestra dos de los patrones generados:

| Nº | Patrón | Días | Soporte - Confianza |
|---|---|---|---|
| 1 | [/administration/user/systemuser.jsp,]---->/private/mycourses/website/folders/content_view.jsp | 2 | 895 - 94% |
| 2 | [/administration/user/systemuser.jsp, /private/mycourses/website/folders/multimedia_view.jsp, /public/team.jsp,]---->/private/mycourses/website/folders/forums_view.jsp,/administration/user/systemuser.jsp | 2 | 92 - 98,91% |

**Tabla 5.3 Patrones Secuenciales para prueba 1**

El patrón número uno que nos presenta el algoritmo nos indica que el 94% de los usuarios que visitaron: [administration/user/systemuser.jsp luego de 2 días visitaron /private/mycourses/website/folders/content_view.jsp.

**Clusterización:** para poder realizar el algoritmo se ingresó soporte =40%. Se han generado 3 clusters y fueron necesarias 15 iteraciones para encontrar dichos clusters.

A continuación se muestra los clusters generados:

Cluster #1
/private/mycourses/website/calendar/index.jsp
/index.jsp
/private/mycourses/website/folders/question/exam_view.jsp
/private/mycourses/website/folders/documents_view.jsp
/private/mycourses/website/folders/content.jsp
/private/mycourses/website/index.jsp
/private/mycourses/website/email/index.jsp
/private/mycourses/website/folders/forums_view.jsp
/private/mycourses/website/folders/announcement_view.jsp
/private/mycourses/website/folders/assignment/assignment.jsp
/private/mycourses/website/folders/content_view.jsp
/private/mycourses/website/syllabus.jsp
/private/mycourses/website/scores/index.jsp
/private/mycourses/website/folders/assignment/assignment_view.jsp
/private/mycourses/website/folders/link_view.jsp
/private/mycourses/website/folders/multimedia_view.jsp

Cluster #2
/private/mycourses/website/index.jsp

Cluster #3
/private/mycourses/website/calendar/index.jsp
/js/tiny_mce/blank.htm
/index.jsp
/private/mybriefcase/index.jsp
/help/help.jsp
/private/mycourses/website/folders/question/exam_view.jsp
/private/mycourses/website/folders/groupView.jsp
/private/mycourses/website/folders/documents_view.jsp
/public/findUsers.jsp
/private/mycourses/website/folders/content.jsp
/help/portal_es/index.html
/loginError.jsp
/private/mycourses/website/folders/assignment/homework.jsp
/private/mycourses/index.jsp

/private/mycourses/website/index.jsp
/private/mycourses/website/email/index.jsp
/public/portalDocument.jsp
/private/mycourses/website/folders/announcement.jsp
/private/mycourses/website/folders/forums_view.jsp
/private/mycourses/website/folders/announcement_view.jsp
/public/portalFolder.jsp
/private/myprofile/index.jsp
/private/mycourses/website/folders/assignment/assignment.jsp
/private/mycourses/website/folders/content_view.jsp
/private/mycourses/website/syllabus.jsp
/private/mycourses/website/scores/index.jsp
/private/mycourses/website/folders/assignment/assignment_view.jsp
/private/mycourses/website/folders/link_view.jsp
/private/mycourses/website/chat/index.jsp
/private/mycourses/website/folders/multimedia_view.jsp

Los resultados nos muestran que los usuarios encuentran una relación entre las páginas: /private/mycourses/website/scores/index.jsp, /private/mycourses/website/calendar/index.jsp, /private/mycourses/website/folders/question/exam_view.jsp, /index.jsp, /private/mycourses/website/folders/documents_view.jsp, /private/mycourses/website/index.jsp /private/mycourses/website/folders/content.jsp, /private/mycourses/website/email/index.jsp, /private/mycourses/website/syllabus.jsp, y las demás pertenecientes todas al cluster #1.

**Prueba #2**

Información perteneciente al 1 de enero del 2006 hasta el 3de enero del 2006.

| **Proceso** | **Tiempo de respuesta** |
|---|---|
| Ingreso | 10 min |
| Limpieza | 40 seg |
| Sesionización | 6min |
| Reglas de Asociación | 40seg |
| Patrones Secuenciales | 15 min |
| Clusterización | 10seg |

**Tabla 5.4 Tiempos de respuesta para prueba 2**

**Resultados**

**Ingreso**: Para esta prueba se ingresaron archivos con información de 3 días (1 de enero del 2006 hasta el 3 de enero).

**Limpieza**: Con la limpieza se pudo reducir la información del 100% al 16%, con esta reducción se logra ahorrar tiempo de proceso para los siguientes algoritmos.

**Sesionización**: para el proceso de sesionización se ha seleccionado como tiempo máximo 40 minutos y como páginas válidas a las de extensión html, jsp, php y htm. Con esto se han encontrado 461 sesiones.

Una vez que realizamos los procesos anteriores procedemos a realizar los algoritmos de minería.

**Reglas de asociación:** para poder realizar el algoritmo se ingresó los parámetros soporte=90% y confianza=90%. Se han generado 0 reglas.

En vista de estos resultados se procede a establecer un nuevo valor de soporte=16% y manteniendo la confianza al 90%, en donde el algoritmo nos genera 2 reglas detalladas a continuación:

| Nº | Regla | Soporte - Confianza |
|---|---|---|
| 1 | [/private/mycourses/website/folders/forums_view.jsp]----->/private/mycourses/website/folders/content_view.js | 18% - 90,90% |
| 2 | [/private/mycourses/website/folders/content.jsp,/private/mycourses/website/folders/content_view.jsp]--->/public/userDetail.js | 16,5% - 98,61% |

**Tabla 5.5 Reglas de asociación para prueba 2**

Con la primera regla podemos descubrir que el 90,90% de los usuarios que visitaron [/private/mycourses/website/folders/forums_view.jsp] también visitaron /private/mycourses/website/folders/content_view.js.

De manera similar la regla número 2 nos indica que el 98,61% que visitaron [/private/mycourses/website/folders/content.jsp y

/private/mycourses/website/folders/content_view.jsp] también visitaron /public/userDetail.js.

**Patrones secuenciales:** para poder realizar el algoritmo se ingresó los parámetros soporte=85 y confianza=90%. Se han generado 8 patrones.

A continuación se muestra tres de los patrones generados:

| Nº | Patrón | Días | Soporte - Confianza |
|---|---|---|---|
| 1 | [/administration/user/systemuser.jsp,]----> /private/mycourses/website/folders/content_view.jsp | 2 | 895 - 94,36% |
| 2 | [/administration/user/systemuser.jsp, /private/mycourses/website/folders/bookmark_view.jsp, /public/portalFolder.jsp,]----> /private/mycourses/index.jsp | 1 | 329 - 94,90% |
| 3 | [/administration/user/systemuser.jsp, /private/mycourses/website/folders/multimedia_view.jsp, /public/team.jsp,]----> /private/mycourses/website/folders/forums_view.jsp,/administration/user/systemuser.jsp | 2 | 92% - 98,91% |

**Tabla 5.6 Patrones Secuenciales para prueba 2**

El patrón número uno que nos presenta el algoritmo nos indica que el 94,36% de los usuarios que visitaron la siguiente secuencia: /administration/user/user/systemuser.jsp, /private/mycourses/website/folders/bookmark_view.jsp, /public/portalFolder.jsp, luego de 1 día visitaron /private/mycourses/index.jsp.

**Clusterización:** para poder realizar el algoritmo se ingresó soporte=80%. Se han generado 3 clusters. Fueron necesarias 9 iteraciones para encontrar los clusters
A continuación se muestra los clusters generados:

Cluster #1

/private/mycourses/website/calendar/index.jsp
/private/mycourses/website/folders/documents_view.jsp
/private/mycourses/website/index.jsp
/private/mycourses/website/folders/announcement_view.jsp
/private/mycourses/website/folders/content_view.jsp
/private/mycourses/website/scores/index.jsp
/private/mycourses/website/folders/assignment/assignment_view.jsp

Cluster #2
/private/mycourses/website/folders/assignment/assignment_view.jsp

Cluster #3
/private/mycourses/website/index.jsp

Los resultados nos muestran que los usuarios encuentran una relación entre las páginas: /private/mycourses/website/calendar/index.jsp, /private/mycourses/website/index.jsp, /private/mycourses/website/folders/documents_view.jsp y las demás pertenecientes todas al cluster #1.

**Prueba #3**
Información perteneciente al 1 de enero del 2006.

| **Proceso** | **Tiempo de respuesta** |
|---|---|
| Ingreso | 1,5 min |
| Limpieza | 13 seg |
| Sesionización | 10 seg |
| Reglas de Asociación | 40seg |
| Patrones Secuenciales | 10 seg |
| Clusterización | 3seg |

**Tabla 5.7 Tiempos de respuesta para prueba 3**

**Resultados**

**Ingreso**: Para esta prueba se ingresó un archivo con información del 1 de enero del 2006.

**Limpieza**: Con la limpieza se pudo reducir la información del 100% al 13,79%, con esta reducción se logra ahorrar tiempo de proceso para los siguientes algoritmos.

**Sesionización**: para el proceso de sesionización se ha seleccionado como tiempo máximo 30 minutos y como páginas válidas a las de extensión html, jsp, php y htm. Con esto se han encontrado 42 sesiones.

Una vez que realizamos los procesos anteriores procedemos a realizar los algoritmos de minería.

**Reglas de asociación:** para poder realizar el algoritmo se ingresó los parámetros soporte=10% y confianza=85%. Se han generado 3 reglas.

En vista de estos resultados se procede a establecer un nuevo valor de soporte=70 en donde el algoritmo nos genera 2 reglas, detalladas a continuación:

| Nº | Regla | Soporte - Confianza |
|---|---|---|
| 1 | [/private/mycourses/website/index.jsp]----> /private/myprofile/index.js | 14,2% - 83,33% |
| 2 | [/public/portalFolder.jsp]----> /public/userDetail.js | 14,2% - 83,33% |
| 3 | [/public/about.jsp,/public/userDetail.jsp]----> /private/myprofile/index.js | 14,2% - 100% |

**Tabla 5.8 Reglas de asociación para prueba 3**

Con la primera regla podemos descubrir que el 83,33% de los usuarios que visitaron /private/mycourses/website/index.jsp también visitaron /private/myprofile/index.js.

De manera similar la regla número 3 nos indica que el 100% delos usuarios que visitaron /public/about.jsp,/public/userDetail.jsp también visitaron /private/myprofile/index.js.

**Patrones secuenciales:** para poder realizar el algoritmo se ingresó los parámetros soporte=90 y confianza=90. Se han generado 2 patrones.

A continuación se muestra los dos patrones generados:

| Nº | Patrón | Días | Soporte - Confianza |
|---|---|---|---|
| 1 | [/private/mybriefcase/document.jsp,]---->/private/mycourses/website/folders/document.jsp | 0 | 94 - 95,91% |
| 2 | [/private/mybriefcase/document.jsp,]---->/private/mycourses/website/folders/document.jsp | 0 | 94 - 95,91% |

**Tabla 5.9 Patrones Secuenciales para prueba 3**

El patrón número uno que nos presenta el algoritmo nos indica que el 94,36% de los usuarios que visitaron la siguiente secuencia: /administration/user/user/systemuser.jsp, /private/mycourses/website/folders/bookmark_view.jsp, /public/portalFolder.jsp, luego de 1 día visitaron /private/mycourses/index.jsp.

**Clusterización:** para poder realizar el algoritmo se ingresó soporte=85%. Se han generado 3 clusters. Fueron necesarias 4 iteraciones para encontrar los clusters

A continuación se muestra los clusters generados:

Cluster #1
/private/mycourses/website/index.jsp

Cluster #2
/private/mycourses/website/folders/announcement.jsp
/private/mycourses/website/folders/announcement_view.jsp
/private/mycourses/website/folders/assignment/assignment_view.jsp
/private/mycourses/website/folders/content_view.jsp
/private/mycourses/website/folders/documents_view.jsp
/private/mycourses/website/index.jsp
/private/mycourses/website/scores/index.jsp
Cluster #3
/private/mycourses/website/email/index.jsp
/private/mycourses/website/folders/assignment/assignment_view.jsp
/private/mycourses/website/folders/content_view.jsp
/private/mycourses/website/index.jsp

Los resultados nos muestran que los usuarios encuentran una relación entre las páginas: /private/mycourses/website/email/index.jsp, /private/mycourses/website/folders/content_view.jsp, /private/mycourses/website/index.jsp pertenecientes todas al cluster #3.

### 5.5. Prueba de Aceptación del usuario

Fueron presentados varios ejemplos de logs de diferentes dias al administrador Web del sitio en pruebas cuyos resultados fueron los esperados en la adquisición de un conocimiento útil para diferentes propósitos.

## 6. DEFINICION DEL PRODUCTO

El desarrollo de Internet y la reducción de sus costos, representa una oportunidad para las empresas o PYMES ecuatorianas que desean ofrecer sus servicios en línea. Las que ya lo hicieron son participes de un canal alternativo o complementario a los ya existentes, como medios de comunicación o promoción dirigido tanto a consumidores como a intermediarios. Sin embargo, hoy en día no se asegura rentabilidad tan solo con tener presencia en Internet. Por tanto, se necesita encontrar o establecer estrategias, dentro del contexto de la actividad de la empresa, que contribuyan a mejorar cada vez su presencia en Internet.

Las estrategias deben estar enfocadas en beneficio de los clientes, es decir de los navegantes del sitio Web de la empresa. Observando esta necesidad empresarial nace nuestra solución “MineroWeb”.

Producto

Cada acción o movimiento de un cliente en un sitio Web genera datos, MineroWeb es una aplicación que analiza y explora estos datos con la finalidad de presentar luego resultados que permitirán establecer estrategias pensadas en la mejora de servicios o el contenido para sus clientes. Las empresas que utilicen nuestra aplicación tendrán un paso adelante de sus competidores directos pues gracias a la tecnología actual podrán volverse lo bastante competitivo frente a sus competencias directas.

MineroWeb es una solución pensada para el análisis de todo tipo de sitio Web desde los que solo proporcionan contenido hasta los más sofisticados que permiten realizar transacciones comerciales en línea.

Las empresas detrás de estos sitios no conocen el tipo de visitantes, ni la forma en como son accedidas sus páginas y por tanto no tienen claro donde realizar mejoras que ayuden a sus objetivos de negocio.

MineroWeb ayuda en esta tarea respaldado con el uso de algoritmos de minería de datos y obteniendo resultados como:

- Grupos de usuarios que prefieren ciertas páginas con algún contenido en particular. Por ejemplo, si se analiza un sitio que vende electrodomésticos en línea podemos descubrir que existen grupos de usuario que se interesan por las ofertas del mes u otro grupo que se interesa por ciertas marcas.
- La forma de navegar por el sitio. Podemos descubrir cual será la pagina siguiente que visitará un navegante. Lo que se podría aprovechar de distintas maneras según el contexto del negocio.
- El posible tiempo en que los usuarios volverán a navegar por ciertas páginas.
- Estadísticas de tráfico. Lo que permitirá conocer parámetros de navegación mediante gráficos. Por ejemplo las páginas mas populares.

MineroWeb es una aplicación que ayuda a los administradores del sitio Web comprender de que forma los usuarios usan el sitio para luego decidir sobre mejoras del servicio que ofrece.

## 6.1. Segmento de mercado

El mercado objetivo inicial son las PYMES de cualquier tipo de actividad comercial. Las PYMES en el Ecuador son aproximadamente 15.000 con un promedio de 22 empleados, cuya concentración se encuentra en mayor proporción en las ciudades de Guayaquil y Quito con un 77%, según la Fundación para la Ciencia y Tecnologías (www.fundacyt.org.ec)

Nuevos segmentos de mercado

"MineroWeb" analiza los movimientos o huellas digitales que dejan los navegantes en un sitio Web, por tanto, se puede fácilmente analizar todo tipo de sitio Web. Pensando en esto podemos decir que nuestro nicho de clientes también se puede concentrar en los sitios Web registrados en la Corporación Ecuatoriana de Comercio Electrónico (CORPECE), donde podemos encontrar variadas categorías de sitios Web como: entretenimiento, noticias, educación, turismo, buscadores, Internet, sociedad y cultura, gobierno y negocios. Cada uno de estos sitios representa un potencial cliente para nuestra aplicación.

Como punto ilustrativo, se podrá observar en el capitulo de pruebas que se trabajo con un sitio Web del área de la educación: www.sidweb.espol.edu.ec.

## 6.2. Análisis económico

En esta sección se realizará un análisis económico que mostrará los costos involucrados en el desarrollo de la aplicación que analiza los datos provenientes de la Web. La duración del desarrollo de la aplicación denominada MineroWeb es de cuatro meses.

### Análisis de costos

Presupuesto del Personal

El equipo humano estará conformado por tres personas que actuarán tanto como investigadores del tema y desarrolladores. Todos cubrirán las necesidades operativas y trabajarán de forma permanente. La tabla 6.1 muestra la distribución del presupuesto para el pago al personal.

| | | Meses | | | |
|---|---|---|---|---|---|
| | **No.Personas** | **1** | **2** | **3** | **4** |
| **Personal operativo** | | | | | |
| Lider | 1 | 150 | 200 | 200 | 200 |
| Desarrollador | 2 | 150 | 200 | 200 | 200 |

| | | | | | |
|---|---|---|---|---|---|
| **Subtotal** | | 450 | 600 | 600 | 600 |
| **Total** | | 2250 | | | |

**Tabla 6.1 Pagos de personal**

Gastos de operación

En la tabla 6.2 se muestra los gastos de operación equivalentes al periodo de desarrollo propuesto.

| **Detalle** | **Mensual** | **Periodo (4 meses)** |
|---|---|---|
| Transporte | 30 | 120 |
| Insumos de oficina | 20 | 80 |
| Telefonía fija | 15 | 60 |
| Telefonía movil | 20 | 80 |
| Internet (Alegro-NIU) | 40 | 160 |
| **Total** | | 500 |

**Tabla 6.2 Gastos operativos**

Gastos de administración y ventas

Para la difusión de la aplicación se realizará anuncios gráficos en portales del medio que pertenezcan al área informática. Para esto se contará con un pequeño soporte de personal entendido para lograr potenciar la imagen de nuestra aplicación. Se contará con un presupuesto mostrado en la tabla 6.3

| **Detalle** | **Mensual** | **Periodo (4 meses)** |
|---|---|---|
| Publicidad | 200 | 800 |
| **Total** | | 800 |

**Tabla 6.3 Gastos administrativos y ventas**

**Costos del Sistema**

Como se menciono, la duración del desarrollo e implementación de la aplicación MineroWeb es de 4 meses. Para describir los costos involucrados se usará algunos componentes de costos ofrecidos en el modelo TCO de The GartnerGroup. En la siguiente tabla se resume estos valores.

| | **Mes 1** | **Mes 2** | **Mes 3** | **Mes 4** |
|---|---|---|---|---|

| **1.- Costos fijos** | | | | |
|---|---|---|---|---|
| Hardware | 25 | 25 | 25 | 25 |
| Software | 100 | 100 | 100 | 100 |
| Desarrollo | 450 | 600 | 600 | 600 |
| Gastos de operación | 125 | 125 | 125 | 125 |
| **Total Costos fijos** | **700** | **850** | **850** | **850** |
| | | | | |
| **2.- Costos variables** | | | | |
| Asesorías | 30 | 0 | 0 | 0 |
| Gastos de Publicidad | 200 | 200 | 200 | 200 |
| **Total Costos variables** | **230** | **200** | **200** | **200** |
| **Costo Total** | **930** | **1050** | **1050** | **1050** |

**Tabla 6.4 Costos para desarrollo del sistema**

Explicación de costos del sistema.

Los costos de Hardware se refieren únicamente al alquiler de equipos para establecer una pequeña infraestructura de red. Las computadoras utilizadas son de propiedad de los desarrolladores y no aporta costo adicional para el proyecto.

Los costos de Software se refieren a las licencias de terceros usados para el desarrollo de MineroWeb: ASP.net y SQL Server 2000.

Los costos de Desarrollo se refieren a los de mano de obra en la programación, pruebas y documentación. Estos rubros están referidos en la tabla 6.1.

Los gastos de operación resultantes para poder trabajar en el desarrollo de la aplicación son detallados en la tabla 6.2.

La tabla 6.3 muestra los gastos de publicidad y en cuanto a las Asesorías es un valor adicional por establecer la infraestructura de la red de tipo inalámbrica.

### 6.2.2. Análisis de ingresos

Hemos mencionado que el mercado objetivo son los sitios Web registrados en la Corporación Ecuatoriana de Comercio Electrónico (CORPECE). Inicialmente se trabajará para obtener clientes que pertenezcan al área comercial, pues aunque nuestro producto MineroWeb puede analizar sin distinción cualquier sitio Web, será mas propicio iniciar con empresas que ofrecen sus productos o servicios en línea.

En el primer año de ventas y operación contaremos con un grupo aproximado de 48 empresas que tienen su sitio Web de tipo comercial. El precio de venta de MineroWeb es de $400 que incluye instalación, entrenamiento y soporte.

La siguiente tabla muestra un presupuesto estimado de ventas que se plantea obtener para el primer año una vez concluido el desarrollo de la aplicación. En la misma, se muestra la meta mínima de clientes con su respectiva aplicación que deseamos obtener por cada trimestre del primer año.

| | **Trimestre 1** | **Trimestre 2** | **Trimestre 3** | **Trimestre 4** |
|---|---|---|---|---|
| **Volumen Estimado de Ventas** | | | | |

| | | | | |
|---|---|---|---|---|
| Aplicación MineroWeb | 5 | 6 | 8 | 8 |
| **Precio de Ventas** | | | | |
| Aplicación MineroWeb | 2000 | 2400 | 3200 | 3200 |
| **Total ventas** | **2000** | **2400** | **3200** | **3200** |

**Tabla 6.5 Análisis de ingreso**

Aunque iniciaremos la operación con clientes que vendan en la Web nuestros esfuerzos publicitarios serán para atraer todo el mercado registrado en CORPECE, por lo cual la tabla 6.5 podría variar favorablemente para los intereses de sus gestores.

# CONCLUSIONES Y RECOMENDACIONES

Actualmente son cientos de usuarios que utilizan la Web para diversos propósitos. Este hecho acrecentó nuestro interés por realizar la presente tesis y de la cual destacamos las siguientes conclusiones:

- La aplicación es capaz de descubrir interesantes reglas de asociación sobre los hábitos de visitas de páginas por parte de los usuarios, encontrar diferentes grupos de comportamiento entre los usuarios , encontrar diferentes secuencias de patrones para conocer el tiempo de retorno de un visitante y revelar detalles estadísticos sobre el tráfico del servidor Web. Tal como se evidencia en el capitulo de pruebas.

- Por último, la aplicación que hemos desarrollado cumple con los objetivos y expectativas propuestas, MineroWeb es una herramienta que permitirá analizar en profundidad el comportamiento de los visitantes de su sitio Web a través de los ficheros log.

*Como recomendación importante destacamos:*

- Nueva generación de salidas: Nuestro sistema proporciona salidas en las pantallas de la aplicación. Una posibilidad atractiva consiste en que la salida se proporcione usando un documento xml para que pueda ser guardado o guardar los resultados en una base de datos. Esto supondría una ventaja para el posterior análisis de la salida de MineroWeb, lo cuál ayudaría a tomar decisiones eficientes sobre la gestión de la página Web.

- Nuevas funcionalidades: Se podría incluir para futuras versiones el uso de otros formatos de archivo log como W3C y nuevos reportes estadísticos que muestren mas parámetros de tráfico de un servidor Web.

# BIBLIOGRAFIA

[1] Kosala & Blockeel 2000, Kosala,R.; Blockeel, H. "Minería Web Research: A Survey", ACM SIGKDD Exploration, Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining, vol 2, pp:1-15,2000

[2] R. Cooley, B. Mobasher, and J. Srivastava. "Minería Web: Information and Pattern Discovery on the World Wide Web",1997.

[3,5,6,8] José Hernández Oralla, Ma. José Ramírez Quintana, César Ferri Ramírez. "Introducción a la Minería de Datos", Capítulo 21.

[4] Mariano Silva, "Introducción al Minería Web" www.webmining.cl

[7] Mariano Silva V. "Minería Web: definiciones y aplicaciones" www.webmining.cl

[9] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, Pang-Ning Tan. "Minería del uso de la Web: Discovery and Applications of Usage Patterns from Web Data", 2000.

[10] Lic. Gustavo D. Koblinc. Universidad de Buenos Aires. Minería Web: Estado Actual de Investigación.

[11] R. Cooley, B Mobasher, Jaideep Srivastava. Data Preparation For Mining World Wide Web Browsing Patterns. Minneapolis, USA,1999.

[12] Rakesh Agrawal and Ramakrishnan Srikant, *Fast algorithms for mining association rules*, 1994

[13] Óscar Marbán Gallego. Estudio de perfiles de visitantes de un website a partir de los logs de los servidores web aplicando técnicas de Data mining (Minería Web). En resumen del trabajo de investigación, Universidad Politécnica de Madrid, España,200.

[14] Rakesh Agrawal and Ramakrishnan Srikant, *Fast algorithms for mining association rules*, Proc. 20th Int. Conf. Very Large Data Bases, VLDB (citeseer.nj.nec.com/agrawal94fast.html) (Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, eds.), Morgan Kaufmann, 12–15 1994,
pp. 487–499.

# ANEXOS

## *ANEXO A*

## *ESTRUCTURA DEL LOG SERVIDOR WEB*

Log de acceso:

Pueden ser guardados en un formato de fichero log comun (Common Log File CLF) o en un formato de fichero log extendido (Extended Log File ELF).

**Ejemplo de CLF:**

130.2.0.203 - - [01/Aug/1995:00:00:01 -0400] "GET /shuttle/missions/sts-68/news/sts-68-mcc-05.txt HTTP/1.0" 200 1839

Los elementos de los que consta el CLF son:

- Remotehost: (host remoto). En el ejemplo: 130.2.0.203
- Rfc931: El nombre de log remoto del usuario. En el ejemplo: -
- Authuser: El nombre con el que el usuario se ha identificado. En el ejemplo: -
- [fecha]: Fecha y hora de la solicitud. En el ejemplo: [01/Aug/1995:00:00:01 -0400]
- “Solicitud”: La linea exacta de petición según viene solicitada desde el cliente. En el ejemplo: “GET /shuttle/missions/sts-68/news/sts-68-mcc-05.txt HTTP/1.0"
- Estado: El codigo de estado del HTPP devuelto al cliente. En el ejemplo: 200
- Bytes: La longitud que tiene el documento contenido. En el ejemplo: 1838

**Ejemplo de ELF:**

217.216.54.238 - - [13/Oct/2005:10:25:34 +0200] "GET /portalcpp/web/estilos.css HTTP/1.1" 304 0 "http://altair.ugr.es/portalcpp/web/form.php" "Mozilla/4.0

(compatible; MSIE 6.0; Windows NT 5.1)"

Los elementos de los que consta el ELF son:

- Remotehost: (host remoto). En el ejemplo: 217.216.54.238
- Rfc931: El nombre de log remoto del usuario. En el ejemplo: -
- Authuser: El nombre con el que el usuario se ha identificado. En el ejemplo: -
- [fecha]: Fecha y hora de la solicitud. En el ejemplo: [13/Oct/2005:10:25:34 +0200]
- “Solicitud”: La linea exacta de petición según viene solicitada desde el cliente. En el ejemplo: “GET /portalcpp/web/estilos.css HTTP/1.1"
- Estado: El código de estado del HTPP devuelto al cliente. En el ejemplo: 304
- Bytes: La longitud que tiene el documento contenido. En el ejemplo: 0
- Referente: URL desde donde se ha realizado la petición. En el ejemplo: "http://altair.ugr.es/portalcpp/web/form.php"
- Agente: tipo de navegador y sistema operativo usado. En el ejemplo: "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"

# *ANEXO B*

# *DEFINICIÓN DE ALGORITMOS DE MINERÍA DE DATOS*

En esta sección definiremos los aspectos teóricos de los algoritmos de minería de datos que hemos utilizado en la realización de la tesis, además de la definición en pseudocódigo de cada uno de ellos.

## Reglas de Asociación

El descubrimiento de reglas de asociación es generalmente aplicado a Bases de Datos transaccionales, donde cada transacción consiste en un conjunto de ítems. En este modelo, cada item es representado por el url de las páginas del sitio, el problema consiste en descubrir todas las asociaciones y correlaciones de páginas donde la presencia de un conjunto de páginas en una transacción implica (con un grado de confianza) la presencia de otras páginas.

En el contexto de Minería Web este problema tiende a descubrir la correlación entre los accesos de los clientes a varios archivos disponibles en el servidor. Cada transacción está compuesta por un conjunto de URL accedidas por el cliente en una visita al servidor.

Para descubrir las reglas de asociación hemos empleado el algoritmo A priori,

Ideas básicas:

Partimos de conjunto de URLs y sesiones obtenidas en la fase de preprocesamiento.

Minamos de las sesiones los grupos de URLs que aparecen juntos con cierta frecuencia.

Lo que obtenemos son reglas de asociación.

**Nomenclatura a utilizar**

$L_k$: Conjunto de grupos grandes (de tamaño k) de urls (conjunto de large urlsets, aquellos que tienen el soporte mínimo).

$C_k$: Conjunto de grupos de urls de tamaño k (k-urlsets) candidatos (large urlsets potenciales). Al conjunto total de sesiones D se le denomina base de datos.

$\overline{C}_k$**:** Conjunto de k-urlsets candidatos tal y como se almacenan en las sesiones

El pseudocódigo de este algoritmo tomado de [14] es el siguiente:

$L_1$={large 1-urlsets};
$\overline{C}_1$ = base de datos $D$ de sesiones;
for ($k = 2$; $L_{k-1} \neq 0$; $k$++) do begin
  $C_k$ = apriori-gen($L_{k-1}$); // Nuevos candidatos
  $\overline{C}_k = 0$;
  for (todas las sesiones $t \in \overline{C}_{k-1}$) do begin
    // determinar grupos de urls candidatos en $C_k$ contenidos en la
    // sesión con identifi cador $t.TID$
    $C_t = \{c \in C_k | (c - c[k]) \in$ conj. de urlsets de $t \wedge (c - c[k-1]) \in$ conj. de urlsets de t$\}$
    for (todos los candidatos $c \in C_t$) do
      $c.count$++;
    if ($C_t \neq 0$) then $\overline{C}_k \longleftarrow < t.TID, C_t >$;
  end;
  $L_k = \{c \in C_k | c.count \geq$ soporte mínimo$\}$
end;
Respuesta = $\bigcup_k L_k$

**Apriori-gen**

Este método consta de dos etapas: una de unión, y otra de poda de elementos innecesarios. Con una notación similar a SQL, podemos describirlo de la siguiente manera:

Fase de unión.

insert into $C_k$
select $p.url_1, p.url_2, \cdots, p.url_{k-1}, q.url_1, q.url_2, \cdots, q.url_{k-1}$ from $L_{k-1}$ $p$, $L_{k-1}$ $q$
where $p.url_1 = q.url_1, \cdots, p.url_{k-2} = q.url_{k-2}, p.url_{k-1} < q.url_{k-1}$;

Fase de poda (eliminamos todos los grupos de urls perteneciente a $C_k$ tales que algún subconjunto de tamaño k -1 de c no esté en $L_{k-1}$):

for (todos los grupos de urls $c \in C_k$) do
  for (todos los subconjuntos $s$ de tamaño $k-1$ de $c$) do
    if ($s \notin L_{k-1}$) then
      Borrar $c$ de $C_k$;

**Generando las reglas**

fi nalrules($l_k$: large $k$-itemset, $H_m$: conjunto de consecuentes con $m$-items)
  if ($k > m+1$) then begin
    $H_{m+1}$ = apriori-gen($H_m$);
    for (todos los $h_{m+1} \in H_{m+1}$) do begin
      if ($soporte(l_k)/soporte(l_k - h_{m+1}) \geq minconf$) then
        Devolver la regla $(l_k - h_{m+1}) \Rightarrow h_{m+1}$
        con confi anza = $conf$ y soporte = $soporte(l_k)$;
      else
        Borrar $h_{m+1}$ de $H_{m+1}$;
    end
    fi nalrules($l_k$, $H_{m+1}$);
  end

## Clustering

Las técnicas de clustering permiten desarrollar un perfil para las páginas pertenecientes a un grupo particular de acuerdo con sus atributos comunes. Este perfil luego puede ser utilizado para clasificar nuevas páginas que se agreguen en la base de datos. Para desarrollar en la tesis este método hemos escogido al algoritmo de K-medias para ser implementado.

El algoritmo de k-medias es el algoritmo de agrupamiento más popular. También es llamado el *algoritmo de las medias móviles* porque en cada iteración se recalculan los centros de los agrupamientos. Por esta razón incorporamos el índice $t$ a la notación que estamos empleando de manera que con $S_i(t)$ indicamos el conjunto de patrones asociados al agrupamiento $S_i$ en la iteración $t$ y mediante $Z_i(t)$ indicamos el valor de su centro en esa iteración.

Este algoritmo requiere un único parámetro, $K$, el número de agrupamientos que debe encontrar.

Ideas básicas:

**1.Inicialización**: Consiste en inicializar arbitrariamente los centros de los $K$ grupos.

**2.Asignación y actualización de los centros**. En este paso se asigna cada patrón al grupo más cercano y se recalculan los centros en base a esta asignación.

**3.Convergencia** En el paso anterior algunos patrones pueden cambiar de agrupamiento y en consecuencia, los centros de éstos. Si ésto ocurre, se trata de repetir el paso 2 hasta que no se modifiquen los centros. Cuando no hayan modificaciones se considera que se ha encontrado una buena partición y se termina la clusterización.

**Algoritmos de las K-medias: K-media(X,K)**

*Entrada*

Un conjunto de N patrones (clusters)

K E N  Número de clusters.

*Salidas*

S1,S2,...,Sk  K conjuntos de patrones(clusters)

Z1,Z2,...Zk  Los centros de los K clusters

*Inicialización*

.t ← 0

Seleccionar Zt(t),  i=1,2,...,K. {Inicializar los centros}

*Asignación y actualización de los centros*

. t ← t + 1

Repetir para X = X1,X2,...,XN

.m← GrupomasCercano(X)  {Asignar cada X}

Sm(t) ← Sm(t-1)U {X}  {Al grupo más cercano}

Fin-para // para X=X1,X2,..,XN

Repetir para i= 1,2,...,K

.Z(t) ← RecalcularCentro(i)  {Recalcular los centros}

Fin-para // para i=1,2,..,K

*Convergencia*

Si Zi(t) =Zi(t-1), para i=1,2,...,K

Terminar.

Si-no {Algún Zi<>Zi(t-1)}

Ir al paso 2.

Fin-si // Si Zi(t) =Zi(t-1), para i=1,2,...,K

## Secuencia de Patrones

En general en la base de datos que obtuvimos a partir del archivo log se tienen disponibles los datos en un período de tiempo y se cuenta con la fecha en que se realizó la transacción; la técnica de patrones secuenciales se basa en descubrir patrones en los cuales la presencia de un conjunto de páginas es seguido por otra página en orden temporal.

Para desarrollar este método hemos escogido al algoritmo caminos frecuentes en la conducta del usuario (Frequent Behavior Paths) para ser implementado.

Para definir el algoritmo caminos frecuentes en la conducta del usuario es necesario saber algunos antecedentes:

Un camino $P = (p_{(0)}, p_{(1)}, \ldots, p_{(n-1)})$ se dice que es *frecuente* si *el sup(P)* > $\varepsilon$. Un camino $P = (p_{(0)}, p_{(1)}, \ldots, p_{(n-1)})$ se dice que es *comportamiento-frecuente* si la probabilidad de llegar a la página $p_{(n-1)}$ habiendo visitado las páginas $p_{(0)}, p_{(1)}, \ldots, p_{(n-2)}$ es mayor que un umbral establecido. Lo que es lo mismo, $\forall 0 \leq i < n$ / $P(\square_{p(i)} | \square_{p(0)} \tilde{} \square_{p(i-1)}) > \delta$. El objetivo del algoritmo es obtener los caminos *comportamiento-frecuentes*.

Dado dos caminos, $P_{IND}$ y $P_{DEP}$, una *regla de comportamiento-frecuente* es una regla de la siguiente forma:

$$P_{IND} \rightarrow P_{DEP}$$

donde $P_{IND}$ es un camino frecuente, denominado parte independiente de la regla y $P_{DEP}$ es un camino *comportamiento-frecuente*, denominándose parte dependiente de la regla. Las reglas de *comportamiento-frecuente* deben tener la propiedad siguiente:

$$P(P_{DEP} \mid P_{IND}) >$$

La formula anterior indica que si un usuario recorre el camino $P_{IND}$, entonces habrá una cierta probabilidad de que el usuario visite las páginas del camino $P_{DEP}$. LA confianza de una *regla de comportamiento-frecuente* se representa como $conf(P_{IND} \rightarrow P_{DEP})$ y define la probabilidad de recorrer el camino $P_{DEP}$ una vez se ha recorrido el camino $P_{IND}$.

Ideas básicas

Los pasos básicos del algoritmo son los siguientes:

Construir la *matriz de transición TM*. La matriz TM es una matriz N*x*N siendo N el número de páginas. TM[*i,j*] representa el número de veces que los usuarios han visitado la página $_j$ después de la página $_i$.

$$\forall S \in L \,/\, TM[i,j]=|s(k)=i \wedge s(k+1)=j|$$

Una vez construida la matriz TM, todas las celdas con valor menor a un umbral determinado toman el valor 0. La nueva matriz se llama FTM.

Una vez construida la matriz FTM, hay que obtener los caminos frecuentes. La matriz de FTM se utiliza durante la generación de los caminos para realizar una poda de caminos (espacio de búsqueda). Esto se hace así teniendo en cuanta que si un 2-camino no es frecuente un 3-camino del que sea subcamino el 2-camino tampoco lo será (propiedad Apriori). Los caminos se almacenan en una estructura multiárbol que se ha denominado FBP-tree.

Usando el FBP-tree se calcularán las *reglas comportamiento-frecuentes*. Para calcular las reglas se hace uso de los soportes de los *caminos frecuentes*.

**Construcción de la matriz TM**

**Entrada:** L: la lista de sesiones,N: el número de páginas

**Salida:** TM: la matriz de transición (NxN)

```
Para cada s en L
{
  para cada par (as(i), as(i+1))en s
    TM[s(i),s(i+1)]++;
}
```

**Construcción de la matriz FTM**

***Entrada:*** *TM: matriz de transición (NxN), e: umbral de frecuencia mínimo*

***Salida:*** *FTM: matriz de transición frecuente (NxN)*

```
para i en 0.. N
{
  para j en 0.. N
    si (TM[i,j] < e)
      FTM[i,j]=0
    sino
      FTM[i,j]=TM[i,j]
}
```

**Construcción del FBP-tree**

**Entrada:** FTM: matriz de transición frecuente (NxN), L: lista de sesiones

**Salida:** FBP-tree: árbol de camino de conducta frecuente (cada nodo del FBP-tree o cada hoja tiene un contador de hit).

```
para cada s en L
{
```

```
para i en N
{
 j = i + 1
 mientras j < N y FTM[s(j-1),s(j)] ¹ 0
 {
   sub-s = {as(i),..., as(j)}
   si ("k/i£k<j:FTM[s(k),s(k+1)]¹0)
   {
     si ($sub-s in FBP-Tree)
      FBP-Tree.increment_hit(sub-s)
     resto
      FBP-Tree.insert_path(sub-s)
   }
 }
}
```

**Construcción de reglas de conducta frecuentes**

**Entrada:** FBP-tree: árbol de camino de conducta frecuente,s : soporte mínimo de la regla, k : confianza mínima de la regla

**Salida:** Reglas-FBP: reglas de camino de conducta frecuentes

```
Para cada l en FBP-Tree.leaves
{
 mientras l ¹ FBP-Tree.root
 {
   si (l.hit < s)
     FBP-Tree.prune(l)
```

```
    sino
    {
      q=cola{}
      mientras(l.hit/l.parent.hit > k)
      {
        q.append(l.page)
        l=l.parent()
      }
      if(q.empty())
      {
        r=rule {}
        r.PIND({FBP-Tree.root.page(),.., l.page})
        r.PDEP(q)
        FBP-Rules.add(r)
      }
      FBP-Tree.prune(l)
    }
}
```

# *ANEXO C*

# *DISEÑO*

En esta sección mostramos información adicional del diseño del sistema, lo que incluye descripción de los escenarios, diagramas de interacción de objetos de cada escenario.

**Descripción de escenarios**
En esta sección se muestran los escenarios resultantes de los 6 casos de uso del sistema, información de los requisitos planteados y los resultados emitidos.

**Caso de uso 1**: Ingresar archivo del log del servidor

En este caso de uso se crearon 4 escenarios detallados a continuación:

**Escenario 1.1**.- Ingreso exitoso del archivo del log del servidor

Requisitos:

- El administrador seleccionó la ruta correcta del archivo log del servidor.
- El archivo log del servidor es del formato correcto.

Resultados:

- Se muestra un mensaje indicando el éxito de la operación. Se almacena la información del archivo. Se preparan los datos guardados para ser analizados

**Escenario 1.2.-** Ingreso no exitoso del archivo del log por ruta incorrecta del archivo.

Requisitos:

- El administrador seleccionó mal la ruta correcta del archivo log del servidor.

Resultados:

- No se guardan los datos y se presentará un mensaje indicando cual fue el error ocurrido.

**Escenario 1.3.-** Ingreso no exitoso del archivo del log por formato de archivo incorrecto.

Requisitos:

- El archivo log del servidor tiene un formato incorrecto. O no contiene datos.

Resultados:

- No se guardan los datos y se presentará un mensaje indicando cual fue el error ocurrido y los formatos de archivos válidos.

**Escenario 1.4**.- Ingreso no exitoso por fallas técnicas en el ingreso del archivo del log.

Requisitos:

- Se tiene problemas al momento del ingreso del archivo del log.

Resultados:

- Mensaje que indique la causa del problema.

**Caso de uso 2**: Seleccionar parámetros para procesamiento

En este caso de uso se crearon 7 escenarios detallados a continuación:

**Escenario 2.1**.- Selección exitosa de limpieza de archivo log para procesamiento

Requisitos:

- El administrador ingresó correctamente el archivo log.

Resultados:

- Se muestra un mensaje indicando el éxito de la operación. Se almacena la información del archivo log realizada la limpieza. Se preparan los datos guardados para ser analizados con la respectiva Técnica de Minería de Datos.

**Escenario 2.2**.- Selección no exitosa de limpieza de archivo log por no ingreso de archivo log.

Requisitos:

- El administrador no ingresó el archivo log con anterioridad.

Resultados:

- No se guardan los datos y se presentará un mensaje indicando cual fue el error ocurrido y como repararlo.

**Escenario 2.3**.- Selección no exitosa de limpieza de archivo log por fallas técnicas..

Requisitos:

- Se tiene problemas al momento de realizar la limpieza del archivo log.

Resultados:

- No se guardan los datos y se presentará un mensaje indicando cual fue el error ocurrido.

**Escenario 2.4**.- Ingreso exitoso de parámetros para procesamiento

Requisitos:

- El administrador seleccionó correctamente el parámetro de tiempo máximo.

- El 4administrador seleccionó correctamente los archivos Web que serán considerados.

Resultados:

- Se muestra un mensaje indicando el éxito de la operación. Se almacena la información de los parámetros. Se preparan los datos guardados para ser analizados con la respectiva técnica de minería de datos.

**Escenario 2.5.-** Ingreso no exitoso de parámetros para procesamiento por falta de tiempo máximo.

Requisitos:

- El administrador no seleccionó el parámetro de tiempo máximo.

Resultados:

- Se muestra un mensaje indicando el error. Se pide ingresar el parámetro faltante.

**Escenario 2.6.-** Ingreso no exitoso de parámetros para procesamiento por falta de archivos permitidos

Requisitos:

- El administrador no seleccionó los archivos Web que serán considerados.

Resultados:

- Se muestra un mensaje indicando el error. Se pide ingresar el parámetro faltante.

**Escenario 2.7**.- Ingreso no exitoso por fallas técnicas en el ingreso de parámetros para procesamiento.

Requisitos:

- Se tiene problemas al momento del ingreso de parámetros.

Resultados:

- Mensaje que indique la causa del problema.

**Caso de uso 3**: Presentar reporte

En este caso de uso se crearon 2 escenarios detallados a continuación:

**Escenario 3.1**.- Presentación exitosa del reporte para estadísticas de uso.

Requisitos:

- Se ingresó previamente exitosamente el archivo log del servidor.
- Se ingresó exitosamente todos los parámetros generales para el procesamiento.
- Los datos fueron analizados con éxito para emitir el reporte.

Resultados:

- Se visualiza el reporte.

**Escenario 3.2.-** Presentación no exitosa del reporte para estadísticas de uso.

Requisitos:

- Previamente no se ingresó exitosamente el archivo log del servidor.

- No se ingresó exitosamente todos los parámetros necesarios para el procesamiento.
- Se tiene problemas al momento de obtener la información de la base de datos para generar el reporte.
- Existen problemas en la base al momento de obtener los datos necesarios para generar el reporte.

Resultados:

No se puede visualizar el reporte y se presenta un mensaje indicando el problema.

**Caso de uso 4**: Reglas de asociación

En este caso de uso se crearon 5 escenarios detallados a continuación:

**Escenario 4.1**.- Presentación exitosa del reporte generación de reglas de asociación.

Requisitos:

- Se ingresó previamente exitosamente el archivo log del servidor.
- Se ingresó exitosamente todos los parámetros generales para el procesamiento.
- Se ingresó exitosamente el soporte y confianza para el procesamiento.
- Los datos fueron analizados con éxito para emitir el reporte.

Resultados:

- Se visualiza el reporte.

**Escenario 4.2**.- Presentación no exitosa del reporte generación de reglas de asociación por falta de archivo log.

Requisitos:

- No se ingresó previamente exitosamente el archivo log del servidor.

Resultados:

- Mensaje que indique la causa del problema..

**Escenario 4.3**.- Presentación no exitosa del reporte generación de reglas de asociación por falta de parámetros generales.

Requisitos:

- No se ingresó exitosamente todos los parámetros generales para el procesamiento.

Resultados:

- Mensaje que indique la causa del problema..

**Escenario 4.4**.- Presentación no exitosa del reporte generación de reglas de asociación por falta de parámetros soporte y confianza .

Requisitos:

- No se ingresó exitosamente todos los parámetros soporte y confianza para el procesamiento.

Resultados:

- Mensaje que indique la causa del problema..

**Escenario 4.5**.- Presentación no exitosa del reporte generación de reglas de asociación por falla técnica.

Requisitos:

- Se tiene algún problema al realizar el proceso para generación del reporte.

Resultados:

- Mensaje que indique la causa del problema..

**Caso de uso 5**: Secuencia de patrones

En este caso de uso se crearon 5 escenarios detallados a continuación:

**Escenario 5.1**.- Presentación exitosa del reporte secuencia de patrones.

Requisitos:

- Se ingresó previamente exitosamente el archivo log del servidor.
- Se ingresó exitosamente todos los parámetros generales para el procesamiento.
- Se ingresó exitosamente el confianza para el procesamiento.
- Los datos fueron analizados con éxito para emitir el reporte.

Resultados:

- Se visualiza el reporte.

**Escenario 5.2**.- Presentación no exitosa del reporte secuencia de patrones por falta de archivo log.

Requisitos:

- No se ingresó previamente exitosamente el archivo log del servidor.

Resultados:

- Mensaje que indique la causa del problema.

**Escenario 5.3**.- Presentación no exitosa del reporte secuencia de patrones por falta de parámetros generales.

Requisitos:

- No se ingresó exitosamente todos los parámetros generales para el procesamiento.

Resultados:

- Mensaje que indique la causa del problema..

**Escenario 5.4**.- Presentación no exitosa del reporte secuencia de patrones por falta de parámetro confianza .

Requisitos:

- No se ingresó exitosamente todos los parámetro confianza para el procesamiento.

Resultados:

- Mensaje que indique la causa del problema..

**Escenario 5.5**.- Presentación no exitosa del reporte secuencia de patrones por falla técnica.

Requisitos:

- Se tiene algún problema al realizar el proceso para generación del reporte.

Resultados:

- Mensaje que indique la causa del problema.

**Caso de uso 6:** Clustering

En este caso de uso se crearon 5 escenarios detallados a continuación:

**Escenario 6.1**.- Presentación exitosa del reporte para agrupamiento.

Requisitos:

- Se ingresó previamente exitosamente el archivo log del servidor.
- Se ingresó exitosamente todos los parámetros generales para el procesamiento.
- Se ingresó exitosamente el soporte para el procesamiento.
- Los datos fueron analizados con éxito para emitir el reporte.

Resultados:

- Se visualiza el reporte.

**Escenario 6.2**.- Presentación no exitosa del reporte para agrupamiento por falta de archivo log.

Requisitos:

- No se ingresó previamente exitosamente el archivo log del servidor.

Resultados:

- Mensaje que indique la causa del problema.

**Escenario 6.3**.- Presentación no exitosa del reporte para agrupamiento por falta de parámetros generales.

Requisitos:

- No se ingresó exitosamente todos los parámetros generales para el procesamiento.

Resultados:

- Mensaje que indique la causa del problema.

**Escenario 6.4**.- Presentación no exitosa del reporte para agrupamiento por falta de parámetro soporte.

Requisitos:

- No se ingresó exitosamente todos los parámetro soporte para el procesamiento.

Resultados:

- Mensaje que indique la causa del problema..

**Escenario 6.5**.- Presentación no exitosa del reporte para agrupamiento por falla técnica.

Requisitos:

- Se tiene algún problema al realizar el proceso para generación del reporte.

Resultados:

- Mensaje que indique la causa del problema..

### *Diagramas de Diseño de interacción de objetos*

A continuación los diagramas de diseño de interacción de objetos para cada escenario detallado en la sección anterior.

## **Caso de uso 1**: Ingresar archivo del log del servidor

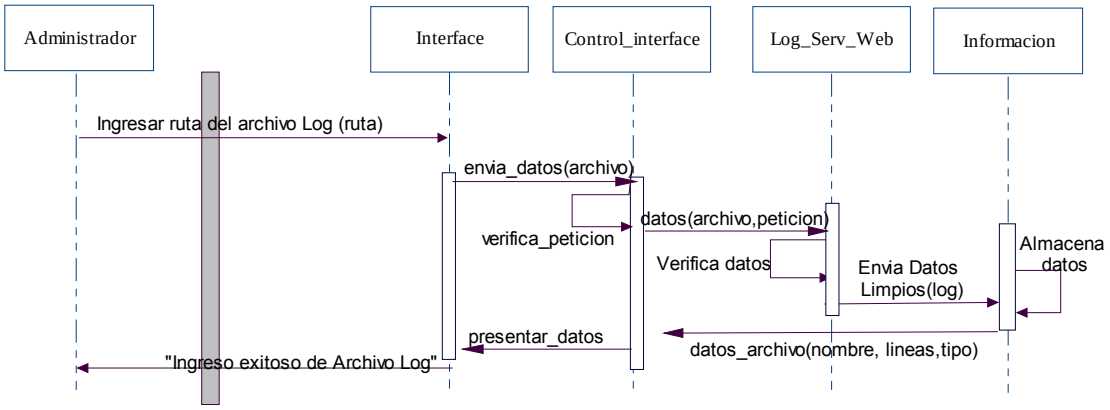



**Figura C.1 Escenario 1.1: Ingreso exitoso del archivo del log del servidor**

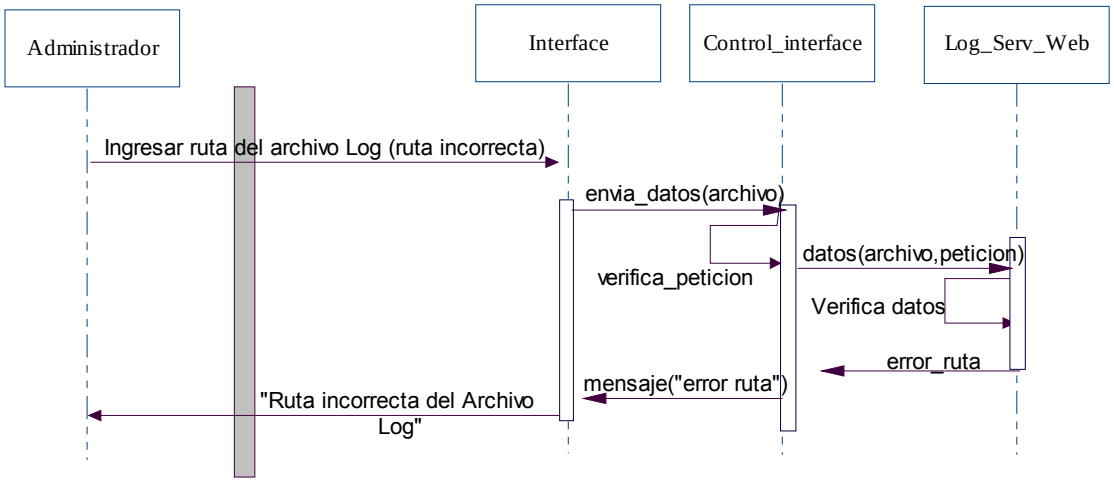



**Figura C.2 Escenario 1.2: Ingreso no exitoso del archivo del log por ruta incorrecta del archivo.**

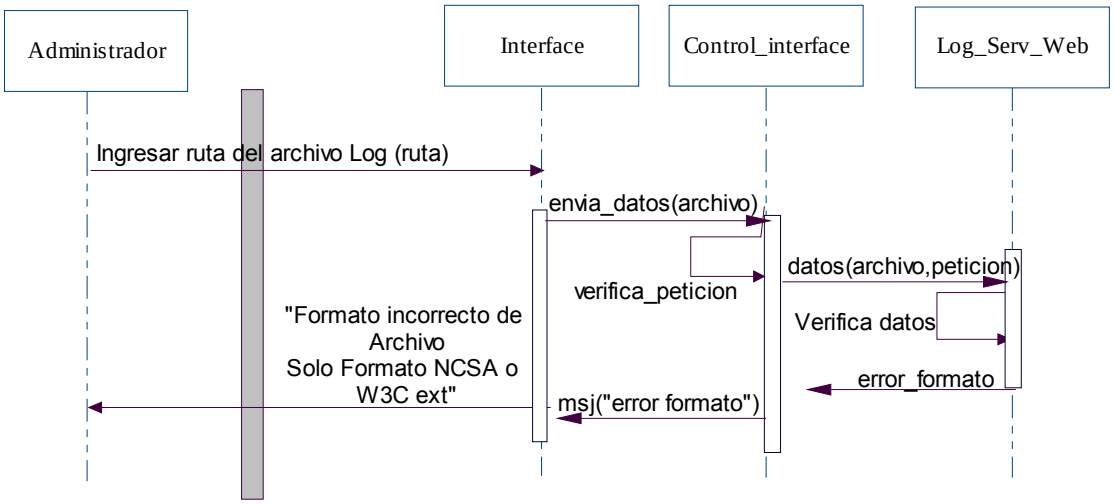



**Figura C.3 Escenario 1.3: Ingreso no exitoso del archivo del log por formato de archivo incorrecto.**

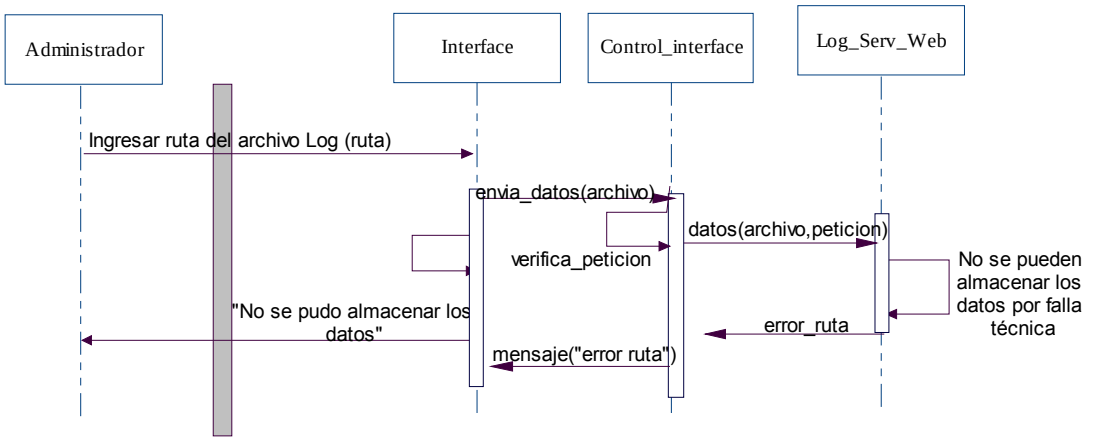



**Figura C.4 Escenario 1.4: Ingreso no exitoso por fallas técnicas en el ingreso del archivo del log.**

**Caso de uso 2**: Seleccionar Parámetros para Procesamiento

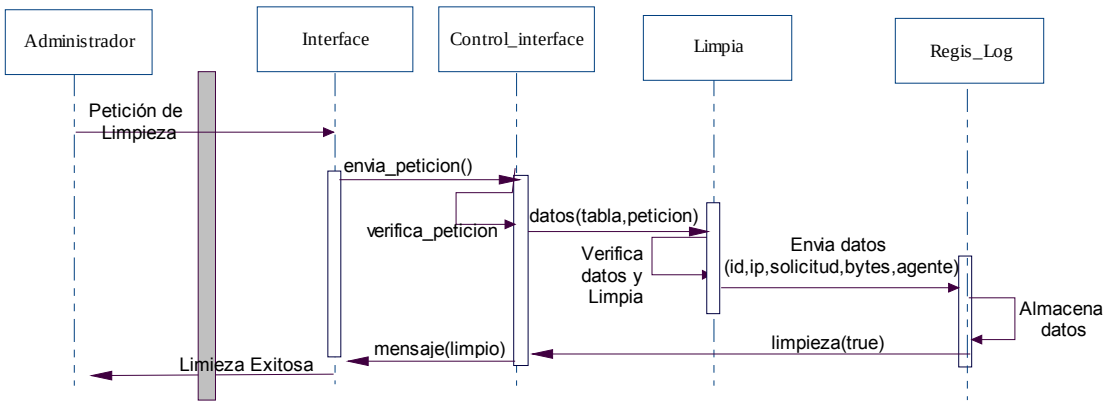



**Figura C.5 Escenario 2.1: Selección exitosa de limpieza de archivo log para Procesamiento**

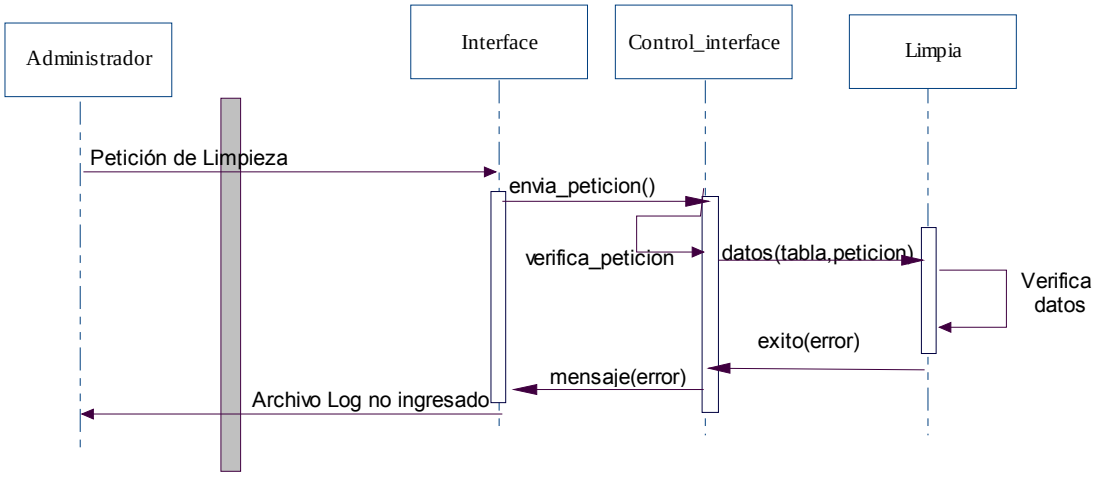



**Figura C.6 Escenario 2.2: Selección no exitosa de limpieza de archivo log por no ingreso de archivo log.**

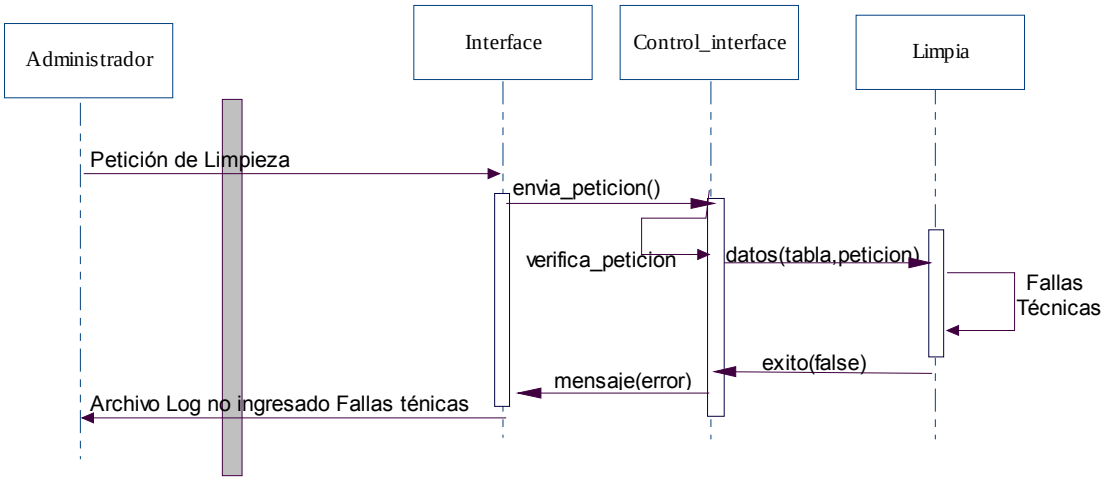



**Figura C.7 Escenario 2.3: Selección no exitosa de limpieza de archivo log por fallas técnicas.**

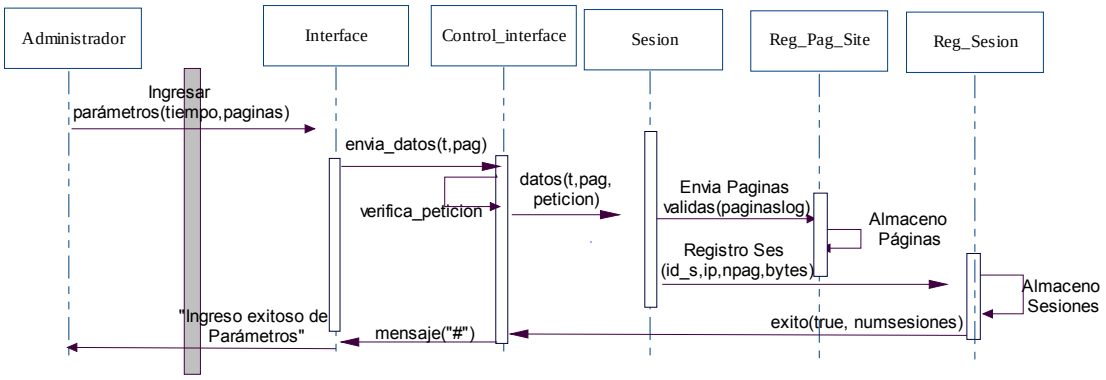



**Figura C.8 Escenario 2.4: Ingreso exitoso de parámetros para procesamiento**

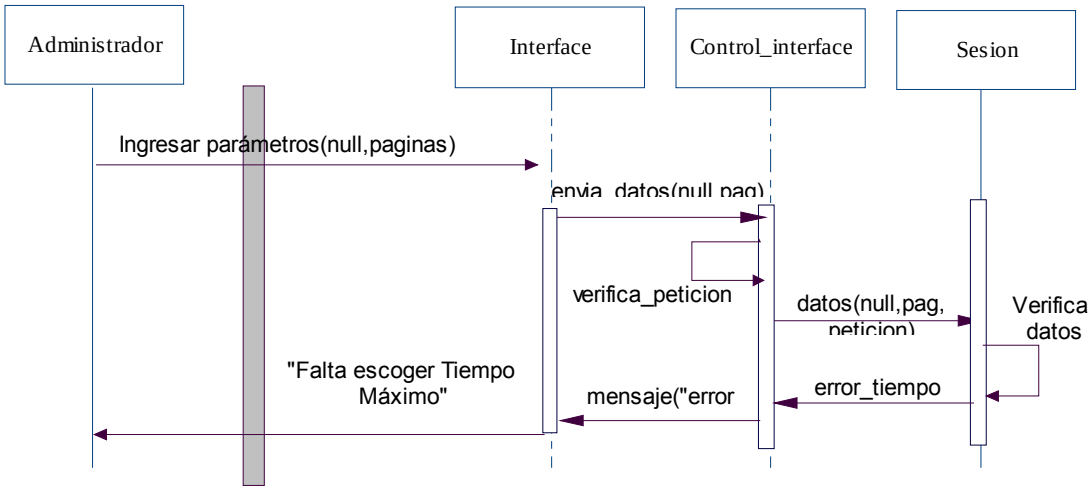



**Figura C.9 Escenario 2.5: Ingreso no exitoso de parámetros para procesamiento por falta de Tiempo Máximo.**

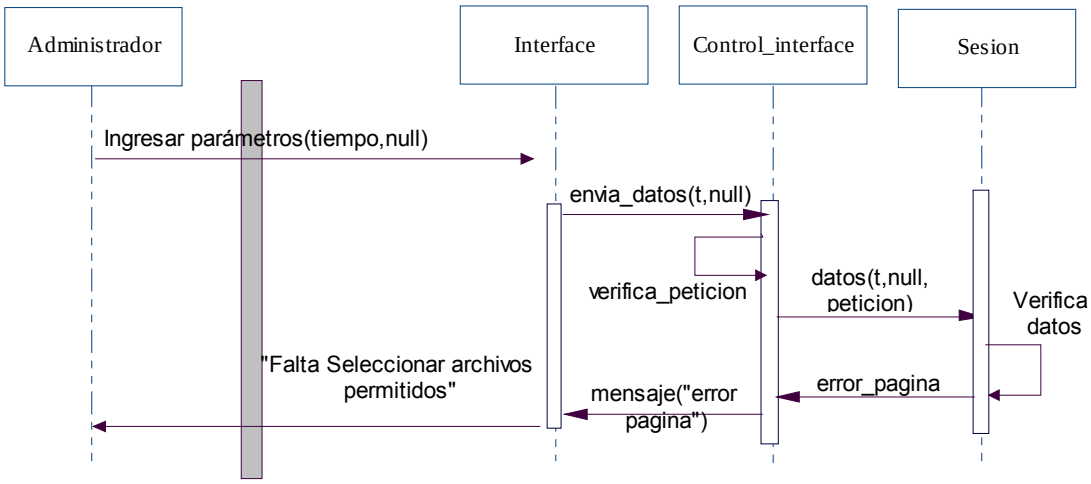



**Figura C.10 Escenario 2.6: Ingreso no exitoso de parámetros para procesamiento por falta de archivos permitidos**

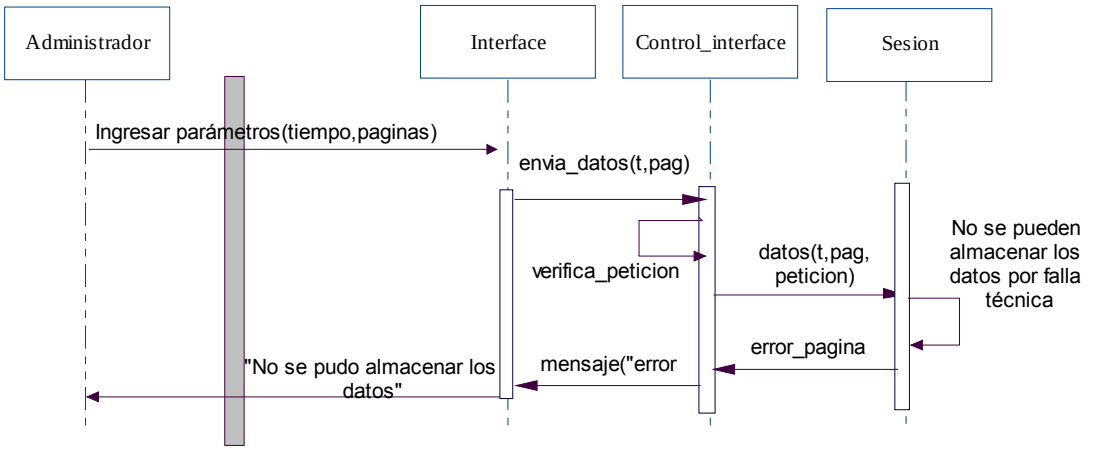



**Figura C.11 Escenario 2.7: Ingreso no exitoso por fallas técnicas en el ingreso de parámetros para procesamiento.**

## **Caso de uso 3**: Presentar reporte

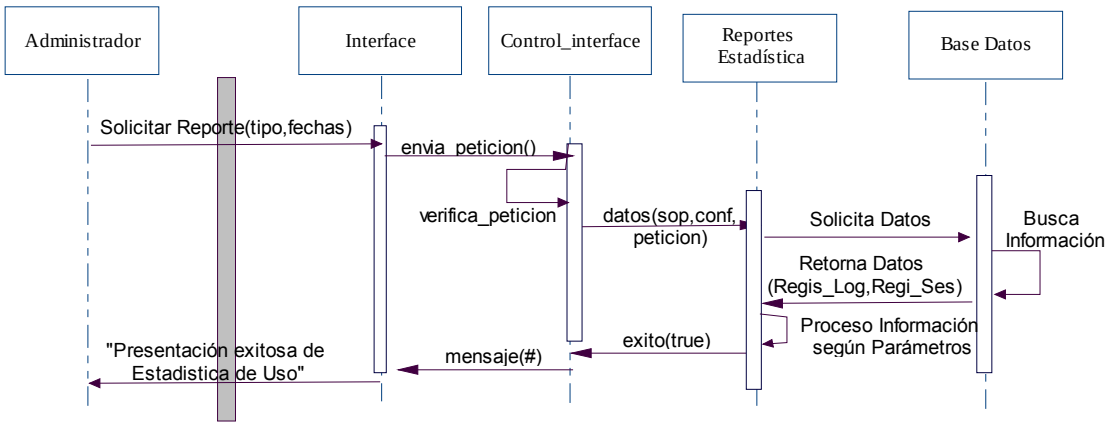



**Figura C.12 Escenario 3.1: Presentación exitosa del reporte para estadísticas de uso.**

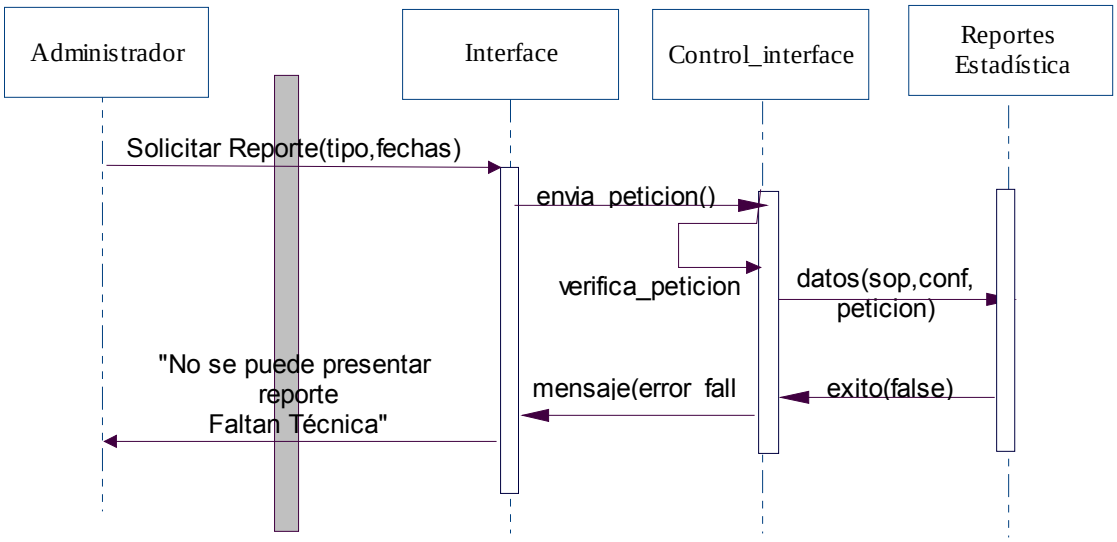



**Figura C.13 Escenario 3.2: Presentación no exitosa del reporte para estadísticas de uso.**

## **Caso de uso 4**: Reglas de asociación

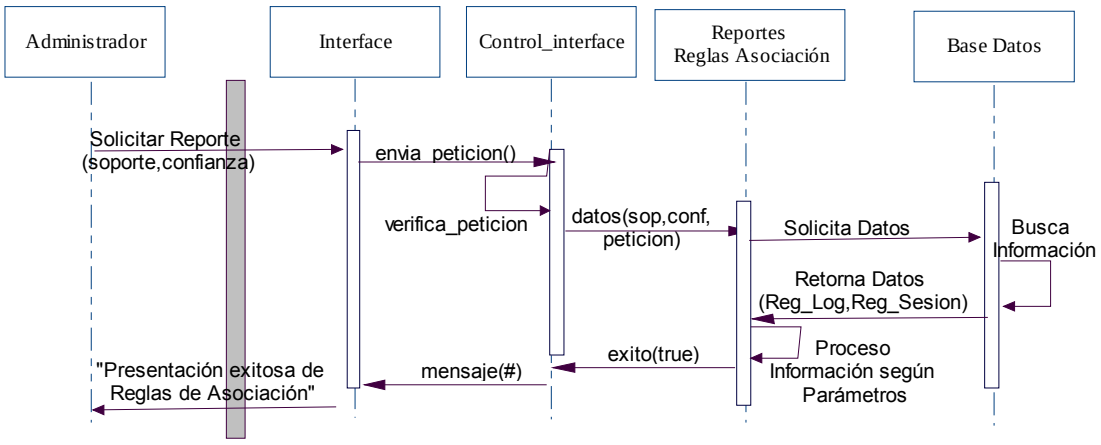



**Figura C.14 Escenario 4.1: Presentación exitosa del reporte generación de reglas de asociación.**

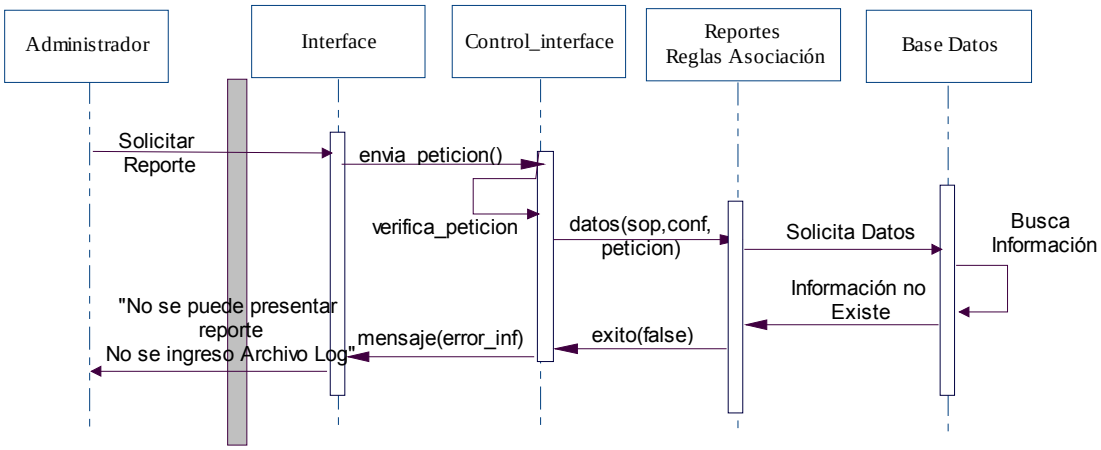



**Figura C.15 Escenario 4.2: Presentación no exitosa del reporte generación de reglas de asociación por falta de archivo log.**

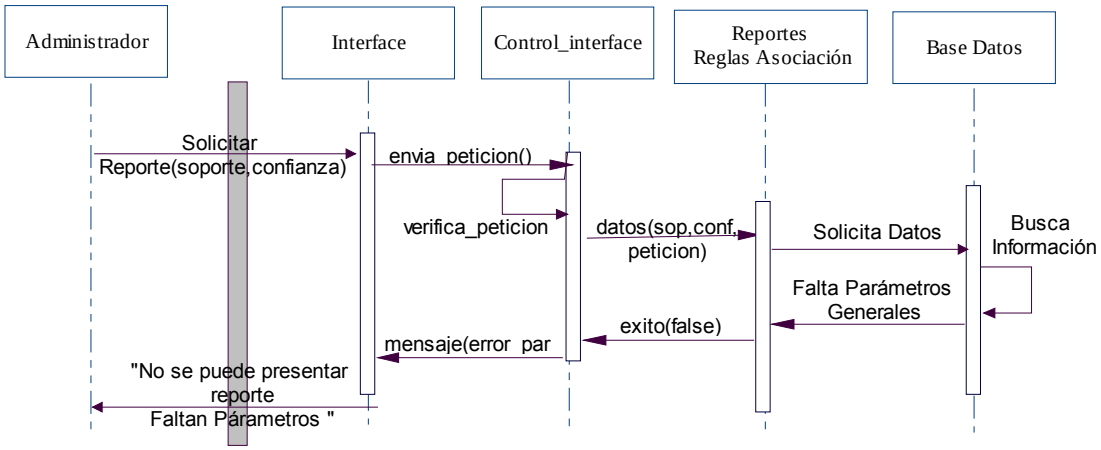



**Figura C.16 Escenario 4.3: Presentación no exitosa del reporte generación de reglas de asociación por falta de parámetros generales.**

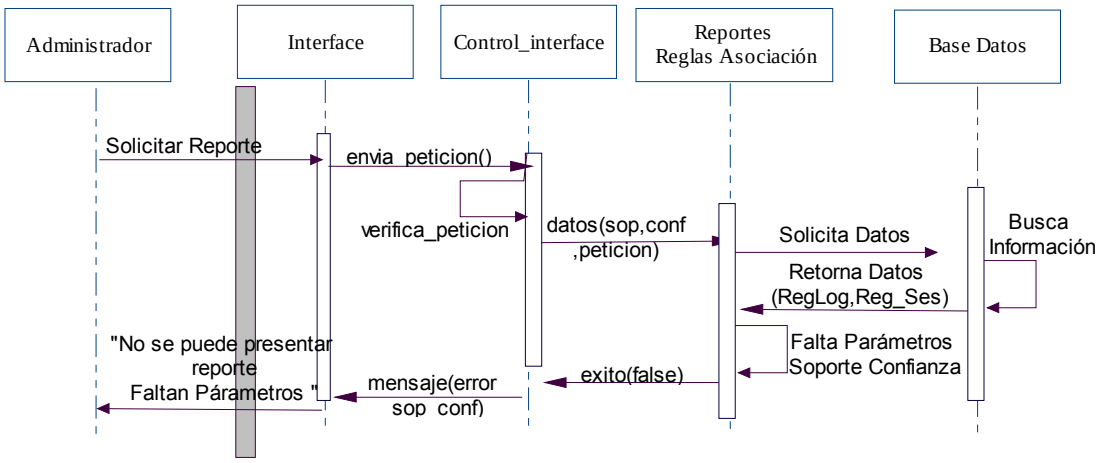



**Figura C.17 Escenario 4.4: Presentación no exitosa del reporte generación de reglas de asociación por falta de parámetros soporte y confianza.**

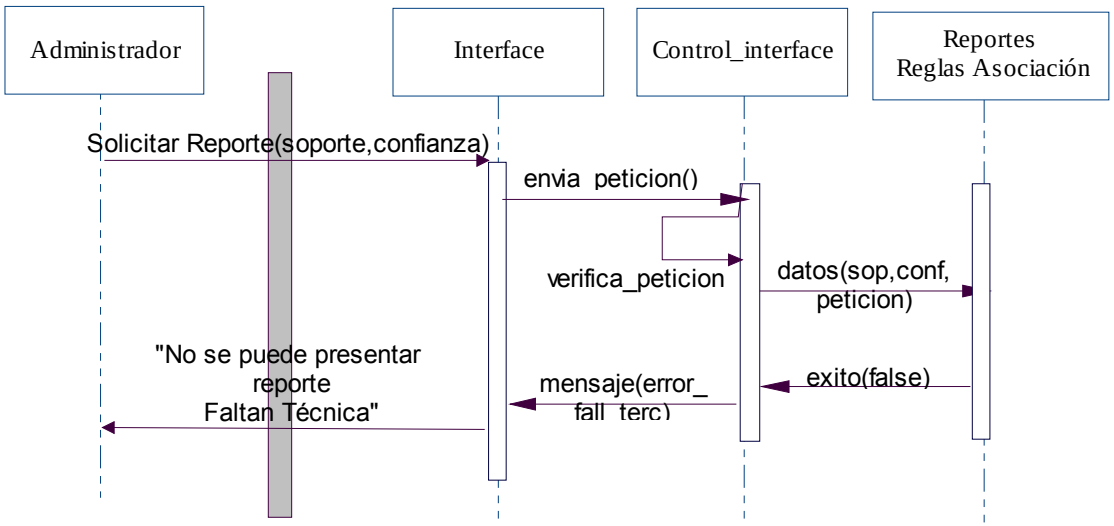



**Figura C.18 Escenario 4.5: Presentación no exitosa del reporte generación de reglas de asociación por falla técnica.**

**Caso de uso 5**: Secuencia de patrones

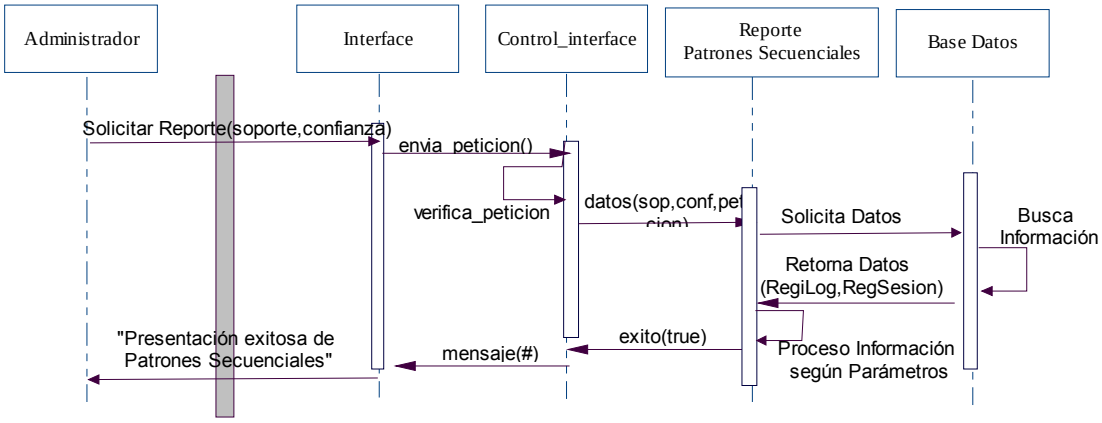



**Figura C.19 Escenario 5.1: Presentación exitosa del reporte secuencia de patrones.**

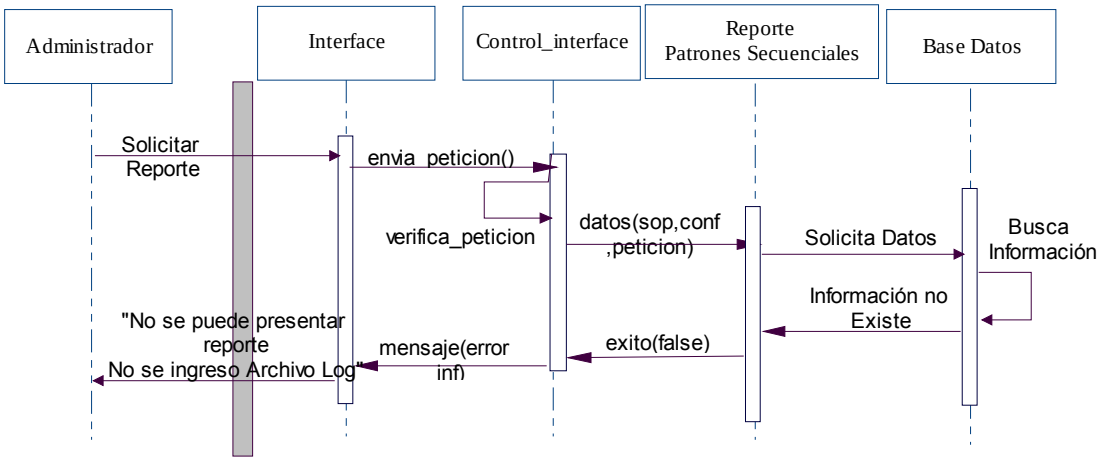



**Figura C.20 Escenario 5.2: Presentación no exitosa del reporte secuencia de patrones por falta de archivo log.**

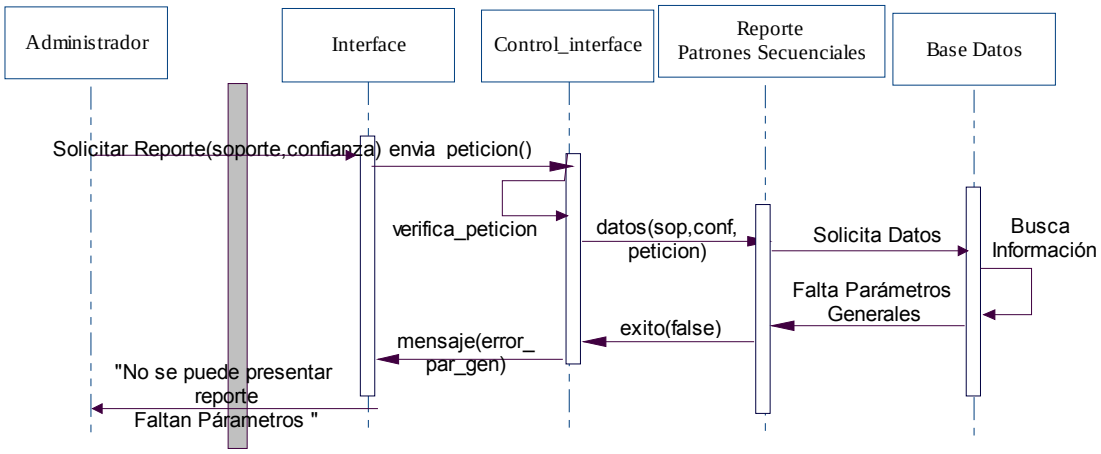



**Figura C.21 Escenario 5.3: Presentación no exitosa del reporte secuencia de patrones por falta de parámetros generales.**

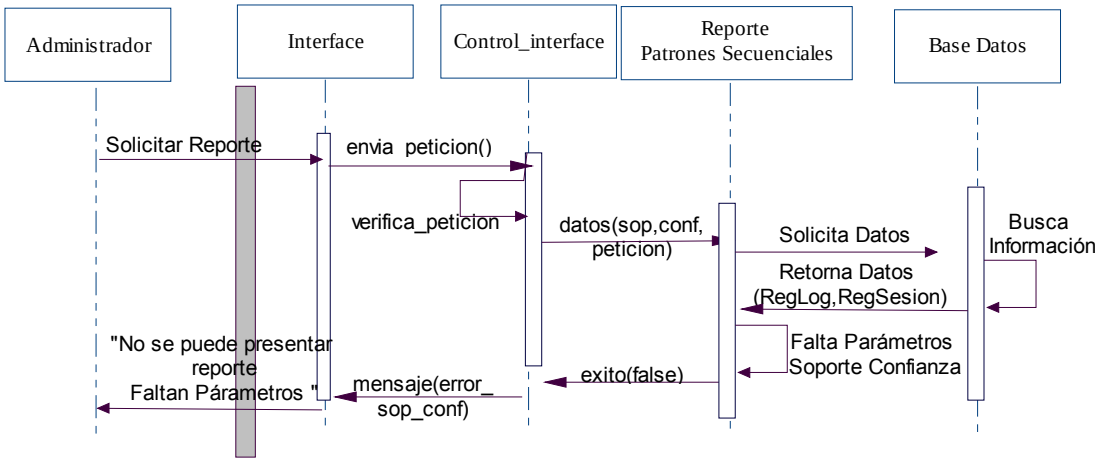



**Figura C.22 Escenario 5.4: Presentación no exitosa del reporte secuencia de patrones por falta de parámetro confianza.**

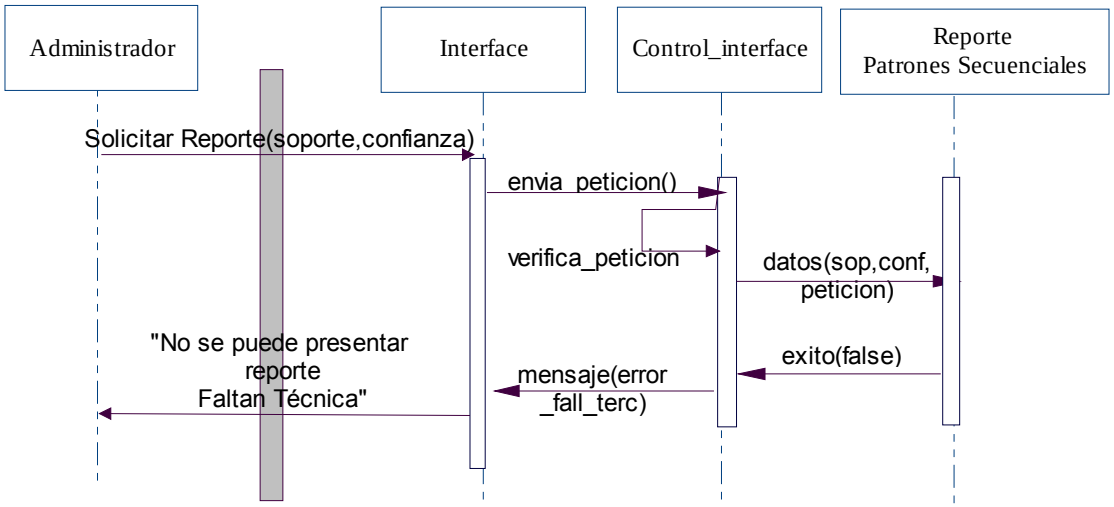



**Figura C.23 Escenario 5.5: Presentación no exitosa del reporte secuencia de patrones por falla técnica.**

**Caso de uso 6:** Agrupamiento

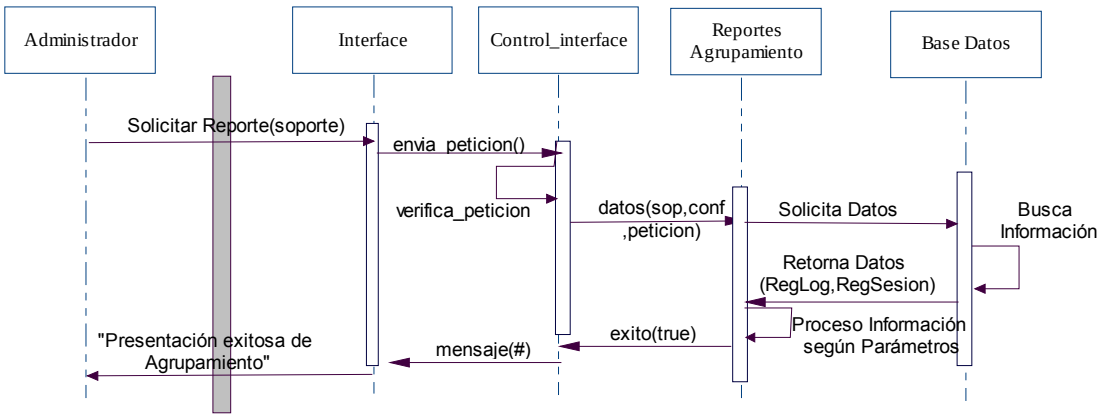



**Figura C.24 Escenario 6.1: Presentación exitosa del reporte para agrupamiento.**

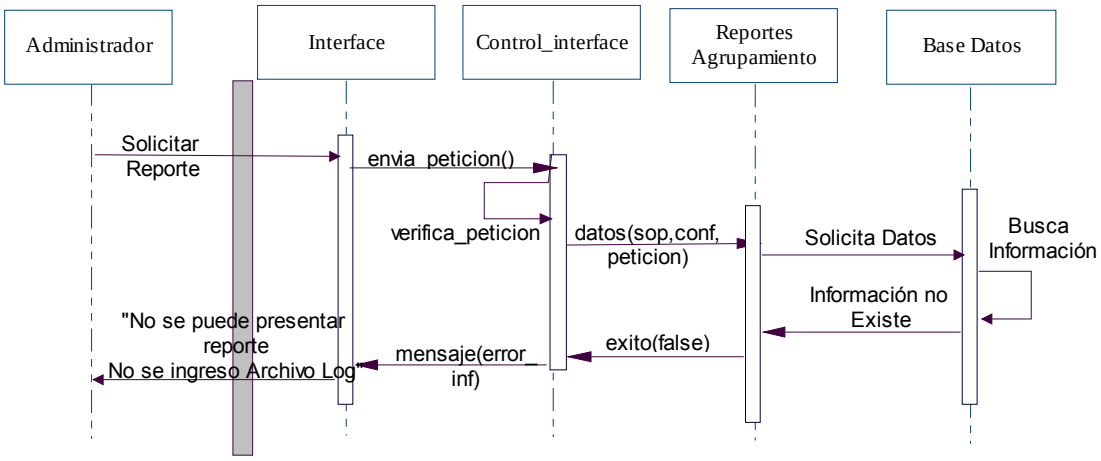



**Figura C.25 Escenario 6.2: Presentación no exitosa del reporte para agrupamiento por falta de archivo log.**

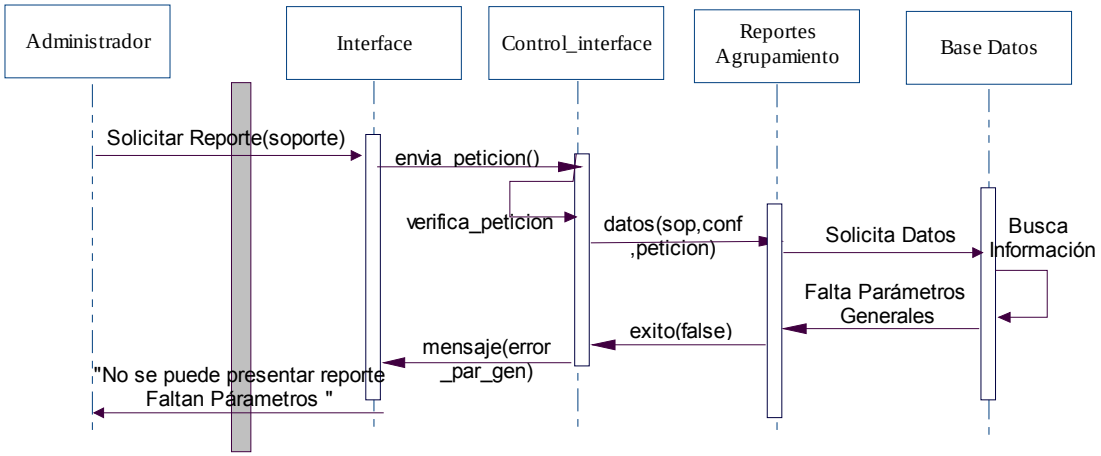



**Figura C.26 Escenario 6.3: Presentación no exitosa del reporte para agrupamiento por falta de parámetros generales.**

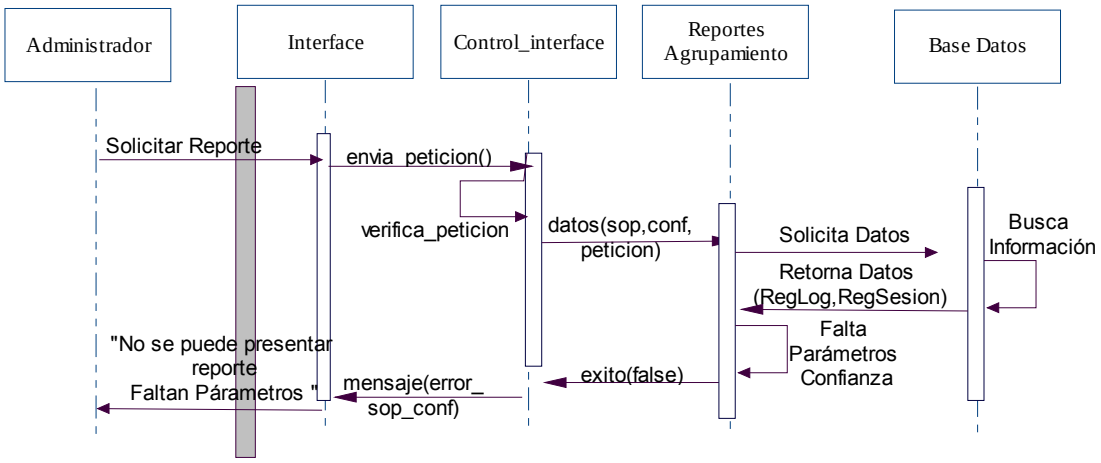



**Figura C.27 Escenario 6.4: Presentación no exitosa del reporte para agrupamiento por falta de parámetro soporte.**

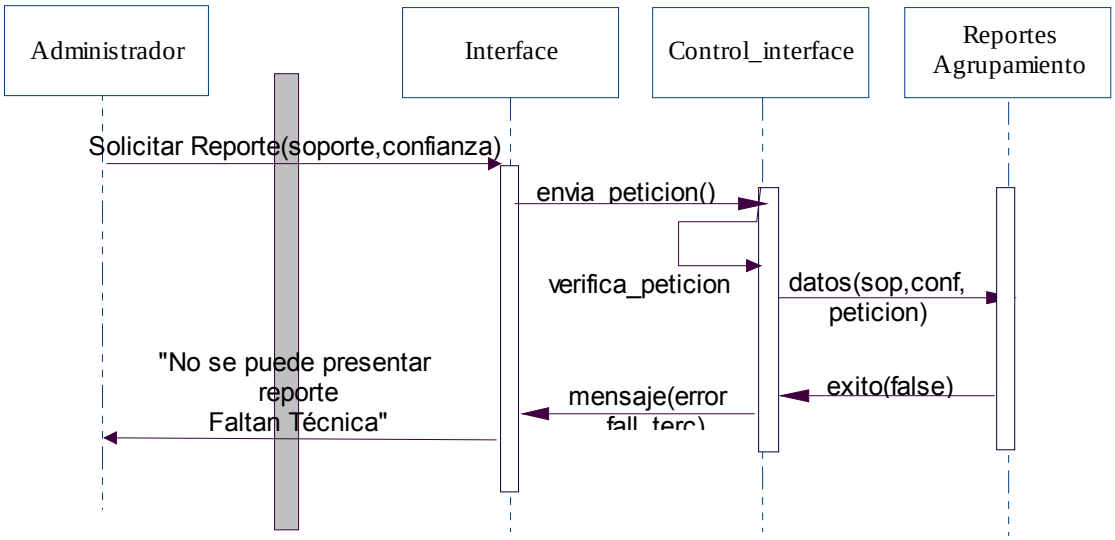



**Figura C.28 Escenario 6.5: Presentación no exitosa del reporte para agrupamiento por falla técnica.**

# *ANEXO D*

# *MANUAL DE USUARIO*

Esta información ha sido diseñada para ayudarlo a iniciarse en el Sistema Minero Web, para guiarlo en su uso y aplicación

**¿Qué es el Sistema MineroWeb?**

El sistema Minero Web, es una aplicación web diseñada para aquellas personas que mantienen un negocio en la Internet, es una herramienta útil y eficaz para la extracción de patrones de conducta de los visitantes de su sitio web.

El sistema ofrece como resultado, además de gráficos estadísticos de los movimientos de su sitio, patrones de conducta que predicen formas de comportamiento de los usuarios o visitantes de su sitio.

Para hacer uso del sistema en primer lugar debe disponer de el o los archivos log generados por su servidor, en este archivo se guarda información histórica de los visitantes y requerimientos a su sitio.

Al ingresar al sistema se le presentará una pantalla de bienvenida (Figura D.1)

La pantalla de bienvenida presenta una breve descripción de las características y funcionamiento del sistema.

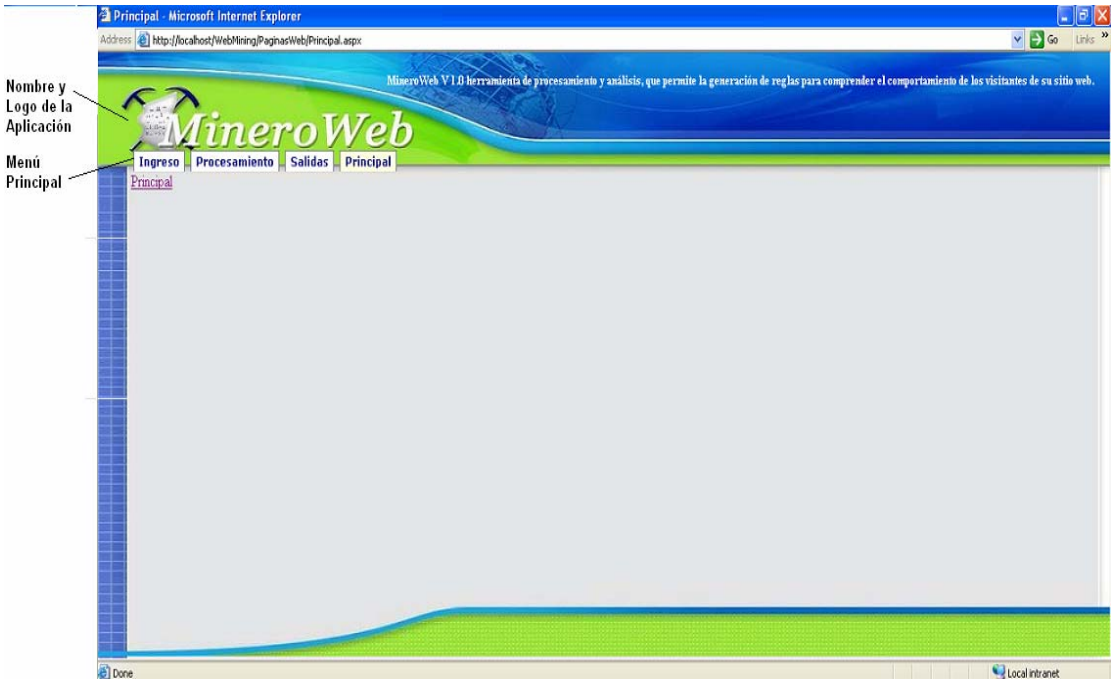



**Figura D.1 Página de bienvenida**

### Ingresando los archivos log

Como primer paso usted debe realizar el ingreso de  o los archivos log de su sitio, para ello:

Seleccione del menú principal la opción Ingreso de Archivo.  La pantalla que se le presenta esta representada en la figura D.2.

Ingrese la dirección y el nombre del archivo log que se quiera analizar.

Si desconoce la ruta del archivo, pulse el botón  XXXX para buscarlo.

Seleccione el archivo y pulse el botón Abrir.

Cuando tenga lista la dirección del archivo log pulse el botón Cargar Archivo.

La aplicación mostrará una tabla con datos del archivo como:

- Número de archivo log agregado
- Nombre del archivo log.
- Formato del archivo log.
- Número de líneas que tiene el archivo log

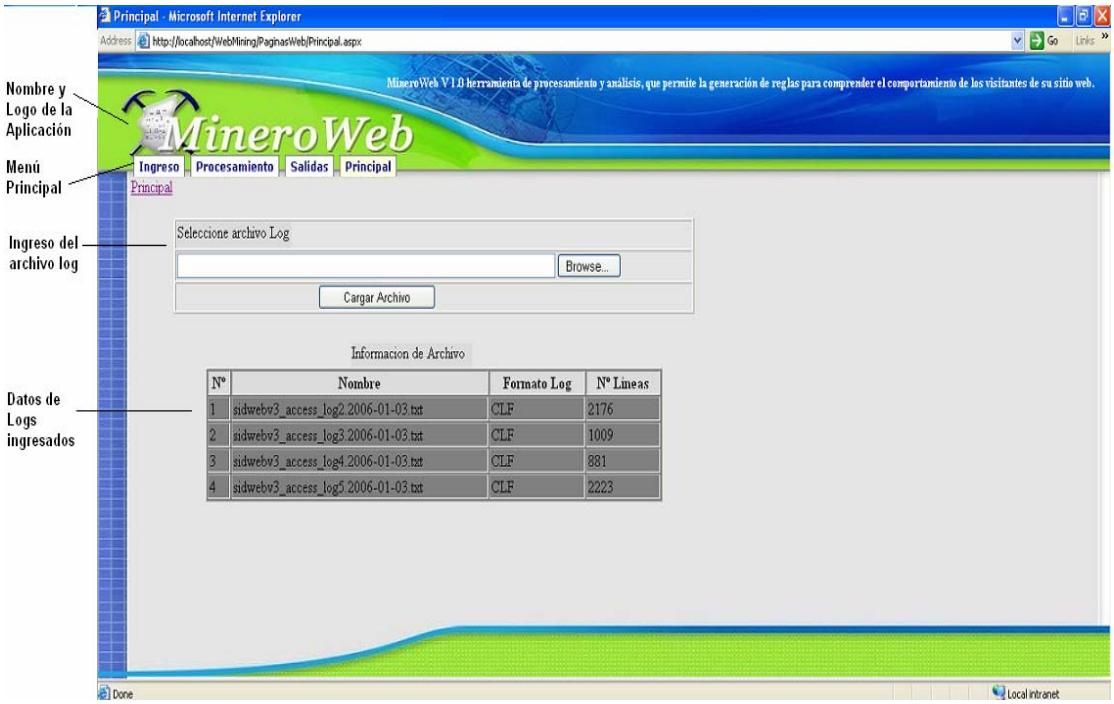



**Figura D.2 Ingreso del archivo log del servidor**

## Realizando el procesamiento de la información

Una vez ingresado y cargado el archivo, debe realizar dos procedimientos. Sin estos procedimientos, ningún reporte podrá ser presentado.

Seleccione la opción Preprocesamiento del menú principal Procesamiento. Esta elección da inicio a un proceso de limpieza sobre los datos.

Active el botón “Limpieza” en la pantalla que muestra la aplicación.

Cuando se ha concluido la limpieza, se presenta un gráfico estadístico que muestra el contraste de tamaños entre la data original del archivo log y la data resultante de la limpieza que es la que se va a analizar. Esta página se encuentra plasmada en la figura D.3 que se encuentra a continuación.

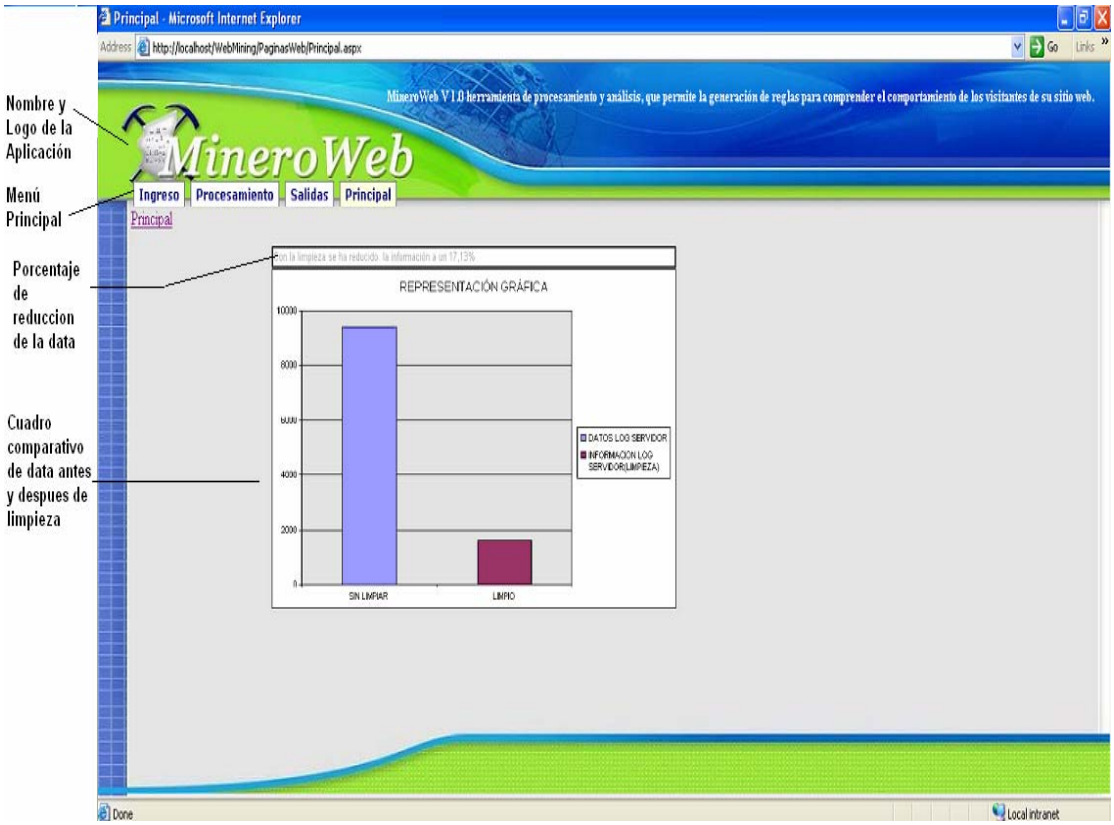



**Figura D.3 Página del proceso de limpieza del archivo log**

Cuando se ha realizado la limpieza seleccione la opción de “Parámetros” del menú principal Procesamiento.

La página que la aplicación activa se encuentra representada por la figura D.4.

Seleccione que páginas dentro de su sitio Web deben ser consideradas como un nuevo requerimiento.

Escoja el tiempo máximo de sesión (tiempo que limita que requerimientos realizados por un mismo visitante pertenecen a una misma sesión)

Pulse el botón Sesionización.

La aplicación mostrará cuantas sesiones fueron descubiertas a partir de los datos ingresados.

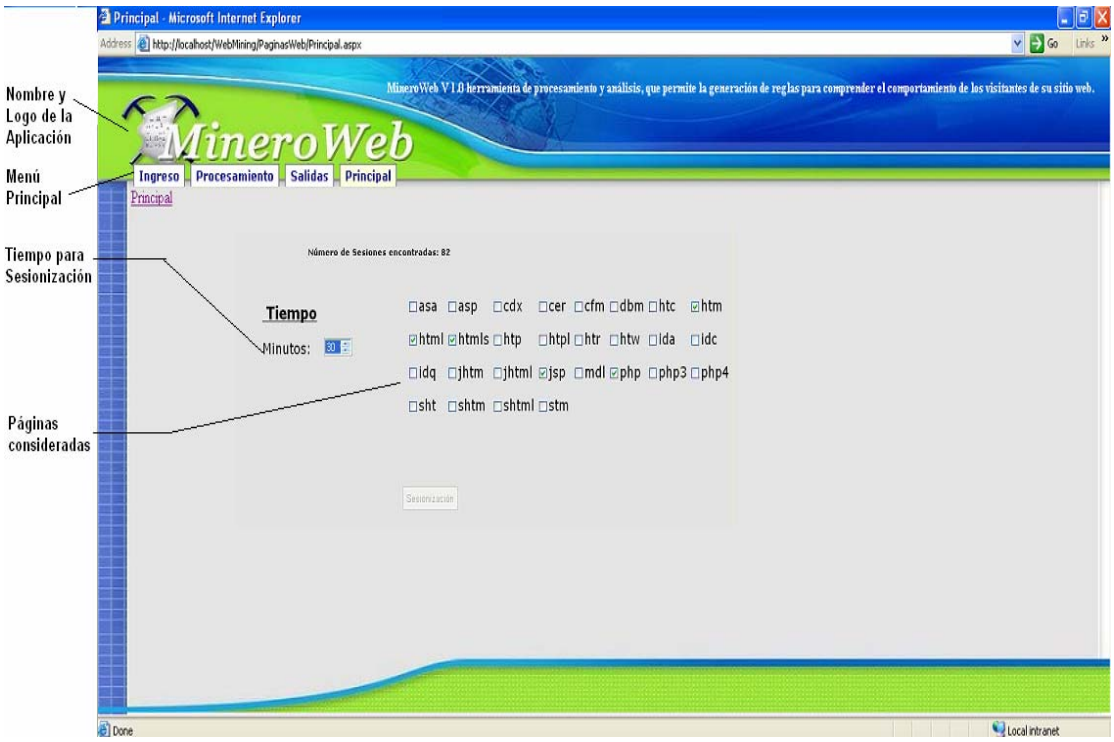



**Figura D.4 Página de sesionización y selección de parámetros generales**

## Emitiendo reportes

El sistema Minero Web genera una gama de reportes una vez que se han realizado los procesos de limpieza y selección de parámetros:

Figura D.5 Reporte para generación de reglas de asociación

Figura D.6 Reporte para secuencia de patrones.

Figura D.7 Reporte para clustering

Figura D.8 Reporte para estadísticas de uso

**Reporte de reglas de asociación**

Para que pueda visualizar los reportes de reglas de asociación es necesario que:

Escoja la opción Salidas del menú principal del sistema

Escoja la opción Reglas de Asociación

Es necesario que ingrese dos parámetros: soporte y confianza. Estos parámetros serán los valores mínimos de soporte y confianza que puede tener cada regla generada.

Presione el botón Reglas.

El número de reglas solo dependerá de la información existente y de los valores de soporte y confianza requeridos.

En el reporte emitido se presenta el número de reglas generadas, el antecedente y precedente de cada regla, la confianza y soporte real de cada regla.

Como ejemplo el reporte que se presenta en la figura D.5 nos dice que se han generado 4 reglas. La regla número nos dice, que el 85.71% de los usuarios que visitaron private/mycourses/website/email/index.jsp también visitaron private/mycourses/website/index.jsp.

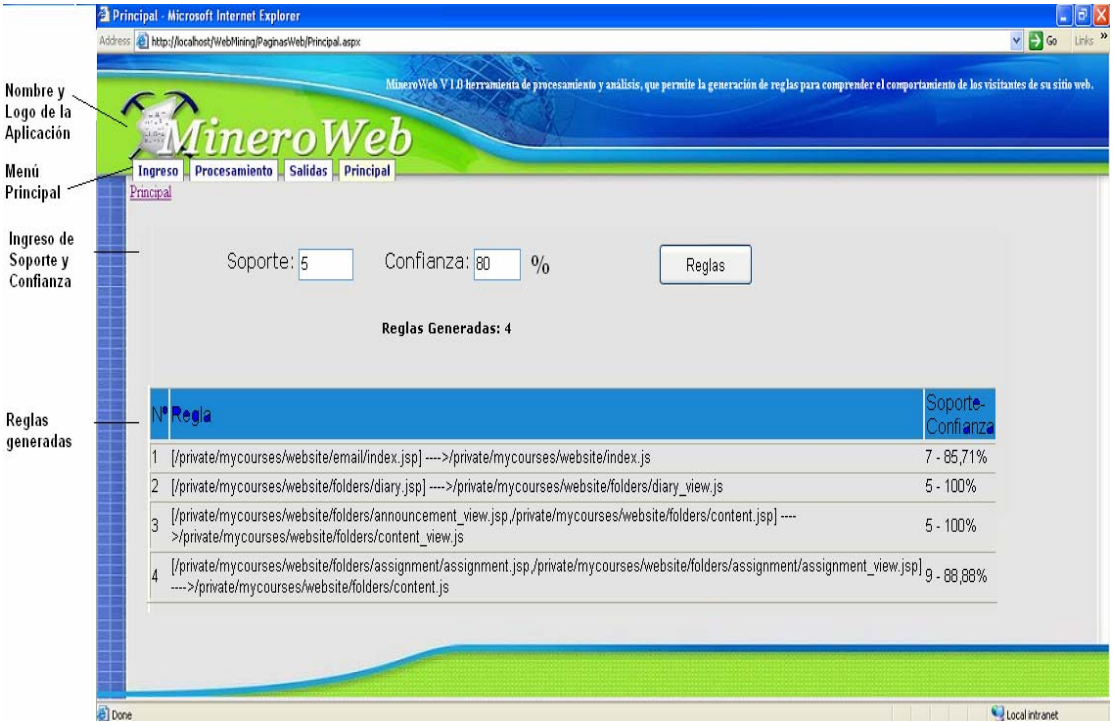



**Figura D.5 Reporte para generación de reglas de asociación**

**Reporte de Secuencia de Patrones**

Para que pueda visualizar los reportes de secuencia de patrones es necesario que:

Escoja la opción Salidas del menú principal del sistema

Escoja la opción Secuencia de Patrones

Es necesario que ingrese un parámetro: la confianza, será el valor mínimo de confianza que puede tener cada patrón generado.

Presione el botón Patrones.

El número de patrones presentados solo dependerá de la información existente y de la confianza requerida.

En el reporte se muestra distintos patrones que se generan al existir alguna secuencia entre las transacciones de un mismo usuario a lo largo del tiempo.

Cada reporte presentará el número de patrones generados, el antecedente y precedente de cada patrón, y la confianza real de cada uno.

El patrón número uno que se presenta en la figura D.6 nos dice, que el 88.88% de los usuarios que visitan la secuencia de páginas private/mycourses/website/folders/assignment/assignment.jsp, private/mycourses/website/folders/content.jsp, van a visitar private/mycourses/website/folders/groupView.jsp en aproximadamente 0 dias.


**Figura D.6 Reporte para secuencia de patrones.**

**Reporte para clustering**

Para que pueda emitir los reportes de clustering siga los siguientes pasos:

Escoja la opción Salidas del menú principal del sistema

Escoja la opción Clusterización.

Es necesario que ingrese un parámetro: el soporte mínimo que debe presentar cada cluster en que divide el sistema a las páginas del sitio.

Presione el botón Iniciar.

El reporte representado en la figura D.7 mostrará datos como: número de sesiones encontradas, número de iteraciones necesarias para determinar los clusters, número de clusters encontrados, soporte real de cada cluster. Así como los clusters con cada una de las páginas que lo conforman.

Para su guía, el reporte nos indica que existe grupos con características similares de navegación, en este caso se han generado 4 grupos o clusters, en donde el cluster número uno define similitudes entre las páginas private/mycourses/website/folders/announcement_view.jsp, private/mycourses/website/ index.jsp y private/mycourses/website/scores/index.jsp

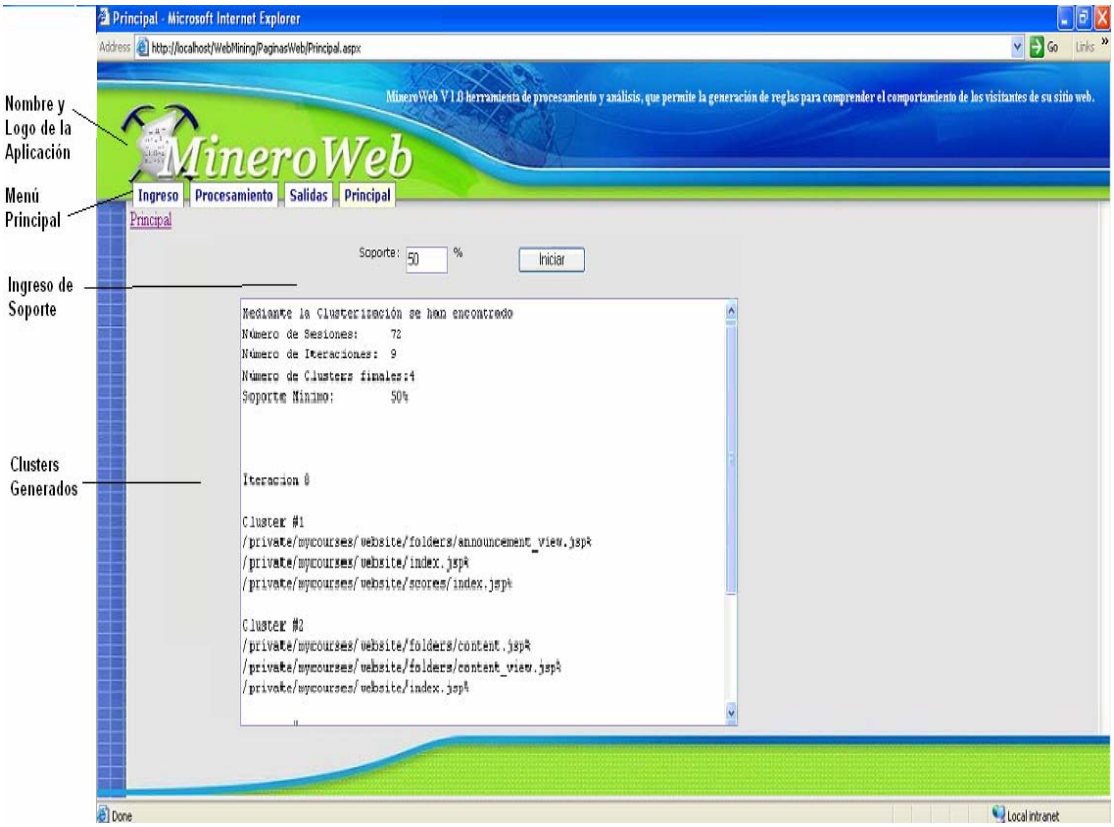



**Figura D.7 Reporte para Clustering**

**Reporte para estadísticas de uso**

Para que pueda emitir cualquiera de los reportes de estadísticas de uso siga los siguientes pasos:

Escoja la opción Salidas del menú principal del sistema

Escoja la opción Estadísticas de Uso.

Se le presentará una pantalla, en donde deberá escoger el tipo de reporte que desea generar, y la fecha tanto inicial como final del reporte.

Una vez que haya escogido el reporte y las fechas límites, presiones el botón Graficar.

El nuevo reporte es generado en la misma página.

El reporte mostrará un cuadro comparativo con la información previamente almacenada y procesada.

En la figura D.8 se aprecia el el reporte de páginas más visitadas , en donde podemos apreciar como /private/mycourses/website/calendar/index.jsp es la página más visitada dentro del sitio web.


**Figura D.8 Página de reportes estadísticos**