POLARIS VERSIÓN 3.0
Una herramienta para la minería de uso y estructura de la web y análisis estadístico de tráfico web

DIEGO ARMANDO TOBAR IBARRA

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2012

POLARIS VERSIÓN 3.0
Una herramienta para la minería de uso y estructura de la web y análisis estadístico de tráfico web

DIEGO ARMANDO TOBAR IBARRA

Trabajo de grado presentado como requisito parcial para optar al título de Ingeniero de Sistemas

Director del Proyecto
RICARDO TIMARÁN PEREIRA, Ph. D.

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2012

NOTA DE RESPONSABILIDAD

Las ideas y conclusiones aportadas en el siguiente trabajo son responsabilidad exclusiva del autor.

Artículo 1ro del Acuerdo No. 324 de octubre 11 de 1966 emanado del Honorable Consejo Directivo de la Universidad de Nariño.

Nota de aceptación:

_____
_____
_____
_____
_____
_____

_____
Firma del presidente del jurado

_____
Firma del jurado

_____
Firma del jurado

San Juan de Pasto, Noviembre de 2012

AGRADECIMIENTOS

Agradezco a Dios, por ser quien ha estado a nuestro lado en todo momento, brindándonos energía y fortaleza constantes en cada paso durante nuestra carrera.

A mis familiares, por su apoyo, comprensión y colaboración incondicional y permanente.

Al Ingeniero Ricardo Timaran Pereira Ph. D., Director de mi proyecto, por su tiempo, paciencia y asistencia continúa durante el desarrollo del proyecto.

A mis amigos y compañeros, por ser partícipes en el proceso de esta etapa de nuestra vida.

A mis profesores, por compartir con nosotros sus conocimientos y experiencias.

Y a todas aquellas personas que contribuyeron en la realización de este proyecto de investigación.

DEDICATORIA

A Dios, porque en su infinita sabiduría me dotó de herramientas maravillosas; puso un sueño, una ilusión y un soplo de esperanza. Me dio unas inmensas alas para ir en busca de mi meta y guió mi camino.

A mis padres, que cultivaron en forma perseverante en mí el espíritu de lucha y superación, me infundieron la confianza y me han fortalecido constantemente dándome una palabra de aliento, una voz de consuelo en los momentos difíciles y enseñándome que no hay limitación más grande que la que se ponga uno mismo. Alcanzar este peldaño en mi vida es mérito también de ellos.

A mis hermanos, que han creído en mí y me han apoyado incondicionalmente en el sendero hacia este logro.

A mis docentes, por su sabiduría, conocimiento y ejemplo, porque me motivaron a continuar.

A todos quienes estuvieron cerca y me brindaron toda la colaboración que necesité.

Hoy les dedico este trabajo de grado, porque en cada página hay algo de cada uno de ustedes.

RESUMEN

Este documento contiene el análisis y diseño del trabajo de grado: **POLARIS Versión 3.0** – Una Herramienta de Minería de Uso y Estructura Web y Análisis Estadístico de Tráfico Web.

.

"**POLARIS Versión 3.0**" es una herramienta que se desarrolló pensando en la necesidad de crear una suite de minería y estadísticas web, básicamente busca descubrir patrones de uso de la web, mejorar la estructura de un portal web y realizar un análisis estadístico de tráfico web de un portal web determinado, dicha herramienta se encuentra orientada hacia los administradores Web.

Esta herramienta se encuentra dividida en tres grandes módulos: El módulo de Minería de Uso Web, implementado en la primera versión de Polaris, el módulo de minería de estructura web implementado en su segunda versión y el módulo de análisis estadístico de tráfico web, y sobre la cual trata el presente documento.

La principal característica de **POLARIS V 3.0** es que tiene la posibilidad de trabajar con diferentes formatos de archivos log de servidor web, los cuales son cargados registro a registro en una base de datos, permitiendo el análisis dinámico de estos archivos mediante el uso un lenguaje de consulta estructurado SQL. Además, tiene la posibilidad de realizar un proceso previo de identificación de sesiones, el cual permite agrupar en sesiones, las entradas registradas por un visitante de la misma máquina en un lapso de tiempo definido, permitiendo la identificación de visitantes únicos.

Por otra parte, en la categoría de estadísticas web, cuenta con diferentes procesos que permiten generar información útil con datos sobre los visitantes de un sitio web, su actividad estadística, archivos a los que acceden, mensajes de error, sistemas operativos e información de las aplicaciones que acceden al sitio web como navegadores web, crawlers, proxys, link chekers o spams.

Finalmente se tiene las diferentes vistas que presenta la herramienta, donde encontramos tablas y gráficos estadísticos, las cuales permiten observar de forma amigable y detallada el resultado de la información recolectada y procesada.

ABSTRACT

This document contains the analysis and design of the grade: **POLARIS Version 3.0** - A Mining Tool Use and Web Structure and Statistical Analysis of Web Traffic.

"**POLARIS Version 3.0**" is a tool that was developed with the need to create a suite of Mining and Web Statistics basically seeks to discover patterns of web usage, improve the structure of a Web site and perform a statistical analysis of Web traffic of a particular Web site, this tool is oriented towards the Webmaster.

This tool is divided into three modules: Module Web Usage Mining, implemented in the first version of Polaris, the module deployed Web Structure Mining in its second version and the Statistical Analysis Module Web Traffic, and which is this document.

The main feature of POLARIS VERSION 3.0 is that it has the ability to work with different formats of Web Server log files, which are loaded row by row in a database, allowing the dynamic analysis of these files by using a language of SQL Structured Query. It also has the possibility of a previous process of identifying sessions, which allows you to group sessions, entries made by a visitor to the same machine in a defined time, allowing the identification of unique visitors.

Moreover, in the category of Web statistics, with different processes that can generate useful information with data about visitors to a website, activity statistics, accessed files that, error messages, operating systems and information from the applications that access the web site as web browsers, crawlers, proxies, link Checkers or spam.

Finally there is the different views presented by the tool, we find statistical tables and graphs, which allow to observe in a friendly and detailed the result of the information collected and processed.

CONTENIDO

Pág.

LISTA DE TABLAS

Pág.

LISTA DE FIGURAS

Pág.

GLOSARIO

Cliente: Computadora o programa que se conecta a servidores para obtener información. Un cliente sólo obtiene datos, no puede ofrecerlos a otros clientes sin depositarlos en un servidor. La mayoría de las computadoras que las personas utilizan para conectarse y navegar por Internet son clientes.

Cliente/Servidor: Sistema de organización de interconexión de computadoras según el cual funciona Internet, así como otros tantos sistemas de redes. Se basa en la separación de las computadoras miembros en dos categorías: las que actúan como servidores (oferentes de información) y otras que actúan como clientes (receptores de información).

Dominio: Un dominio de Internet es una red de identificación asociada a un grupo de dispositivos o equipos conectados a la red Internet. El propósito principal de los nombres de dominio en Internet y del sistema de nombres de dominio (DNS), es traducir las direcciones IP de cada nodo activo en la red, a términos memorizables y fáciles de encontrar. Esta abstracción hace posible que cualquier servicio (de red) pueda moverse de un lugar geográfico a otro en la red Internet, aun cuando el cambio implique que tendrá una dirección IP diferente.

Hiperenlace (Link): Link, hipervínculo, vínculo. Conexión entre dos equipos o nodos. Conexión de una página Web con otra mediante una palabra que representa una dirección de Internet (Url). Generalmente un enlace está subrayado y es azul. También sirve para descarga de ficheros, abrir ventanas, etc.

Hit: Unidad de medición de accesos a determinado recurso. Forma de registrar cada pedido de información que un usuario efectúa a un server. Por ejemplo, en el caso de un sitio Web, la solicitud de cada imagen, página y frame genera un hit. Por lo tanto, para conocer en realidad cuántos accesos hubo, debe dividirse la cantidad de hits por la cantidad de objetos independientes (texto, frames e imágenes) que una página contiene, o usar un contador de accesos.

Page View: Una vista de la página (PV) o impresión de la página es una solicitud para cargar un solo HTML archivo ('page') de un sitio de Internet .

Proxy Server: Utilizado en relación a Internet, hace referencia a un servidor que media entre el usuario (su computadora) y otro servidor de la Red. El Proxy Server puede hacer, por ejemplo, un pedido de información para un cliente en lugar de que el cliente lo haga directamente.

Request: Solicitud de información o datos que una computadora cliente efectúa a un servidor.

Sitio Web: Se lo utiliza para definir un conjunto coherente y unificado de páginas y objetos intercomunicados, almacena[illegible]vidor.

Spam: Mensaje electrónico no solicitado enviado a muchas personas.

Spiders: Complejos programas autónomos que recorren la Web siguiendo enlace tras enlace en cada página; almacena estas últimas para que más tarde sean catalogadas en las enormes bases de datos de los índices de búsqueda.

URI: Un Uniform Resource Identifier o URI (en español «identificador uniforme de recurso») es una cadena de caracteres corta que identifica inequívocamente un recurso (servicio, página, documento, dirección de correo electrónico, enciclopedia, etc.). Normalmente estos recursos son accesibles en una red o sistema. Los URI pueden ser localizadores uniformes de recursos (URL), Uniform Resource Name (URN), o ambos.

URL: Un localizador uniforme de recursos, más comúnmente denominado URL (sigla en inglés de Uniform Resource Locator), es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación, como por ejemplo documentos textuales, imágenes, videos, presentaciones digitales, etc.

URN: Es un acrónimo inglés de Uniform Resource Name, en español "Nombre de recurso uniforme". Un URN funciona de manera similar a un URL (Localizadores Uniformes de Recursos).

User agent: Un agente de usuario es una aplicación informática que funciona como cliente en un protocolo de red; el nombre se aplica generalmente para referirse a aquellas aplicaciones que acceden a la World Wide Web

Usuario único*: U*n usuario único se corresponde con cada persona física que accede al servidor en el período de tiempo estudiado. Cuando no existe autenticación en el servidor
(habitualmente realizada mediante un nombre y una contraseña), la identificación de usuarios reales se suele basar en la dirección IP, aunque dicha correspondencia no es totalmente fiable.

Visita: Una visita (o sesión de usuario) está formada por el conjunto de páginas accedidas por un usuario durante una misma sesión de trabajo; generalmente se considera que la sesión de trabajo se mantiene mientras el tiempo entre la vista de dos páginas consecutivas no supere un determinado umbral o *timeout.*

.

INTRODUCCIÓN

La internet es sin duda el mecanismo más importante, práctico y altamente difundido para el intercambio de información de todo tipo convirtiéndose en un recurso de disposición pública y general al cual acceden las personas en busca de satisfacer sus necesidades de información, obtener algún recurso en particular o realizar algún tipo de transacción, sin mencionar la gran cantidad de negocios que se manejan por internet, la competencia y la creciente necesidad de mejorar los servicios para poder sobrevivir en un ambiente competitivo.

En los últimos años se ha planteado el desarrollo de nuevas tecnologías que buscan el ofrecer un mejor servicio de la internet, haciendo uso de tecnologías como lo es la minería de datos orientada a la web o Web Mining, cuyo objetivo es aprovechar la información que circula en Internet para definir los diferentes patrones de los usuarios que navegan en la red [40].

Sin duda alguna, el conocimiento obtenido después de aplicar técnicas de Minería Web sobre los datos contenidos en el World Wide Web es sumamente valioso, pero para obtener información que pueda dar una visión más completa sobre el acceso de usuarios a un determinando sitio, es necesario también realizar un análisis estadístico del tráfico Web.

El tráfico web puede ser analizado con ver las estadísticas encontradas en el archivo del servidor de la página, el cual genera automáticamente una lista de todas las páginas vistas. La cantidad de tráfico en un sitio web sirve para medir su popularidad. Analizando las estadísticas de visitantes es posible saber qué aspectos de diseño, contenido y estructura de un sitio web se han desarrollado correctamente y qué aspectos se debe mejorar. También es posible aumentar la popularidad del sitio y la cantidad de gente que lo visita.

Una herramienta para el análisis estadístico del tráfico Web permite la medición de varios criterios: número de visitantes, promedio de páginas vistas por un usuario (un promedio alto indica que los usuarios exploran constantemente la página), promedio de tiempo de un usuario en el sitio, promedio de duración de la página, clases dominantes (niveles de direcciones IP requeridas para abrir páginas), hora pico (el mayor tiempo de popularidad de la página puede mostrarse cuando se hacen campañas promocionales), páginas más requeridas (más populares) [49].

En este documento se presenta el resultado del trabajo de grado para optar el título de Ingeniero de Sistemas que tiene como propósito desarrollar el módulo de análisis estadístico de tráfico web y acoplarlo a la herramienta Polaris.

ALCANCE Y DELIMITACIÓN

El desarrollo de la herramienta POLARIS inició a partir de la segunda versión de POLARIS, en la cual se implementa el módulo de análisis estadístico de tráfico web y cuyo resultado es POLARIS VERSIÓN 3.0.

Las pruebas de rendimiento de la herramienta se realizaron utilizando conjuntos de datos reales del portal de la Universidad de Nariño y otros archivos públicos obtenidos de sitios web correspondientes a herramientas de software de análisis de tráfico y minería web entre los cuales se encuentran Analog [3], Awstats [4] y Web Log Expert [45].

PLANTEAMIENTO DEL PROBLEMA

El Grupo de Investigación GRIAS del programa de Ingeniería de Sistemas de la Universidad de Nariño, con la dirección del Doctor RICARDO TIMARAN PEREIRA Ph.D, y la participación de las ingenieras DIANA PATRICIA ANGULO URBANO, JENNY JOHANA DAZA BURBANO y ALEJANDRA ZULETA MEDINA, en la línea KDD, desarrollaron la Herramienta POLARIS Versión 1.0 [40], con el módulo de minería de uso de la web. Posteriormente, con la participación de los ingenieros JHON CRISTIAN ILES LÓPEZ y CARLOS ANDRÉS RUBIO CORAL se desarrolló la segunda versión de Polaris, donde se implemento el módulo de minería de estructura de la web.

POLARIS en la versión actual (Versión 2.0), se ve limitada al momento de obtener información que se encuentre relacionada con el contenido de la web o con el análisis estadístico del tráfico web, información que resulta muy relevante a la hora de tomar decisiones que permitan optimizar procesos en sitios web.

Continuando con el proyecto del Grupo de Investigación GRIAS, se propuso en este proyecto complementar la herramienta POLARIS, con el Módulo de Análisis Estadístico de Tráfico Web, ampliando así la cobertura y el campo de acción de ésta herramienta, obteniendo la versión 3 de POLARIS.

OBJETIVOS

Objetivo General.

Potenciar la herramienta POLARIS a través del desarrollo del módulo de análisis estadístico de tráfico web, con el fin de obtener información que permita complementar y reforzar la información obtenida mediante técnicas de minería web de uso y estructura.

Objetivo específicos:

Analizar las versiones anteriores de la herramienta Polaris y las diferentes herramientas de análisis estadístico de tráfico web existentes.

Desarrollar procedimientos y técnicas que permitan realizar un análisis estadístico completo de tráfico web.

Desarrollar procesos que permitan la visualización gráfica de los resultados obtenidos en con respecto al análisis estadístico de tráfico web e integrarlos en la herramienta de software POLARIS 3.0.

Realizar las pruebas de funcionalidad del módulo de análisis estadístico de tráfico web de la herramienta Polaris 3.0, utilizando conjunto de datos reales.

JUSTIFICACIÓN

Se puede definir Web Mining como el proceso de extracción de patrones potencialmente útiles e interesantes y de información implícita (Data Mining) de datos provenientes de la Web o, de manera más sencilla, como “la aplicación de técnicas de Data Mining a grandes depósitos de datos Web” [6].

El análisis estadístico de tráfico web permite medir y cuantificar diferentes criterios relacionados con los de datos enviados y recibidos por los visitantes de un sitio web, mediante el análisis de los registros almacenados en los archivos log generados por el servidor web.

En la actualidad no existen herramientas que permitan realizar de manera integrada tareas de minería web y análisis estadístico de la web, impidiendo así la posibilidad de relacionar información relevante y complementaria.

Con base en lo anterior, nace la necesidad de actualizar POLARIS VERSIÓN 2.0 en POLARIS VERSIÓN 3.0, una herramienta de minería web de uso y estructura y análisis estadístico de tráfico web, con el grupo GRIAS del programa de Ingeniería de Sistemas de la Universidad de Nariño y bajo la dirección de RICARDO TIMARAN PEREIRA Ph. D.

El desarrollo de POLARIS VERSIÓN 3.0 como una herramienta software bajo licencia pública GPL para el análisis del tráfico web, permitirá a todo tipo de empresas, entidades y personas del común, utilicen tecnologías Web Mining como un mecanismo de apoyo en la toma acertada de decisiones.

POLARIS VERSIÓN 3.0, se convierte en un aporte significativo en cuanto a investigación científica de tecnología Web Mining, un reconocimiento mas para el área de descubrimiento de conocimiento en base de datos de la Universidad de Nariño que a través del programa de Ingeniería de Sistemas contribuye al desarrollo de la región y del país.

ORGANIZACIÓN DEL DOCUMENTO

El resto del documento está organizado en capítulos. El capitulo 1 conforma el marco teórico que sirve de base de conocimiento para el desarrollo de este proyecto de investigación y en el cual se describen los conceptos básicos sobre la web y sobre minería y tráfico web. En el capítulo 2 se presenta un estudio sobre las herramientas de tráfico web como antecedentes de este proyecto. En el capítulo 3 se presenta la conceptualización de la metodología de desarrollo así como las herramientas de software utilizadas para la construcción del módulo de tráfico web de POLARIS VERSIÓN 3.0. En el capítulo 4 se describe el análisis y diseño del módulo de tráfico web de la nueva versión de POLARIS. En el capítulo 5 se muestra la arquitectura de POLARIS VERSIÓN 3.0 y del nuevo módulo de tráfico web y se detallan los aspectos de su implementación. En el capítulo 6 se presentan los resultados de las pruebas de funcionalidad del módulo implementado y finalmente en el ultimo capitulo se dan las conclusiones y recomendaciones de este proyecto de investigación.

1. MARCO TEÓRICO

1.1 LA WORLD WIDE WEB

1.1.1 Definición y características. En informática, la World Wide Web (WWW) o Red informática mundial es un sistema de distribución de información basado en hipertexto o hipermedios enlazados y accesibles a través de Internet. Con un navegador web, un usuario visualiza sitios web compuestos de páginas web que pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces [46].

La World Wide Web es un sistema de información integrado por agentes. Los agentes son programas que actúan en nombre de otra persona, entidad, o proceso de intercambio y procesamiento de información. Básicamente, en la tecnología del World Wide Web, hay dos tipos de agentes: los agentes de servidores y agentes de usuario. Un agente de servidor es un programa que ofrece servicios a los agentes de usuario, y un agente de usuario es un programa que utiliza los servicios ofrecidos por los agentes de servidor. La World Wide Web se considera una tecnología de la red que ha heredado muchos de los principios de diseño, de la interoperabilidad, la evolución y la descentralización [30].

La Web tiene muchos usos y aplicaciones. Para el presente estudio de investigación se hará referencia a la Web como base documental, es decir, la Web es un conjunto de información con características particulares cuyo fin es ser consultada por diversos usuarios. Dicha información posee determinadas características que hacen compleja la labor de recuperación [14]. Según Fuentes y Pavón, dichas características son las siguientes:

- Está distribuida en diversos servidores.
- Es dinámica, cambia constantemente tanto de forma como de contenido.
- No tiene estructura común en todos los documentos. Los documentos carecen de modelo conceptual que estructure semánticamente el contenido.
- Es heterogénea, es decir, está compuesta por partes de diversa naturaleza, ya que soporta grandes cambios de formato y lenguaje.
- Es altamente volátil, aparece información nueva y desaparece la información desactualizada a gran velocidad.

- Es redundante. En numerosas ocasiones se encuentra repetida la información en distintas páginas.

- La información no es únicamente textual.

1.1.2 Arquitectura de la World Wide Web. El diseño del World Wide Web sigue el modelo cliente-servidor: un paradigma de división del trabajo informático en el que las tareas se reparten entre un número de clientes que efectúan peticiones de servicios de acuerdo con un protocolo, y un número de servidores que las atienden. En el Web, las estaciones de trabajo son clientes que demandan hipertextos a los servidores. Según G. Malkin, para poner en marcha un sistema como éste ha sido necesario [24]:

- Diseñar e implementar un nuevo protocolo que permitiera realizar saltos hipertextuales, esto es, de un nodo de origen a un nodo de destino, que podría ser un texto o parte de un texto, una imagen, un sonido, una animación, fragmento de vídeo, etc. Es decir, cualquier tipo de información en formato electrónico. Este protocolo se denomina HTTP (HyperText Transfer Protocol) y es el lenguaje que hablan los servidores del WWW.

- Inventar un lenguaje para representar hipertextos que incluyera información sobre la estructura y el formato de representación y, especialmente, indicar origen y destino de saltos hipertextuales. Este lenguaje es el HTML o (HyperTextex markup Language).

- Idear una forma de codificar las instrucciones para los saltos hipertextuales de un objeto a otro de la Internet. Dada la variedad de protocolos, y por tanto, formas de almacenamiento y recuperación de la información, en uso en la Internet, esto es vital para que los clientes puedan acceder a dicha información.

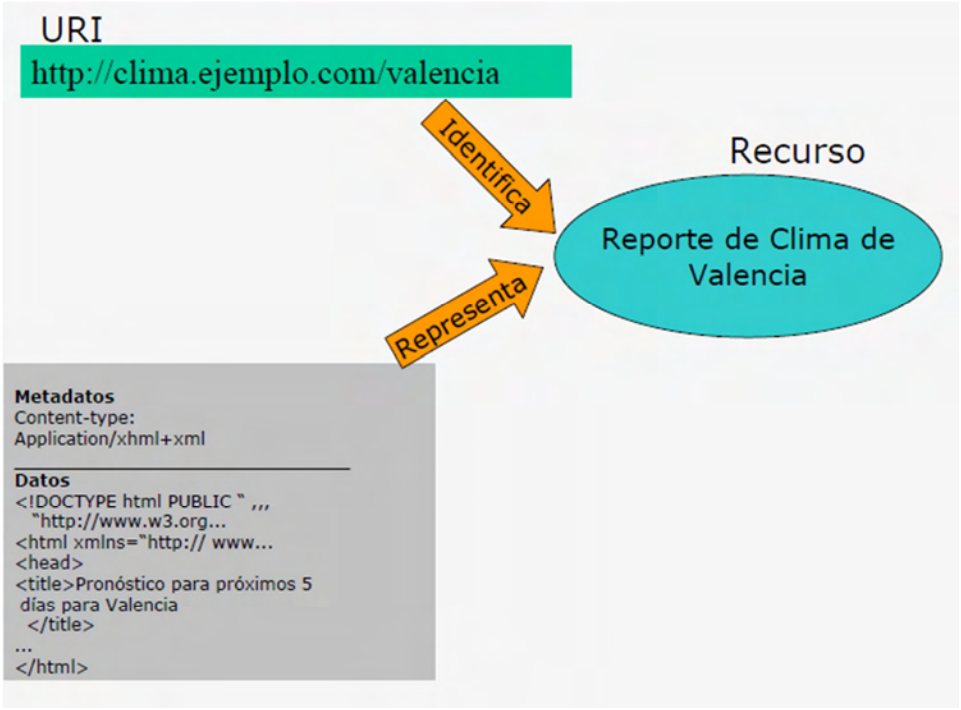
- Desarrollar aplicaciones cliente para todo tipo de plataforma y resolver el problema de cómo acceder a información que está almacenada y es accesible a través de protocolos diversos (FTP, NNTP, Gopher, HTTP, X.500, WAIS, etc.) y representar información multiformato (texto, gráficos, sonidos, fragmentos de vídeo, etc.). A este fin se han desarrollado diversos clientes, entre los que destaca la familia Mosaic, del NCSA (National Center for Supercomputer Applications) de la Universidad de Chicago, y su sucesor Netscape Navigator, de Netscape Communications Corporation.

Recientemente, el W3C ha establecido una Recomendación para fijar la arquitectura de la World Wide Web. Se trata de la recomendación "Architecture of the World Wide Web, Volume One" [35] y en ella se fijan los 3 aspectos básicos en

los que se concreta la arquitectura de la World Wide Web actual que, ha dado un fuerte impulso al desarrollo no sólo de aspectos técnicos como los protocolos y lenguajes, sino también otros aspectos relacionados con el contenido y la semántica de la información, como el uso de metadatos para describir dicha información y su uso por parte de los agentes inteligentes con el fin de poder recuperarla de forma automática.

Según el World Wide Web Consortium (*W3C*) la arquitectura Web consiste de tres conceptos fundamentales [35]: identificación (URIs), interacción (protocolos como HTTP y SOAP) y representación (formatos como HTML, SVG y PNG). Estas tres ramas se encuentran representadas por la experiencia familiar del usuario que utiliza un navegador para pulsar sobre un enlace que identifica un sitio Web, llevándole a interactuar con el sitio Web (al que se refiere genéricamente como "recurso Web"), y a mostrar la información en el navegador. (Ver figura 1)

Figura 1. Arquitectura de la web



1.1.2.1 Identificación. Se generaliza y normaliza el uso de URIs para identificar y localizar los recursos de la Web.

Con el fin de comunicarse internamente, una comunidad está de acuerdo (en una medida razonable) en un conjunto de términos y sus significados. Uno de los objetivos de la Web, desde su creación, ha sido la de construir una comunidad global en la que cualquiera de las partes puede compartir información con cualquier otra parte. Para lograr este objetivo, la Web hace uso de un sistema de identificación único y global: el URI. Los URI son una piedra angular de la arquitectura Web, que facilita la identificación que es habitual en la Web. El alcance global de la URI promueve a gran escala "efectos de red": el valor de un identificador aumenta cuanto más se usan de manera habitual. [35]

1.1.2.1.1 Uniform resource locator (URL). Es el sistema más común de localización de documentos dentro de la web. Lo que describe este recurso no es el nombre del documento, sino la forma de acceder a él. Se describe en la RFC1737 [20] y su sintaxis del URL consta de varios bloques separados por barras inclinadas que indican en primer lugar el protocolo mediante el que se localizará el documento, seguidas del servidor en el que está alojado, del directorio en que se encuentra, y en su caso, el subdirectorio o subdirectorios, y finaliza con el nombre del archivo y extensión del documento que se pretende identificar y localizar [22].

Ejemplo:

http://www.w3.org/Addressing/URL/Overview.html
Protocolo/ Nombre de dominio internacional/ Directorio/ Subdirectorio/Documento.extensión

1.1.2.1.2 Uniform resource name (URN). El URN fue una iniciativa de la Internet Engineering Task Force IETF [34], la rama de desarrollo de ingeniería y protocolos de Internet, con la premisa de conseguir una forma universal de identificación de recursos, para que cada recurso fuera único y constante. Se trataba de un identificador paralelo al URL. Una característica importante de este sistema es que trabaja junto con Uniform Resource Characteristics/Citacion (URC), un sistema para la descripción de metadatos. La sintaxis del URN, explicada en la RFC2141 [20] consta de 3 bloques separados por dos puntos: el identificador URN, el NID o nombre de la categoría en la que se incluye el documento (por ejemplo, inet para documentos de Internet) y el NSS o cadena específica que indica primero la ruta y a continuación el documento concreto [22].

Ejemplo:

urn:inet:dtsc.edu.au:tr0088

Lo interesante de URN es que puede subsumir todos los identificadores

bibliográficos existentes tales como ISSN (International Standard Serial Number) para publicaciones seriadas, ISBN (International Standard Books Number) para libros y SICI (Serial Item and Contribution Identifier) que identifica no sólo la revista, sino también el número de un publicación seriada.

El primitivo URN, junto con el identificador URL, ha sido uno de los pilares para la creación del URI, que se está convirtiendo en el identificador global más utilizado, ya que engloba a ambos [22].

1.1.2.1.3 Uniform resource identifier (URI). URI también ha sido desarrollado por el IETF (The Internet Engineering Task Force) [34] y pretende crear un sistema mundial para identificar recursos de todo tipo en la web: documentos, imágenes, programas, servicios, correos electrónicos, etc. Este método combina URNs y URLs, esto es, nombres/direcciones. Se trata de identificar los documentos mediante una secuencia de sintaxis controlada que identifica cada documento de una forma única. Los URIs hacen posible encontrar los recursos bajo una gran variedad de esquemas definidos y métodos de acceso tales como HTTP, FTP, Gopher, news, telnet o correos electrónicos localizables siempre de la misma manera, ya que a un mismo documento se puede acceder desde distintos protocolos. Ya se han establecido una serie de schemes o esquemas direccionados. Los esquemas definidos URI coinciden con los protocolos más usados de Internet [22]. Estos son unos ejemplos de esquemas URI:

Ejemplos:

ftp://ftp.is.co.za/rfc/rfc1808.txt
(esquema ftp para servicios de File Transfer Protocol)

gopher://spinaltap.micro.umn.edu/00/Weather/California/Los%20Angeles
(esquema gopher para Gopher y servicios Gopher+ Protocol)

http://www.math.uio.no/faq/compression-faq/part1.html
(esquema http para servicios de Hypertext Transfer Protocol)

mailto:mduerst@ifi.unizh.ch
(esquema de mailto para direcciones de correo electrónico)

news:comp.infosystems.www.servers.unix
(esquema news para grupos de noticias y artículos de USENET)

telnet://melvyl.ucop.edu/
(esquema telnet para servicios interactivos vía Protocolo TELNET)

1.1.2.2 Interacción. Los agentes web se comunican usando protocolos estandarizados que hacen posible la interacción mediante el intercambio de mensajes que se adhieren a una sintaxis y semántica definidas. Mediante la introducción de un URI dentro de un diálogo de recuperación o seleccionando un enlace de hipertexto, un usuario le dice a su navegador que realice una acción de recuperación del recurso identificado por el URI. El navegador envía una petición de HTTP GET (parte del protocolo HTTP) al servidor, vía TCP/IP puerto 80, y el servidor devuelve un mensaje que contiene lo que éste determina que es una representación del recurso, como por ejemplo, la fecha en que la representación se generó. Este ejemplo es específico para un navegador de información hipertextual -pero son posibles otras clases de interacción-, ambas desde los navegadores y por medio del uso de otros tipos de agentes Web [35].

1.1.2.2.1 HyperText transfer protocol (HTTP). El HTTP (HyperText Transfer Protocol) es el protocolo de alto nivel del World Wide Web que rige el intercambio de mensajes entre clientes y servidores del Web. Un protocolo es:

"Una descripción formal de los formatos de los mensajes y reglas que deben seguir dos ordenadores para intercambiar dichos mensajes. Los protocolos pueden describir detalles de bajo nivel de los interfaces de máquina a máquina (por ejemplo, el orden en el cual deben enviarse bits y bytes a través de un cable) o intercambios de alto nivel entre programas (por ejemplo, la forma en que dos programas transfieren un fichero a través de la Internet)." [24]

El HTTP es un protocolo genérico orientado a objetos que no mantiene la conexión entre transacciones. Ha sido especialmente diseñado para atender las exigencias de un sistema hipermedia distribuido como es el World Wide Web [9]. Según Berners-Lee, sus características principales son:

- Ligereza: Reduce la comunicación entre clientes y servidores a intercambios discretos, de modo que no sobrecarga la red y permite saltos hipertextuales rápidos.

- Generalidad: Puede utilizarse para transferir cualquier tipo de datos. Esto incluye también los que desarrollen en el futuro, ya que el cliente y el servidor pueden negociar en cualquier momento el modo de representación de los datos: el cliente notifica al servidor una lista de formatos que entiende, y en adelante el servidor sólo remitirá al cliente datos que este sea capaz de manejar. El cliente debe aceptar al menos dos formatos: text/plain (texto normal) y text/html (hipertexto codificado en HTML: el lenguaje en el que se escriben los hipertextos de la Web).

- Extensibilidad: Contempla distintos tipos de transacción entre clientes y

servidores ("métodos", en la jerga HTTP), y la futura implementación de otros nuevos. Esto abre posibilidades más allá de la simple recuperación de objetos de la red: búsquedas, anotaciones, etc.

El esquema básico de cualquier transacción HTTP entre un cliente y un servidor es el siguiente [9]:

- Conexión: El cliente establece una conexión con el servidor a través del puerto 80 (puerto estándar), u otro especificado.

- Petición: El cliente envía una petición al servidor.

- Respuesta: El servidor envía al cliente la respuesta (esto es, el objeto demandado o un código de error).

- Cierre: Ambas partes cierran la conexión.

- La eficiencia del HTTP posibilita la transmisión de objetos multimedia y la realización de saltos hipertextuales con una rapidez razonable.

1.1.2.3 Representación. Muchos de los protocolos usados para recuperar la representación y/o introducirla, hacen uso de una secuencia de uno o más mensajes, que tomados juntos contienen un conjunto de datos de representación y metadatos, para transferir la representación entre agentes. La elección de los lugares del protocolo de interacción pone límites a los formatos de representación de datos y metadatos que pueden ser transmitidos. HTTP, por ejemplo, transmite típicamente un octeto simple de transmisión de metadatos y usa los campos de la cabecera Content-Type y Content-Encoding para identificar además el formato de la representación. Por ejemplo, la representación transferida puede ser en XHTML, identificado por el campo de la cabecera HTTP Content-type que contiene el nombre del tipo de media registrado en Internet, application/xhtml+xml. Este nombre del tipo de media en Internet indica que los datos representados pueden ser procesados de acuerdo con la especificación XHTML [35].

1.1.2.3.1 HyperText Markup Language (HTML). El HTML (HyperText Markup Language) es el lenguaje en el que se escriben los hipertextos del World-Wide Web. Cumple la norma SGML [39], y permite añadir a un documento de texto:

- La especificación de estructuras del texto. Por ejemplo, títulos, encabezamientos, límites de los párrafos, listas de elementos.

- Estilos: texto enfatizado, citas, etc.

- Objetos multimedia: imágenes o sonido, pongamos por caso.

- Conexiones hipertextuales a otros objetos de la red: partes sensibles del documento desde dónde podríamos saltar otras partes del Web.

- Todo este "valor añadido" al texto se codifica como etiquetas ("tags", en la jerga) que se insertan en el propio texto. Un ejemplo nos permitirá hacernos una idea de todo ello. (Ver figura 2)

Figura 2. Documento HTML

```
<HTML>

<HEAD>
<TITLE>IT stories contents
</TITLE>
<BASE HREF="http://www.uji.es/CPI/">
</HEAD>

<BODY>

<H1><IMAGE SRC="et.gif" ALIGN=middle>
    <I>Contes per a extraterrestres</I></H1>

<P> Revista electr&ograve;nica / E-zine

<MENU>
<LI><A HREF="signatures/extraterrestres.html">
    Qu&egrave; s&oacute;n els <I>Contes per a Extraterrestres</I> /
    About the <I>IT Stories</I></A>
</MENU>

<H2>Novetats / What's new</H2>

<MENU>
<LI><A HREF="dossiers/index.html">Dossiers / Dossiers</A>
<LI><A HREF="contes/index.html">Contes / Short Stories</A>
<LI><A HREF="microcontes/index.html">Microcontes / Short short stories</A>
<LI><A HREF="news/index.html">Not&iacute;cies del m&oacute;n / World-Wide News</A>
<LI><A HREF="web/index.html">Ficcions en la xarxa / Fiction resources over the Web</A>
</MENU>

<HR>
<ADDRESS>
<A HREF="signatures/extraterrestres.html">
 Contes per a Extraterrestres</A>
<A HREF="mailto:extraterrestres@guest.uji.es">
&lt;extraterrestres@guest.uji.es&gt;</A>
</ADDRESS>

</BODY>

</HTML>
```

Como se observa en la Figura 2 las etiquetas del HTML se delimitan por medio de los signos < y >. Por ejemplo, la etiqueta <P> marca el inicio de cada párrafo. Otras, la mayor parte, van por parejas: <TITLE> y </TITLE> abren y cierran, respectivamente, el título del documento.

Los links se abren y cierran con las etiquetas <A> y </A>. El objeto de la red a donde nos lleva el link se codifica en la etiqueta de apertura por medio de una notación que se ha convertido de hecho en un estándar de Internet: los llamados URL.

1.1.3 Aspectos tecnológicos de la web. La World Wide Web, WWW o Web se ha convertido en el principal servicio de Internet. La WWW utiliza la estructura de comunicación existente en Internet y comparte protocolos de comunicaciones comunes, estándares y otras notaciones de protocolos de comunicación que permiten el acceso universal a los servicios de información presentes en la Web a través del modelo cliente-servidor, esto es, mediante la conexión remota entre una red de ordenadores o máquinas llamadas servidores, y los ordenadores clientes [22].

La WWW intercambia la información vía Internet y reparte entre los ordenadores clientes y servidores las operaciones de conexión. Por su parte, el navegador web o browser realiza la presentación de las páginas web en la máquina u ordenador cliente, una vez que éste ha consultado la información contenida en los servidores [22].

Una de las características principales de la Web es la independencia en la visualización y presentación de la información, lo que permite que los sistemas de hipertexto sean construidos independientemente del desarrollo de nuevos avances en la representación de los datos. Para la visualización sólo se precisa de un navegador web [22].

Todas estas características técnicas trabajan de manera ensamblada en la World Wide Web.

Según Lamarca Lapuente cabe resaltar tres aspectos fundamentales en la tecnología de la Web [22]:

- El modelo cliente-servidor.
- Los protocolos web.
- Los navegadores web.

1.1.3.1 El modelo cliente-servidor. El modelo cliente-servidor se basa en los siguientes elementos [22]:

- Cliente: en una red cliente/servidor, se trata de un nodo de la red que emplea

los recursos que proporciona un servidor.

- Servidor: nodo de red que proporciona servicios a PCs cliente; por ejemplo, acceso a archivos, formación de trabajos de impresión o ejecución remota.

- Nodo: cada uno de los ordenadores individuales u otros dispositivos conectados a la red.

- Paquete: grupo de bits de datos de información asociada, incluidos la dirección de origen y de destino, formateadas para transmitirse de un nodo a otro.

- Ruteador: dispositivo que conecta dos redes y que opera como un puente, pero que también puede elegir rutas a través de una red.

La World Wide Web utiliza las comunicaciones establecidas en Internet entre "clientes" y "servidores" para el acceso y el intercambio de información y recursos. La máquina u ordenador cliente se conecta a la máquina u ordenador servidor WWW para realizar una consulta y el servidor le devuelve una respuesta. Si, por ejemplo, pinchamos sobre un enlace en el navegador, el programa de acceso a la WWW utiliza la dirección correspondiente al enlace y se conecta al servidor de nombres de dominio (DNS) que le permite enrutar la respuesta hacia el servidor WWW correspondiente. El servidor recoge la demanda y devuelve los archivos de texto, imágenes, etc. al cliente. El protocolo HTTP (HyperText Transfer Protocol) es el que hace posible esta relación. El cliente recoge el documento y éste se visualiza a través del navegador. Así, la carga de trabajo se reparte entre el ordenador cliente (el demandante de información) y el servidor (quien ofrece la información).

1.1.3.2 Los protocolos web. Tres protocolos, gobiernan el funcionamiento de la Web. Son los estándares que permiten generalizar los mecanismos de intercambio y presentación de archivos y documentos y que proveen los mecanismos de direccionamiento universal. Podemos definir un protocolo como cualquier conjunto definido de procedimientos, convenciones o métodos que permiten inter-operar a dos dispositivos [22].

Estos 3 protocolos son:

- La URL o Universal Resource Locator: Se trata de una definición única o dirección permanente de localización de un documento.

- HTML o HyperText Markup Languaje: Es un lenguaje o sintaxis específica para la WWW que describe la estructura de los documentos a través de marcas y etiquetas, y que posibilita los enlaces a otras páginas o informaciones.

estructurada.

- HTTP o HyperText Transfer Protocol: Es un protocolo que permite el intercambio de información en la world wide web, el método mediante el cual se transfieren las páginas web a un ordenador.

1.1.3.3 Los navegadores web. Un navegador web o web browser es una aplicación que opera a través de Internet, permitiendo interpretar y leer la información de documentos y archivos alojados en la Web.
El navegador interpreta el código, HTML generalmente, en el que está escrita la página web y lo presenta en pantalla permitiendo al usuario interactuar con su contenido y navegar hacia otros lugares de la red mediante enlaces o hipervínculos [22].

La funcionalidad básica de un navegador web es permitir la visualización de documentos de texto, posiblemente con recursos multimedia incrustados. Los documentos pueden estar ubicados en la computadora en donde está el usuario, pero también pueden estar en cualquier otro dispositivo que esté conectado a la computadora del usuario o a través de Internet, y que tenga los recursos necesarios para la transmisión de los documentos.

Tales documentos, comúnmente denominados páginas web, poseen hipervínculos que enlazan una porción de texto o una imagen a otro documento, normalmente relacionado con el texto o la imagen.

1.1.4 Servidores web. Un servidor web o servidor HTTP es un programa informático que sirve datos en forma de Páginas Web, hipertextos o páginas HTML con textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. La comunicación de estos datos entre cliente y servidor se hace por medio un protocolo, concretamente del protocolo HTTP [22].

Con esto, un servidor Web se mantiene a la espera de peticiones HTTP, que son ejecutadas por un cliente HTTP; lo que solemos conocer como un Navegador Web. El servidor responde al cliente enviando el código HTML de la página; el navegador cuando recibe el código, lo interpreta y lo muestra en pantalla. El Cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página. El servidor se encarga de transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

1.1.4.1 Funcionamiento. El Servidor web se ejecuta en un ordenador manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y que responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error. A modo de ejemplo, al teclear www.wikipedia.org en el navegador, éste realiza una petición HTTP al servidor de dicha dirección. El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo exhibe en pantalla. Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma [44].

Además de la transferencia de código HTML, los Servidores web pueden entregar aplicaciones web. Éstas son porciones de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

- Aplicaciones en el lado del cliente, donde el cliente web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java "applets" o Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts). Comúnmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje javascript y java, aunque pueden añadirse más lenguajes mediante el uso de plugins.

- Aplicaciones en el lado del servidor, donde el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

Las aplicaciones de servidor muchas veces suelen ser la mejor opción para realizar aplicaciones web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste no necesita ninguna capacidad añadida, como sí ocurre en el caso de querer ejecutar aplicaciones javascript o java. Así pues, cualquier cliente dotado de un navegador web básico puede utilizar este tipo de aplicaciones.

1.1.4.2 Peticiones web. Durante una sesión normal de trabajo en la World Wide Web un cliente (navegador) solicita un documento de un servidor Web y una vez obtenido lo muestra al usuario que hizo la solicitud. Si este documento contiene un enlace a otro documento (en el mismo o en distinto servidor), y el usuario activa el enlace el cliente Web efectuará la petición y mostrará el nuevo documento. [44]

El navegador por medio de la interfaz de usuario permite al usuario realizar una o varias peticiones web. La interfaz de usuario o entorno de usuario es el conjunto de elementos del navegador que permiten realizar la petición de forma activa. Una petición Web no sólo puede ser realizada mediante un navegador sino con cualquier herramienta habilitada para tal fin, como una consola de comandos Telnet [44].

Ejemplo:

Petición típica del navegador:
GET /about-mit.html HTTP/1.1
Host: web.mit.edu
Accept: text/html, text/plain, image/jpeg, image/gif, */*

Respuesta típica del servidor:
HTTP/1.1 200 OK
Server: Apache/1.3.3 Ben-SSL/1.28 (Unix)
Content-Type: text/html
Content-Length: 8300

1.1.4.2.1 Peticiones en el protocolo HTTP: GET, HEAD y POST. La primera línea de una petición contiene los comandos HTTP, conocidos como métodos. Existen varios, pero los más conocidos y utilizados son tres: GET, HEAD y POST. [11]

El método GET se utiliza para recuperar información identificada por un URI por parte de los navegadores. Si el URI se refiere a un proceso generador de datos como un programa CGI, se devuelven los datos generados por el programa. El método GET también se puede utilizar para pasar una pequeña cantidad de información al servidor en forma de pares atributo-valor añadidos al final del URI detrás de un símbolo de interrogación (?) [11].

Ejemplo:

GET /cgi/saludar.pl?nombre=pepe&email=pepe@infor.uva.es HTTP/1.0

La longitud de la petición GET está limitada por el espacio libre en los buffers de entrada. Por lo que para mandar una gran cantidad de información al servidor ha de utilizarse el método POST.

El método HEAD es idéntico al GET excepto que el servidor no devolverá el cuerpo del mensaje en la respuesta a un método HEAD. Esto es útil para obtener información sobre las entidades implicadas en la petición sin que tengan que transferirse. Sirve para comprobar si los enlaces son válidos o para saber cuándo

fue la última modificación de la entidad solicitada.

El método POST se refiere normalmente a la invocación de procesos que generan datos que serán devueltos como respuesta a la petición. Además se utiliza para aportar datos de entrada a esos programas. En este caso los pares atributo-valor son incluidos en el cuerpo de la petición separados por ampersand.

Ejemplo:

POST /cgi/saludar.pl HTTP/1.0

Una petición HTTP está formada por:

Línea de la petición: contiene el recurso que se solicita. La línea de la petición está formada por estos elementos:

Método: nombre del método HTTP utilizado (GET, POST, etc.).
Identificador del recurso: URL ("Uniform Resource Locator").
Versión del protocolo utilizado.

Cabecera de la petición: contiene la información adicional sobre el cliente que hace la solicitud. Los identificadores más importantes son:

Host: nombre del servidor.
User-Agent: nombre del navegador o del programa usado para acceder al recurso solicitado.
Accept: se indican los formatos de texto e imagen aceptados por el User-Agent.
Accept-Language: idiomas que soporta (preferentemente) el cliente.

Cuerpo de la petición: en peticiones de tipo POST y otras contiene más información adicional.

1.1.4.2.2 Parámetros de la petición. Una petición HTTP puede contener parámetros, por ejemplo, como respuesta a un formulario de registro o a una selección de entre los productos en una tienda virtual. Tales parámetros pueden pasarse de 2 formas [11]:

Formando parte de la propia cadena de la petición, codificados como parte de la misma URL.

Como datos añadidos a la petición.

Para codificar los parámetros como parte incluida en la URL, éstos deben añadirse a la URL detrás del nombre del recurso, separándolos de éste mediante el

caracter "?". Los parámetros se separan entre sí mediante el caracter "&". Los espacios se sustituyen por el caracter "+". Los caracteres especiales (los mencionados antes de "&", "+" y "?", y los caracteres que no son imprimibles, etc.) se representan mediante "%xx", donde "xx" representa el código en codificación ASCII en hexadecimal del caracter en cuestión [11].

Por ejemplo:

http://www.ejemplo.com/indice.jsp?nombre=Fulano+Mengano&OK=1

En la petición HTTP quedaría:

GET /indice.jsp?nombre=Fulano+Mengano&OK=1 HTTP/1.0
Host: www.unejemplo.com
User-Agent: Mozilla/4.5 [en]
Accept: image/jpeg, image/gif, text/html
Accept-language: en
Accept-Charset: iso-8859-1

Para pasar los parámetros como datos añadidos, se envían al servidor en el cuerpo del mensaje de la petición. Por ejemplo, siguiendo con la petición:

POST /indice.jsp HTTP/1.0
Host: www.unejemplo.com
http://www.unejemplo.com/indice.jsp?nombre=Fulano+Mengano&OK=1
User-Agent: Mozilla/4.5 [en]
Accept: image/jpeg, image/gif, text/html
Accept-language: en
Accept-Charset: iso-8859-1

nombre=Perico+Palotes&OK=1

Se debe destacar que para pasar los parámetros en el cuerpo de la petición, ésta se debe realizar como POST (no como GET), aunque una petición POST puede llevar parámetros en la línea de petición (igual que una GET). Los parámetros pasados en el cuerpo de la petición están codificados igual que si los pasamos mediante la URL, o pueden usar una codificación específica derivada del formato MIME ("Multipurpose Internet Mail Extensions"), conocida como codificación multiparte.

El ejemplo anterior en formato multiparte quedaría:

POST /indice.jsp HTTP/1.0
Host: www.unejemplo.com
User-Agent: Mozilla/4.5 [en]

Accept: image/jpeg, image/gif, text/html
Accept-language: en
Accept-Charset: iso-8859-1
Content-Type: multipart/form-data,
delimiter="----ALEATORIO----"

----ALEATORIO----

Content-Disposition: form-data; name="nombre"
Fulano Mengano

----ALEATORIO----
Content-Disposition: form-data; name="OK"

----ALEATORIO------

Esta codificación es propia y exclusiva del método POST. Se emplea para el envío de ficheros al servidor.

1.1.4.2.3 Respuestas en el protocolo HTTP. Las respuestas en el protocolo HTTP son similares a las peticiones. Una respuesta estándar sería similar a esto [11]:

HTTP/1.1 200 OK
Date: Mon, 04 Aug 2003 16:25:10 GMT
Server: Apache/2.0.40 (Red Hat Linux)
Last-Modified: Tue, 26 Mar 2004 08:53:53 GMT
Accept-Ranges: bytes
Content-Length: 428
Connection: close
<HTML>
...

Se puede observar que la primera línea responde con la versión del protocolo utilizada para enviar la página, seguida por el código de estado HTTP y lo que se denomina una frase de retorno.

Después del estatus aparecen unos campos de control, que tienen el mismo formato que en las cabeceras de la petición.

A continuación se incluye el contenido del recurso solicitado.

1.1.4.3 Servidores web más utilizados. La compañía inglesa Netcraft realiza mensualmente mediciones de uso de los distintos servidores. El estudio se hace comprobando la cantidad de dominios en los que se encuentra presente cada uno de los servidores. Las estadísticas proporcionadas por Netcraft son las más reputadas por su dimensión, y permiten conocer qué tipo de sistemas operativos y especialmente de servidores web funcionan en Internet en cada momento [28].

Según el informe publicado por Netcraft en enero de 2012 [28], el servidor que tuvo un mayor cantidad de sitios alojados, dadas las cifras de hostnames, fue el servidor Apache que logra un total de sitios de más de 378 millones, el que lo secunda es el servidor Microsoft con más de 84 millones de hostnames, en tercer lugar encontramos a Ngnix con un total de más de 56 millones de sitios y en cuarto lugar esta Google con una cifra superior a los 18 millones. Los detalles de este informe se observan en la Figura 3 y en la Tabla 1.

Figura 3. Participación en el mercado de los servidores principales en todos los dominios. (Agosto de 1995 – Enero de 2012)

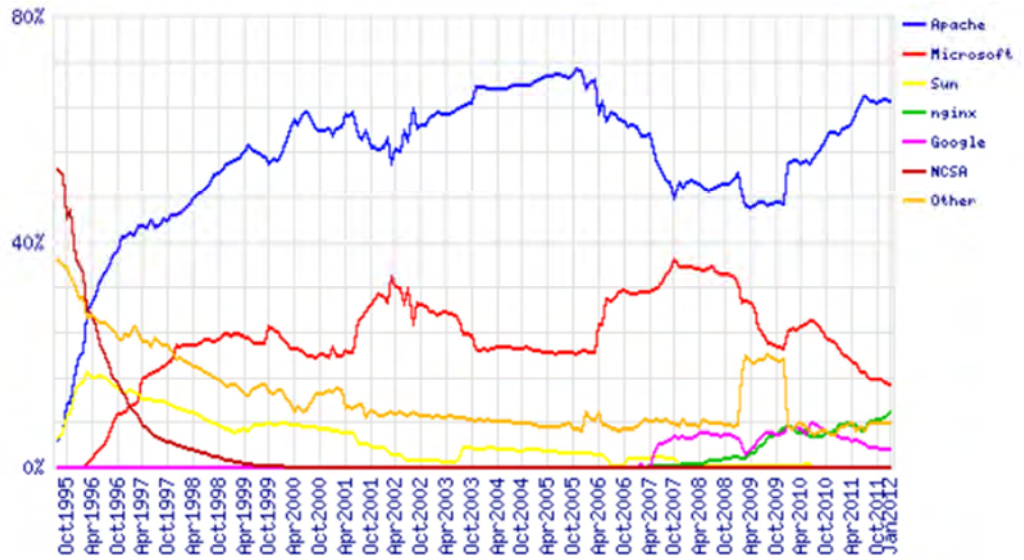


Tabla 1. Participación en el mercado de los servidores principales en todos los dominios. (Agosto de 1995 – Enero de 2012)

Developer	December 2011	Percent	January 2012	Percent	Change
Apache	362,267,922	65.22%	378,267,399	64.91%	-0.30
Microsoft	82,521,809	14.86%	84,288,985	14.46%	-0.39
nginx	49,143,289	8.85%	56,087,776	9.63%	0.78
Google	18,464,148	3.32%	18,936,381	3.25%	-0.07

Como se observa en la Figura 4 y en la Tabla 2, en las cifras de sitios activos (sitios que se encuentran en constante actualización) existe una variación en el ranking de lugares, donde el servidor Nginx asciende al segundo lugar, dejando a Microsoft en el tercer lugar.

Figura 4. Totales de sitios activos en todos los dominios. (Junio de 2000 – Enero de 2012)

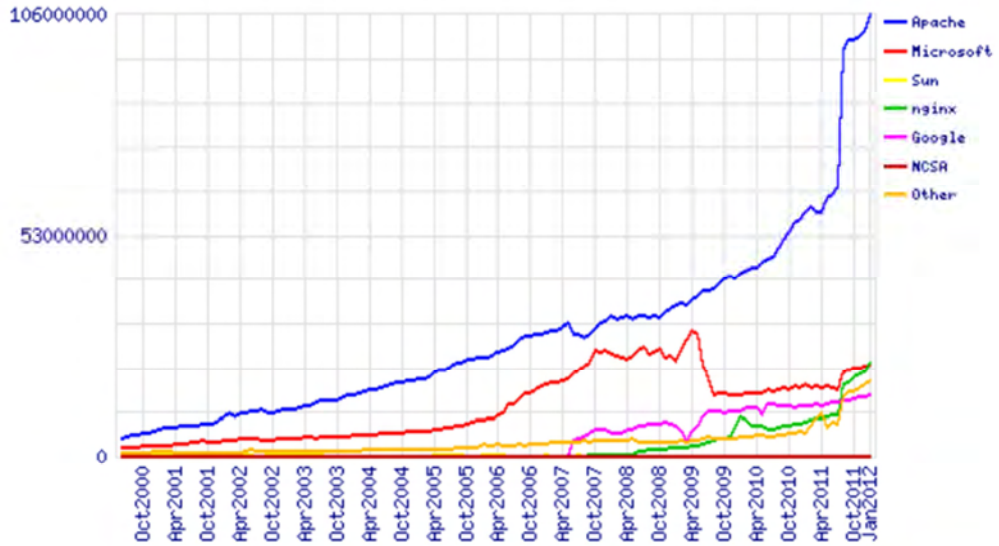


Tabla 2. Totales de sitios activos en todos los dominios. (Junio de 2000 – Enero de 2012)

Developer	December 2011	Percent	January 2012	Percent	Change
Apache	102,005,032	58.21%	105,684,049	57.93%	-0.28
nginx	20,342,324	11.61%	22,221,514	12.18%	0.57
Microsoft	21,572,870	12.31%	22,142,114	12.14%	-0.17
Google	14,240,979	8.13%	14,412,926	7.90%	-0.23

Teniendo en cuenta la información anterior, es importante realizar una breve descripción de los servidores web más utilizados:

- **Apache**: El servidor Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

- **Microsoft IIS**: Internet Information Services o IIS es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS. Los servicios de IIS proporcionan las herramientas y funciones necesarias para administrar de forma sencilla un servidor web seguro. Este servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas. Por ejemplo, Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

- **Nginx**: Es un servidor web/proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico (IMAP/POP3). Es software libre y de código abierto, licenciado bajo la Licencia BSD simplificada. Es multiplataforma, por lo que corre en sistemas tipo Unix (GNU/Linux, BSD, Solaris, Mac OS X, etc.) y Windows. El sistema es usado por una larga lista de sitios web conocidos, como: WordPress, Hulu, GitHub, Ohloh, SourceForge, TorrentReactor y partes de Facebook.

- **Google**: Google Web Server (GWS) es el nombre del servidor web que utiliza Google en sus infraestructuras y servidores. Google es intencionadamente vago acerca de GWS, simplemente se limitó a decir que es un servidor

personalizado de desarrollo propio que se ejecuta en sistemas UNIX cómo GNU/Linux. Adicionalmente existen especulaciones sobre que GWS es una versión modificada y adaptada de Apache HTTP Server que Google utiliza para su propia explotación.

1.2 MINERÍA WEB

La Minería Web o Web Mining consiste en aplicar las técnicas de minería de datos para descubrir y extraer automáticamente información de los documentos y servicios de la Web. [12] En particular, la creación, extracción y mantenimiento de los modelos de usuarios en Sistemas de Recomendación en Internet mejora la experiencia del usuario en relación con la información que le es relevante reduciendo el problema conocido como sobrecarga de la información. [17]

Sin embargo, las técnicas de minería de datos no son fácilmente aplicables a datos de la Web debido a problemas relacionados tanto con la tecnología subyacente como con la ausencia de estándares en el diseño e implementación de páginas Web. La información contenida en archivos log y otras fuentes de información debe ser procesada previamente a la obtención de los modelos. [25]

Etzioni señala que la minería Web está compuesta por tres tareas: descubrimiento de las fuentes, selección y preprocesamiento de la información y descubrimiento de patrones generales desde los sitios Web, en esta última es donde se realiza el proceso de minería en sí. En investigaciones posteriores se consideró una cuarta etapa dirigida al análisis para la validación y/o interpretación de los patrones minados. [29]

La minería Web puede dividirse en tres áreas o categorías principales: minería de contenido, minería de estructura y minería de uso, en función de los datos utilizados para inducir los modelos.

1.2.1 Tipos de minería web. En el caso de la minería Web los datos pueden ser obtenidos desde el lado del servidor, del cliente, de los servidores proxy o de la base de datos corporativa de la entidad a la cual pertenece el sitio. Desde este punto de vista, los datos encontrados en un sitio Web en particular, pueden ser clasificados en tres tipos predominantes [7], como se indica en la Figura 5.

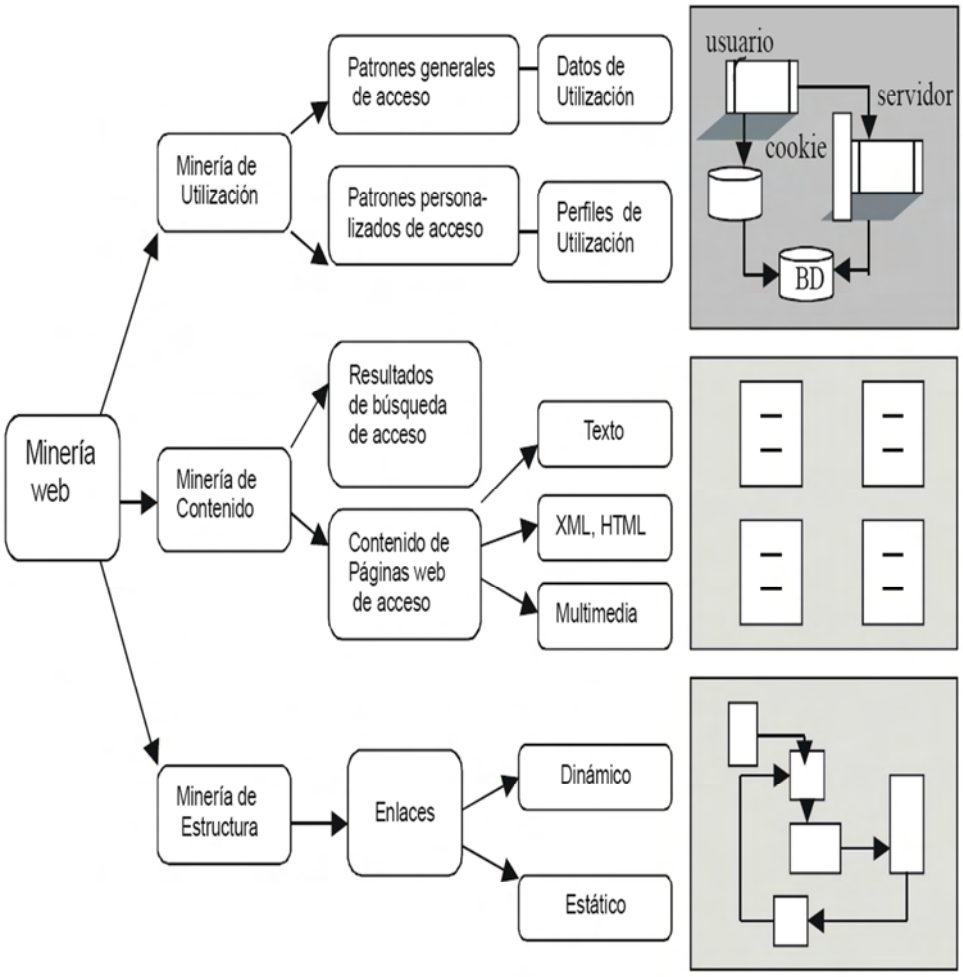
- **Minería de contenido de la web**: Son los datos reales que se entregan a los usuarios. Es decir, los datos que almacenan los sitios Web, los cuales consisten generalmente en textos e imágenes u otros medios. Este tipo de dato es el más importante y difícil de procesar, por ser multimedia.

- **Minería de uso de la web**: Son aquellos datos que describen el uso al cual se

ve sometido un sitio, registrado en los logs de acceso de los servidores Web.

- **Minería de la estructura de la Web**: Son los datos que describen la organización del contenido en el interior de un sitio. Esto incluye, la organización dentro de una página, la distribución de los enlaces tanto internos al sitio como externos, y la jerarquía de todo el sitio. (Ver figura 5)

Figura 5. Mapa conceptual de la Minería Web



1.2.1.1 Minería del contenido de la web. Web Content Mining (Minería de Contenido Web) se centra en el contenido, por lo que se pueden obtener datos acerca de la forma de escribir que sea más atractiva para el usuario, de si la catalogación que se usa sirve para mejorar la relevancia del sitio, si los temas que

se tratan interesan o no [15].

Esta área de la minería Web tiene dos vertientes: recuperación de la información y base de datos.

Como se conoce, los sitios de la Web están compuestos de colecciones de documentos de hipertexto. La recuperación de la información se realiza a través de la exploración semántica de los documentos, mediante dos enfoques: la minería de textos y el análisis semántico de los textos.

Considerando que los sitios de la Web también son colecciones de documentos semiestructurados, se pueden descubrir y extraer esquemas para formularios que capturen información semántica relevante de fuentes de datos heterogéneas. Los enfoques están basados en lenguajes de consultas para Web (XML, WebSQL, WebML), base de datos múltiples y descubrimiento de jerarquías.

1.2.1.2 Minería de la estructura de la web. Web Structure Mining (Minería de Estructura Web) se refiere al grado de dificultad que tienen los usuarios para encontrar la información, si la estructura del sitio es simple o muy profunda, si los elementos están colocados en los lugares adecuados dentro de la página, si la navegación es comprensible, cuáles son las secciones menos visitadas y su relación con el lugar que ocupan en la página central [15].

La minería de estructura Web revela más información que simplemente la información contenida en los documentos. Por ejemplo, enlaces o eslabones que apuntan a un documento indican su nivel de popularidad, mientras que los enlaces o eslabones que salen de un documento indican la riqueza o quizás la variedad de temas que se abarcan en el documento. Esto fue resaltado por Kleinberg en el algoritmo HITS, (del inglés Hypertext Induced Topic Selection), es un algoritmo diseñado para valorar y de paso clasificar la importancia de una página Web. [15]

Este tipo de minería Web está muy relacionada con la temática de los grafos tanto en la parte visual como los diferentes procesos que se llevan a cabo en cada uno de ellos, la teoría de grafos y la visualización de los mismos son campos muy amplios que en muchos estudios e investigaciones se trabajan totalmente independientes.

1.2.1.3 Minería uso de la web. Web Usage Mining (Minería de uso Web) se refiere a la técnicas de data mining para descubrir pautas de conducta a la hora de utilizar la Web por parte de los usuarios [7].

Intenta encontrar patrones sobre el uso que se le da a la Web a través del análisis de los registros de los servidores (Log files) sobre todas las transacciones

informáticas realizadas.

Según Yates y Poblate, podemos distinguir también aquí [7], de las tres áreas en las que se suele dividir la Minería Web o Web Mining, la que más éxito ha tenido es la Minería Web de Uso o Web Usage Mining, que se caracteriza por la aplicación de técnicas de Data Mining para la obtención de patrones acerca del uso que los usuarios le dan a la Web.

La Minería de Uso en la Web - Web Usage Mining – es la aplicación de las técnicas de minería de datos para descubrir patrones de uso desde los datos web, con el fin de entender y mejorar el servicio basado en las necesidades web.

Incluye típicamente el desarrollo de tres fases [7]:

Preparación y transformación de datos.
Descubrimiento de patrones.
Análisis de patrones.

- **Preparación y transformación de datos**. Una de las tareas más importantes para el uso de técnicas de minería de datos es la creación de un grupo de datos sobre el cual sus algoritmos puedan ser aplicados. Este proceso involucra el preprocesamiento de los datos originales, la integración con múltiples recursos (si es el caso) y la transformación de los datos integrados, en una forma adecuada, para ser utilizados en operaciones específicas de la minería.

  En el caso particular de minería Web de uso, la preparación y transformación de datos incluyen subprocesos específicos como limpieza de datos, identificación de páginas vistas, identificación de usuarios, identificación de sesiones (también llamado sesionalización), inferencia de referencias pérdidas debido a problemas de caché y la identificación de transacciones (ó episodios).

- **Descubrimiento de patrones**. Una vez los datos han sido preparados, métodos estadísticos y de Maquina de Aprendizaje (Machine Learning), son usados para extraer patrones de uso. Existe una gran variedad de modelos de aprendizaje utilizados para el descubrimiento de patrones en minería web de uso. Los modelos más comunes son: Descubrimiento de asociaciones y patrones secuenciales, clasificación y agrupación (clustering). Los primeros modelos de minería de datos utilizados en minería Web de uso fueron aquellos relacionados con clasificación, sin embargo, debido a la gran dificultad que representa etiquetar grandes cantidades de datos para realizar sobre ellos un aprendizaje supervisado, las técnicas más usadas hoy en día son aquellas que pertenecen a aprendizaje no supervisado, como agrupación. A continuación se describen las técnicas utilizadas en el proceso de descubrimiento de patrones

- Reglas de asociación es un modelo para encontrar patrones frecuentes, asociaciones y correlaciones entre grupos de ítems. Las reglas de asociación son usadas para revelar correlaciones entre páginas accedidas durante una sesión. Dichas reglas indican, además, la posible relación existente entre páginas visitadas simultáneamente aunque no se encuentren conectadas de una forma directa, y relaciones entre grupos de usuarios sin intereses específicos.

- El descubrimiento de patrones secuenciales es una extensión de minería de reglas de asociación que refleja patrones de concurrencia incorporando la noción de secuencia de tiempo. En el dominio Web, dichos patrones podrían ser grupos de páginas Web accedidas inmediatamente luego de otro grupo de páginas. A través de esta aproximación se pueden descubrir diferentes tendencias de los usuarios y usarlas para realizar predicciones.

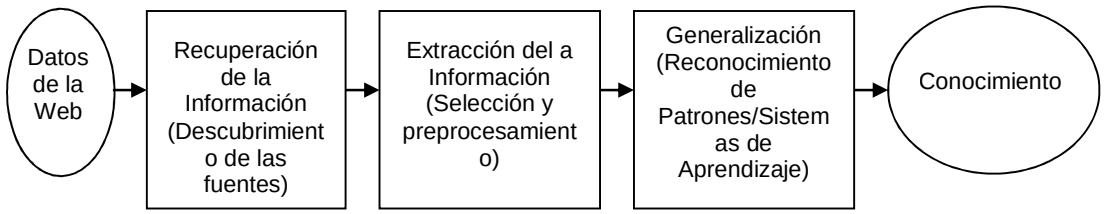
- Técnicas de agrupación son usadas para reunir ítems que tienen características similares. En el contexto de minería Web, se pueden distinguir dos casos: agrupación de usuarios y agrupación de páginas. La agrupación de páginas identifica grupos de páginas que, desde el punto de vista de los usuarios, se relacionan conceptualmente. Por otra parte, la agrupación de usuarios genera grupos de usuarios que exhiben un comportamiento de navegación similar en la web.

- Clasificación es un proceso que transforma ítems de datos en una de diferentes categorías preestablecidas. De esta manera, en el dominio Web se realizan diferentes tipos de categorizaciones como aquellas a documentos, tipos de usuarios, entre otras. Análisis de Patrones. Dependiendo de los objetivos que se quieran alcanzar con la aplicación de técnicas y modelos de minería web de uso, deberá llevarse a cabo una fase de análisis y estudio de los resultados obtenidos. Es importante mencionar que el trabajo con minería web de uso es independiente del dominio de aplicación sobre el que se está realizando el estudio, sin embargo para casos en los cuales el fin de este trabajo está enfocado en descubrir intereses de usuarios en un sitio web particular.

- **Análisis de patrones**. Dependiendo de los objetivos que se quieran alcanzar con la aplicación de técnicas y modelos de minería web de uso, deberá llevarse a cabo una fase de análisis y estudio de los resultados obtenidos. Es importante mencionar que el trabajo con minería web de uso es independiente del dominio de aplicación sobre el que se está realizando el estudio, sin embargo para casos en los cuales el fin de este trabajo está enfocado en descubrir intereses de usuarios en un sitio web particular.

1.2.2 Etapas de la minería web. Según Talwar y Mitra las etapas para la aplicación de la minería web son, descubrimiento de las fuentes, selección y preprocesamiento y generalización [29] como se observar en la **¡Error! No se encuentra el origen de la referencia.** 6.

Figura 6. Fases de la Minería Web



1.2.2.1 Descubrimiento de las fuentes. El descubrimiento de las fuentes o la recuperación de la información (RI) consisten en la recuperación automática de los documentos relevantes, mientras que se asegura al mismo tiempo, tanto como sea posible, que los no relevantes no sean considerados. El proceso de RI principalmente incluye la representación del documento, uso de índices y búsqueda de documentos [29].

El gran número de páginas en la Web, el dinamismo, y la actualización frecuente hace a las técnicas de uso de índices aparentemente imposible. En la actualidad, existen cuatro enfoques para poner índices a los documentos en la Web: índice humano o manual, índice automático, índice inteligente o basado en agentes e índices basados en metadatos.

Los futuros sistemas de descubrimiento de las fuentes harán uso de la tecnología de categorización de texto automática para clasificar los documentos de la Web en categorías. Esta tecnología podría facilitar la construcción automática de directorios de la Web como Yahoo, que localiza documentos y los presenta en categorías. Alternativamente, esta tecnología podría ser usada para filtrar los resultados de consultas a los índices de búsqueda.

Los estudios en la recuperación de información incluyen la modelación, desarrollo de interfaces con el usuario, visualización de los datos, y filtros. [8]

1.2.2.2 Selección/extracción y preprocesamiento. Una vez los documentos se han recuperado, el desafío es extraer conocimiento automáticamente y otras informaciones requeridas sin la interacción humana. La extracción de la información (EI) es la tarea de identificar fragmentos específicos de un documento

que constituyen su contenido semántico fundamental. Hasta ahora, los principales métodos de extracción de información involucran wrappers de escritura (codificación de la escritura) que asigna los documentos a algún modelo de datos [29].

Los wrappers actúan como interfaces de cada fuente de datos, proporcionando una semiestructura a aquellas fuentes no estructuradas o bien mapean la estructura de datos original en la búsqueda de un patrón común [1]. Los métodos wrapper, aunque son eficaces para eliminar atributos irrelevantes y redundantes, son muy lentos, variando en cada ejecución el número de atributos, siguiendo algún criterio de búsqueda y de parada. [21]

La extracción de información tiene como objetivo extraer el nuevo conocimiento de los documentos recuperados en la estructura y representación del documento mediante la conversión en mayúsculas, teniendo en cuenta que los expertos de la recuperación de información consideran que el texto del documento es una bolsa de palabras y no prestan atención a la estructura del documento. La escalabilidad es el mayor desafío más para los expertos de extracción de información; no es factible construir sistemas de extracción de información que sean escalables al tamaño y dinamismo de la Web. Por tanto, la mayoría de los sistemas de Extracción de información extraen información de sitios específicos y se enfocan en áreas definidas.

Algunos de los agentes inteligentes de la Web (software robots) han sido desarrollados para buscar información relevante usando características de un dominio particular (y posiblemente un perfil de usuario) para organizar e interpretar la información descubierta.

Es necesario un sistema de procesamiento robusto para extraer cualquier tipo de conocimiento, incluso a partir bases de datos medianas. Cuando un usuario solicita una página Web se accede a una variedad de archivos como imágenes, sonido, video, ejecutables y HTML. Como resultado, el log del servidor contiene muchas entradas que son redundantes o irrelevantes para las tareas de minería. Esto significa, que estas entradas deben eliminarse a través del preprocesamiento. Una de las técnicas de preprocesamiento usadas para la extracción de información es el índice semántico latente, del inglés latent semantic index, que busca transformar los vectores del documento original a un espacio dimensional más bajo mediante el análisis de la estructura correlacional en esa colección del documento de modo que documentos similares que no comparten los mismos términos se colocan en la misma categoría (tema).

1.2.2.3 Generalización. Una vez que se ha automatizado el descubrimiento y la extracción de la información procedente de los sitios Web, el siguiente paso es tratar de generalizar a partir de la experiencia acumulada. Para ello, la Minería

Web ha adaptado técnicas de minería de datos (reglas de asociación, clustering, entre otras), de la recolección de información y ha desarrollado algunas técnicas propias, como por ejemplo el análisis de caminos, que ha sido usado para extraer secuencias de patrones de navegación desde archivos log. [18]

Actualmente, la mayoría de los sistemas de aprendizaje desplegados en la Web se dedican más a aprender acerca de los intereses de sus usuarios, en lugar de aprender sobre el propio contenido y organización de la Web. El problema del etiquetado es un obstáculo importante cuando se aprende sobre la Web, ya que los datos son abundantes pero no están etiquetados. [12]

Algunas técnicas, como las pruebas de incertidumbre, reducen la cantidad de datos no etiquetados necesarios, pero no eliminan el problema del etiquetado. Un enfoque para resolver este problema se basa en el hecho de que la Web es mucho más que una colección enlazada de documentos, es un medio interactivo. Por ejemplo, Etzioni y Langheinrich [13] toman como entrada el nombre de una persona y su afiliación y se intenta localizar el home page de esa persona, de esta forma se les pregunta a los usuarios acerca de las páginas recuperadas, para etiquetar sus respuestas como correctas o incorrectas. [29]

Las técnicas de clustering no requieren las entradas etiquetadas y se han aplicado con éxito a las grandes colecciones de documentos. De hecho, la Web ofrece un terreno fértil para investigaciones de clustering y clasificación de documentos.

Las reglas de asociación también son una parte íntegra de esta fase. Básicamente, las reglas de asociación son expresiones del tipo $X \Rightarrow Y$, donde X e Y son conjuntos de elementos [29].

$X \Rightarrow Y$ expresa que siempre que una transacción T contenga a X entonces probablemente T también contiene a Y. La probabilidad o confianza de la regla se define como el porcentaje de transacciones que contienen a Y y además a X en relación con el total de transacciones que contienen a X.

1.2.2.4 Análisis. El análisis es un problema de manipulación de información que requiere que existan datos suficientes disponibles para que la información potencialmente útil se pueda extraer y analizar. Los humanos juegan un papel importante en el proceso de descubrimiento del conocimiento en la Web, considerando que la Web es un medio interactivo. Esto es especialmente importante para la validación y/o interpretación de los patrones minados que tienen lugar en esta fase. Una vez que los patrones se han descubierto, los analistas necesitan herramientas apropiadas para entender, visualizar e interpretar estos patrones. Algunos usan el Procesamiento Analítico en Línea, del inglés Online Analytical Processing (OLAP) con el propósito de simplificar el análisis de

las estadísticas a partir de los logs de los servidores, y otros mecanismos SQL, como el sistema WEBMINER que propone un lenguaje de consultas, similar a SQL, que posee un mecanismo de consultas para preguntar acerca del conocimiento descubierto (en forma de reglas de la asociación y modelos secuenciales) [29].

1.2.3 Minería de datos. La Minería de Datos (DM) por las siglas en inglés Data Mining es el proceso de extraer conocimiento útil y comprensible, previamente desconocido, desde grandes cantidades de datos almacenados en distintos formatos [23]. Las herramientas de Data Mining predicen futuras tendencias y comportamientos, permitiendo en los negocios la toma de decisiones.

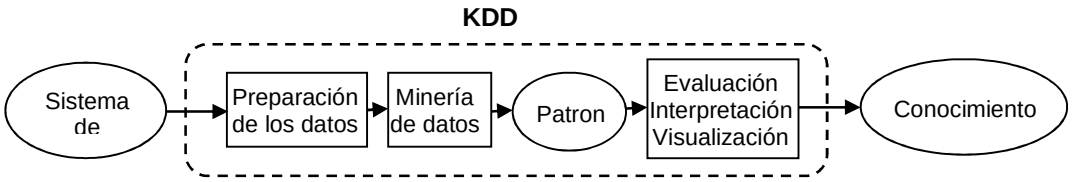
Existen términos que se utilizan frecuentemente como sinónimos de la minería de datos. Uno de ellos se conoce como "análisis (inteligente) de datos" [10], que suele hacer un mayor hincapié en las técnicas de análisis estadístico. Otro término muy utilizado, y el más relacionado con la minería de datos, es la extracción o "descubrimiento de conocimiento en bases de datos" (Knowledge Discovery in Databases o KDD, según sus siglas en inglés). [18]

Aunque algunos autores usan los términos Minería de Datos y KDD indistintamente, como sinónimos, existen claras diferencias entre los dos. Así la mayoría de los autores coinciden en referirse al KDD como un proceso que consta de un conjunto de fases, una de las cuales es la minería de datos. De acuerdo con esto, el proceso de minería de datos consiste únicamente en la aplicación de un algoritmo para extraer patrones de datos y se llamará KDD al proceso completo que incluye pre-procesamiento, minería y post-procesamiento de los datos.

El KDD según es la extracción automatizada de conocimiento o patrones interesantes, no triviales, implícitos, previamente desconocidos, potencialmente útiles y predictivos de la información de grandes Bases de Datos.

La
muestra las fases del proceso de KDD, una de las cuales es la Minería de Datos

Figura 7. Fases del proceso KDD

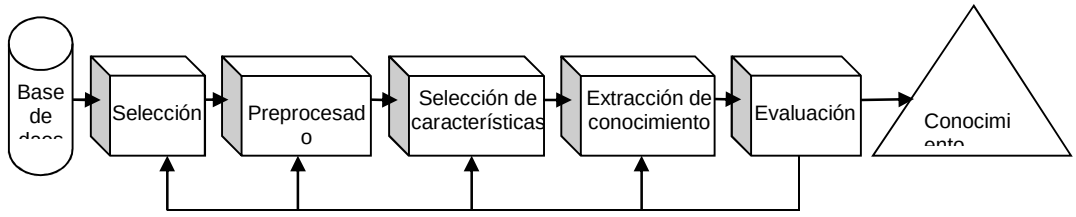


Fuente: González Díaz, Ernesto; Pérez Hernández, Zady y Espinosa Conde, Ivet. monografias.com. Técnicas de minería de datos. [En línea] [Citado el: 8 de Mayo de 2009.] http://www.monografias.com/trabajos55/mineria-de-datos/mineria-de-datos.shtml. [38]

Las investigaciones en temas de KDD incluyen análisis estadístico, técnicas de representación del conocimiento y visualización de datos, entre otras. Algunas de las tareas más frecuentes en procesos de KDD son la clasificación y clustering, el reconocimiento de patrones, las predicciones y la detección de dependencias o relaciones entre los datos.

1.2.3.1 Fases de la minería de datos. Los pasos a seguir para la realización de un proyecto de minería de datos son siempre los mismos, independientemente de la técnica específica de extracción de conocimiento usada. (Ver figura 8)

Figura 8. Fases dentro de un proceso de minería de datos



Fuente: González Díaz, Ernesto; Pérez Hernández, Zady y Espinosa Conde, Ivet.monografias.com. Técnicas de minería de datos. [En línea] [Citado el: 8 de Mayo de 2009.] http://www.monografias.com/trabajos55/mineria-de-datos/mineria-de-datos.shtml. [16]

Según Gonzales y Zady [16], el proceso de minería de datos pasa por las siguientes fases:

- **Comprensión del problema**: Se debe comprender plenamente el problema al cual se le quiere dar solución.

- **Filtrado de datos**: El formato de los datos contenidos en la fuente de datos nunca es el correcto, y la mayoría de las veces no es posible ni siquiera utilizar algún algoritmo de minería sobre los datos iníciales sin que requieran alguna transformación. En este paso se filtran los datos con el objetivo de eliminar valores incorrectos, no válidos o desconocidos; según las necesidades del algoritmo a utilizar. Además se obtienen muestras de los datos en busca de mayor velocidad y eficiencia de los algoritmos, o se reducen el número de valores posibles para los atributos de análisis.

- **Selección de variables**: Después de realizar la limpieza de los datos, en la mayoría de los casos se tiene una gran cantidad de variables o atributos. La selección de características reduce el tamaño de los datos, sin apenas sacrificar la calidad del modelo de conocimiento obtenido del proceso de minería; seleccionando las variables más influyentes en el problema. Los métodos para la selección de los atributos que más influencia tienen en el problema son básicamente dos: aquellos basados en la elección de los mejores atributos del problema y quellos que buscan variables independientes mediante test de sensibilidad, algoritmos de distancia o heurísticos.

- **Extracción de conocimiento**: La extracción del conocimiento es la esencia de la Minería de Datos donde mediante una técnica, se obtiene un modelo de conocimiento, que representa patrones de comportamiento observados en los valores de las variables del problema o relaciones de asociación entre dichas variables. Los modelos que se generan son expresados de diversas formas: reglas, arboles y redes neuronales. También pueden usarse varias técnicas a la vez para generar distintos modelos, aunque generalmente cada técnica obliga a un pre-procesado diferente de los datos.

- **Interpretación y evaluación**: Una vez obtenido el modelo, se procede a su validación; donde se comprueba que las conclusiones que arroja son válidas y suficientemente satisfactorias. En el caso de haber obtenido varios modelos mediante el uso de distintas técnicas, se deben comparar los modelos para buscar el que se ajuste mejor al problema. Si ninguno de los modelos alcanza los resultados esperados, debe alterarse alguno de los pasos anteriores para generar nuevos modelos.

1.3 ANÁLISIS ESTADÍSTICO DE TRÁFICO WEB

1.3.1 Descripción y características generales. El tráfico web hace referencia a la cantidad de datos enviados y recibidos por los visitantes de un sitio web. El tráfico web se produce cada vez que un cliente interactúa con un servidor HTTP, proceso en el que se realiza una transferencia de recursos desde el servidor hacia el cliente. Por cada recurso enviado al cliente (esto es, cada página HTML y cada elemento no textual que contiene, como botones, separadores, iconos, etc.), el servidor escribe una línea en un archivo de registro de accesos.

El tráfico web puede ser analizado mediante la generación de estadísticas sobre los datos encontrados en el archivo del servidor de la página web, el cual genera automáticamente una lista de todas las páginas vistas.

El análisis web establece un conjunto de técnicas relacionadas con el análisis de datos relativos al tráfico en un sitio web con el objetivo de entender su tráfico como punto de partida para optimizar diversos aspectos del mismo. En el análisis de tráfico web es importante tener en cuenta criterios de medición como los siguientes:

- Número total de solicitudes (Request).
- Solicitudes fallidas.
- Solicitudes redireccionadas
- Número de visitantes.
- Promedio de páginas vistas.
- Promedio de transferencia de datos.
- Promedio de duración de la página (por cuánto tiempo es vista).
- Numero de direcciones Ip únicas.
- Páginas más requeridas (más populares).
- Horas pico (Horarios de mayor audiencia).

Los datos registrados en los archivos log son la materia prima para desarrollar procesos que permitan analizar el tráfico web de un sito. Estos archivos incluyen

información sobre errores, tiempo de procesamiento, ancho de banda utilizado, dirección IP del visitante, de dónde procedían las visitas (referentes) así como información sobre el sistema operativo y el navegador empleado por los visitantes. El análisis estadístico de tráfico web permite realizar un análisis del registro de la actividad del servidor y generan informes compuestos por tablas de datos y/o gráficas, con el objetivo de responder preguntas como:

¿Cuál es el archivo más solicitado?

¿Cuáles son los principales dominios web de origen: países, comerciales, etc.?

¿Cuáles son los navegadores y sistemas operativos más utilizados?

¿Cuál es el número de páginas vistas?

¿Cuál es el número de usuarios que se conectan a un archivo específico?

¿Qué tipo de usuarios se conectan a un archivo?

¿Qué hora del día es la "hora pico" de un sitio web?

¿Cuáles son las imágenes más vistas?

1.3.2 Análisis de tráfico web versus minería web. Un sitio web puede ser estudiado desde varios puntos de vista, el estadístico por ejemplo en el cual se refiere a la utilización de técnicas estadísticas que permitan medir audiencias en internet, o en otras palabras análisis descriptivos de usabilidad de los recursos almacenados un sitio web; la tarea de este tipo de método es un análisis de datos dirigido a la verificación, lo que en palabras más precisas se trata de análisis estadísticos de ficheros web logs, técnica que probablemente sea la más difundida en todo el planeta. Los análisis estadísticos de logs persiguen objetivos como determinar la cantidad de páginas vistas o hits realizados por los usuarios del sitio, y determinar por ejemplo el uso de un recurso contenido en una página. Una respuesta posible de obtener por medio de herramientas estadísticas de análisis de archivos logs puede ser la siguiente: El sitio publiquegratis.net tiene un total de 10 visitas sobre un recurso publicitario contratado y alojado en él. Esta respuesta se basa en las peticiones realizadas al servidor web que aloja la pagina web, peticiones que se transforman en eventos registrados en los archivos logs. El problema se presenta cuando se necesita relacionar la página de interés con grupos de usuarios o usuarios, dado que para obtener respuestas de tipo todos los usuarios X visitan Y es necesario determinar el usuario ip que realizo la petición; esta ip puede ser la misma para muchos usuarios o distinta para los mismos (cuando estos emplean direcciones ip dinámicas o servidores proxy corporativos). Es posible resolver este problema mediante el uso de Web Mining o Minería de Datos aplicada a los ficheros logs, metodología que es conocida con el nombre de

Web Usage Mining. Cualquier intento de dar respuesta a determinar relaciones como por ejemplo usuarios X visitan Y, necesariamente se enfrenta al problema de la determinación de la ip del cliente o usuario, problema que ha llevado al uso de metodologías de Web Usage Mining planteada en [33] referidas a la identificación de sesiones, determinación de episodios, determinación de caminos de búsquedas completos y definición de que es un "pageview", propuestas que tienen por objeto asimilar los registros contenidos en los archivos logs a una base de datos tradicional; para luego aplicar técnicas de minería de datos que permitan encontrar patrones de comportamiento sobre el sitio web. Polaris – Herramienta de Mineria de Uso de la Web [40] propone algoritmos de minería de datos como A priori, FPGrowth y EquipAsso que se encargan de determinar grandes grupos de ítems y asociarlos a un cliente, un ítem puede en analogía ser asociado a una página web y el cliente a una ip o a un visitante identificado mediante el proceso de sesionalización.

1.3.3 Análisis de archivos logs de servidores web. Un log es un registro oficial de eventos durante un rango de tiempo en particular. Para los profesionales en seguridad informática es usado para registrar datos o información sobre quién, qué, cuándo, dónde y por qué (who, what, when, where y why) un evento ocurre para un dispositivo en particular o aplicación. [48]

La mayoría de los logs son almacenados o desplegados en el formato estándar, el cual es un conjunto de caracteres para dispositivos comunes y aplicaciones. De esta forma cada log generado por un dispositivo en particular puede ser leído y desplegado en otro diferente.

A su vez la palabra log se relaciona con el término evidencia digital. Un tipo de evidencia física construida de campos magnéticos y pulsos electrónicos que pueden ser recolectados y analizados con herramientas y técnicas especiales, lo que implica la lectura del log y deja al descubierto la actividad registrada en el mismo [48].

Los servidores web crean y administran automáticamente archivos de texto, en los cuales se almacena toda la actividad que se hace sobre el servidor. El ejemplo más típico de log de servidor es el log de accesos de un servidor web, en donde se almacenan datos como la dirección IP, navegador, fecha y hora, etc., de cada acceso al mismo. En el caso de sitios web con gran tráfico, los archivos web llagan a superar los 100 megas diarios. De toda esta avalancha de datos es posible obtener estadísticas de acceso y estudiar las preferencias de los visitantes.

1.3.3.1 Datos almacenados en archivos logs de accesos. Cuando se establece una comunicación entre un cliente y un servidor, se produce un intercambio de información. En este intercambio de información de datos intervienen distintos

sistemas de comunicación, programas de software y protocolos de comunicación. Este proceso va generando distintos eventos relacionados con los recursos requeridos, el destino y con el éxito o fracaso de la comunicación entre el cliente y servidor o viceversa. Estos eventos pueden ser capturados y registrados por medio de programas que se ejecutan en os servidores, siendo estas aplicaciones las encargadas de capturar los distintos eventos de importancia que ocurren tanto a nivel de hardware como en los distintos niveles de software que intervienen entre la comunicación entre cliente y servidor.

Un evento puede ser entendido como el resultado de una operación realizada por un servidor ante una solicitud o requerimiento por parte de un cliente, como por ejemplo la solicitud y posterior retorno de un documento, la ejecución de un script, la bajada de un archivo, u otras solicitudes de servicios. Estos requerimientos de los usuarios hacia el servidor y las clases de respuestas que este entrega como resultado de las operaciones solicitadas pueden ser registradas empleando programas que capturan el resultado de la operación, almacenando los distintos componentes asociados al tipo de solicitud y las respuesta entregadas por el servidor en un archivo de registro de eventos. Considerando la arquitectura de Web más sencilla, cada vez que un cliente realiza un requerimiento a un servidor de red, es enviado un paquete HTTP (Hypertext Transfer Protocol) sobre la red, desde el cliente hacia el servidor nombrado en el campo URL (Universal Resource Locator) de la solicitud. Luego el servidor retorna uno o más respuestas o bien un código de error.

Cada vez que se produce un evento, este es capturado y registrado en un archivo de registro de eventos o log file, en estos archivos se almacenan todios los sucesos relevantes que ocurren en un servidor web. En estos archivos por ejemplo pueden quedar registradas las páginas requeridas por un usuario, el tiempo de conexión del usuario con el servidor web, los puertos de acceso utilizados u otros parámetros referidos al uso del sitio web o del servidor de red que lo almacena. A partir de la información almacenada en los archivos logs, es posible establecer relaciones u asociaciones posteriores como por ejemplo: la hora y fecha cuando fue requerida una página web por parte de un cliente, el total de páginas requeridas por un usuario en particular o la página más visitada.

Los servidores Web pueden recolectar las solicitudes de documentos, objetos y otros servicios por parte de sus clientes, registrando cada solicitud HTTP y sus resultados. Este proceso de captura de información es realizado de manera automática por una o varias aplicaciones que se ejecutan en tiempo real en el servidor, los resultados obtenidos ante un requerimiento de un cliente del servidor web son capturadas por estas aplicaciones sean estos exitosos o hayan terminado en un error, resultando eventos que son almacenados en forma de registros en un archivo log dedicado a esta única función. Es importante mencionar que la información almacenada en estos archivos no es del todo confiable ya que las aplicaciones encargadas de capturar los eventos pueden ser personalizadas por el

administrador del servidor, tanto en el tamaño máximo en bytes del archivo en donde son almacenados los eventos, como en otros parámetros relacionados con la captura de los mismos. Pr ejemplo, si el archivo log completa su capacidad máxima de almacenamiento algunos sucesos de importancia pueden no ser registrados. Por otra parte, el empleo de Servidores Proxy afecta la calidad de los datos registrados en los archivos de registro de eventos del servidor, ya que es posible que las solicitudes de los clientes queden registradas como anónimas o con el nombre del proxy. Otro problema que se puede presentar es que la información almacenada en los logs del servidor excluye aquellos documentos (páginas web) que están almacenado en un "cache de páginas". En caso de existir más de un servidor que proporciona un servicio, existirá la posibilidad que un mismo evento sea almacenado en dos archivos diferentes. Se deberá tener especial cuidado que el registro de los eventos en caso de dos o más servidores que prestan el mismo servicio, no se interfieran unos con los otros almacenando registros de eventos redundantes. Existe la posibilidad que para dar cumplimiento a normativas relativas a la privacidad, no sea posible recolectar los nombres de los clientes.

La información que contienen estos archivos logs es principalmente de tipo numérica y se refiere a datos como fecha, tiempo de conexión, dirección ip del cliente, el fichero o página de destino en el servidor, puerto de destino del servidor. A modo de ejemplo el Servidor Web Apache registrará cualquier evento por causa de peticiones de recursos, objetos y servicios sobre los archivos access_log y error_log.

1.3.3.2 Formatos de archivos logs de accesos. Una de las principales fuentes de información que puede usarse para realizar estadísticas Web son los datos que provienen de los procesos que capturan y registran los eventos ocurridos sobre un Servidor Web, eventos que son almacenados en los archivos logs. La información generada por un evento es capturada y posteriormente almacenada bajo la forma de un registro que se almacena en un archivo ASCII de texto, empleando para este fin un formato conocido; el formato corresponde a la forma en cómo se almacenan los datos provenientes del evento como por ejemplo el formato de la hora y la fecha [40].

Uno de los formatos más empleados corresponde al desarrollado por NCSA (National Center for Supercomputing Applications) denominado Formato Común de Registro o Common Logs Format (CLF) y su extensión o ampliación denominada Formato de Registro Combinado o Combined Logs Format (XCLF), por medio del empleo de estos formatos el Servidor Web Apache almacena los registros que contienen la información de un evento. La empresa Microsoft en cambio, para su servidor web denominado Microsoft Internet Information Server o IIS, suministra la posibilidad de grabar en un fichero log o de registro todos los eventos o entradas de los usuarios que se conectan al mismo en tres formatos diferentes: Microsoft IIS Log File Format; el NCSA Common Log File Format; y

W3C Extended Log File Format; además de la posibilidad de grabar los datos directamente en un servidor de datos mediante un enlace ODBC con la opción ODBC Logging [27]. El formato que más información ofrece al administrador del servidor web es mediante la utilización del formato W3C Extended, ya que permite recoger hasta un total de 20 datos diferentes, incluyendo las direcciones IP del cliente y del servidor, fecha y hora de la conexión, bytes enviados y recibidos, recursos visitaos y otras opciones.

Los formatos de los archivos logs son relativamente libres y descriptivos del tipo de suceso, de la aplicación que los captura y de las características de diseño del servidor web aplicadas por su fabricante o desarrollador. En términos generales todos los formatos incluyen datos como la fecha y hora del evento, la dirección ip del cliente u otra información relevante. La mayoría de los archivos de registro emplean un formato estándar para identificar la fecha y la hora del suceso, basándose en la norma ISO 8601 que se refiere a recomendaciones para le representación de fecha y la hora.

1.3.3.2.1 Formatos NCSA. Los archivos de registro con formato NCSA pueden ser guardados en un formato de archivo log común llamado Formato Común de Registro o Common Logs Format (CLF) o en un formato de archivo log extendido llamado Formato de Registro Combinado o Combined Logs Format (XCLF).

El formato Common Log Format (CLF) fue utilizado originalmente por el servidor Web del NCSA, llegando a convertirse en el estándar a utilizar por los servidores Web en sus registros. Los registros CLF contienen casi toda la información necesaria para realizar estudios exhaustivos sobre la actividad de un servidor Web. En la Figura 9, se observa un registro de un archivo log con formato CLF.

Figura 9. Formato de archivo Common Log Format

```
in24.inetnebr.com - - [01/Aug/1995:00:00:01 -0400] "GET /shuttle/missions/sts-68/news/sts-68-mcc-05.txt HTTP/1.0" 200 1839
```

La estructura del archivo CLF se describe a continuación [38]:

remotehost rfc931 authuser [date] "request" status bytes

- **Remotehost** (host remoto). Dirección IP o nombre DNS del cliente.

  Ejemplo: in24.inetnebr.com.

- **Rfc931**. Identificación remota del cliente. No se aplica. "identd" es un servicio exclusivo de UNIX. La búsqueda de identidad consume mucho ancho de

banda, y sobrecarga innecesariamente a los servidores. Raramente se utiliza. Si no existe se escribe un guión ("-").

- **Authuser** (Autenticacion de Usuario). Nombre de usuario del cliente. Es el nombre con el que el usuario se ha identificado. Normalmente este campo no se llena y aparece como un guión ("-").

- [**Date**](fecha). Fecha y hora de la solicitud.

  Ejemplo: [01/Aug/1995:00:00:01 -0400].

- "**Request**" (Solicitud). La línea exacta de petición según viene solicitada desde el cliente.

  Ejemplo: "GET /shuttle/missions/sts-68/news/sts-68-mcc-05.txt HTTP/1.0".

- **Status** (Estado). El código de estado del HTPP devuelto al cliente.

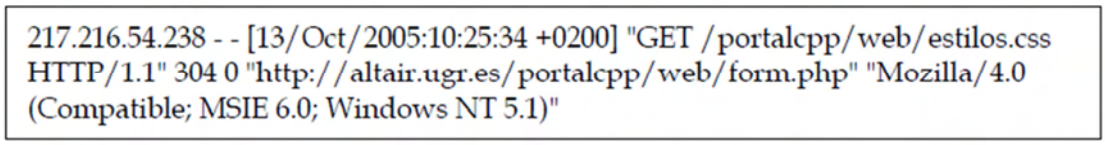
  Ejemplo: 200.

- **Bytes**. Numero de bytes enviado al cliente. Se refiere a la longitud que tiene el archivo transferido.

  Ejemplo: 1838.

El formato de registro de NCSA combinado o combined log format (XCLF) es una extensión del formato de registro común NCSA common log format (CLF). Como se observa en la Figura 10, el formato combinado contiene la misma información que el formato de registro común, más tres campos adicionales: el campo de referencia y el campo user_agent. (Ver figura 10)

Figura 10. Formato de archivo combined log format



Fuente: TIMARÁN, R., DAZA, J., ZULETA, A., ANGULO, D. Polaris: Una Herramienta para Minería de Uso de la Web. Trabajo de Grado. San Juan de Pasto. Universidad de Nariño. Facultad de Ingeniería. Departamento de Ingeniería de Sistemas. 2007. 271 p. [40]

Los siguientes son los campos del formato de registro combinado (con los tres campos adicionales que aparecen en negrita):

- **Host**
- **Rfc931**
- **Username**
- **Date:time**
- **Request**
- **Statuscode**
- **Bytes**
- **Referrer**
- **User_agent**

Los campos adicionales que aparecen en el formato de registro combinado se describna a continuación:

- "**Referrer**" (Referente). URL desde donde se ha realizado la petición.

Ejemplo: "http://altair.ugr.es/portalcpp/web/form.php".

- "**User-Agent**". Tipo de navegador y sistema operativo usado.

Ejemplo: "Mozilla/4.0 (compatible; MSIE 6.0; Windows™ NT 5.1)".

1.3.3.2.2 Formato W3C extended log file format. La mayoría de servidores web ofrecen la opción de almacenar ficheros de registro, ya sea en el formato de registro común o un formato propietario. En muchos casos es deseable grabar más información. Sitios sensibles a las cuestiones de datos personales podría omitir el registro de ciertos datos. Además ambigüedades surgen al analizar el formato de archivo de registro común, ya que los caracteres separadores de campo en algunos casos puede ocurrir dentro de los campos. [36]

Un archivo de registro extendido Extended Log File Format contiene una secuencia de líneas que contienen caracteres ASCII. Las entradas consisten en una secuencia de campos relacionados con una sola transacción HTTP. Los campos están separados por espacios en blanco. Si un campo no se utiliza en una cadena de entrada en particular "-" marca el campo omitido. Directivas registrar información sobre el proceso de registro en sí.

Las líneas que comienzan con el carácter # contienen directivas (Figura 11). Las directivas se definen los siguientes:

Version: La versión del formato de archivo de registro extendido utilizado.
Fields: Especifica los campos registrados en el registro.
Software: identifica el software que generó el registro.
Start-Date: La fecha y hora en que se inició el registro.
End-Date: La fecha y hora en que se terminó el registro.
Date: La fecha y la hora en que se agregó la entrada.
Remark: la información de comentario. Los datos registrados en este campo debe ser ignorada por las herramientas de análisis. (Ver figura 11)

Figura 11. Ejemplo de encabezado de Extended Log File Format

```
#Software: Microsoft Internet Information Services 5.0
#Version: 1.0
#Date: 2002-12-08 17:05:01
#Fields: date time c-ip cs-username s-ip s-port cs-method cs-uri-stem cs-uri-query sc-status cs(User-Agent)
```

El prefijo de algunos nombres de estos campos, está formado de la siguiente forma [36]:

s- =Relativo al servidor
c- =Relativo al cliente
cs-=Flujo del cliente al servidor
sc-=Flujo del servidor al cliente

En la Tabla 3, se observa los nombres de los campos descritos en la linea #Fields de los archivos de logs y una breve descripción de cada uno de ellos.

Tabla 3. Campos de extended log file format

CAMPO	DESCRIPCIÓN
Fecha (date)	Fecha de la petición de página.
Hora (time)	Hora de la petición de página.
Dirección IP del cliente (c-ip)	IP de la petición.
Nombre de usuario (cs-name)	Nombre de usuario (autentificación en el servidor web). Tendrá el login de la cuenta del servidor Windows. Si el sitio es publico, tendrá un guion - si es un acceso anónimo.
Nombre de servicio (cs-username)	W3SVC1 en general. Es el nombre del servicio de ese sitio web. Lo normal es tener logs de cada sitio web separados, así este campo es redundante.
Nombre de servidor (s-computername)	Nombre de la maquina donde esta alojado el Servidor que ofreció los datos.
Dirección IP del servidor (s-ip)	IP del servidor que sirvió la página (puede existir varias IP en un mismo dominio).
Puerto del servidor (s-ip)	Puerto local del servidor, por el cual se sirve los datos (80 en general).
Método (cs-method)	GET, POST. Método de acceso del cliente.
Recurso (URI) visitado (cs-uri-stem)	Pagina visitada. Se sustituyen los espacios por signos +.
Consulta (URI) solicitada (cs-uri-query)	Variables enviadas con el método post. Se logeara a partir del ? sin incluir este signo
Estado del protocolo (sc-status)	Estado de respuesta del servidor, dependiendo del estado de la pagina. (Por ej: 200 si todo ha ido bien)
Estado Win32 (sc-win32-status)	Estado de respuesta del servidor, con códigos de Windows.
Bytes enviados (sc-bytes)	Numero de bytes enviados por el servidor.

Tabla 3. Campos de extended log file format

CAMPO	DESCRIPCIÓN
Bytes recibidos (cs-bytes)	Numero de bytes recibidos por el servidor.
Tiempo consumido (time-taken)	Tiempo que se uso para llevar a cabo la petición.
Versión del protocolo (cs-versión)	Versión del protocolo que uso el cliente (HTTP 1.0 por ejemplo).
Host (cs-host)	Nombre host por el cual accedió el cliente al servidor.
Agente de usuario (cs(User-Agent))	Nombre del cliente de páginas web (IExplorer, mozilla,...).
Cookie (cs(Cookie))	Contenido de las cookies usadas.
Sitio de referencia (cs(Referer))	Sitio del cual se procede. (el que vinculo al cliente a la pagina solicitada).

1.3.3.3 Análisis de archivos logs de accesos. Como se menciono anteriormente los servidores web (los de FTP, proxy-cache, etc. son fuentes primarias) guardan registros de eventos que ocurren por peticiones de recursos realizadas vía http, estos ficheros son almacenados en el disco duro en donde se aloja el sistema operativo, o en la ubicación seleccionada por el administrador del sistema. En estos archivos logs se anotan todos los eventos que ocurren durante el funcionamiento normal del servicio web o sitio web.

En el registro de accesos al servidor o log de acceso se almacenan las peticiones de recursos y objetos realizadas por los usuarios reales o virtuales al servidor.

Los datos que se encuentran almacenados en los archivos logs y el formato de estos datos depende del tipo de servidor web utilizado en la implementación de un sitio web, como por ejemplo Apache, Internet Information Server o AOLServer. Por lo general, estos servidores guardan los eventos en los archivos logs utilizando formatos como Common Log Format (CLF), Combined Logs Format (XCLF) o W3C Extended Log File Format, en términos generales la información factible de obtener en este tipo de archivos se resume en la Tabla 4.

Tabla 4. Resumen de información almacenada en un archivos Logs

1	Número de peticiones recibidas (hits).
2	Volumen total de bytes de datos y archivos servidos.
3	Número de peticiones por tipo de archivo.
4	Direcciones de clientes diferentes atendidas y peticiones para cada una de ellas.
5	Número de peticiones por dominio (o direccion IP).
6	Número de peticiones por directorio o archivo.
7	Número de peticiones por código de estado HTTP.
8	Direcciones de procedencia (referrer).
9	Navegadores usados.

En términos generales las aplicaciones para analizar logs, se diseñan para trabajar con los distintos formatos disponibles como son el NCSA, W3C u otros formatos empleados para almacenar registros en archivos de texto; estas aplicaciones generan diversas estadísticas a partir de la información rescatada de estos archivos, estadísticas que se basan principalmente en la información factible de obtener a partir de los datos almacenados en los archivos log, en la Tabla 4 se observa un resumen de la información contenida en los formatos de archivos logs mas utilizados. A pesar de que la información que se puede obtener del análisis de archivos logs es importante y numerosa, existe información que no se puede obtener por medio de un análisis estadístico de logs, donde se hace indispensable el uso y aplicación de técnicas de Minería de Uso de la Web. Esta información se observa en la Tabla 5.

Tabla 5. Información complicada de obtener

1	Identidad de los usuarios, excepto en aquellos casos en los que el usuario se identifique por petición del servidor.
2	Número de usuarios. A pesar de tener el número de IP distintas es posible saber de forma absoluta el número de usuarios. Una dirección IP puede representar varios usuarios.
3	Un robot, araña u otro programa de navegación automático.
4	Un usuario individual con un navegador en su ordenador.
5	Un servidor proxy-cache que puede ser usado por cientos de usuarios.
.6	Ficheros no vistos.
	Otras…

1.3.4 Análisis de tráfico web en polaris Versión 3.0.

1.3.4.1 Criterios de medición. Los usuarios de un sitio web mantienen un comportamiento que se ve reflejado en los objetos o recursos que solicitan al servidor que aloja el sitio web, este comportamiento puede ser analizado desde el punto de vista estadístico, es decir, la cantidad de recursos solicitados, la cantidad de usuarios que acceden al sitio, la cantidad de bytes transferidos (tamaño del recurso) o identificación de los usuarios que se conectan al sitio.

Para realizar un análisis de datos desde el punto de vista de la estadística descriptiva, la cual se encarga de recolectar, ordenar, analizar y representar un conjunto de datos, con el fin de describir apropiadamente las características de ese conjunto [50], es preciso identificar criterios de medición que puedan asociarse a los diferentes recursos y objetos representativos del tráfico web, que poseen propiedades cuantitativas.

Así entonces, es posible recoger datos que se pueden analizar con la estadística, desde páginas y objetos de un sitio web empleando los siguientes parámetros de medición:

- Cantidad de Solicitudes (hits).
- Transferencia de datos (Tamaño de la información transferida)
- Cantidad de visitantes (Sesiones de usuario)

Un hit es una petición a un servidor web de un archivo (página web , imágenes, JavaScript , hojas de estilo en cascada , etc.). Cuando una página web se carga de un servidor el número de "hits" es igual al número de archivos solicitados. Por lo tanto, una página cargada no siempre es un hit, porque a menudo las páginas están formadas por otras imágenes y archivos. Debido a que una página cargada no es igual a un hit, esta es una medida imprecisa de la popularidad de un sitio web o de tráfico web. Una medida más precisa de tráfico web es el número de visitas realizadas a un sitio web.

Una visita (o sesión de usuario) está formada por el conjunto de páginas accedidas por un usuario durante una misma sesión de trabajo; generalmente se considera que la sesión de trabajo se mantiene mientras el tiempo entre la vista de dos páginas consecutivas no supere un determinado umbral o *timeout* no siempre fácil de determinar.

A modo de ejemplo, si tenemos los siguientes registros de un archivo log:

Remote Host	Date Time		Request	Bytes
192.168.5.10	15/Dec/2011	10:35:00	/xy.html	80
192.168.5.10	15/Dec/2011	10:36:00	/xy.html	20

Es posible asociar el objeto xy.html con los parámetros de medición mencionados así:

Hits	2
Transferencia de datos	100 bytes
Visitantes	1

Según esta información se puede afirmar que el objeto xy.html fue requerido 2 veces, con un transferencia total de 100 bytes y se asocia a 1 vista, lo que quiere decir que este archivo fue solicitado 2 veces en la misma sesión de usuario.

1.3.4.2 Proceso de sesionalización. Para realizar un análisis confiable de los eventos o peticiones registradas en archivos logs, es importante desarrollar procesos que permitan identificar los objetos requeridos y los sujetos que los solicitan con la finalidad de determinar por ejemplo el uso de un objeto en particular y su relación con los objetos que lo solicitan [40].

Por medio de la determinación de una sesión de usuario o visita al servidor es posible determinar por ejemplo la conducta o comportamiento de este en su interacción con el sitio web, identificando así la secuencia de transacciones realizadas en una visita en particular. Una sesión de usuario puede ser obtenida a partir de los requerimientos que un visitante realiza sobre los objetos contenidos en un sitio web, objetos que son contenidos en las paginas desplegadas en el navegador que usa el usuario en un instante e tiempo dado, su navegación corresponderá por tanto a la secuencia de páginas que un usuario en particular solicita al servidor y el tiempo total desde que inicia una visita hasta que abandona el sitio. Es importante mencionar que en un archivo log se almacena toda la actividad que los usuarios realizan sobre el sitio web, por lo tanto un registro de este tipo puede contener múltiples sesiones de un usuario o múltiples visitas; el concepto de sesión de usuario se refiere por tanto a la actividad de segmentar un archivo log en tiempos de visita para establecer una sesión real de usuario, el usuario puede ser identificado por todas las secuencias de objetos o recursos requeridos en una visita.

Antes de comenzar cualquier tipo de análisis de tráfico web basado en los datos almacenados en archivos logs, es necesario extraer las transacciones que pertenecen a usuarios individuales. Una transacción se puede considerar como sola entrada en el registro o un sistema de entradas alcanzadas por un visitante de la misma máquina en un lapso de tiempo definido o sesión. La transacción deseada puede ser el sistema de entradas de registro de un visitante en una sola visita. Sin embargo, el uso de los servidores Proxys y la utilización de las Caches, hace difícil la identificación de las sesiones de usuarios. [40]

Una sesión de host se define como la secuencia de peticiones al servidor 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. Es importante establecer el tiempo máximo que debe transcurrir entre una petición de un host y otra petición del mismo host para que se considere como parte de la misma visita.

El proceso de identificación de sesiones consiste en dividir los accesos de un determinado usuario en sesiones de navegación independientes, agrupando los pertenecientes a una misma sesión de trabajo y que se han realizado de manera ininterrumpida por parte del usuario. El método a utilizar para realizar este proceso es el de *timeout*, en el cual se utiliza umbral o tiempo límite (*timeout*), de modo que si el tiempo transcurrido entre dos peticiones consecutivas del mismo usuario supera ese límite, se considera que ha iniciado una nueva sesión; medidas empíricas [32] sitúan este *timeout* en aproximadamente 30 minutos.

Mediante la implementación de este proceso es posible identificar visitas (o sesiones de usuarios) y posteriormente asociar cada objeto o recurso solicitado (páginas web, imágenes, videos, etc.) a una visita en particular. Los registros de

sesiones y objetos asociados a estas se registran en tablas independientes. En la Tabla 6, se observa cómo se registran las peticiones en un archivo log, Tabla 7 muestra los registros de de las sesiones con sus respectivas horas de inicio y fin y en la Tabla 8 se observan los registros de las solicitudes de objetos asociadas a cada sesión.

Tabla 6. Registro de un archivo log

Id Peticion	Host Remoto	Fecha - Hora
1	A	F1
2	A	F2
3	A	F3
4	B	F4
5	B	F5
6	B	F6
7	C	F7
8	C	F8
9	C	F9
10	C	F10

Tabla 7. Reagistro de sesiones

Id Sesion	Host Remoto	Inicio	Fin
1	A	F1	F3
2	B	F4	F6
3	C	F7	F10

Tabla 8. Registro de sesiones y objetos asociados

Host Remoto	Id Petición	Id Sesión
A	1	1
A	2	1
A	3	1
B	4	2
B	5	2
B	6	2
C	7	3
C	8	3
C	9	3
C	10	3

Al almacenar estos registros en una base de datos, es posible obtener valiosa información mediante la utilización del lenguaje SQL (Structured Query Language) que permite obtener información extra mediante diferentes procesos como el cruce de registros, esta información es la base para realizar un análisis estadístico completo.

1.3.4.3 Estadísticas por periodos de tiempo. Un aspecto muy importante que se debe tener en cuenta al realizar un análisis estadístico de tráfico web de un sitio web en particular, es determinar periodos de tiempo en los que se observan los niveles de audiencia más altos o bajos sobre el sitio web, es decir, intervalos de tiempo en los que los usuario realizan el mayor y/o número de peticiones de objetos o recursos al servidor web.

Un evento registrado en un archivo log está constituido por tres parámetros principales: una dirección Ip que identifica el sujeto o cliente, un objeto o recurso solicitado y un campo que indica la fecha y hora. Esto permite asociar cada fecha y hora con un objeto solicitado y con el cliente o usuario que realizo la solicitud. Por lo general los servidores web registran la fecha y hora de cada petición con un formato aaaa-mm-dd hh:mm:ss el cual pude variar dependiendo del formato del archivo log; de cualquier forma, es posible extraer el nombre del día, la fecha y la hora del dia y asociarlos a los parámetros de medición mencionados anteriormente. Así establecemos los siguientes periodos de tiempo:

- Periodo hora del día.
- Periodo día de la semana.
- Periodo fecha.

De acuerdo a esto es posible crear estadísticas que permitan conocer el número de visitas, el número de hits y la cantidad de bytes transferidos en cada periodo de tiempo establecido. Por ejemplo, si se tiene una secuencia de registros como se observa en la Tabla 8 y además se asocia cada registro a su fecha correspondiente y se registra la cantidad de datos trasferidos en cada periodo de tipo, es posible obtener información como la registrada en la Tabla 9.

Tabla 9. Objetos y periodos de tiempo asociados

Host Remoto	Id Petición	Id Sesión	Transferencia	Fecha	Día	Hora
A	1	1	bytes	F1	"nombre_dia"	hh
A	2	1	bytes	F2	"nombre_dia"	hh
A	3	1	bytes	F3	"nombre_dia"	hh
B	4	2	bytes	F4	"nombre_dia"	hh
B	5	2	bytes	F5	"nombre_dia"	hh
B	6	2	bytes	F6	"nombre_dia"	hh
C	7	3	bytes	F7	"nombre_dia"	hh
C	8	3	bytes	F8	"nombre_dia"	hh
C	9	3	bytes	F9	"nombre_dia"	hh
C	10	3	bytes	F10	"nombre_dia"	hh

Si se tiene esta información es posible obtener la cantidad de hits, visitantes o bytes transferidos por periodos de tiempo mediante un proceso de agrupación de registros, en el cual se agrupa cada registro de acuerdo a un día, fecha u hora en específico.

1.3.4.4 Estadísticas de accesos. Los usuarios que interactúan con un sitio web, lo hacen básicamente con la finalidad de acceder a los objetos que este contiene. Una página web puede ser definida como un objeto contenedor que puede almacenar un número determinado de objetos. Estos objetos representan archivos de diversos tipos tales como documentos, imágenes, animaciones, videos, archivos de audio, archivos comprimidos, etc.

Como se menciono anteriormente, un registrado en un archivo log está constituido principalmente por una dirección Ip que identifica el sujeto o cliente, un objeto o recurso solicitado y un campo que indica la fecha y hora. De esta forma el servidor registra la información sobre el objeto solicitado, en un campo que en su formato

básico tiene la siguiente sintaxis:

Método URI Versión

El método le indica al servidor que hacer con el URI, por último la versión simplemente indica el número de versión del protocolo HTTP utilizado. Una petición habitual utiliza el método GET para pedirle al servidor que devuelva el URI solicitado. A manera de ejemplo, el registro de la petición de un archivo index.html haciendo uso del método GET y el protocolo HTTP 1.0 quedaría registrada en el archivo log como GET /index.html HTTP/1.0.

En la URI se registra la dirección, el nombre y la extensión del objeto u archivo solicitado, esta información es de gran utilidad ya que permite identificar los diferentes archivos solicitados, clasificarlos de acuerdo a su formato y generar las estadísticas de accesos correspondientes a cada archivo. Para este fin, se clasifico los archivos en archivos de página web, archivos de imagen, archivos de audio, archivos de video, archivos de texto, archivos comprimidos y archivos de internet.

En la Tabla 10, se observa la extensión y la descripción de archivos de tipo página web.

Tabla 10. Archivos tipo página web

Extensión de Archivo	Tipo de Archivo	Descripción
.aspx	Web Page	Página ASP
.jsp	Web Page	Jsp File
.php4	Web Page	Personal Home Page
.htm	Web Page	Pagina Web HTML
.html	Web Page	Pagina Web HTML
.asp	Web Page	Pagina ASP
.php	Web Page	Personal Home Page

En la Tabla 11, se tiene la descripción de archivos de tipo imagen.

Tabla 11. Archivos tipo imagen

Extensión de Archivo	Tipo de Archivo	Descripción
.tga	Image	Imagen
.tif	Image	Imagen
.tiff	Image	Imagen
.pic	Image	Imagen
.pcx	Image	Imagen
.ico	Image	Icono
.emf	Image	Imagen
.bmp	Image	Mapa de bits
.gif	Image	Formato de intercambio gráfico
.dib	Image	Imagen
.jpg	Image	Fotografía comprimida
.png	Image	Imagen

En la Tabla 12, se tiene la extensión y la descripción de archivos de tipo audio.

Tabla 12. Archivos tipo audio

Extensión de Archivo	Tipo de Archivo	Descripción
.aif	Audio	Reproductor Winamp
.mid	Audio	Música sintetizada
.mp3	Audio	Música comprimida
.ra	Audio	Música de Real Audio
.ash	Audio	Música Windows Media
.cda	Audio	Reproductor Winamp
.snd	Audio	Reproductor Winamp
.midi	Audio	Música sintetizada
.voc	Audio	Reproductor Winamp
.amf	Audio	Reproductor Winamp
.wma	Audio	Reproductor Winamp

En la Tabla 13, se tiene la extensión y la descripción de archivos de tipo video.

Tabla 13. Archivos tipo video

Extensión de Archivo	Tipo de Archivo	Descripción
.mov	Video	QuickTime
.mp4	Video	Formato mpeg-4
.mpg	Video	Formato mpeg
.avi	Video	Más extendido
.wmv	Video	Windows Media
.mpeg	Video	Formato mpeg

En la Tabla 14 se observa la extensión y la descripción de archivos de tipo texto.

Tabla 14. Archivos tipo texto

Extensión de Archivo	Tipo de Archivo	Descripción
.docx	Text	Microsoft Word
.doc	Text	Microsoft Word
.txt	Text	Bloc de Notas
.pdf	Text	Adobe Acrobat
.rtf	Text	Microsoft Word

En la Tabla 15, se tiene la extensión y la descripción de archivos de tipo comprimido.

Tabla 15. Archivos tipo comprimido

Extensión de Archivo	Tipo de Archivo	Descripción
.bz2	Compressed	Izarc - WinRar
.bz	Compressed	Izarc - WinRar
.cab	Compressed	Cab - Station
.gz	Compressed	IZarc - WinRar
.iso	Compressed	WinRar
.rar	Compressed	WinRar
.tar	Compressed	IZarc - WinRar
.tgz	Compressed	Izarc - WinRar
.zip	Compressed	WinZip

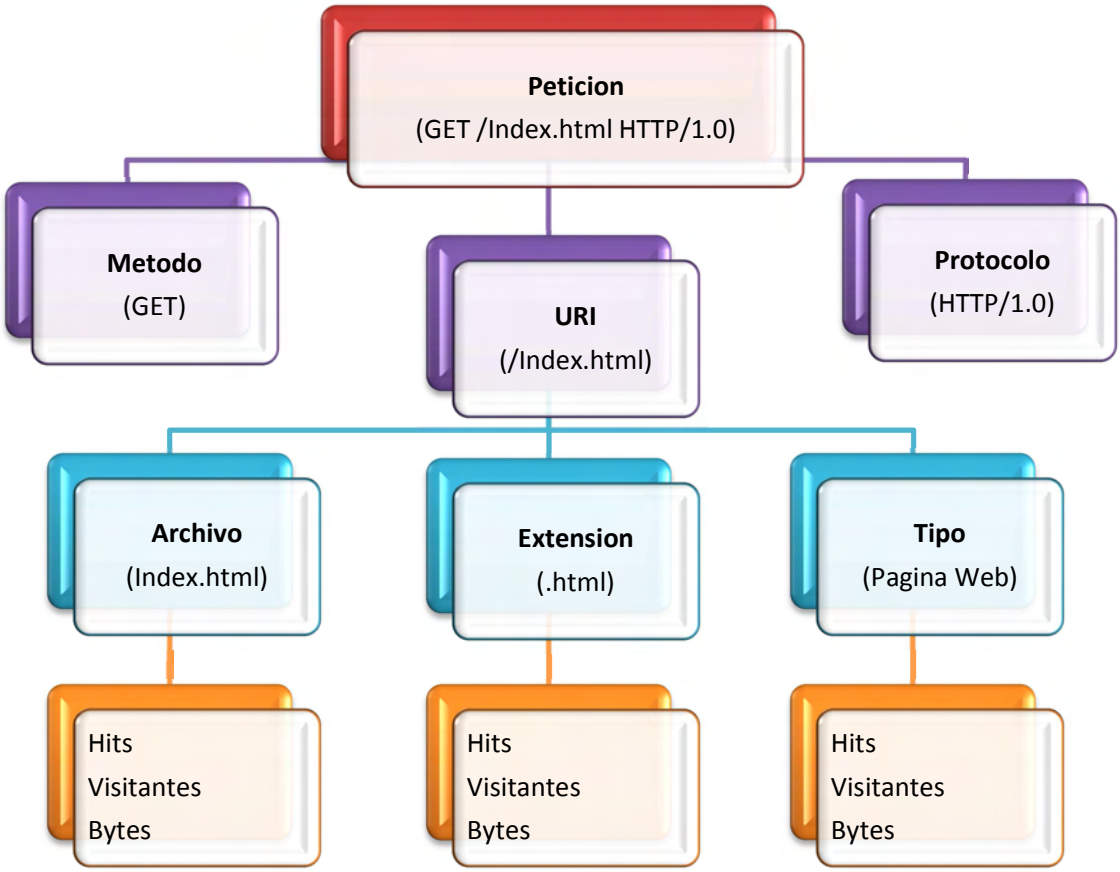
En la Tabla 16, se tiene la extensión y la descripción de archivos de tipo internet.

Tabla 16. Archivos tipo internet

Extensión de Archivo	Tipo de Archivo	Descripción
.js	Internet	Java Script
.swf	Internet	Objeto Macromedia
.css	Internet	Hoja de Estilos

Con la información descrita en estas tablas y la información que proporciona el campo URI que describe el objeto o recurso solicitado por el cliente, se puede obtener tres grupos de datos: un grupo de datos que representa el nombre de los diferentes archivos solicitados, un grupo de datos donde se identifica las extensiones de estos archivos y por último, un grupo de datos con los tipos de archivo asociados a cada objeto solicitado. Una vez realizado este proceso, se crea las estadísticas de acceso relacionando cada uno de los objetos obtenidos (nombre de archivo, extensión de archivo y tipo de archivo) con los parámetros de medición establecidos (Hits, Visitantes, Transferencia) como se muestra en la Figura 12.

Figura 12. Estadísticas de accesos



1.3.4.5 Análisis de agentes de usuario. Otra información importante que es almacenada en los archivos de registro de los servidores web, es la información que identifica y describe a la aplicación que accede a los objetos o recursos de la web. Esta información generalmente hace referencia al navegador web que usa un usuario para acceder a la web y se conoce como User Agent o agente de usuario. Es destacable mencionar que junto con los usuarios reales existen usuarios virtuales o programas especializados como robots o arañas que son agentes inteligentes que generan eventos idénticos al solicitar un objeto o recurso de la web, por esta razón, es de suma importancia diferenciar usuarios reales y usuarios virtuales al realzar procesos de análisis de tráfico web.

Un agente de usuario es una aplicación informática que funciona como cliente en un protocolo de red; el nombre se aplica generalmente para referirse a aquellas aplicaciones que acceden a la World Wide Web. Los agentes de usuario que se

conectan a la Web pueden ser desde navegadores web hasta los web crawler de los buscadores. Cuando un usuario accede a una página web, la aplicación generalmente envía una cadena de texto que identifica al agente de usuario ante el servidor. Este texto forma parte del pedido a través de HTTP, llevando como prefijo User-agent: o User-Agent: y generalmente incluye información como el nombre de la aplicación, la versión, el sistema operativo, y el idioma. Los bots, como los web crawlers, a veces incluyen también una URL o una dirección de correo electrónico para que el administrador del sitio web pueda contactarse con el operador del mismo.

Para realzar análisis confiable sobre la información contenida en el campo User Agent, es indispensable identificar y clasificar estos datos, ya que como se menciono anteriormente el agente de usuario registra diferentes características de la aplicación que genero el evento en el servidor. Para este fin se clasificó los agentes de usuario de la misma forma como se muestra en la Tabla17.

Tabla 17. Clasificación de User Agents

Identificador	Tipo de Aplicacion
B	Browsers.
C	Link -, bookmark -, server - checking.
D	Downloading tool.
P	Proxy server, web filtering.
R	Robot, crawler, spider.
S	Spam or badbot

El cliente que accede a un recurso de la web, ya sea un usuario real o un usuario virtual, envía al servidor una cadena de texto que identifica la aplicación utilizada para acceder al recurso; el formato y el contenido de esta cadena dependen del tipo de aplicación. En la Tabla 18 se observan algunos ejemplos de las cadenas de User Agents generadas por los diferentes tipos de aplicaciones descritas en la Tabla 18.

Tabla 18. Ejemplos de cadena User Agent

Aplicación	Cadena User Agent
Browsers.	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:11.0) Gecko Firefox/11.0
Link -, bookmark -, server - checking.	Mozilla/5.0 (compatible; AbiLogicBot/1.0; +http://www.abilogic.com)
Downloading tool.	DA 4.0 (www.downloadaccelerator.com)
Proxy server, web filtering.	DoCoMo/1.0/P502i/c10 (Google CHTML Proxy/1.0)
Robot, crawler, spider.	Googlebot/2.1 (+http://www.google.com/bot.html)
Spam or badbot	Mozilla/4.0 (compatible; Advanced Email Extractor v2.xx)

Antes de realizar un análisis de tráfico web en relación con el agente de usuario, es importante conocer la definición de cada uno de los tipos de aplicaciones mencionadas anteriormente. Las aplicaciones mencionadas en la categoría C (Link -, bookmark -, server - checking), hacen referencia a herramientas de comprobación. Un Link Checker es una aplicación que permite comprobar el estado de los enlaces o hipervínculos en las páginas web de un sitio web; un Bookmark Checker es una herramienta que se usa para comprobar la validez de los marcadores de internet (almacenan la localización de una página web) y un Server Cheker permite comprobar si un sitio web está funcionando o si un servidor está en línea. En la categoría D se encuentran las aplicaciones tipo Downloadin Tool o herramientas de descarga que son herramentas que permiten gestionar los procesos de descargas desde la web. En la categoría S se tiene a los Spam que son mensajes no solicitados, no deseados o de remitente desconocido. En la categoría P se tiene a los Servidores Proxy (Proxy Server) que son programas o dispositivos que acceden a un servidor en lugar de en logar del usuario y los Filtros Web (Web Filter) que permiten filtrar el contenido de sitios web dependiendo de los requerimientos del usuario. Las aplicaciones que mencionaron anteriormente no son grandes generadores de eventos como si ocurre con los navegadores web y los robots o arañas que recorren la web con el objetivo de registrar los documentos, por lo que es importante observar de manera más detallada el comportamiento de estas aplicaciones.

Cuando un cliente accede a un recurso web por medio de un navegador web (usuario real), el navegador envía una serie de información hacia el servidor que aloja el sitio web que se está visitando, como se muestra en el siguiente ejemplo:

GET / HTTP/1.1
Accept: text / html, application / xhtml + xml, * / *

Accept-Language: en-US
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)
Accept-Encoding: gzip, desinfla
Proxy-Connection: Keep-Alive
Host: microsoft.com

En la Tabla 19, se describen los campos de la cadena de agente de usuario que se muestra en el ejemplo:

Tabla 19. Agente de usuario tipo navegador web

Campo	Descripción
Mozilla/5.0	Nombre de la aplicación y la versión.
compatible	Se indica que el navegador web es compatible con un conjunto de características comunes.
MSIE 9.0	Identifica el nombre del navegador y el número de versión.
Windows NT 6.1	Identifica el sistema operativo y versión.
Trident/5.0	Identifica la versión del MSHTML (Trident).

Así pues, a partir de la información registrada en la cadena User Agent es posible extraer dos grupos de datos importantes: un grupo de datos que reúne los diferentes sistemas operativos utilizados y otro grupo donde se identifican los navegadores web. Para obtener estos grupos de datos es indispensable conocer los sistemas operativos y navegadores web que los usuarios utilizan para acceder a la web. También es de suma importancia identificar la cadena de caracteres que representa a cada sistema operativo o navegador web utilizado en el gente de usuario. Esto permitirá mediante un cruce de registros extraer de la cadena de agente de usuario únicamente la información requerida. La información referente a navegadores web se describe en la Tabla 20 y la información relacionada con sistemas operativos se observa en la Tabla 21.

Tabla 20. Lista de navegadores web

Nombre	String	Tipo
LeechCraft	LeechCraft	Browser
Konqueror	Konqueror	Browser
Links	Links (	Browser
lolifox	lolifox	Browser
Lorentz	Lorentz/	Browser
Lynx	Lynx/	Browser
Minefield	Minefield/	Browser
myibrow	myibrow/	Browser
Namoroka	Namoroka/	Browser
Navscape	Navscape/	Browser
NetNewsWire	NetNewsWire	Browser
NetPositive	NetPositive	Browser
Netscape	Netscape/	Browser
NetSurf	NetSurf/	Browser
OmniWeb	OmniWeb/	Browser
Opera	Opera	Browser
Oregano	Oregano	Browser
osb-browser	osb-browser	Browser
QtWeb Internet Browser	QtWeb Internet Browser/	Browser
…		

Tabla 21. Lista de sistemas operativos

Sistema Operativo	String
BeOS	BeOS
Linux	Linux
Linux	X11
Mac OS	Mac OS
Mac OS	Macintosh
Mac OS	Mac_PowerPC
Open BSD	OpenBSD
OS/2	OS/2
QNX	QNX
Sun OS	SunOS
Windows 2000	Windows NT 5.0
Windows 2000	Windows 2000
Windows 3.11	Win16
Windows 7	Windows NT 6.1
Windows 95	Win95
Windows 95	Windows_95
Windows 95	Windows 95
Windows 98	Windows 98
Windows 98	Win98
Windows CE	Windows CE
Windows ME	Windows ME
Windows ME	win 9x
Windows NT 4.0	WinNT4.0
Windows NT 4.0	WinNT
Windows NT 4.0	Windows NT)
Windows NT 4.0	Windows NT 4.0
Windows Server 2003	Windows NT 5.2
Windows Vista	Windows NT 6.0
Windows XP	Windows NT 5.1
Windows XP	Windows XP

Por ejemplo, para determinare el número de hits o peticiones que se realizaron desde el navegador Opera (Navegador Web creado por la empresa noruega Opera Software), se deberá revisar los archivos log del servidor web y buscar los registros o eventos almacenados en estos archivos en los que aparezca la cadena de caracteres *Opera*.

Como se mencionó anteriormente, al igual que los navegadores, las aplicaciones descritas en la categoría C de agentes de usuario (Robot, Crawler, Spider) generan una cantidad relativamente alta de eventos sobre los servidores web. Este tipo de aplicaciones o agentes inteligentes difieren de un navegador web en un aspecto, que es la autonomía en la búsqueda; un browser debe ser operado por un usuario real, en cambio un agente inteligente es un dispositivo de software que tiene incorporado técnicas de aprendizaje y realiza búsqueda de documentos de acuerdo a ciertos criterios y genera resultados que están destinados a la construcción de listas de índice. Los robots recorren la web de forma metódica y automatizada y su uso consiste en crear una capia de todas las páginas web visitadas para posteriormente ser procesadas por un motor de búsqueda que indexa las paginas proporcionando un sistema de búsqueda rápido. Como se menciono anteriormente, los robots o spiders se consideran como grandes fuentes generadoras de eventos siendo necesario identificarlos, ya sea para bloquear su acceso al servidor web o como en este caso para generar estadísticas de acceso a la web.

Cuando un agente inteligente (Crawler, robot o spider) realiza una visita a un sitio web genera un evento que es registrado en los archivos log junto con su correspondiente cadena de agente de usuario. Por ejemplo Google emplea el robot conocido como googlebot, el cual registra cadenas de agente de usuario como las siguientes:

Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
Googlebot/2.1 (+http://www.googlebot.com/bot.html)
Googlebot/2.1 (+http://www.google.com/bot.html)

Con esta información es posible determinar el número de veces que el robot GoogleBot ha visitado un sitio web, se deberá revisar los archivos log del servidor y buscar los registros o eventos almacenados en esos archivos en los que aparezca la cadena de caracteres “Googlebot”. Obviamente existen un gran número de agentes inteligentes diferentes a GoogleBot, por lo que es necesario identificar los más importantes con el fin de impedir que estos filtren o se confundan con otras aplicaciones al momento de realizar un análisis de registros. En la Tabla 22, se muestra a manera de resumen los robots, crawlers o spiders más utilizados.

Tabla 22. Lista de robots, crawlers y spiders

Nombre	String	Tipo
BeslistBot	BeslistBot	Robot, crawler, spider
Covario-IDS	Covario-IDS/	Robot, crawler, spider
DataparkSearch	DataparkSearch/	Robot, crawler, spider
DiamondBot	DiamondBot	Robot, crawler, spider
Discobot	discobot/	Robot, crawler, spider
DotBot	DotBot/	Robot, crawler, spider
EARTHCOM.info	EARTHCOM.info	Robot, crawler, spider
EsperanzaBot	EsperanzaBot	Robot, crawler, spider
envolk[ITS]spider	envolk[ITS]spider/	Robot, crawler, spider
Exabot	Exabot/	Robot, crawler, spider
Googlebot	Googlebot/	Robot, crawler, spider
Gigabot	Gigabot/	Robot, crawler, spider
Googlebot-Image	Googlebot-Image/	Robot, crawler, spider
GurujiBot	GurujiBot/	Robot, crawler, spider
HappyFunBot	HappyFunBot/	Robot, crawler, spider
hl_ftien_spider	hl_ftien_spider	Robot, crawler, spider
Holmes	holmes/	Robot, crawler, spider
IssueCrawler	IssueCrawler	Robot, crawler, spider
findlinks	findlinks/	Robot, crawler, spider
RufusBot	RufusBot	Robot, crawler, spider
…		

1.3.4.6 Análisis de sitios de procedencia o referentes. Una petición HTTP genera un campo conocido como HTTP referer que identifica, la dirección de la página web desde donde se origino la solicitud. En la situación más común, esto significa que cuando un usuario hace clic en un enlace en un navegador web , el navegador envía una solicitud al servidor que contiene la página web de destino. La solicitud incluye el campo de referencia, que registra información sobre la última página en la que el usuario estaba antes de acceder al sito (en la que él usuario hace clic en el enlace).

El registro referer se utiliza para permitir a sitios web y servidores web identificar la procedencia de las visitas, con fines promocionales o de seguridad. El referer permite obtener información valiosa desde el punto de vista estadístico.

Al visitar una página web , la referencia o página de referencia es la URI de la página web anterior de la que fue seguido de un enlace.
En términos más generales, un referer es la URI de un elemento anterior que dio

lugar a esta solicitud. El referer para una imagen, por ejemplo, es generalmente la pagina HTML en la que se va a mostrar. El campo de referencia es una parte opcional de la petición HTTP enviada por el navegador web al servidor web.

Un usuario puede acceder a un objeto o recurso de un sitio web básicamente de dos formas: acceder sede un sitio web en particular al hacer click en un enlace o hipervínculo o realizando un proceso de búsqueda utilizando un motor de búsqueda como por ejemplo Google. Teniendo en cuenta esto es importante en análisis de archivos log identificar los eventos en los que se se accedió al sitio web por medio de un enlace y los eventos en los que se accedió al sitio utilizando un motor de búsqueda. Esto permite obtener información importante como la siguiente:

- Sitios web referentes.
- Moros de busque utilizados.
- Palabras clave (secuencia de palabras utilizadas en la búsqueda)

Una vez se ha identificado el modo y el sitio desde el cual se accedió a determinado objeto o recurso, es posible relacionar esta información directamente con el recurso u objeto solicitado, permitiendo generar estadísticas interesantes.

Independientemente de la forma y el sitio de procedencia de una petición http, el servidor web permite registrar la dirección URI de referencia. A modo de ejemplo, si un usuario ingresa a al sitio web de la Universidad de Nariño desde el portal web de Universia, el servidor web registra el referente de la siguiente manera:

http://estudios.universia.net/colombia/institucion

Del mismo modo, si el usuario accede al sitio web desde un enlace generado mediante un proceso de búsqueda en Google, el servidor web almacenara el la URI del referente de la siguiente manera:

http://www.google.com.co/#hl=es&gs_nf=1&cp=4&gs_id=91&xhr=t&q=udenar&pf=p&output…

Ahora bien, si al analizar los registros del *referer* almacenados en un archivo log podemos identificar y diferenciar motores de búsqueda y sitios web referentes, es posible agrupar las peticiones de objetos o recursos solicitado en dos grupos de datos que representan el tipo de objeto referente (motor de búsqueda o sitio web). Para realizar este proceso es necesario identificar los motores de búsqueda utilizados en la web, ya que es posible determinar el motor de búsqueda utilizado en una petición en particular si se tiene una lista de los motores de búsqueda más

utilizados, esta información se muestra en la Tabla 23.

Tabla 23. Motores de búsqueda

Motor de busqueda	String
Google	google.
Altavista	.altavista.
Lycos	.lycos.
Yahoo	.yahoo.
Mamma	.mamma.
Bing	.bing.
Ask	.ask.
AOL	.aol.
HOTBOT	.hotbot.
Alexa	.alexa.
Search	.search.
Excite	.excite.

Así entonces, para determinar las solicitudes procedentes desde motores de búsqueda, es necesario comparar las cadenas que identifican a cada motor de búsqueda (por ejemplo .google.) con cada uno de los registros del archivo log donde aparece el campo referrer; si la comparación es positiva, se dirá que la solicitud procede de un motor de búsqueda, de lo contrario se dirá que es procedente de otro sitio web.

Al haber identificado motores de búsqueda y sitios web referentes, es posible generar estadísticas asociando cado uno de estos con el número de visitas, el número de hits o la cantidad de bytes transferidos.

Ahora bien, si se ha determinado que un grupo de solicitudes registradas en un archivo log, proceden de motores de búsqueda, es posible obtener información extra muy importante, esto es, las palabras clave o palabras utilizadas por el usuario en el proceso de búsqueda.

Las palabras clave o cadenas de búsqueda es lo que escribe un usuario en un buscador, que a su vez le devuelve una lista de páginas web que contienen los vocablos o frases expresadas en la cadena de búsqueda. Es importante desarrollar un proceso que permita obtener las cadenas de búsqueda que emplearon los visitantes que llegaron desde buscadores.

El motor de búsqueda (Google, Bing, Altavista, etc) transfiere esta información a al sistema de regitro de accesos. Y lo hace añadiendo un parámetro y un valor al campo REFERER en la cabecera HTTP que se envía cuando se accede a una página al hacer click en el link que aparece en la lista de resultados de la búsqueda.

El campo REFERER contiene la URI del documento desde donde siguiendo un link o enlace se accedió a una página web. En el caso de los buscadores se introduce una pequeña variante, que es lo que permite transmitir la cadena de búsqueda. Un URI combina en una dirección una serie elementos básicos de información necesarios para recuperar un recurso desde cualquier parte en la Internet:

- El protocolo de comunicación.
- El servidor con el que se comunica.
- El puerto de red en el servidor para conectarse (opcional).
- La ruta al recurso en el servidor (por ejemplo, nombre de archivo).
- La cadena de búsqueda.

Un URI típico generado por un motor de busque puede verse como:

http://www.google.es/search?q=%22Estad%A1sticas%20Web%22

Donde *Http* es el protocolo, *www.google.es* es el servidor y *?q=%22Estad%A1sticas%20Web%22* es la cadena de búsqueda.

Los motores de búsqueda transmiten las palabras clave por medio de la URI después de un identificador que generalmente es el caracter ? o el caracter & seguido de la variable de búsqueda (que varía dependiendo del buscador utilizado, ver Tabla 24), un signo igual y la cadenas de búsqueda (que puede tener caracteres codificados) finalizando con el carácter &. De este modo es posible obtener la cadena de búsqueda, extrayendo del campo referir la cadena de caracteres que se encuentra entre los caracteres mencionados. Una vez se haya obtenido esta cadena es necesario identificar los caracteres codificados y convertirlos a un formato legible por el usuario (Unicode). La información sobre caracteres y códigos se observa en la Tabla 25.

Tabla 24. Variables de búsqueda

Motor de busqueda	Variable de busqueda
Google	&q
Altavista	?p
Lycos	&query
Yahoo	?p
Mamma	&q
Bing	?q
Ask	?q
AOL	&q
HOTBOT	?q
Alexa	?q
Search	?q
Excite	?q

Tabla 25. Caracteres Unicode y codificación UTF- 8

Carácter	UTF-8	Nombre
!	21	EXCLAMATION MARK
"	22	QUOTATION MARK
#	23	NUMBER SIGN
$	24	DOLLAR SIGN
%	25	PERCENT SIGN
&	26	AMPERSAND
'	27	APOSTROPHE
(	28	LEFT PARENTHESIS
)	29	RIGHT PARENTHESIS
*	2a	ASTERISK
+	2b	PLUS SIGN
,	2c	COMMA
-	2d	HYPHEN-MINUS
.	2e	FULL STOP
/	2f	SOLIDUS
…		

A partir del campo referrer también es posible identificar el código del país del cual procede una petición. A este identificador se conoce como Dominio de Nivel Superior Geográfico, el cual identifica los dominios, basados en los dos caracteres

de identificación de cada territorio de acuerdo a las abreviaciones del ISO-3166. (Ej. *.co, *.mx) y se denomina ccTLD (Dominio de nivel superior de código de país ó Country Code Top level Domain) [47]. En Tabla 26 se observa un resumen de la lista de países y su respectivo identificador.

Tabla 26. Lista de dominios de nivel superior geográfico

ccTLD	Pais / Territorio independiente / Region
.al	Albania
.ac	Ascensiòn
.ad	Andorra
.ae	Emiratos A□rabes Unidos
.as	Samoa Americana
.at	Austria
.aw	Aruba
.az	Azerbaiyán
.be	Bélgica
.bg	Bulgaria
.bo	Bolivia
.bs	Bahamas
.by	Bielorrusia
.bz	Belize
.ca	Canadá¡
.cc	Islas Cocos
.cd	República Democrática del Congo
.cg	República del Congo
.ci	Costa de Marfil
.cl	Chile
.cm	Camerón
.cn	China
.co	Colombia
…	

1.3.4.7 Análisis de códigos de estado HTTP. Cuando un usuario accede a un sitio web, por cada petición realizada sobre un objeto o recurso, el servidor web responde con un código que informa sobre el estado de la transacción y registrando esta información en el archivo log de acceso. Al analizar los códigos de estado registrados, es posible asociar la respuesta del servidor con cada objeto solicitado o visitante del sitio web, lo que permite obtener estadísticas como el número de peticiones o visitantes relacionados con un código en particular y

determinar el comportamiento del servidor cada vez que un usuario realiza una petición.

Para este fin es necesario identificar las diferentes respuestas (códigos de estado HTTP), que un servidor puede dar cuando un cliente realice una petición. A continuación se realiza una descripción de los códigos de estado comúnmente utilizados [37]:

- Respuestas informativas (1xx): Petición recibida, continuando proceso. Esta clase de código de estatus indica una respuesta provisional, que consiste únicamente en la línea de estatus y en encabezados opcionales, y es terminada por una línea vacía. Ya que HTTP/1.0 no definía códigos de estatus 1xx, los servidores no deben enviar una respuesta 1xx a un cliente HTTP/1.0, excepto en condiciones experimentales.

- Peticiones Correctas (2xx): Esta clase de código de estado indica que la petición fue recibida correctamente, entendida y aceptada.

- Redirecciones (3xx): El cliente tiene que tomar una acción adicional para completar la petición. Esta clase de código de estado indica que una acción subsecuente necesita efectuarse por el agente de usuario para completar la petición. La acción requerida puede ser llevada a cabo por el agente de usuario sin interacción con el usuario si y sólo si el método utilizado en la segunda petición es GET o HEAD.

- Errores de cliente (4xx): La solicitud contiene sintaxis incorrecta o no puede procesarse. La intención de la clase de códigos de respuesta 4xx es para casos en los cuales el cliente parece haber errado la petición. Excepto cuando se responde a una petición HEAD, el servidor debe incluir una entidad que contenga una explicación a la situación de error, y si es una condición temporal o permanente. Estos códigos de estado son aplicables a cualquier método de solicitud (como GET o POST). Los agentes de usuario deben desplegar cualquier entidad al usuario.

- Errores de servidor (5xx): El servidor falló al completar una solicitud aparentemente válida. Los códigos de respuesta que comienzan con el dígito "5" indican casos en los cuales el servidor tiene registrado aún antes de servir la solicitud, que está errado o es incapaz de ejecutar la petición. Excepto cuando está respondiendo a un método HEAD, el servidor debe incluir una entidad que contenga una explicación de la situación de error, y si es una condición temporal o permanente. Los agentes de usuario deben desplegar cualquier entidad incluida al usuario. Estos códigos de repuesta son aplicables a cualquier método de petición.

En la Tabla 27, se observa la lista de códigos de respuesta HTTP y las frases estándar asociadas, destinadas a dar una descripción corta del código de estado.

Tabla 27. Lista de códigos de estado HTTP

Codigo	Descripcion	Tipo
100	Continuar	Informational
101	Switching Protocols	Informational
102	Processing	Informational
103	Checkpoint	Informational
122	Request-URI too long	Informational
200	OK	OK
201	Created	OK
202	Acepted	OK
203	Non-Authoritative Information (since HTTP/1.1)	OK
204	No Content	OK
226	IM Used (RFC 3229)	OK
300	Multiple Choices	Redirection
301	Moved Permanently	Redirection
302	Found	Redirection
303	See Other (since HTTP/1.1)	Redirection
304	Not Modified	Redirection
305	Use Proxy (since HTTP/1.1)	Redirection
308	Resume Incomplete	Redirection
400	Bad Request	Client Error
401	Unauthorized	Client Error
402	Payment Required	Client Error
444	No Response	Client Error
500	Internal Server Error	Server Error
501	Not Implemented	Server Error

2. ANÁLISIS DE HERRAMIENTAS

2.1 POLARIS VERSIÓN 1.0

Descripción preliminar. Polaris es una herramienta de minería Web de uso desarrollada por estudiantes pertenecientes al Grupo de Investigación GRIAS (Grupo de Investigación Aplicado a Sistemas) dirigido por el Dr. Ricardo Timarán Pereira, Ph.D. del Programa de Ingeniería de Sistemas de la Universidad de Nariño de la ciudad de Pasto.

Polaris es un software que realiza minería Web de uso a partir de un log de acceso de servidores Web: Apache y IIS Server. Polaris tiene una completa variedad de algoritmos de asociación y clasificación, y un algoritmo de minería de uso llamado HPG (Gramática Probabilística de Hipertexto), también tiene siete formatos de visualización que ayuda a entender todo el proceso de minería.[40]

Requisitos del sistema. Para el correcto funcionamiento de la herramienta se necesita tener instalado el siguiente software:

- Postgres 8.2 o superior.
- La Maquina Virtual de Java jre o superior.
- Componente de Java3D 1.5.0 o superior.
- Usuario del SGBD Postgres con permiso de creación de base de datos.
- Instalador de Polaris v1.0

Ventajas.

Tras realizar un análisis de la herramienta Polaris Versión 1.0 se observan las siguientes ventajas:

- Es una herramienta libre.
- Requerimientos mínimos de software y hardware.
- Genera reportes gráficos.
- Trabaja sobre sistemas operativos Windows y Linux.
- Posee una interfaz gráfica amigable.
- Únicamente soporta Minería Web de Uso.
- Permite el descubrimiento de patrones.

Desventajas.

Las desventajas observadas durante el análisis de la herramienta Polaris Versión

1.0 son las siguientes:

- Únicamente soporta Minería Web de Uso.
- No genera informes estadísticos.

2.2 ANALOG

Descripción Preliminar. Analog es un completo analizador de logs para servidores apache e ISS. Genera automáticamente una página web XHTML con el resultado del análisis, que puede configurarse con todo detalle. La documentación está en inglés, pero el programa crea informes en más de 30 idiomas, incluido el castellano.

El programa no contiene una interfaz gráfica o GUI como suele llamarse, si no que consta de un archivo ejecutable (analog.exe), y este funciona desde la línea de comandos mediante el paso de parámetros para configurarlo y poder crear los reportes y listados con la información. [3]

Requerimientos.

Requerimientos de Hardware:

- Mínimo de memoria RAM libre de 32 MB.
- Procesador Pentium-100 MHz o superior.
- 4 MB de espacio libre en disco.

Requerimientos de Software

- Analog está disponible para Windows 95/98/ME/NT/2000/XP.

Ventajas.

El análisis de la herramienta Analog perite determinar las siguientes ventajas:

- La principal ventaja de Analog es que es totalmente Open Source eso posibilita su estudio y modificación.
- Analog es operable en diversas plataformas tales como: Windows, Linux, Mac.
- Los informes de resultados son ofrecidos hasta en 33 idiomas distintos, incluyendo al español.

Desventajas.

Después de analizar la herramienta Analog, se observan las siguientes

desventajas:

- No posee interfaz gráfica lo que hace que – ya sea en Linux o Windows – se maneje solo a través de comandos y siempre debe modificarse su archivo de configuración para hacer un cambio de log o en el reporte de salida, es por eso que se recomienza iniciar su uso con mínimos cambios de configuración y poco a poco ir avanzando hasta lograr los reportes al susto del usuario.
- El reporte es sencillo y estadístico.

2.3 AWSTATS

Descripción Preliminar. AWStats es un programa que genera estadísticas gráficas para servidores web. Lo que hace es mostrar el contenido del archivo de log del servidor web de forma gráfica. Entre las cosas que muestra están el número de visitas, navegadores usados, sistemas operativos. Es capaz de analizar archivos de registro de todas las herramientas de servidores web importantes como Apache (NCSA combinado / XLF / ELF o formato de registro común / CLF Formato de registro), WebStar, IIS (formato de registro W3C) y otros como: servidores proxy, servidores wap, servidores de correo y algunos servidores de ftp. [4]

Requerimientos.

Requerimientos de Hardware

- Mínimo de memoria RAM libre de 32 MB.
- Procesador Pentium-100 MHz o superior.
- 4 MB de espacio libre en disco.

Requerimientos de Software

- Descargar ActivePerl 5.12 para instalación en Windows.
- AWStats está disponible para la mayoría de los sistemas operativos (SO) y computadoras: Unix, Windows 95, 98 o NT, o MacOS.

Ventajas.

A continuación se observan las ventajas obtenidas como resultado del análisis de la herramienta AWStats.

- Genera atractivos informes en HTML, de forma relativamente más clara y entendible, con la ayuda de barras y gráficas en 3D. El programa usa el navegador para conseguir un atractivo entorno, que sea lo más consistente

posible a través de muchos sistemas operativos.

- AWStats es uno de los analizadores de archivos log para estadísticas web avanzadas más populares del mundo. Es capaz de generar informes rápidos y exactos sobre un sitio web ofreciendo valiosa información sobre sus visitas. Está disponible en 6 idiomas entre los que se encuentra el español.
- Brinda un buen diseño y comprensión de los reportes y de las páginas.

Desventajas.

El análisis de la herramienta AWStats permite determinar las siguientes desventajas:

- El AWStats no realiza procesos de minería de datos con algoritmos de asociación, clasificación, no realiza procesos con el algoritmo de minería de uso HPG (Gramática Probabilística de Hipertexto), por lo tanto no implementa ninguna de sus técnicas.
- AWStats está escrito en el lenguaje de programación Perl; por tal razón se necesita tener instalado el programa anteriormente dicho para poder ejecutar sin problemas el analizador de logs.

2.4 ALTERWIND LOG ANALYZER

Descripción preliminar. AlterWind Log Analyzer Lite es una herramienta para analizar el tráfico que recibe un sitio web. Lo hace a partir de los logs que crea el servidor.

AlterWind Log Analyzer Lite muestra un detallado informe, en el que aparecen: páginas visitadas, imágenes y recursos descargados, hosts y países desde los que se han recibido las visitas, sitios de referencia, motores de búsqueda desde los que procede la visita, frases que se han escrito en los buscadores, navegadores y sistemas operativos utilizados por los visitantes, spiders que visitan constantemente un sitio, y mucho más.

AlterWind Log Analyzer Lite soporta la mayoría de los formatos de archivo de log, interpretando también archivos comprimidos. Los informes pueden ser presentados en diferentes temas y son muy intuitivos. [2]

Requerimientos.

- Sistema operativo Windows 98/Me/NT/2000/XP/2003.

Ventajas.

Tras realizar un análisis de la herramienta Alterwind Log Analizer se observan las siguientes ventajas:

- Es capaz de generar informes rápidos y exactos sobre un sitio web ofreciendo información sobre sus visitas.
- Los datos son presentados visualmente en informes de tablas.
- Soporta el modo de línea de comandos.
- Posibilidad de cambiar el diseño de informes.
- Detecta automáticamente los archivos de registro en cualquier formato: analizador de registro de Apache, IIS analizador de logs, etc.
- Personalización completa de los informes. Se puede cambiar el diseño de informes, personalizar los datos que entra, y ajustar el volumen de los datos del informe.
- El informe es sencillo y estadístico. Los informes de resultados son ofrecidos en idioma ingles, español, portugués, etc.

Desventajas.

Las desventajas observadas en el análisis de la herramienta Alterwind Log Analizer son las siguientes:

- El Alterwind log Analizer no realiza procesos de minería de Datos Data Mining- por lo tanto no implementa ninguna de sus técnicas.
- Solo funciona en plataformas Windows.
- Información sobre agentes de usuario limitada.

2.5 WEBLOG EXPERT

Descripción Preliminar. WebLog Expert Lite, es un analizador de registro de acceso rápido y potente. Brinda información sobre los visitantes de un sitio: estadísticas de actividad, los archivos de estadística de acceso, caminos a través del sitio, información sobre páginas de referencia, motores de búsqueda, navegadores, sistemas operativos, errores y mucho más. El programa produce informes HTML fáciles de leer ya que incluyen tanto información de texto (tablas) como gráficos. [45]

Requerimientos.

Requerimientos de Hardware

- Mínimo de memoria RAM libre de 32MB.

- Procesador Pentium-100 MHz o superior.
- 4 MB de espacio libre en disco.

Requerimientos de Software

- WebLog Expert Lite está disponible para Windows 95/98/ME/NT/2000/XP.

Ventajas.

- Detecta automáticamente el formato de registro.
- WebLog Expert Lite es compatible con archivos de registro de servidores Apache (común y combinado) e IIS y puede leer logs comprimidos en GZ y ZIP.
- Pueden analizar los registros de carga de servidores de equilibrio.
- Se puede crear informes en HTML, PDF y CSV.
- Admite la traducción de los reportes a otros idiomas.
- El programa genera informes basados en HTML con tablas y gráficos.
- La versión gratuita no obstante debe ser suficiente para la mayoría de los sitios web personales y proporciona una buena visión general de la actividad del tráfico en general.

Desventajas.

- WebLog Expert en la versión Lite es libre de uso, sin embargo, sólo provee una cantidad limitada de los informes.
- No presenta informes sobre robots o spiders.

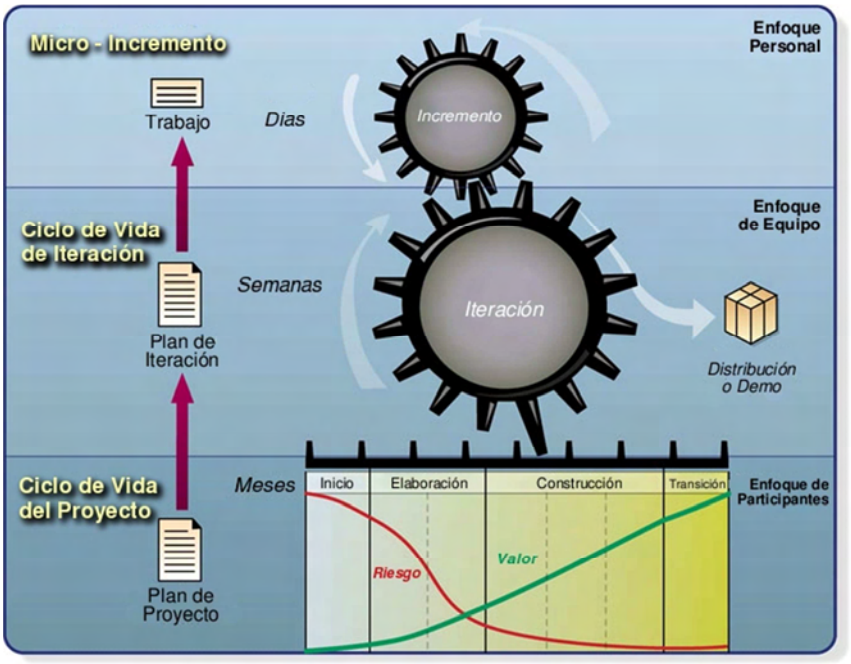
3. METODOLOGÍA Y HERRAMIENTAS DE DESARROLLO

3.1 METODOLOGÍA DE DESARROLLO

3.1.1 Metodologia OpenUP. OpenUP es un método y un proceso de desarrollo de software propuesto por un conjunto de empresas de tecnología, quienes lo donaron en el año 2007 a la Fundación Eclipse. La fundación lo ha publicado bajo una licencia libre y lo mantiene como método de ejemplo dentro del proyecto *Eclipse Process Framework*.

OpenUP es una metodología apoyada en el Proceso Unificado que aplica un enfoque iterativo e incremental dentro de un ciclo de vida estructurado. El *OpenUP* es un proceso mínimo y suficiente, lo que significa que solo el contenido fundamental y necesario es incluido. Por lo tanto no provee lineamientos para todos los elementos que se manejan en un proyecto pero tiene los componentes básicos que pueden servir de base a procesos específicos. La mayoría de los elementos de OpenUP están declarados para fomentar el intercambio de información entre los equipos de desarrollo y mantener un entendimiento compartido del proyecto, sus objetivos, alcance y avances [31]. En la Figura 13, se observa la descripción del proceso OpenUp.

Figura 13. Fases de la metodología OpenUP



Principios del OpenUP

- Colaborar para sincronizar intereses y compartir conocimiento. Este principio promueve prácticas que impulsan un ambiente de equipo saludable, facilitan la colaboración y desarrollan un conocimiento compartido del proyecto.

- Equilibrar las prioridades para maximizar el beneficio obtenido por los interesados en el proyecto. Este principio promueve prácticas que permiten a los participantes de los proyectos desarrollar una solución que maximice los beneficios obtenidos por los participantes y que cumple con los requisitos y restricciones del proyecto.

- Centrarse en la arquitectura de forma temprana para minimizar el riesgo y organizar el desarrollo.

- Desarrollo evolutivo para obtener retroalimentación y mejoramiento continuo. Este principio promueve prácticas que permiten a los equipos de desarrollo obtener retroalimentación temprana y continua de los participantes del proyecto, permitiendo demostrarles incrementos progresivos en la funcionalidad.

Fases del OpenUP

Concepción. Primera de las 4 fases en el proyecto del ciclo de vida, acerca del entendimiento del propósito y objetivos y obteniendo suficiente información para confirmar que el proyecto debe hacer. El objetivo de ésta fase es capturar las necesidades de los stakeholder en los objetivos del ciclo de vida para el proyecto.

Elaboración. Es el segundo de las 4 fases del ciclo de vida del OpenUP donde se trata los riesgos significativos para la arquitectura. El propósito de esta fase es establecer la base la elaboración de la arquitectura del sistema.

Construcción. Esta fase está enfocada al diseño, implementación y prueba de las funcionalidades para desarrollar un sistema completo. El propósito de esta fase es completar el desarrollo del sistema basado en la Arquitectura definida.

Transición. Es la última fase, cuyo propósito es asegurar que el sistema es entregado a los usuarios, y evalúa la funcionalidad y performance del último entregable de la fase de construcción.

3. 2 HERRAMIENTAS DE DESARROLLO

3.2.1 Lenguaje de programación Java. Java es un lenguaje de programación orientado a objetos desarrollado por James Gosling y sus compañeros de Sun Microsystems al inicio de la década de 1990. A diferencia de los lenguajes de programación convencionales, que generalmente están diseñados para ser compilados a código nativo, Java es compilado en un bytecode que es ejecutado (usando normalmente un compilador JIT), por una maquina virtual Java. El lenguaje Java se crea con cinco objetivos principales:

- Usar la metodología de la programación orientada a objetos.
- Permitir la ejecución de un mismo programa en múltiples sistemas operativos.
- Incluir por defecto soporte para trabajo en red.
- Ejecutar código en sistemas remotos de forma segura.
- Ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

3.2.1.1 Características principales de Java:

- Orientado a Objetos: La primera característica, orientado a objetos (”OO”), se refiere a un método de programación y al diseño del lenguaje. Aunque hay muchas interpretaciones para OO, una primera idea es diseñar el software de forma que los distintos tipos de datos que use están unidos a sus operaciones. Así los datos y el código (funciones o métodos) se combinan en entidades llamadas objetos. Un objeto puede verse como un paquete que contiene el “comportamiento” (el código) y el “estado” (datos).

- El principio es separar aquello que cambia de las cosas que permanecen inalterables. Frecuentemente, cambiar una estructura de datos implica un cambio en el código que opera sobre los mismos, o viceversa. Esta separación en objetos coherentes e independientes ofrece una base más estable para el diseño de un sistema software. El objetivo es hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando como consecuencia su calidad y reduciendo el número de proyectos fallidos. Otra de las grandes promesas de la programación orientada a objetos es la creación de entidades más Genéricas (objetos) que permitan la reutilización del software entre proyectos, una de las premisas fundamentales de la Ingeniería del Software. Un objeto genérico ”cliente”, por ejemplo, Deberá en teoría tener el mismo conjunto de comportamiento en diferentes proyectos, sobre todo cuando estos coinciden en cierta medida, algo que suele suceder en las grandes organizaciones. En este sentido, los objetos Podría verse como piezas reutilizables que pueden

emplearse en múltiples proyectos distintos, posibilitando así a la industria del software a construir proyectos de envergadura empleando componentes ya existentes y de comprobada calidad; conduciendo esto finalmente a una reducción drástica del tiempo de desarrollo. Podemos usar como ejemplo de objeto el aluminio. Una vez definidos datos (peso, maleabilidad, etc.), y su "comportamiento" (soldar dos piezas, etc.), el objeto "aluminio" puede ser reutilizado en el campo de la construcción, del automóvil, de la aviación, etc.

- Independencia de la plataforma: La segunda característica, la independencia de la plataforma, significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Es lo que significa ser capaz de escribir un programa una vez y que pueda ejecutarse en cualquier dispositivo, tal como reza el axioma de Java, "write once, run everywhere". Para ello, se compila el código fuente escrito en lenguaje Java, para generar un código conocido como "bytecode" (específicamente Java bytecode) que son instrucciones maquina simplificadas específicas de la plataforma Java. Esta pieza esta "a medio camino" entre el código fuente y el código maquina que entiende el dispositivo destino. El bytecode es ejecutado entonces en la maquina virtual (VM), un programa escrito en código nativo de la plataforma destino (que es el que entiende su hardware), que interpreta y ejecuta el código. Además, se suministran bibliotecas adicionales para acceder a las características de cada dispositivo (como los gráficos, ejecución mediante hebras o threads, la interfaz de red) de forma unificada. Se debe tener presente que, aunque hay una etapa explicita de compilación, el bytecode generado es interpretado o convertido a instrucciones maquina del código nativo por el compilador JIT (Just In Time).

- El recolector de basura: Un argumento en contra de lenguajes como C++ es que los programadores se encuentran con la carga añadida de tener que administrar la memoria de forma manual. En C++, el desarrollador debe asignar memoria en una zona conocida como heap (monticulo) para crear cualquier objeto, y posteriormente desalojar el espacio asignado cuando desea borrarlo. Un olvido a la hora de desalojar memoria previamente solicitada, o si no lo hace en el instante oportuno, puede llevar a una fuga de memoria, ya que el sistema operativo piensa que esa zona de memoria está siendo usada por una aplicación cuando en realidad no es así un programa mal diseñado Podría consumir una cantidad desproporcionada de memoria. Además, si una misma región de memoria es desalojada dos veces el programa puede volverse inestable.

En Java, este problema potencial es evitado en gran medida por el recolector automático de basura (o automatic garbage collector). El programador determina cuando se crean los objetos y el entorno en tiempo de ejecución de Java (Java runtime) es el responsable de gestionar el ciclo de vida de los objetos. El

programa, u otros objetos pueden tener localizado un objeto mediante una referencia a este (que, desde un punto de vista de bajo nivel es una dirección de memoria). Cuando no quedan referencias a un objeto, el recolector de basura de Java borra el objeto, liberando así la memoria que ocupaba previniendo posibles fugas (ejemplo: un objeto creado y únicamente usado dentro de un método solo tiene entidad dentro de este; al salir del método el objeto es eliminado), aun así es posible que se produzcan fugas de memoria si el código almacena referencias a objetos que ya no son necesarios es decir, pueden aun ocurrir, pero en un nivel conceptual superior. En definitiva, el recolector de basura de Java permite una fácil creación y eliminación de objetos, mayor seguridad y frecuentemente más rápida que en C++.

La recolección de basura de Java es un proceso prácticamente invisible al desarrollador. Es decir, el programador no tiene conciencia de cuando la recolección de basura tendrá lugar, ya que esta no tiene necesariamente que guardar relación con las acciones que realiza el código fuente. Debe tenerse en cuenta que la memoria es solo uno de los muchos recursos que deben ser gestionados.

3.2.2 Entorno de desarrollo Netbeans. El entorno en el que se desarrollo POLARIS Versión 3.0 fue Neatbeans 7.0.1. NetBeans es un proyecto exitoso de código abierto con una gran base de usuarios, una comunidad en constante crecimiento. Sun MicroSystems fundó el proyecto de código abierto NetBeans en junio 2000 y continúa siendo el patrocinador principal de los proyectos.

Al día de hoy hay disponibles dos productos: el NetBeans IDE y NetBeans Platform.

NetBeans IDE es un entorno de desarrollo integrado, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

También está disponible NetBeans Platform; una base modular y extensible usada como estructura de integración para crear grandes aplicaciones de escritorio. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. Entre las características de la plataforma están:

- Administración de las interfaces de usuario (ej. menús y barras de herramientas).
- Administración de las configuraciones del usuario.
- Administración del almacenamiento (guardando y cargando cualquier tipo de dato)
- Administración de ventanas.
- Framework basado en asistentes (diálogos paso a paso).

Ambos productos son de código abierto y gratuito para uso tanto comercial como no comercial. El código fuente está disponible para su reutilización de acuerdo con la Common Development and Distribution License (CDDL) v1.0 and the GNU General Public License (GPL) v2.

3.2.3 Postgresql. PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una sola empresa sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

3.2.3.1 Características. Algunas de sus principales características son, entre otras:

Alta concurrencia: Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

Amplia variedad de tipos nativos: PostgreSQL provee nativamente soporte para:

- Números de precisión arbitraria.
- Texto de largo ilimitado.
- Figuras geométricas (con una variedad de funciones asociadas)
- Direcciones IP (IPv4 e IPv6).
- Bloques de direcciones estilo CIDR.
- Direcciones MAC.
- Arrays.

Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. Algunos ejemplos son los tipos de datos GIS creados por el proyecto PostGIS.

Foreign Keys: Claves ajenas también denominadas Llaves ajenas o Claves Foráneas.

Disparadores (triggers): Un disparador o trigger se define en una acción específica basada en algo ocurrente dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica. Ahora todos los disparadores se definen por seis características:

- El nombre del disparador o trigger
- El momento en que el disparador debe arrancar
- El evento del disparador deberá activarse sobre...
- La tabla donde el disparador se activará
- La frecuencia de la ejecución
- La función que podría ser llamada

Entonces combinando estas seis características, PostgreSQL le permitirá crear una amplia funcionalidad a través de su sistema de activación de disparadores (triggers).

Soporte para transacciones distribuidas: Permite a PostgreSQL integrase en un sistema distribuido formado por varios recursos (ejemplo una base de datos PostgreSQL, otra Oracle, una cola de mensajes IBM MQ JMS y un ERP SAP) gestionado por un servidor de aplicaciones donde el éxito ("commit") de la transacción global es el resultado del éxito de las transacciones locales.

Funciones: Bloques de código que se ejecutan en el servidor. Pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos da, desde las operaciones básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientada a objetos o la programación funcional.

Los disparadores (triggers en inglés) son funciones enlazadas a operaciones sobre los datos.
PostgreSQL soporta funciones que retornan "filas", donde la salida puede tratarse como un conjunto de valores que pueden ser tratados igual a una fila retornada por una consulta (query en inglés).

Las funciones pueden ser definidas para ejecutarse con los derechos del usuario

ejecutor o con los derechos de un usuario previamente definido. El concepto de funciones, en otros DBMS, son muchas veces referidas como "procedimientos almacenados" (stored procedures en inglés).

3.2.4 Controlador Jdbc. JDBC es el acrónimo de Java Database Connectivity, un API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java independientemente del sistema de operación donde se ejecute o de la base de datos a la cual se accede utilizando el dialecto SQL del modelo de base de datos que se utilice.

El API JDBC se presenta como una colección de interfaces Java y métodos de gestión de manejadores de conexión hacia cada modelo especifico de base de datos. Un manejador de conexiones hacia un modelo de base de datos en particular es un conjunto de clases que implementan las interfaces Java y que utilizan los métodos de registro para declarar los tipos de localizadores a base de datos (URL) que pueden manejar. Para utilizar una base de datos particular, el usuario ejecuta su programa junto con la biblioteca de conexión apropiada al modelo de su base de datos, y accede a ella estableciendo una conexión, para ello provee en localizador a la base de datos y los parámetros de conexión específicos. A partir de allí puede realizar cualquier tipo de tareas con la base de datos a las que tenga permiso: consultas, actualizaciones, creado modificado y borrado de tablas, ejecución de procedimientos almacenados en la base de datos, etc. Cada base de datos emplea un protocolo diferente de comunicación, protocolos que normalmente son propietarios. El uso de un manejador, una capa intermedia entre el código del desarrollador y la base de datos, permite independizar el código Java que accede a la BD del sistema de BD concreto a la que estamos accediendo, ya que en nuestro código Java emplearemos comandos estándar, y estos comandos serian traducidos por el manejador a comandos propietarios de cada sistema de BD concreto. Si queremos cambiar el sistema de BD que empleamos lo único que deberemos hacer es reemplazar el antiguo manejador por el nuevo, y seremos capaces de conectarnos la nueva BD.

3.2.5 Biblioteca Jfreechart. FreeChart es un marco de software open source para el lenguaje de programación Java, el cual permite la creación de gráficos complejos de forma simple.

JFreeChart es compatible con una serie de gráficas diferentes, incluyendo cuadros combinados. Los tipos de gráficos compatibles son:

- Gráficos XY (línea, spline y dispersión). Es posible usar un eje del tiempo.
- Gráfico circular.
- Diagrama de Gantt.

- Gráficos de barras (horizontal y vertical, apiladas e independientes)
- Histogramas.
- Varias gráficas específicas (tabla de viento, gráfica polar, burbujas de diferentes tamaños, etc.)

Además los gráficos, es posible colocar varios marcadores en el área de gráfica.

JFreeChart dibuja automáticamente las escalas de los ejes y leyendas. Con el ratón informático se puede hacer zoom en la interfaz de la gráfica automáticamente y cambiar algunos ajustes a través del menú local. Las tablas existentes pueden actualizarse fácilmente a través de los oyentes (listeners) que la biblioteca tiene en sus colecciones de datos.

4. ANÁLISIS Y DISEÑO DEL MÓDULO DE ANÁLISIS ESTADÍSTICO DE TRÁFICO WEB

En el presente proyecto se aplicó la metodología OpenUP teniendo en cuenta que sus características para el desarrollo de software son compatibles con las especificaciones propias de este. EL **MÓDULO DE ANÁLISIS ESTADÍSTICO DE TRÁFICO WEB DE LA HERRAMIENTA POLARIS VERSIÓN 3.0** es un proyecto a desarrollarse en un periodo de tiempo relativamente corto y cuenta con un solo desarrollador, además, se trabaja en equipo con el cliente como elemento esencial para el planteamiento de los requerimientos y el éxito del proyecto.

En esta sección se detallan todos los resultados obtenidos al aplicar las fases de la metodología, brindándole al lector una visión general del trabajo realizado y del proceso de construcción que se utilizó para el desarrollo del **MÓDULO DE ANÁLISIS ESTADÍSTICO DE TRÁFICO WEB**.

Es necesario recordar que, los procesos de cada una de las fases de la metodología, no son pasos estrictos que deben darse uno tras otro. Más bien, se tratan de recomendaciones que se relacionan entre sí para la comprensión total de la fase en cuestión.

Debido a eso, en lugar de exponer en detalle los procesos de cada fase, se mencionarán los resultados de los procesos más significativos de éstas, pero haciendo alusión a los demás sub puntos inmersos en la etapa en cuestión. Lo que permitirá ser más específicos y poco redundantes, en la redacción de los resultados de la investigación.

Para la construcción del **MÓDULO DE ANÁLISIS ESTADÍSTICO DE TRÁFICO WEB** se desarrollaron: un módulo de utilidades para realizar la conexión con la base de datos y lectura de archivos log, un módulo de Kernel para la creación de estadísticas y gráficos y un módulo de interfaz gráfica que permita interactuar de forma amigable con la herramienta.

A continuación puede apreciarse la descripción formal del desarrollo de este módulo.

4.1 REQUERIMIENTOS DEL SISTEMA

En la Tabla 28, se describen los requerimientos o funciones del módulo de análisis estadístico de tráfico web.

Tabla 28. Requerimientos del sistema

Ref #	Función	Categoría
R1.1	Abrir archivo log de acceso	Evidente
R1.2	Identificar el formato del archivo log	Evidente
R1.3	Cargar los registros del archivo log a una base de datos	Oculta
R2.1	Filtrar registros poco confiables	Superflua
R2.2	Identificar sesiones de usuario	Evidente
R3.1	Generar resumen general de estadísticas	Evidente
R3.2	Generar estadísticas por fecha	Evidente
R3.3	Generar estadísticas por día de la semana	Evidente
R3.4	Generar estadísticas por hora del día	Evidente
R3.5	Generar estadísticas de archivos solicitados	Evidente
R3.6	Generar estadísticas de direcciones IP	Evidente
R3.7	Generar estadísticas de códigos de estado HTTP	Evidente
R3.8	Generar estadísticas de agentes de usuario	Evidente
R3.9	Generar estadísticas de sistemas operativos	Evidente
R3.10	Generar estadísticas de navegadores web	Evidente
R3.11	Generar estadísticas de Crawlers, robots o spiders	Evidente
R3.12	Generar estadísticas de sitios web referentes	Evidente
R3.13	Generar estadísticas de motores de búsqueda	Evidente
R3.14	Generar estadísticas de palabras y cadenas de búsqueda	Evidente
R3.15	Generar estadísticas de dominios geográficos	Evidente
R4.1	Visualizar resultados por medio de gráficos de barras	Evidente
R4.2	Visualizar resultados por medio de gráficos de líneas	Evidente
R4.3	Visualizar resultados por medio de gráficos de circulares	Evidente
R4.4	Visualizar resultados por medio de tablas	Evidente

4.2 DEFINICIÓN EXTENDIDA DEL CASO DE USO GESTIONAR FUENTE DE DATOS

En la Tabla 29, se describe las características del caso de uso gestionar fuente de datos.

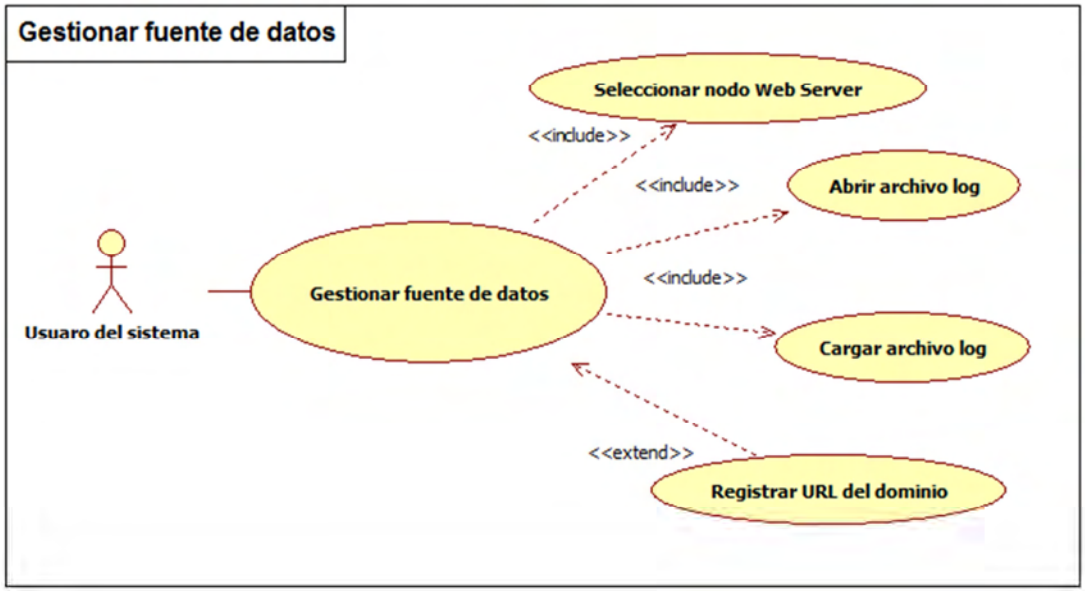
Tabla 29. Definición del caso de uso gestionar fuente de datos

Id Caso de Uso	**CU -001**
Titulo	**Gestionar fuente de datos.**
Objetivo	Permitir al usuario cargar los registros del archivo log que será analizado.
Resumen	El sistema permitirá seleccionar una archivo log de accesos y posteriormente cargar los registros de este archivo en la base de datos del sistema. Adicionalmente el usuario podrá registrar la URL del dominio al cual pertenece el archivo log.
Actores	Act-01 Usuario del sistema
Requisitos	• Abrir archivo log de acceso. • Identificar el formato del archivo log. • Cargar los registros del archivo log a una base de datos.
Precondiciones	
Postcondiciones	Identificar sesiones de usuario.
Flujo de Eventos	**Flujo Básico** El caso de uso inicia cuando el usuario selecciona un nodo Web Server: **1.** El usuario despliega el menú del nodo Web Server **2.** El usuario selecciona la opción Open File. **3.** El usuario selecciona el archivo log desde el sistema de archivos. **4.** El usuario ingresa la URL del dominio al cual pertenece el archivo log. **5.** El usuario despliega el menú del nodo Web Server. **6.** El usuario selecciona la opción Run. **7.** El sistema carga los registros del archivo log en una tabla de la base de datos. **Flujo de Excepción** **1.** 1. Paso 6: Si se presenta un error en el proceso de carga de registros, el sistema el sistema permite reintentar, omitir o cancelar el proceso

Diagrama de Caso de Uso.

En la Figura 14, se observa el diagrama del caso de uso gestionar fuente de datos.

Figura 14. Diagrama de caso de uso gestionar fuente de datos



Prototipo de interfaz gráfica.

En la Figura 15, Figura 16, Figura 17 y Figura 18 se observan los prototipos de la interfaz gráfica referente al caso de uso gestionar fuente de datos:

Figura 15. Prototipo de interfaz gráfica - Menú de un nodo Web Server

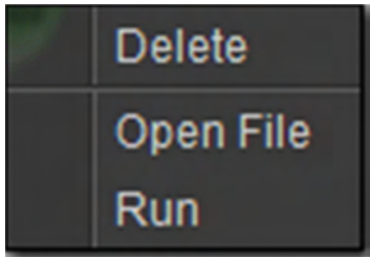


Figura 16. Prototipo de interfaz gráfica – abrir archivo log

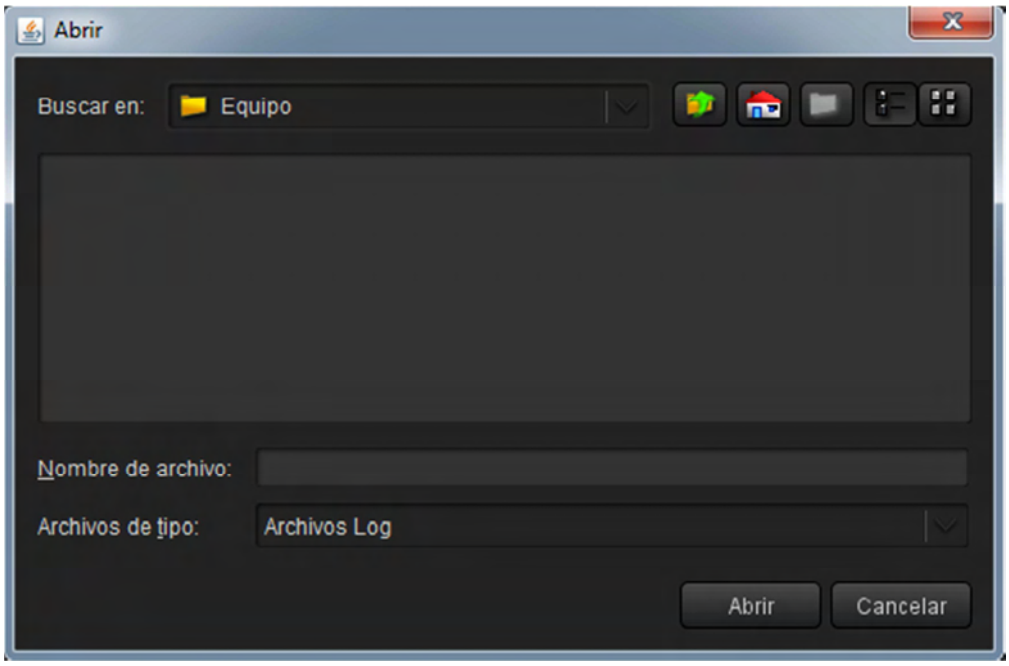


Figura 17. Prototipo de interfaz gráfica – cargar archivo log

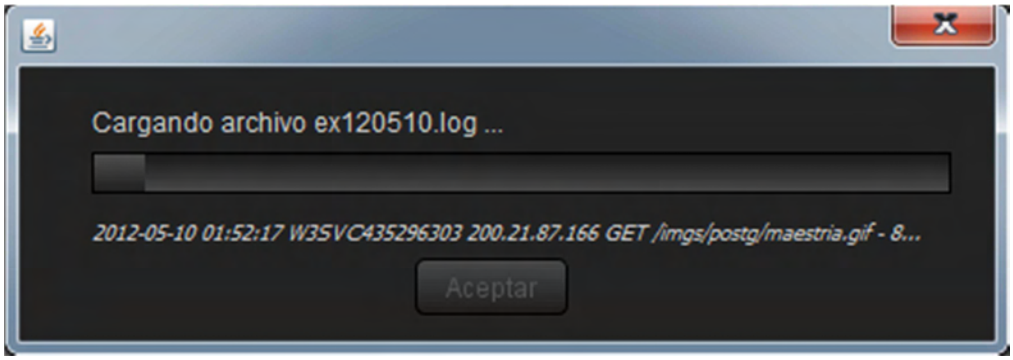


Figura 18. Prototipo de interfaz gráfica – Registrar URL del dominio

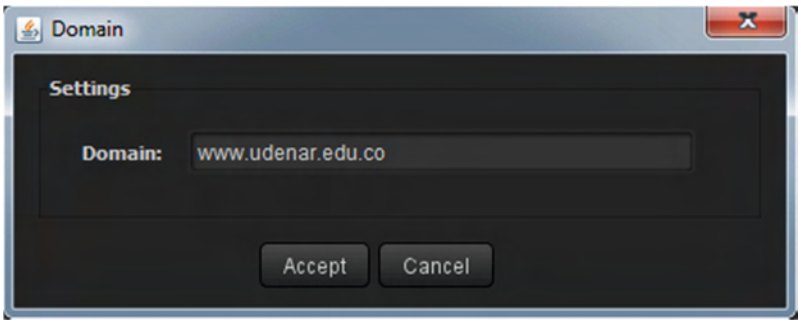
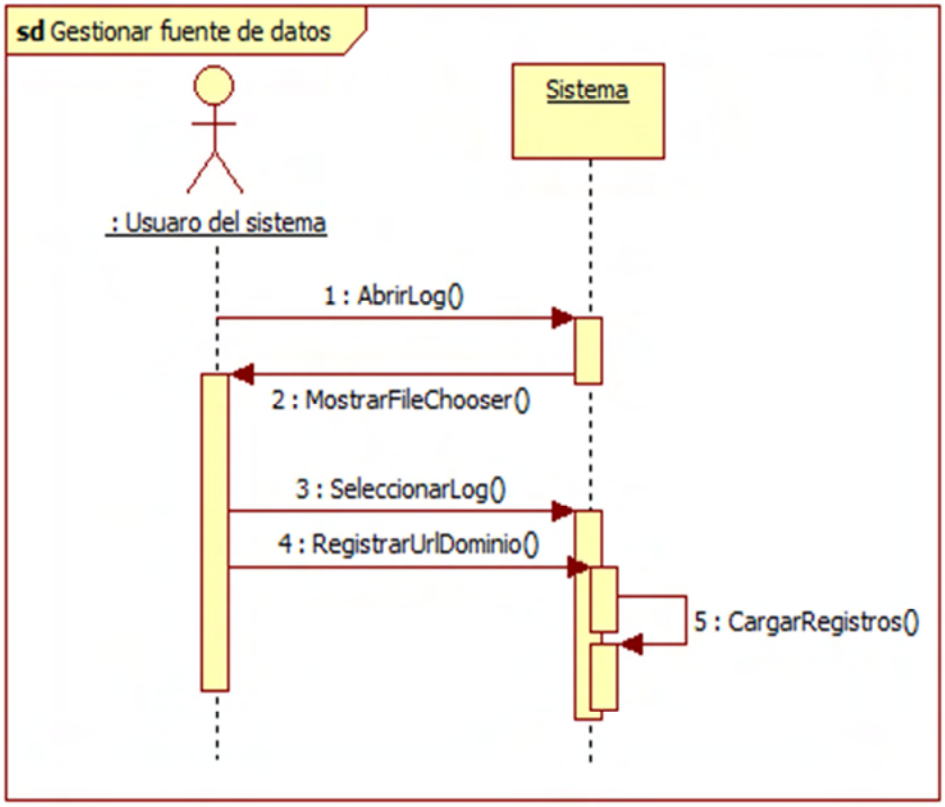


Diagrama de Secuencia.

En la Figura 19, se observa el diagrama de operaciones referente al caso de uso Gestionar Fuente de Datos.

Figura 19. Diagrama de secuencia - gestionar fuente de datos



Casos de pruebas.

En la Tabla 30, se observa la descripción de escenarios de pruebas para el caso de uso Gestionar Fuente de Datos.

Tabla 30. Escenarios de pruebas para el caso de uso gestionar fuente de datos

Código de Escenario	Escenario	Flujo Comienzo	Flujo Alternativo
E1	Selección de archivo log	El usuario selecciona la opción Open File.	
E2	Registro de URL del dominio	El usuario ingresa la URL del dominio.	
E3	Captura de registros	El usuario selecciona la opción Run.	El sistema permite reintentar, omitir o cancelar el proceso.

En la Tabla 31, se observa la descripción de los casos de prueba para el caso de uso Gestionar Fuente de Datos.

Tabla 31. Matriz de casos de prueba para el caso de uso gestionar fuente de datos

Caso de Prueba	Escenario	Condición	Resultado
C1	E1	• Desplegar el menú del nodo Web Server. • Seleccionar la opción Open File.	Ventana de acceso al sistema de archivos local.
C2	E2	• Seleccionar archivo log. • URL del dominio.	Formulario de registro de URL de dominio.
C3	E3	• Desplegar el menú del nodo Web Server. • Seleccionar la opción Run.	Mensaje en pantalla informando el estado del proceso.

4.3 DEFINICIÓN EXTENDIDA DEL CASO DE USO GESTIONAR GENERADOR DE SESIONES

En la Tabla 32, se describe las características del caso de uso gestionar generador de sesiones.

Tabla 32. Definición del caso de uso gestionar generador de sesiones

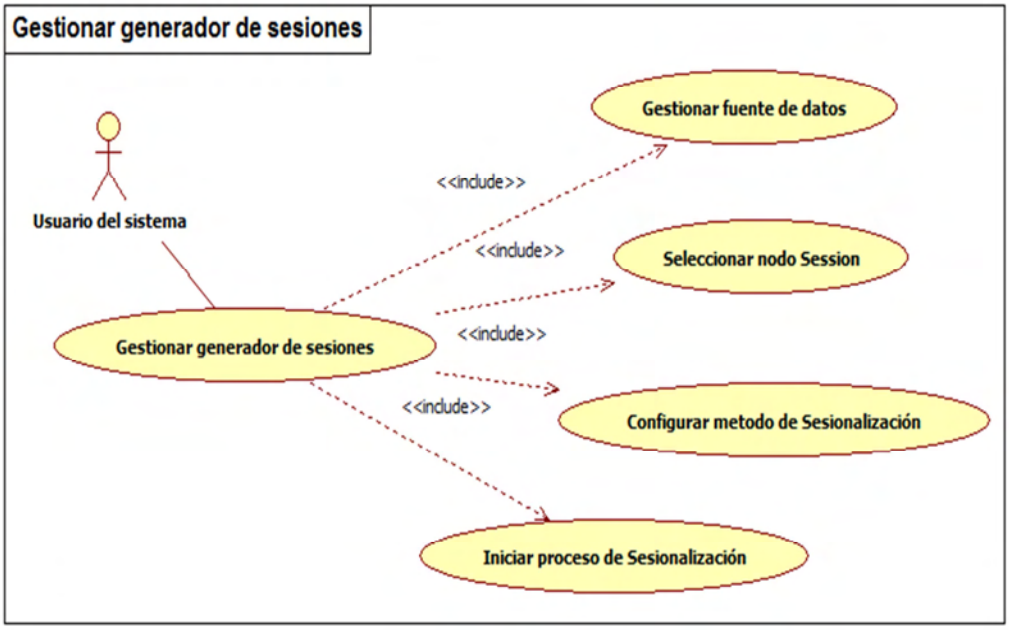
Id Caso de Uso	**CU -002**
Titulo	**Gestionar generador de sesiones.**
Objetivo	Permitir al usuario realizar un proceso de identificación de sesiones de usuario.
Resumen	El sistema permitirá al usuario configurar el método que se usara para generar sesiones de usuario y posteriormente iniciar el proceso de identificación de sesiones.
Actores	Act-01 Usuario del sistema
Requisitos	Identificar sesiones de usuario.

Precondiciones	Cargar un archivo log en la base de datos.
Postcondiciones	Generar estadísticas de tráfico web.
Flujo de Eventos	**Flujo Básico** El caso de uso inicia cuando el usuario selecciona un nodo Session: **1.** El usuario despliega el menú del nodo Session **2.** El usuario selecciona la opción Settings. **3.** El usuario selecciona el método de identificación de sesiones. **4.** El usuario despliega el menú del nodo Session. **5.** El usuario selecciona la opción Run. **6.** El sistema identifica sesiones de usuario y almacena el resultado en una tabla de la base de datos. **Flujo de Excepción** **1.** Paso 6: Si se presenta un error en la conexión con la base de datos, el sistema permite reintentar, cancelar el proceso.

Diagrama de Caso de Uso.

En la Figura 20, se observa el diagrama del caso de uso gestionar generador de sesiones.

Figura 20. Diagrama de caso de uso gestionar generador de sesiones



Prototipo de interfaz gráfica.

En Figura 21 y Figura 22, se observan los prototipos de la interfaz gráfica referente al caso de uso gestionar generador de sesiones.

Figura 21. Prototipo de interfaz gráfica - menú de un nodo session

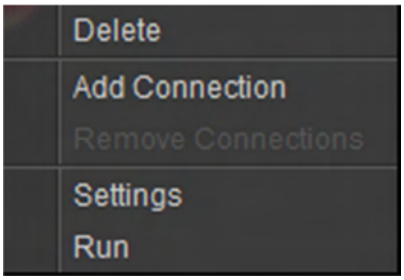
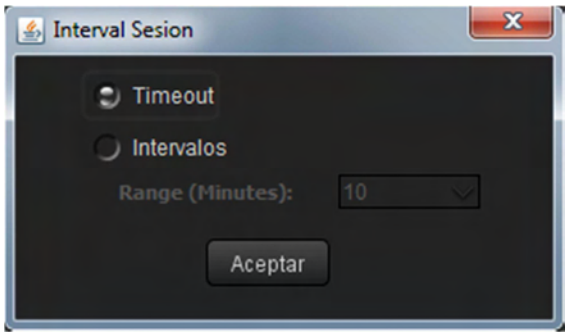


Figura 22. Prototipo de interfaz gráfica – configurar nodo Session

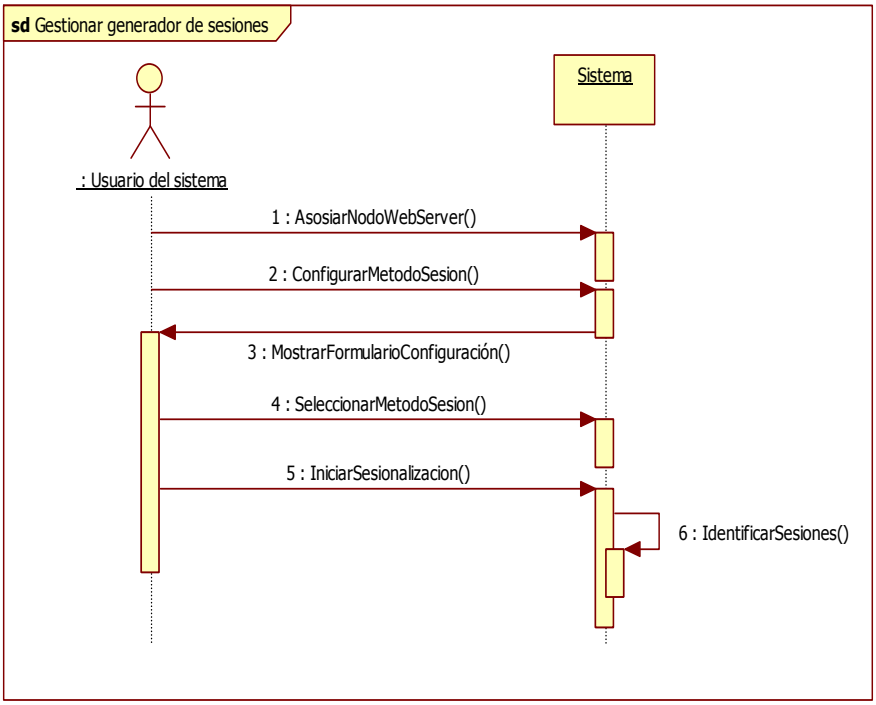


Fuente: La presente investigación

Diagrama de Secuencia.

En la Figura 23, se observa el diagrama de operaciones referente al caso de uso Gestionar Generador de Sesiones.

Figura 23. Diagrama de secuencia - gestionar Generador de Sesiones



Casos de pruebas.

En la Tabla 33, se observa la descripción de escenarios de pruebas para el caso de uso gestionar generador de sesiones.

Tabla 33. Escenarios de pruebas para el caso de uso gestionar generador de sesiones

Código de Escenario	Escenario	Flujo Comienzo	Flujo Alternativo
E1	Configuración de sesiones.	El usuario selecciona la opción Settings.	
E2	Identificación de sesiones.	El usuario selecciona la opción Run.	El sistema permite reintentar o cancelar el proceso.

En la Tabla 34, se observa la descripción de los casos de prueba para el caso de uso gestionar generador de sesiones.

Tabla 34. Matriz de casos de prueba para el caso de uso gestionar generador de sesiones

Caso de Prueba	Escenario	Condición	Resultado
C1	E1	• Desplegar el menú del nodo Session. • Seleccionar la opción Settings.	Se observa el formulario de configuración de sesiones. El nodo cambia a color naranja.
C2	E2	• Desplegar el menú del nodo Session. • Seleccionar la opción Run.	El nodo cambia a color verde.

4.4 DEFINICIÓN EXTENDIDA DEL CASO DE USO GESTIONAR GENERADOR DE ESTADÍSTICAS

En la Tabla 35, se describe las características del caso de uso gestionar generador de estadísticas.

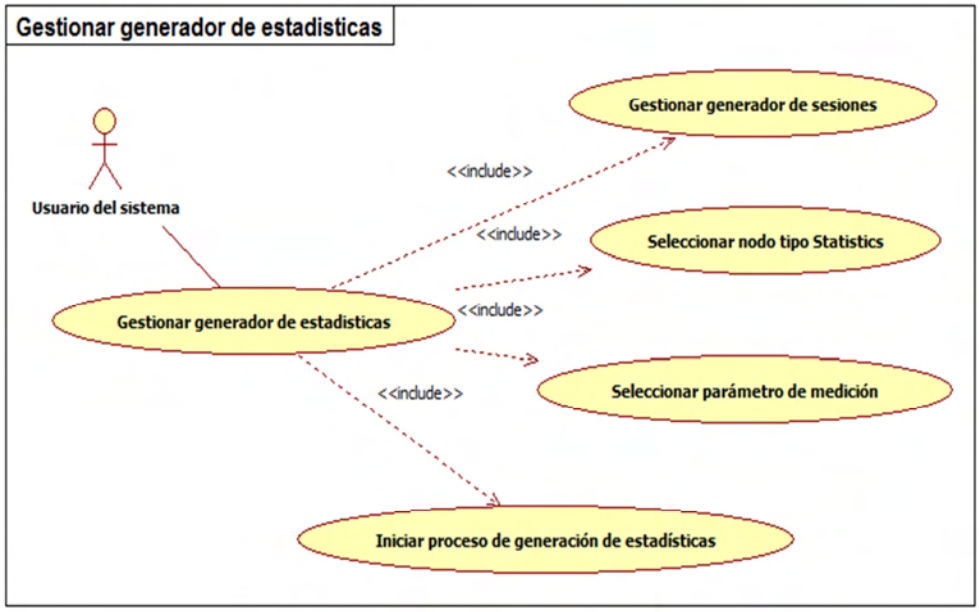
Tabla 35. Definición del caso de uso gestionar generador de estadísticas

Id Caso de Uso	**CU -003**
Titulo	**Gestionar generador de estadísticas.**
Objetivo	Permitir al usuario generar estadísticas de tráfico web
Resumen	Seleccionar los parámetros de medición que se usaran en el análisis estadístico y posteriormente iniciar el proceso de generación de estadísticas.
Actores	Act-01 Usuario del sistema
Requisitos	Generar resumen general de estadísticas. Generar estadísticas por fecha. Generar estadísticas por día de la semana. Generar estadísticas por hora del día. Generar estadísticas de archivos solicitados. Generar estadísticas de direcciones IP. Generar estadísticas de códigos de estado HTTP. Generar estadísticas de agentes de usuario. Generar estadísticas de sistemas operativos. Generar estadísticas de navegadores web. Generar estadísticas de Crawlers, robots o spiders. Generar estadísticas de sitios web referentes
Precondiciones	Identificar sesiones de usuario.
Postcondiciones	Visualizar resultados estadísticos.
Flujo de Eventos	**Flujo Básico** El caso de uso inicia cuando el usuario selecciona un nodo tipo Statistics: **1.** El usuario despliega el menú del nodo tipo Statistics. **2.** El usuario selecciona la opción Settings. **3.** El usuario selecciona el parámetro de medición. **4.** El usuario despliega el menú del nodo tipo Statistics. **5.** El usuario selecciona la opción Run. **6.** El sistema genera estadísticas según el nodo tipo Statistics seleccionado y almacena el resultado en una tabla de la base de datos. **Flujo de Excepción** **1.** Paso 6: Si se presenta un error en la conexión con la base de datos, el sistema permite reintentar, cancelar el proceso.

Diagrama de Caso de Uso.

En la Figura 24, se observa el diagrama del caso de uso gestionar generador de estadísticas.

Figura 24. Diagrama de caso de uso gestionar generador de estadísticas



Prototipo de interfaz gráfica.

En la Figura 25 y Figura 26, se observan los prototipos de la interfaz gráfica referente al caso de uso gestionar generador de sesiones.

Figura 25. Prototipo de interfaz gráfica - Menú de un nodo tipo Statistics

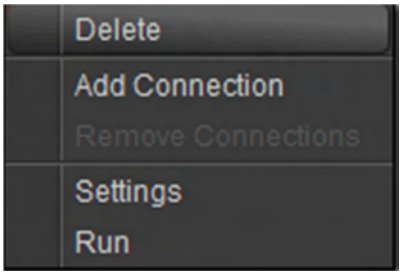


Figura 26. Prototipo de interfaz gráfica – Configurar nodo tipo Statistics

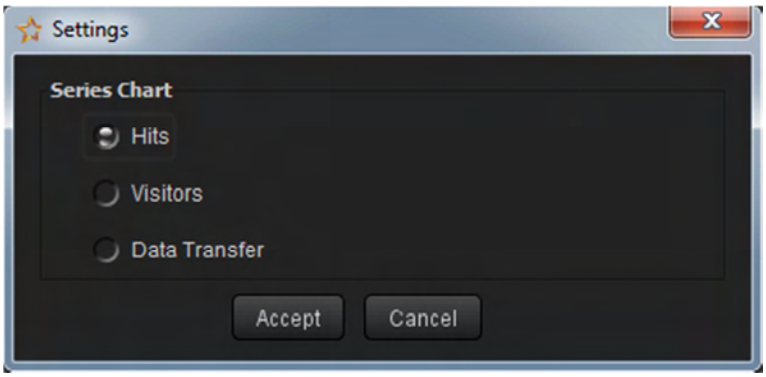
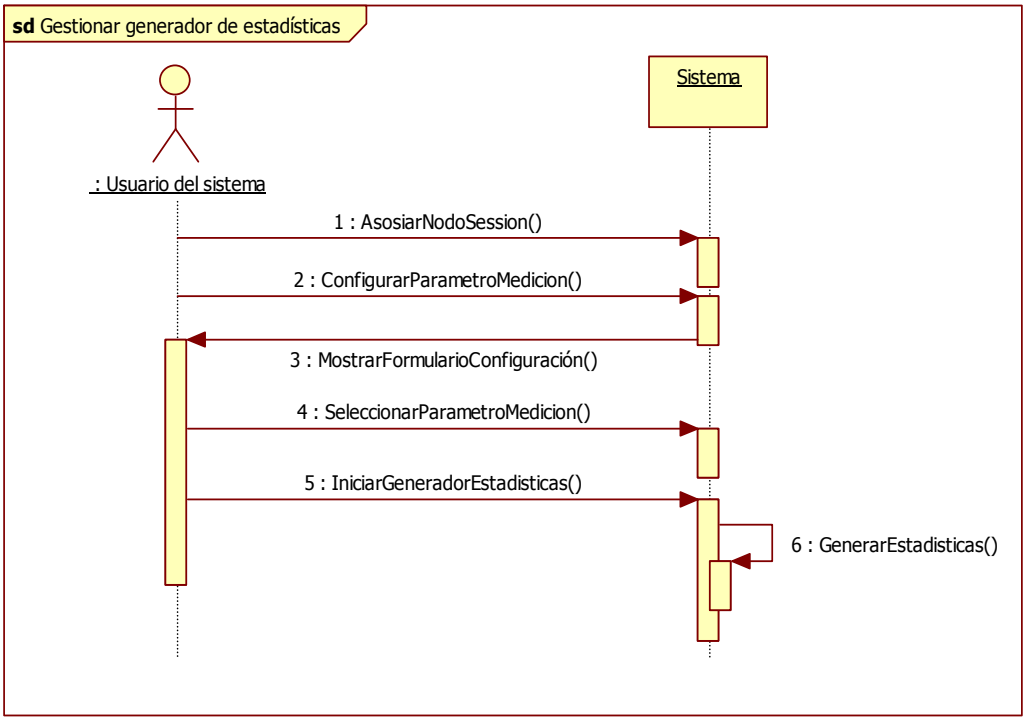


Diagrama de Secuencia.

En la Figura 27, se observa el diagrama de operaciones referente al caso de uso Gestionar Generador de Estadísticas.

Figura 27. Diagrama de secuencia - gestionar generador de estadísticas



Casos de pruebas.

En la Tabla 36, se observa la descripción de escenarios de pruebas para el caso de uso gestionar generador de estadísticas.

Tabla 36. Escenarios de pruebas para el caso de uso gestionar generador de estadísticas

Código de Escenario	Escenario	Flujo Comienzo	Flujo Alternativo
E1	Configuración de estadísticas.	El usuario selecciona la opción Settings.	
E2	Generación de estadísticas.	El usuario selecciona la opción Run.	El sistema permite reintentar o cancelar el proceso.

En la Tabla 37, se observa la descripción de los casos de prueba para el caso de uso gestionar generador de estadísticas.

Tabla 37. Matriz de casos de prueba para el caso de uso gestionar generador de estadísticas

Caso de Prueba	Escenario	Condición	Resultado
C1	E1	• Desplegar el menú del nodo tipo Statistics. • Seleccionar la opción Settings.	Se observa el formulario de configuración de estadísticas. El nodo cambia a color naranja.
C2	E2	• Desplegar el menú del nodo tipo Statistics. • Seleccionar la opción Run.	El nodo cambia a color verde.

4.5 DEFINICIÓN EXTENDIDA DEL CASO DE USO GESTIONAR GENERADOR DE GRÁFICOS

En la Tabla 38, se describe las características del caso de uso gestionar generador de gráficos.

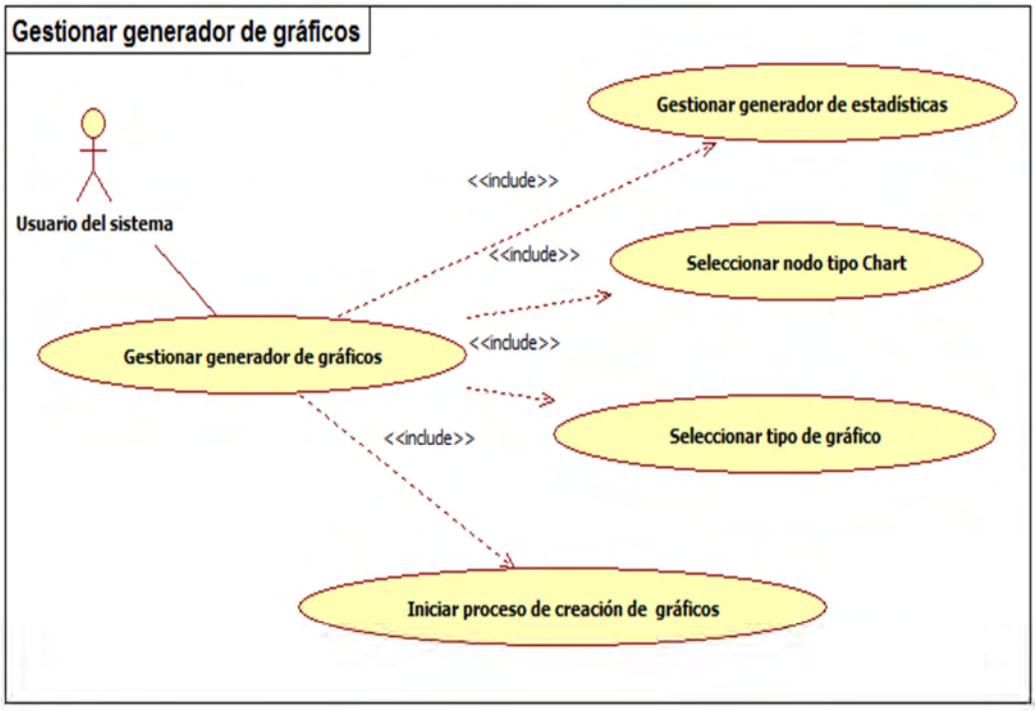
Tabla 38. Definición del caso de uso gestionar generador de gráficos.

Id Caso de Uso	**CU -004**
Titulo	**Gestionar generador de gráficos.**
Objetivo	Permitir al usuario visualizar resultados por medio de gráficos.
Resumen	Configurar el tipo de gráfico y posteriormente iniciar el proceso de creación y visualización de gráficos.
Actores	Act-01 Usuario del sistema
Requisitos	Visualizar resultados por medio de gráficos de barras. Visualizar resultados por medio de gráficos de líneas. Visualizar resultados por medio de gráficos de circulares.
Precondiciones	Gestionar generador de estadísticas.
Postcondiciones	Visualizar resultados estadísticos.
Flujo de Eventos	**Flujo Básico** El caso de uso inicia cuando el usuario selecciona un nodo Chart: **1.** El usuario despliega el menú del nodo Chart. **2.** El usuario selecciona la opción Settings. **3.** El usuario selecciona el tipo de gráfico. **4.** El usuario despliega el menú del nodo Chart. **5.** El usuario selecciona la opción Run. **6.** El sistema crea y muestra el gráfico de estadísticas. **Flujo de Excepción** **1.** Paso 6: Si se presenta un error en la conexión con la base de datos, el sistema permite reintentar, cancelar el proceso.

Diagrama de Caso de Uso.

En la Figura 28, se observa el diagrama del caso de uso gestionar generador de gráficos.

Figura 28. Diagrama de caso de uso gestionar generador de gráficos



Prototipo de interfaz gráfica.

En la Figura 29 y Figura 30, se observa los prototipos de la interfaz gráfica referente al caso de uso gestionar generador de gráficos.

Figura 29. Prototipo de interfaz gráfica - Menú de un nodo tipo Chart

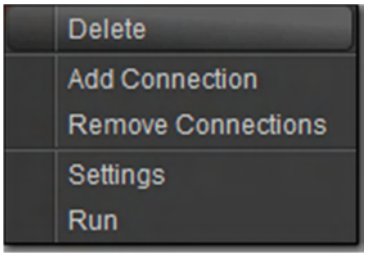


Figura 30. Prototipo de interfaz gráfica – Configurar nodo tipo Chart

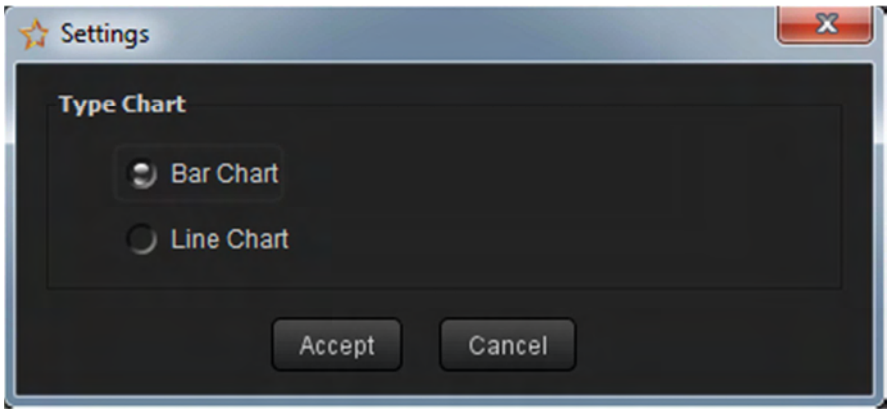
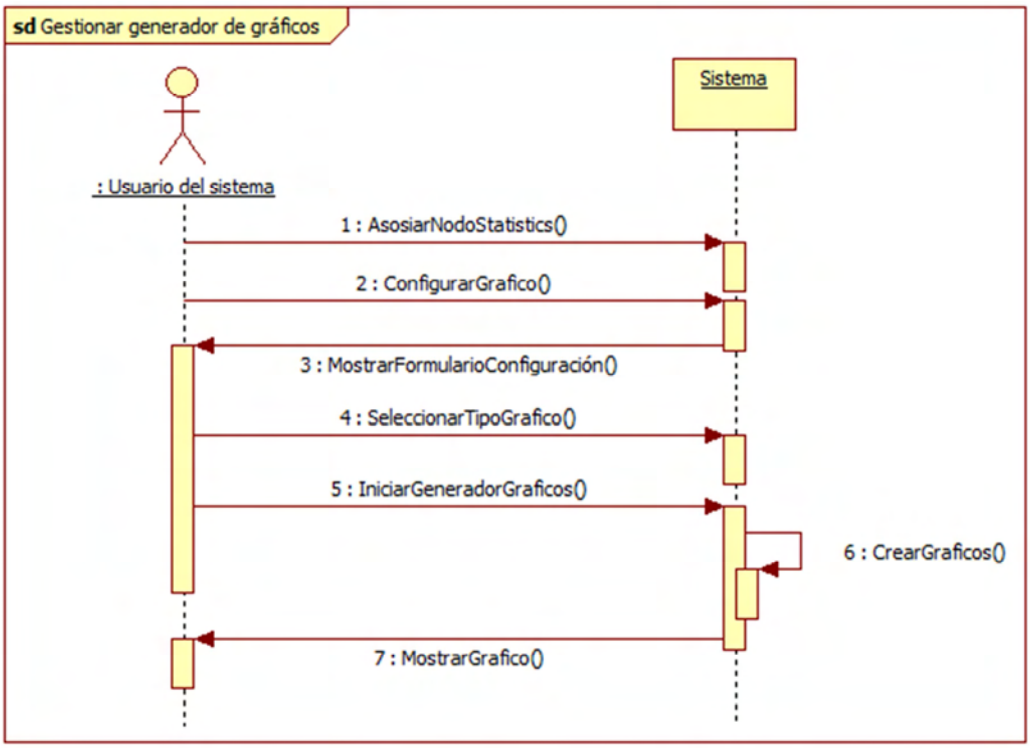


Diagrama de Secuencia.

En la Figura 31, se observa el diagrama de secuencia referente al caso de uso gestionar generador de gráficos.

Figura 31. Diagrama de Secuencia - gestionar generador de gráficos



Casos de pruebas.

En la Tabla 39, se observa la descripción de escenarios de pruebas para el caso de uso Gestionar Generador de Gráficos.

Tabla 39. Escenarios de pruebas para el caso de uso Gestionar Generador de Gráficos

Código de Escenario	Escenario	Flujo Comienzo	Flujo Alternativo
E1	Configuración de gráficos.	El usuario selecciona la opción Settings.	
E2	Creación de gráficos.	El usuario selecciona la opción Run.	El sistema permite reintentar o cancelar el proceso.

En la Tabla 40, se observa la descripción de los casos de prueba para el caso de uso Gestionar Generador de Gráficos.

Tabla 40. Matriz de casos de prueba para el caso de uso Gestionar Generador de Estadísticas

Caso de Prueba	Escenario	Condición	Resultado
C1	E1	• Desplegar el menú del nodo tipo Chart. • Seleccionar la opción Settings.	Se observa el formulario de configuración de gráficos. El nodo cambia a color naranja.
C2	E2	• Desplegar el menú del nodo tipo Chart. • Seleccionar la opción Run.	El nodo cambia a color verde. Se muestra el gráfico de estadísticas.

4.5 DEFINICIÓN EXTENDIDA DEL CASO DE USO GESTIONAR VISUALIZADOR DE TABLAS

En la Tabla 41, se describe las características del caso de uso gestionar generador de tablas.

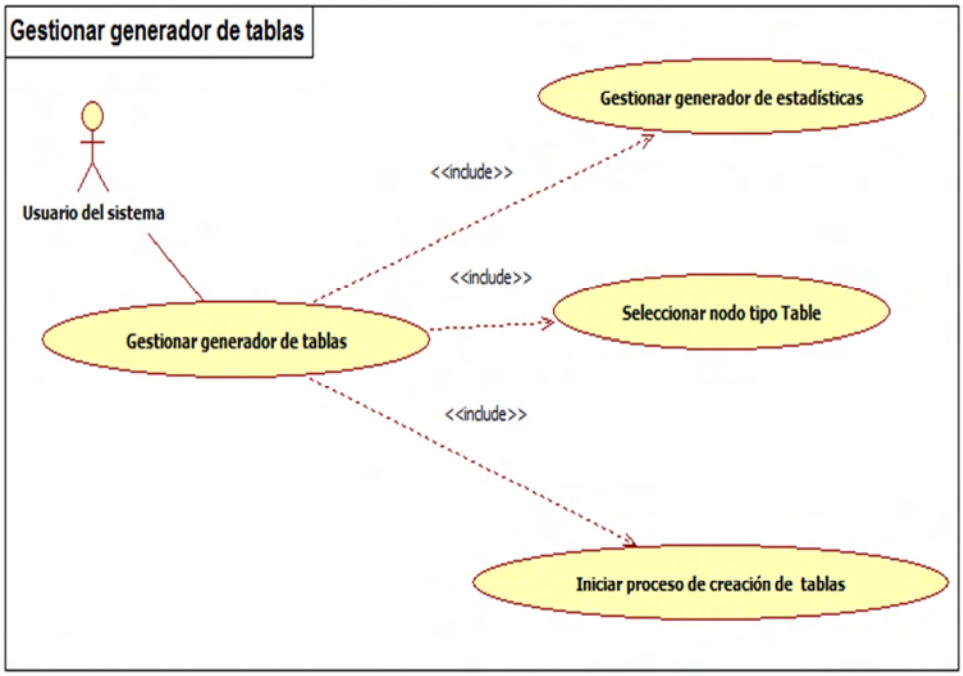
Tabla 41. Definición del caso de uso gestionar generador de tablas.

Id Caso de Uso	**CU -005**
Titulo	**Gestionar generador de tablas.**
Objetivo	Permitir al usuario visualizar resultados por medio de tablas.
Resumen	Realizar el proceso de creación y visualización de tablas.
Actores	Act-01 Usuario del sistema
Requisitos	Visualizar resultados por medio de tablas.
Precondiciones	Gestionar generador de estadísticas.
Postcondiciones	Visualizar resultados estadísticos.
Flujo de Eventos	**Flujo Básico** El caso de uso inicia cuando el usuario selecciona un nodo Table: **1.** El usuario despliega el menú del nodo Table. **2.** El usuario selecciona la opción Run. **3.** El sistema crea y muestra la tabla de estadísticas. **Flujo de Excepción** **1.** Paso 3: Si se presenta un error en la conexión con la base de datos, el sistema permite reintentar, cancelar el proceso.

Diagrama de Caso de Uso.

En la Figura 32, se observa el diagrama del caso de uso gestionar generador de tablas.

Figura 32. Diagrama de caso de uso gestionar generador de tablas



Prototipo de interfaz gráfica.

En la Figura 33, se observa los prototipos de la interfaz gráfica referente al caso de uso Gestionar Generador de Tablas:

Figura 33. Prototipo de interfaz gráfica - Menú de un nodo tipo Table

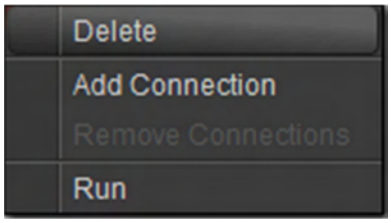
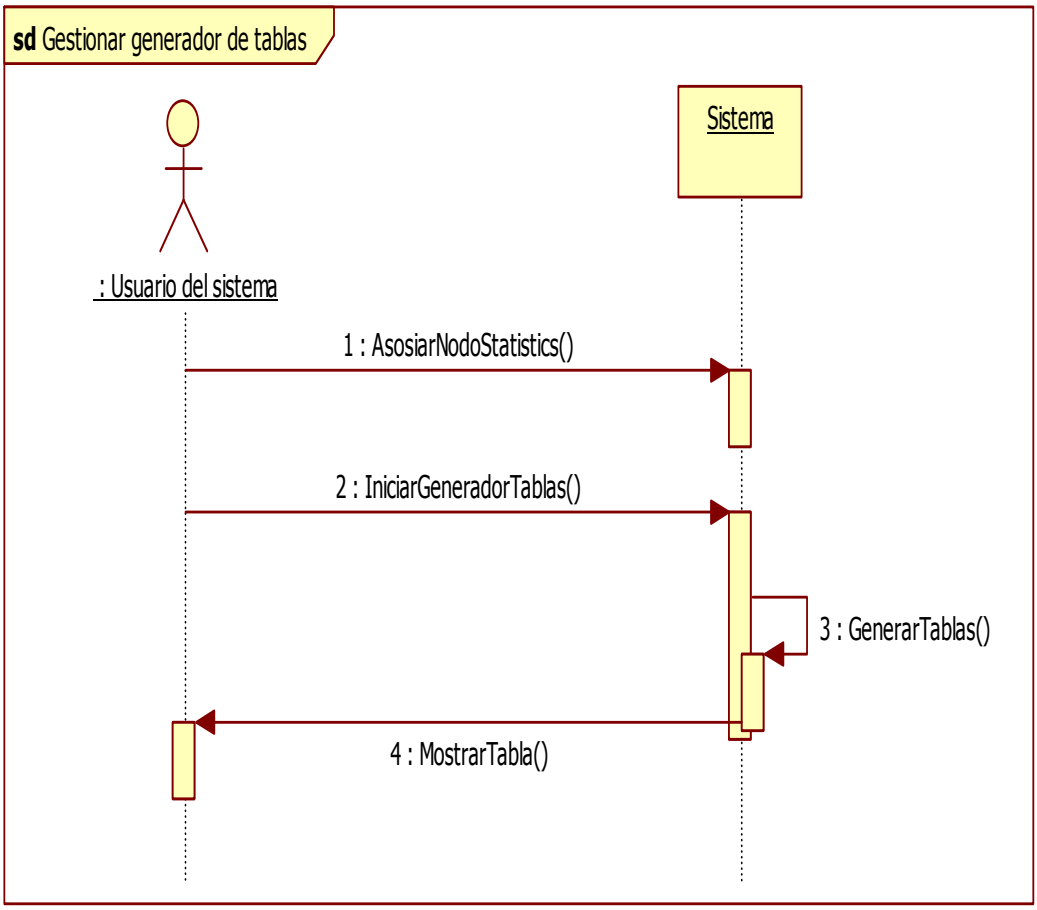


Diagrama de Secuencia.

En la Figura 34, se observa el diagrama de secuencia referente al caso de uso gestionar generador de tablas.

Figura 34. Diagrama de Secuencia - Gestionar Generador de Tablas



Casos de pruebas.

En la Tabla 42, se observa la descripción de escenarios de pruebas para el caso de uso Gestionar Generador de Gráficos.

Tabla 42. Escenarios de pruebas para el caso de uso Gestionar Generador de Gráficos

Código de Escenario	Escenario	Flujo Comienzo	Flujo Alternativo
E2	Creación de Tablas.	El usuario selecciona la opción Run.	El sistema permite reintentar o cancelar el proceso.

En la Tabla 43, se observa la descripción de los casos de prueba para el caso de uso Gestionar Generador de Gráficos.

Tabla 43. Matriz de casos de prueba para el caso de uso Gestionar Generador de Estadísticas

Caso de Prueba	Escenario	Condición	Resultado
C2	E2	• Desplegar el menú del nodo tipo Table. • Seleccionar la opción Run.	El nodo cambia a color verde. Se muestra la tabla de estadísticas.

5. IMPLEMENTACIÓN DE LA HERRAMIENTA POLARIS VERSIÓN 3.0

La implementación de la herramienta software ***POLARIS V3.0*** se realizó sobre sistema operativo *Windows Seven Ultimate de 64 bits* utilizando el lenguaje de programación Java 6.0 y la plataforma de desarrollo NetBeans 7.0.1.

5.1 ARQUITECTURA

Los módulos de software del módulo de análisis estadístico de tráfico web de la herramienta POLARIS Versión 3.0 son:

5.1.1 Módulo de utilidades. Este módulo es el encargado de dos tareas principales:

- Realizar la conexión a la base de datos y poder acceder a la fuente de datos que repose en el disco duro.

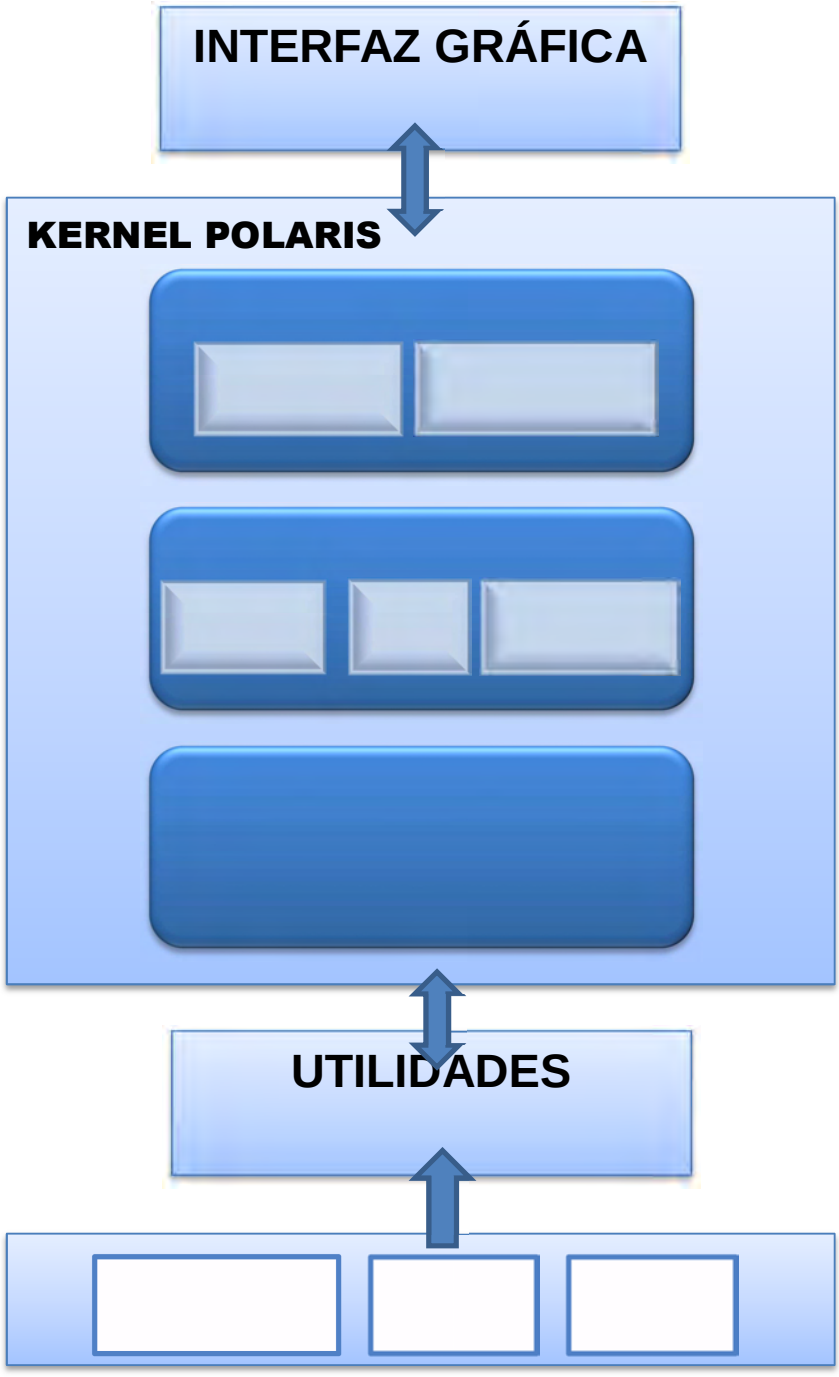
- Contener la colección de clases principales y librerías que son utilizadas por otras clases en la manipulación y visualización de datos, de esta manera puede hacerse la administración de las mismas y se hace factible la reutilización del código.

5.1.2 Módulo de kernel. Este módulo se encarga de realizar las tareas y procedimientos que permiten extraer los registros almacenados en los archivos logs, procesarlos y transformarlos para luego aplicarles técnicas de análisis estadístico correspondiente. Los componentes principales del Kernel son el generador de estadísticas, el generador de sesiones y el generador de gráficos.

5.1.3 Módulo de interfaz gráfica. Este módulo contiene las clases necesarias para construir y desplegar las estructuras para la visualización gráfica y dinámica de los resultados obtenidos después de la aplicación de los diferentes procesos que permiten obtener estadísticas de tráfico web. Además, en el se encuentran las herramientas necesarias para que el usuario pueda interactuar de una forma fácil y agradable con los diferentes componentes de la misma.

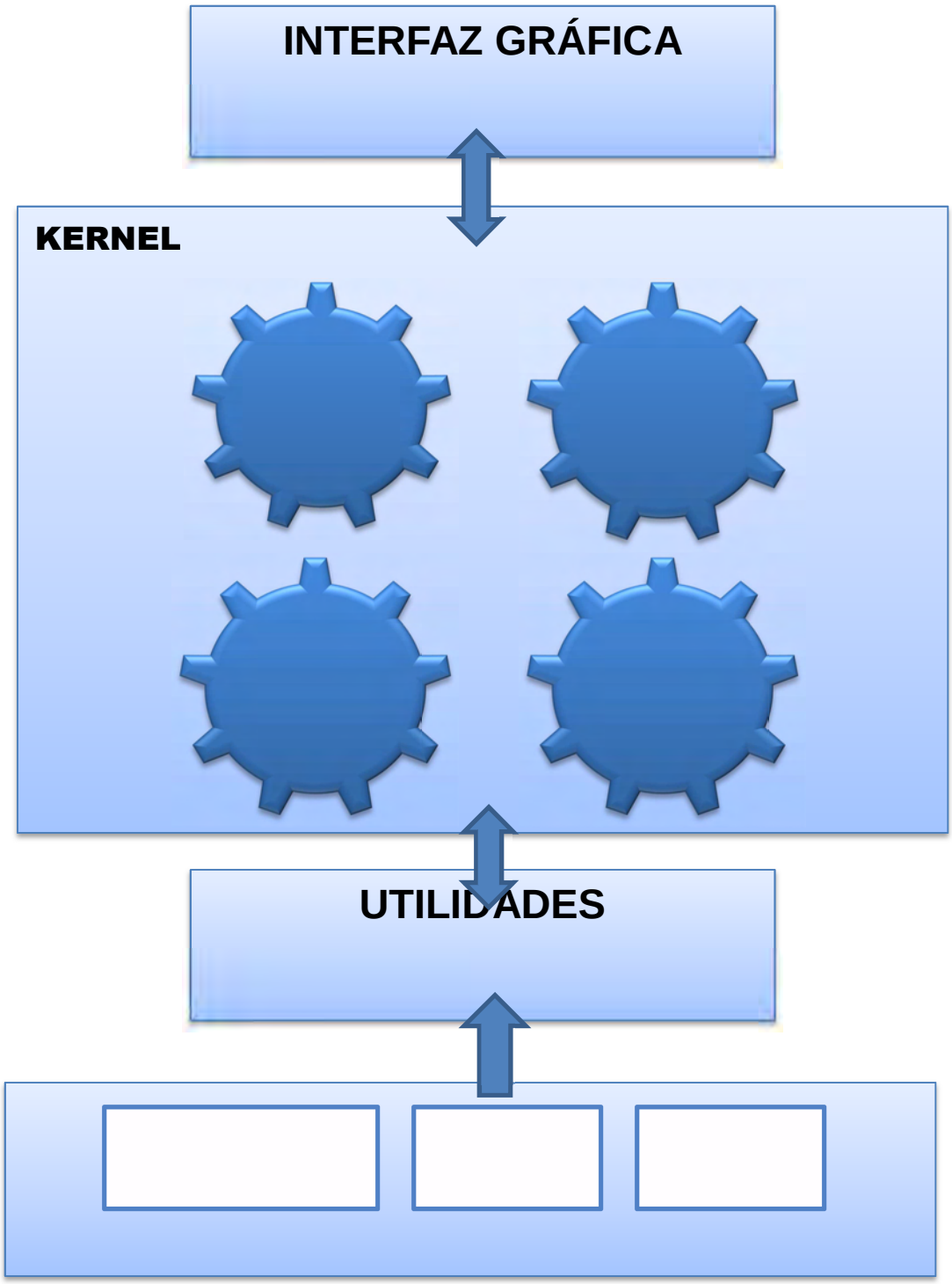
En la **Figura 35,** se observa la arquitectura general de la herramienta Polaris Versión 3.0.

Figura 35. Arquitectura general de Polaris Versión 3.0



En la **Figura 36,** se observa la arquitectura del arquitectura del módulo de análisis estadístico de tráfico web.

Figura 36. Arquitectura del módulo de análisis estadístico de tráfico web



5.2 ESTRUCTURA DE PAQUETES

La estructura general de los paquetes de la herramienta software **Polaris Versión 3.0** para el módulo de análisis estadístico de tráfico web y el acoplamiento con la versión anterior, es la siguiente:

- El **Módulo de utilidades** contiene el paquete polaris3.ReadLogs que contiene las clases necesarias para leer los registros del archivo log y modificarlos para almacenarlos en la base de datos; y le paquete polaris3.DataAcces que contiene las clases que permiten leer y escribir datos en la base de datos.

- El **Módulo de kernel** contiene el paquete polaris3.Session el cual agrupa las clases necesarias para realizar procesos de sesionalización y el paquete polaris3.Statistics que contiene las clases que permiten generar estadísticas de tráfico web.

- El **Módulo de interfaz gráfica** contiene los paquete polaris2 y polaris3.graph que contiene las clases que permiten construir la interfaz de usuario; el paquete polaris3.Chart que contiene clases que permiten visualizar estadísticas en un formato gráfico y el paquete polaris3.table que permite visualizar estadísticas en un formato de tabla.

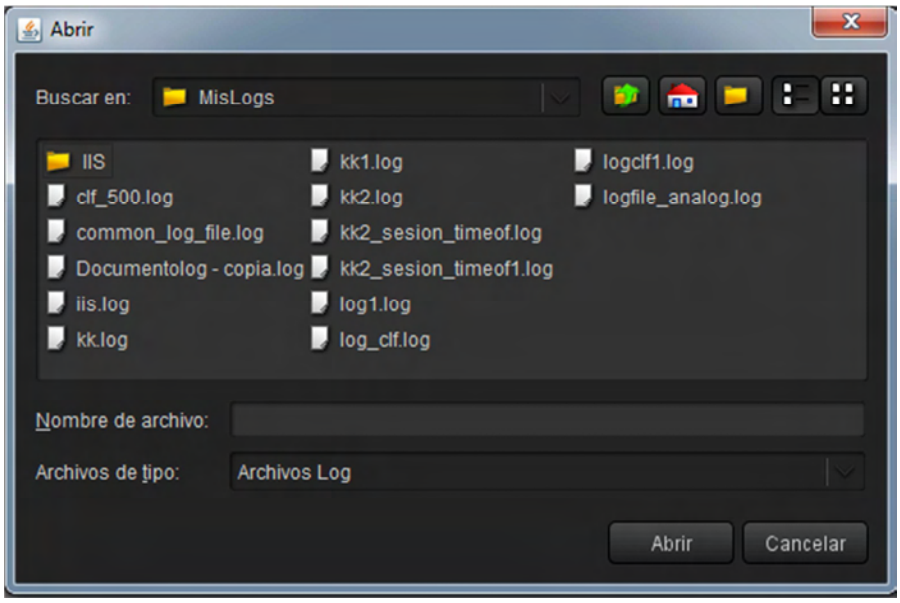
A continuación se amplía y se detallan conceptos sobre los paquetes más importantes dentro del desarrollo de la herramienta.

5.2.1 Paquetes del módulo de utilidades.

5.2.1.1 Paquete polaris3.ReadLogs. Este paquete agrupa las clases necesarias para leer los registros almacenados en los archivos log. Estas clases son:

Clase AbrirLog: Esta clase permite seleccionar el archivo log del sistema de archivos del disco duro y leer los registros almacenados. El método principal de esta clase es LeerArchivo(), el cual permite realizar los procedimientos mencionados anteriormente. En la Figura 37, se aprecia la ventana que permite abrir un archivo log.

Figura 37. Abrir archivo log



Fuente: La presente investigación

Clase IdentificarFormatoLog: Esta clase permite identificar el formato del log de acceso leído. Los principales métodos de esta clase son:

- Es_W3C(): este método permite determinar si el archivo log leído tiene un formato Formato W3C Extended Log File Format.

- Es_CommonLogFile: este método permite determinar si el archivo log leído tiene un formato NCSA.

5.2.1.2 Paqute polaris3.DataAccess. Este paquete contiene las clases que permite insertar y leer registros de la base de datos. Se destacan las siguientes clases:

Clase MConexion: Se encarga de establecer, administrar y cerrar la conexión a una Base de Datos. Esta clase maneja los siguientes métodos:

- AbrirConexion(): este método abre la conexión a la base de datos.

- CerrarConexion(): este método cierra la conexión con la base de datos.

- getConexion(): este método permite obtener una instancia de una conexión abierta a la base de datos.

Clase EjecutarQuerys: Esta clase permite ejecutar diferentes tipos de consultas en la base de datos. Esta clase maneja los siguientes métodos:

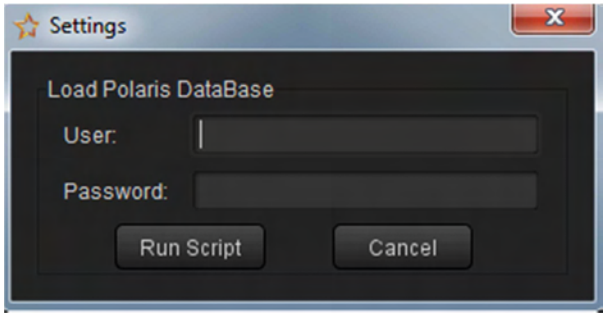
- InsertarDatos(): este método permite insertar datos en la base de datos.
- LeerDatos(): este método permite leer datos de la base de datos.

Clase cargarScript: Esta clase permite ejecutar el archivo sql que contiene el código necesario para crear la base de datos necesaria para la ejecución normal del programa. Esta clase posee el siguiente método principal:

- CorrerScript(String bd, String uN, String uP): este metodo permite leer y ejecutar cada sentencia sql almacenada en el script.

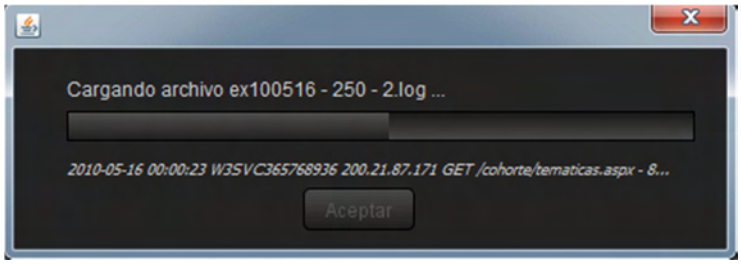
En la Figura 38, se observa la ventana que permite ejecutar el script sql:

Figura 38. Ejecutar script



Clase CargarRegistros: Esta clase permite dar formato a los registro leídos del archivo log y almacenar estos datos en la base de datos con un formato que permita facilitar la tarea de procesamiento de datos. En la Figura 39, se observa la ventana que permite visualizar el proceso de carga de registros:

Figura 39. Cargar registros



5.2.2 Paquetes del módulo de kernel.

5.2.2.1 Paquete polaris3.Session. Este paquete contiene las clases que permiten determinar sesiones de usuario y configurar el método de sesionalización que se utilizara (por intervalos o por *timeout*). Las clases mas importantes asociadas a este paquete son:

Clase RangoSesion: Esta clase permite configurar el método a utilizar para realizar el proceso de sesionalización, el cual puede ser utilizando intervalos de tiempo o utilizando el parámetro *timeout*.

Clase CrearSesiones: Esta clase contiene los procedimientos necesarios para realizar tareas de identificación de sesiones de usuario. Los métodos más importantes de esta clase son:

- CrearTablaSesiones(): este método permite crear una tabla en la base datos, la cual tendrá los campos necesarios para almacenar los datos producto del proceso de sesionalización.

- Sesionalizar*Timeout*(): este método permite realizar el proceso de identificación de sesiones utilizando el parámetro *timeout*, es decir, crea sesiones teniendo en cuenta un tiempo máximo de 30 minutos de diferencia entre accesos.

- SesionalizarIntervalos(): este método permite realizar el proceso de identificación de sesiones estableciendo previamente el tiempo que durara cada sesión. Este intervalo de tiempo es configurado por el usuario de la herramienta.

- CrearTablaResumen(): utilizando este método se crea una tabla en la base de datos en la cual se identifica cada sesión, se le asigna la fecha/hora de inicio y la fecha/hora de finalización, la cantidad de hits por cada sesión y la cantidad de bytes transferidos en cada sesión.

5.2.2.2 Paqute polaris3.Statistics. Este paquete agrupa las clases necesarias para procesar la información de los registros del archivo log y generar las diferentes estadísticas de tráfico web. Las clases que permiten generar estas estadísticas son:

Clase CrearEstadisticasCodigosHttp: Esta clase permite crear estadísticas relacionadas con los códigos de estado HTTP. Esta clase posee solo el siguiente método:

- CrearTablas(): este método genera datos estadísticos sobre códigos de estado

HTTP y crea una tabla en la base de datos donde se registra la información generada. En este método se ejecuta la función pgplsql crear_estadisticas_codigoshttp().

Clase CrearEstadisticasAcceso: Esta clase permite crear estadísticas relacionadas con los diferentes tipos de páginas y archivos solicitados. Esta clase posee solo el siguiente método:

- CrearTablas(): este método genera datos estadísticos sobre archivos solicitados y crea una tabla en la base de datos donde se registra la información generada. En este método se ejecuta la función pgplsql crear_acceso_estadisticas ().

Clase CrearEstadisticasBuscadores: Esta clase permite crear estadísticas relacionadas con los motores de búsqueda utilizados. Esta clase posee solo el siguiente método:

- CrearTablas(): este método genera datos estadísticos sobre los diferentes motores de búsqueda utilizados y crea una tabla en la base de datos donde se registra la información generada. En este método se ejecuta la función pgplsql crear_est_buscadores ().

Clase CrearEstadisticasDominios: Esta clase permite crear estadísticas relacionadas con los dominiios geográficos de los cuales provienen las visitas. Esta clase posee solo el siguiente método:

- CrearTablas(): este método genera datos estadísticos sobre dominios geográficos de los cuales provienten las visitas y crea una tabla en la base de datos donde se registra la información generada. En este método se ejecuta la función pgplsql crear_est_dominios ().

Clase CrearEstadisticasPalabras: Esta clase permite crear estadísticas relacionadas con las palabras utilizadas para realizar la búsqueda del sitio web. Esta clase posee solo el siguiente método:

- CrearTablas(): este método genera datos estadísticos sobre cadenas de búsqueda utilizadas por los usuarios de un sitio y crea una tabla en la base de datos donde se registra la información generada. En este método se ejecuta la función pgplsql crear_est_palabras ().

Clase CrearEstadisticasSitios: Esta clase permite crear estadísticas relacionadas con los sitiios web de los cuales proceden las visitas. Esta clase posee solo el siguiente método:

- CrearTablas(): este método genera datos estadísticos sobre los sitios web de los cuales proceden las visitas y crea una tabla en la base de datos donde se registra la información generada. En este método se ejecuta la función pgplsql crear_est_sitios_referentes ().

Clase CrearEstadisticasUserAgent: Esta clase permite crear estadísticas relacionadas con las características de la aplicación que accede un sitio web, permitiendo obtener información del sistema operativo, del navegador u otro tipo de aplicación. Esta clase posee solo el siguiente método:

CrearTablas(): este método genera datos estadísticos sobre la información de la aplicación que permite al cliente acceder al sitio web y crea una tabla en la base de datos donde se registra la información generada. En este método se ejecuta la función pgplsql crear_useragent_estadisticas ().

5.2.3 Paquetes del módulo de interfaz gráfica.

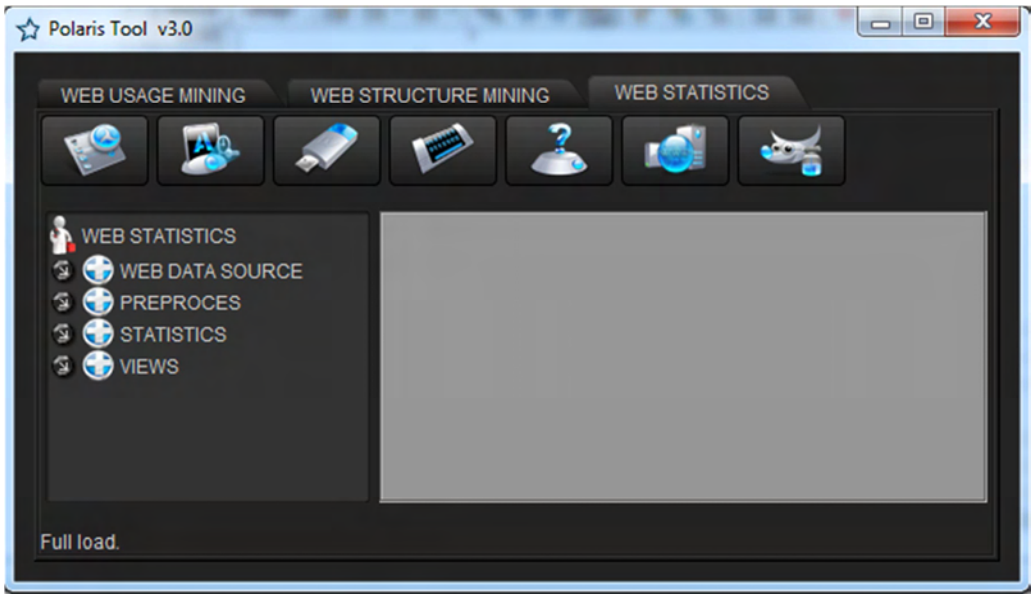
5.2.3.1 Paquete polaris2. En este paquete se encuentra la clase principal para la visualización de la interfaz gráfica implementado el acople de los módulos de minería de uso Web, Minería de estructura Web y Análisis Estadístico de Tráfico Web, y otras clases que permiten el buen funcionamiento del acople del producto final. Es el paquete que permite las diferentes conexiones de los nodos en el área de trabajo, como también las restricciones, eliminación, modificación de los nodos. En este paquete se encuentra la siguiente clase:

Clase ventana: Se encarga de presentar la interfaz gráfica de la herramienta Polaris versión 3.0 como se muestra en la Figura 40, conteniendo en ella los módulos de minería de uso, minería de estructura y análisis estadístico de tráfico web. Los métodos principales de esta clase son:

- eventos_area_trabajo(): Es el método que permite eliminar, modificar, arrastrar y mover los diferentes objetos con los cuales se trabaja dentro del área de trabajo, permite configurar y determinar las diferentes restricciones para los objetos presentes dentro de una área de trabajo y que cada uno de ellos se pueda conectar al objeto correspondiente. Trabaja con diferentes métodos para lograr este objetivo.

- eventos_arbol(): Permite reconocer que icono se está seleccionando y arrastrando a la área de trabajo permitiendo el manejo del Drag and Drop en el momento de seleccionar un icono de la árbol de herramientas.

Figura 40. Ventana principal de la herramienta Polaris



5.2.3.2 Paquete polaris3.graph. Este paquete contiene la clase que se necesitan para unir los diferentes nodos de un gráfico (gráfico es el dibujo que se forma con los diferentes nodos que se pueden arrastrar del árbol de herramientas que está en la parte izquierda de la presentación) y las diferentes conexiones que se realizan entre estos por el usuario.

Clase area_trabajo: El área de trabajo es la parte gris de la ventana principal, esta clase es la que administra todos los nodos que se arrastran desde el árbol hasta el área de trabajo donde se ubican y conectan las diferentes partes del sistema.

Se encarga de eliminar, modificar, mover y presentar los nodos en el área donde el usuario se encuentra trabajando.

Contiene variable de tipo nodo que apunta al primer y último nodo que se encuentra en el área de trabajo y tiene funciones que permiten en el área de trabajo.

- public void parar_nodo(Nodo n): En el área de trabajo cuando un nodo es reconfigurado o eliminado los nodos que estén conectados a este deben volverse a ejecutar, esta función se llama para reconocer los nodos conectados a este nodo.

- public void paint(Graphics g): Función realiza el dibujo de los nodos y conexiones del área de trabajo.

- public void eliminar_nodo(Nodo nod): Elimina un nodo del área de trabajo.

- public void insertar_nodo(String tipo,int pos_x,int pos_y): Inserta un nodo en le área de trabajo, se le envían como parámetros las coordenadas en donde será dibujado.

- public void conectar(Nodo fuente, Nodo destino): Realiza una conexión entre dos nodos (es llamada por insertar_conexion, que es el método encargado de validarlo).

- public int revisar(Conexion c, String t): Revisa si en las conexiones de un nodo existe un nodo tipo "t" configurado y arrancado.

- public int insertar_conexion(Nodo fuente, Nodo destino): Esta función es la que realiza la conexión entre dos nodos, primero verifica de que tipo son los nodos que se van a conectar, luego valida si la conexión entre estos nodos del área de trabajo es permitida, si es permitida los conecta sino devuelve un mensaje indicando por qué no se pudieron conectar.

5.2.3.3 Paquete polaris3.Chart. Este paquete contiene las clases necesarias para generar diferentes tipos de gráficos estadísticos a partir de la información generada por el módulo kernel. El paquete contiene diversas clases, cada una de ellas genera gráficos que difieren dependiendo de las estadísticas que se desee generar.

Estas clase se comportan de manara similar y por tanto comparte los métodos más importantes. Estos métodos son:

- CargarDatosGrafico(): este método permite cargar desde la base de datos la información necesaria para generar un gráfico.

- CrearDatosGrafico(): este método organiza la información obtenida de la base de datos, de tal manera que pueda ser leída por el Api que se utilizo para crear gráficos.

En la Figura 41, Figura 42 y Figura 43, se observan los diferentes tipos de gráfico que pueden generar las clases mencionadas.

Figura 41. Gráfico de barras

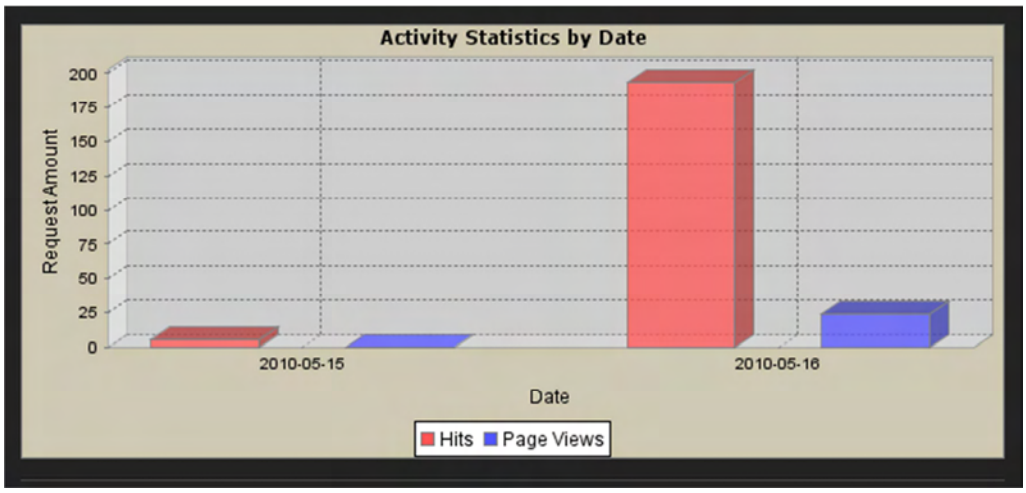


Figura 42. Gráfico de líneas

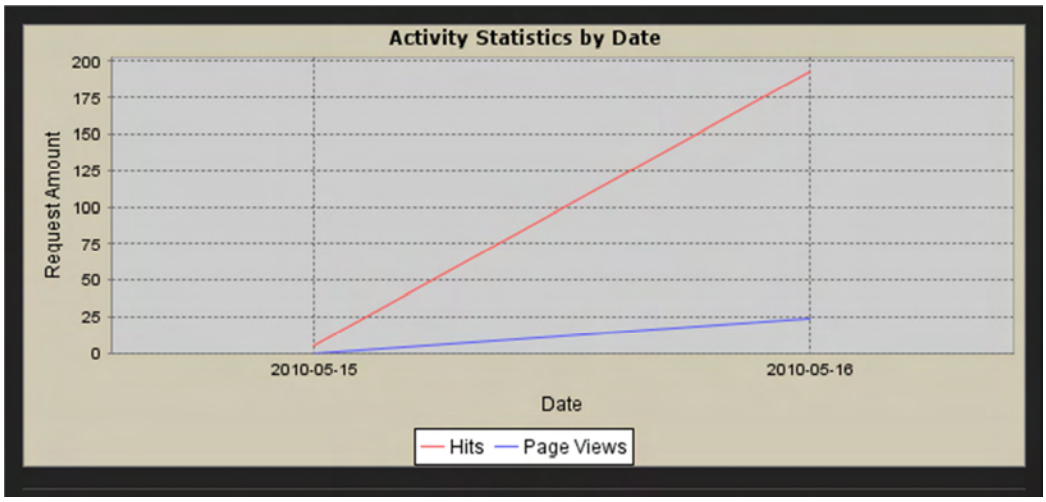
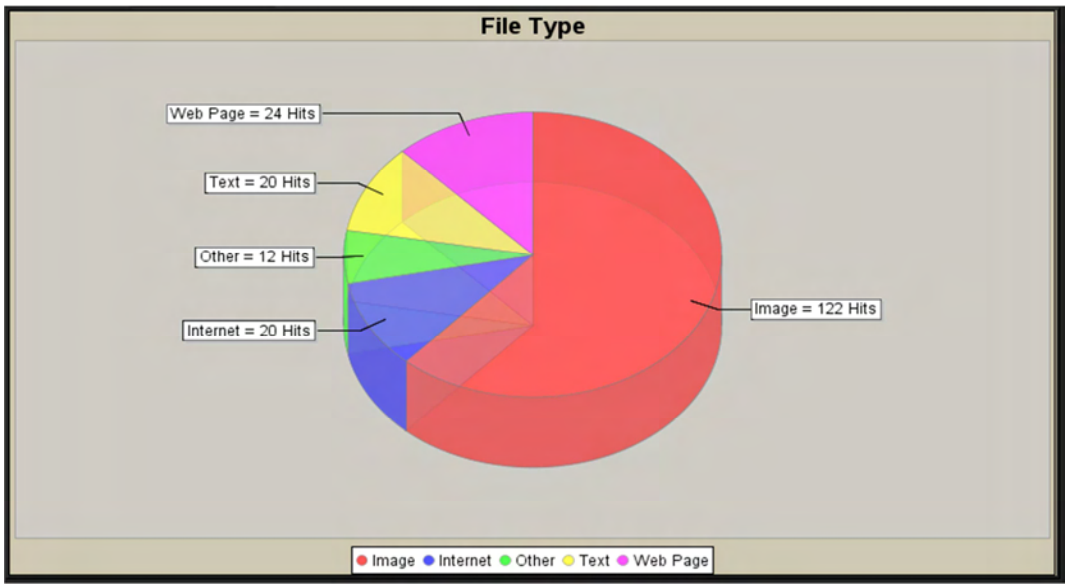


Figura 43. Gráfico circular



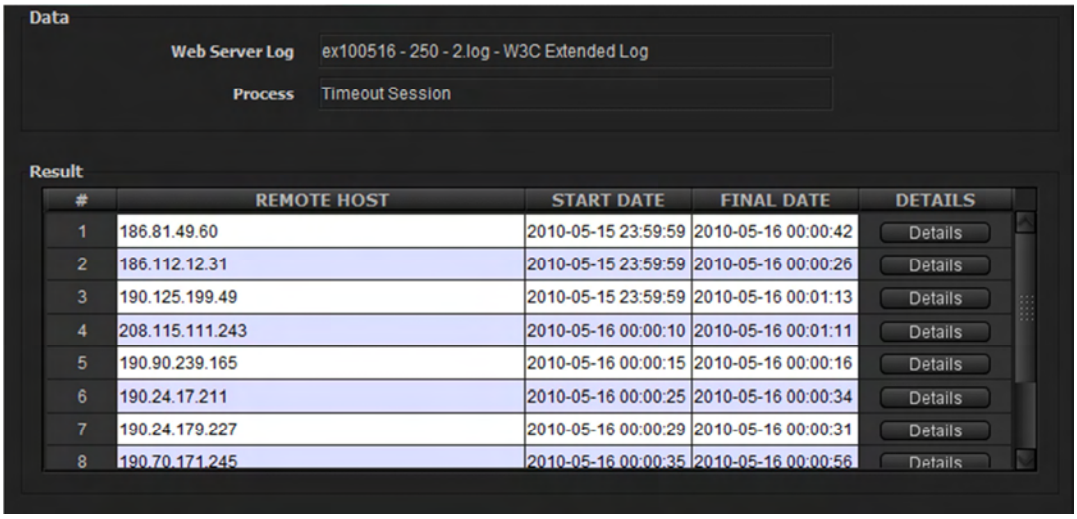
5.2.3.3 Paquete polaris3.Table. Este paquete contiene las clases necesarias para generar reportes en formato de tabla a partir de la información generada por el módulo kernel. El paquete contiene diversas clases, cada una de ellas genera tablas que difieren dependiendo de las estadísticas que se desee generar.

Estas clase se comportan de manara similar y por tanto comparte los métodos más importantes. Estos métodos son:

- CrearTabla(): este metodi crea un componente JTable que permite visualizar los reportes estadísticos en un formato de tabla.
- CargarDatosDatos(): este método permite cargar la información desde la base de datos hacia el componente JTable creado.

En la Figura 44, se observa un reporte estadístico en un formato de tabla.

Figura 44. Reporte en formato tabla



6. PRUEBAS Y EVALUACIÓN DE RESULTADOS

Las pruebas de la herramienta se desarrollaron utilizando un procesador Intel Core I3, Disco Duro de 180 GB y memoria de 4 Gigas.

EL conjunto de datos utilizados para las pruebas está conformado por archivos logs pertenecientes a repositorios reales de la Universidad de Nariño, específicamente del sitio web de la Vicerrectoría de Investigaciones, Postgrados y Relaciones Internacionales.

La ejecución de estas pruebas tiene como objetivos determinar la capacidad del Módulo de Análisis Estadístico de Tráfico Web de la herramienta Polaris, de generar información estadística confiable y útil con respecto al tráfico web realizado sobre un sito en un periodo de tiempo determinado, y evaluar aspectos las funcionalidades de Polaris 3.0 con respecto a otras herramientas de análisis estadístico de tráfico web.

En la Tabla 44, se describen las características generales del archivo log utilizado para realizar las pruebas.

Tabla 44. Archivo log de prueba

Nombre del archivo	**Ex1205.log**
Formato	**W3C Log Format**
Tamaño	**4447 Kilobytes**
Numero de registros	**16138**
Inicio (Fecha - Hora)	**10/Mayo/2012 - 16:29:05**
Fin (Fecha- Hora)	**16/Mayo/2012 - 23:56:00**

El formato de archivo log W3C, puede variar con respecto a los campos que este almacena, dependiendo de la configuración del servidor web. Los campos que registra el archivo log de prueba se describen en la Tabla 45:

Tabla 45. Campos de archivo log de prueba

CAMPO	DESCRIPCIÓN
date	Fecha de la petición
time	Hora de la petición
s-sitename	Nombre de la maquina donde está alojado el Servidor
s-ip	IP del servidor que sirvió la pagina
cs-method	Método de acceso del cliente
cs-uri-stem	Recurso solicitado
cs-uri-query	Variables enviadas con el método post
s-port	Puerto local del servidor, por el cual se sirve los datos
cs-username	Es el nombre del servicio del sitio web
c-ip	IP del cliente
cs(User-Agent)	Aplicación utilizada por el cliente
cs(Referer)	Sitio del cual se procede
cs-host	Nombre host por el cual accedió el cliente al servidor
sc-status	Estado de respuesta del servidor
sc-win32-status	Estado de respuesta del servidor, con códigos de Windows
sc-bytes	Numero de bytes enviados por el servidor.
cs-bytes	Numero de bytes recibidos por el servidor.

6.1 ANÁLISIS DE FUNCIONALIDAD DE LA HERRAMIENTA ANALOG

En la Tabla 46, se observa el resultado del análisis y evaluación de una serie de características funcionales sobre la herramienta **Analog**.

Tabla 46. Análisis de funcionalidad de la herramienta Analog

CARACTERÍSTICA EVALUADA	
Presenta un módulo que permite realizar un instalación sencilla y guiada paso a paso.	**NO**
Presenta una interfaz de usuario amigable y fácil de usar.	**NO**
Permite procesar archivos log en sus diferentes formatos.	**SI**
Permite realizar tareas de preprocesamiento y limpieza de datos.	**NO**
Permite identificar Sesiones de Usuario.	**NO**
Permite realizar procesos de filtrado de datos.	**NO**
Permite realizar análisis estadístico de tráfico web por fechas.	**NO**
Permite realizar análisis estadístico de tráfico web por cada hora del día.	**SI**
Permite realizar análisis estadístico de tráfico web por cada día de la semana.	**SI**
Permite analizar y clasificar los archivos solicitados en cada petición web.	**SI**
Permite identificar y clasificar las direcciones Ip desde las cuales se realizan las peticiones web.	**SI**
Permite identificar las respuestas del servidor web mediante el uso de códigos de estado HTTP.	**SI**
Permite analizar las cadenas los registros de User Agents de los archivos log.	**NO**
Permite identificar peticiones realizadas por programas como crawlers, robots o spiders.	**NO**
Permite identificar los diferentes sistemas operativos desde los cuales se realizaron las peticiones web.	**SI**
Permite identificar los diferentes navegadores web que se usaron para acceder al servidor web.	**SI**
Permite identificar y clasificar los dominios desde los cuales se realizaron las peticiones web.	**SI**
Permite identificar los motores de búsqueda utilizados para encontrar el sitio web.	**SI**
Permite extraer e identificar las cadenas de búsqueda utilizadas para encontrar el sitio web.	**NO**
Permite identificar los sitos web desde los cuales se accedió al sitio web analizado.	**NO**
Permite visualizar los resultados obtenidos de forma gráfica y entendible.	**SI**
Permite ejecutarse en diferentes sistemas operativos.	**SI**
Presenta una documentación adecuada.	**SI**
Es posible configurar el software en diferentes idiomas.	**SI**

6.2 ANÁLISIS DE FUNCIONALIDAD DE LA HERRAMIENTA WEB LOG EXPERT

En la Tabla 47, se observa el resultado del análisis y evaluación de una serie de

características funcionales sobre la herramienta **Analog**.

Tabla 47. Análisis de funcionalidad de la herramienta Web Log Expert

CARACTERÍSTICA EVALUADA	
Presenta un módulo que permite realizar un instalación sencilla y guiada paso a paso.	**SI**
Presenta una interfaz de usuario amigable y fácil de usar.	**SI**
Permite procesar archivos log en sus diferentes formatos.	**SI**
Permite realizar tareas de preprocesamiento y limpieza de datos.	**NO**
Permite identificar Sesiones de Usuario.	**NO**
Permite realizar procesos de filtrado de datos.	**NO**
Permite realizar análisis estadístico de tráfico web por fechas.	**NO**
Permite realizar análisis estadístico de tráfico web por cada hora del día.	**SI**
Permite realizar análisis estadístico de tráfico web por cada día de la semana.	**SI**
Permite analizar y clasificar los archivos solicitados en cada petición web.	**SI**
Permite identificar y clasificar las direcciones Ip desde las cuales se realizan las peticiones web.	**SI**
Permite identificar las respuestas del servidor web mediante el uso de códigos de estado HTTP.	**SI**
Permite analizar las cadenas los registros de User Agents de los archivos log.	**SI**
Permite identificar peticiones realizadas por programas como crawlers, robots o spiders.	**SI**
Permite identificar los diferentes sistemas operativos desde los cuales se realizaron las peticiones web.	**SI**
Permite identificar los diferentes navegadores web que se usaron para acceder al servidor web.	**SI**
Permite identificar y clasificar los dominios desde los cuales se realizaron las peticiones web.	**SI**
Permite identificar los motores de búsqueda utilizados para encontrar el sitio web.	**SI**
Permite extraer e identificar las cadenas de búsqueda utilizadas para encontrar el sitio web.	**SI**
Permite identificar los sitos web desde los cuales se accedió al sitio web analizado.	**SI**
Permite visualizar los resultados obtenidos de forma gráfica y entendible.	**SI**
Permite ejecutarse en diferentes sistemas operativos.	**NO**
Presenta una documentación adecuada.	**NO**
Es posible configurar el software en diferentes idiomas.	**NO**

6.3 ANÁLISIS DE FUNCIONALIDAD DE LA HERRAMIENTA POLARIS 3.0

En la Tabla 48, se observa el resultado del análisis y evaluación de una serie de características funcionales sobre la herramienta **Analog**.

Tabla 48. Análisis de funcionalidad de la herramienta Polaris 3.0

CARACTERÍSTICA EVALUADA	
Presenta un módulo que permite realizar un instalación sencilla y guiada paso a paso.	**SI**
Presenta una interfaz de usuario amigable y fácil de usar.	**SI**
Permite procesar archivos log en sus diferentes formatos.	**SI**
Permite realizar tareas de preprocesamiento y limpieza de datos.	**SI**
Permite identificar Sesiones de Usuario.	**SI**
Permite realizar procesos de filtrado de datos.	**SI**
Permite realizar análisis estadístico de tráfico web por fechas.	**SI**
Permite realizar análisis estadístico de tráfico web por cada hora del día.	**SI**
Permite realizar análisis estadístico de tráfico web por cada día de la semana.	**SI**
Permite analizar y clasificar los archivos solicitados en cada petición web.	**SI**
Permite identificar y clasificar las direcciones Ip desde las cuales se realizan las peticiones web.	**SI**
Permite identificar las respuestas del servidor web mediante el uso de códigos de estado HTTP.	**SI**
Permite analizar las cadenas los registros de User Agents de los archivos log.	**SI**
Permite identificar peticiones realizadas por programas como crawlers, robots o spiders.	**SI**
Permite identificar los diferentes sistemas operativos desde los cuales se realizaron las peticiones web.	**SI**
Permite identificar los diferentes navegadores web que se usaron para acceder al servidor web.	**SI**
Permite identificar y clasificar los dominios desde los cuales se realizaron las peticiones web.	**SI**
Permite identificar los motores de búsqueda utilizados para encontrar el sitio web.	**SI**
Permite extraer e identificar las cadenas de búsqueda utilizadas para encontrar el sitio web.	**SI**
Permite identificar los sitos web desde los cuales se accedió al sitio web analizado.	**SI**
Permite visualizar los resultados obtenidos de forma gráfica y entendible.	**SI**
Permite ejecutarse en diferentes sistemas operativos.	**SI**
Presenta una documentación adecuada.	**SI**
Es posible configurar el software en diferentes idiomas.	**NO**

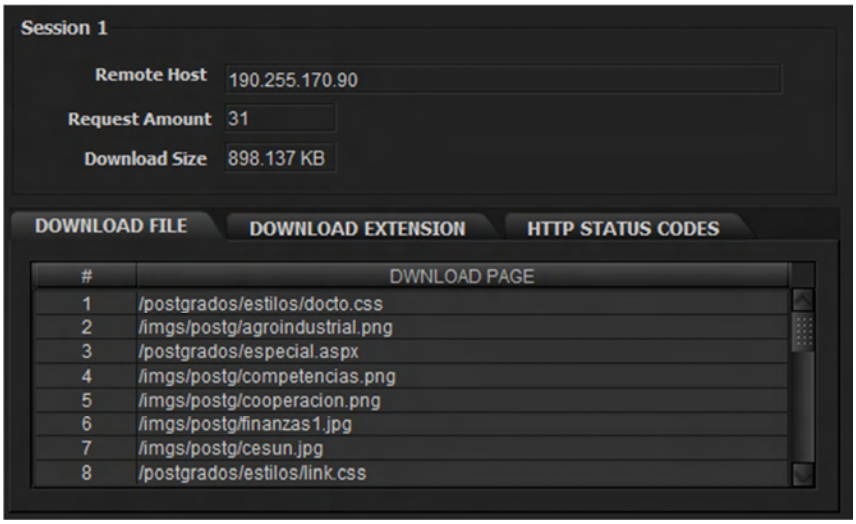
6.3 RESULTADO DE LAS PRUEBAS DE LA HERRAMIENTA POLARIS 3.0

6.3.1 Pruebas de sesiones de usuario. - El objetivo de esta prueba es determinar si la herramienta permite identificar sesiones de usuario y registrar información importante de cada sesión. En las siguientes Figura 45 y Figura 46, se observa la información sobre las sesiones de usuario obtenidas usando el método *timeout*.

Figura 45. Resultado de las pruebas - sesiones de usuario

#	REMOTE HOST	START DATE	FINAL DATE	DETAILS
1977	190.68.52.169	2012-05-16 23:33:54	2012-05-16 23:33:54	Details
1978	200.52.15.206	2012-05-16 23:38:31	2012-05-16 23:38:31	Details
1979	190.71.153.54	2012-05-16 23:41:50	2012-05-16 23:41:50	Details
1980	190.252.162.229	2012-05-16 23:45:28	2012-05-16 23:45:30	Details
1981	190.70.163.231	2012-05-16 23:51:39	2012-05-16 23:51:43	Details
1982	189.177.216.133	2012-05-16 23:52:22	2012-05-16 23:52:22	Details
1983	190.249.121.198	2012-05-16 23:52:54	2012-05-16 23:52:54	Details
1984	64.116.185.9	2012-05-16 23:55:40	2012-05-16 23:56:00	Details

Figura 46. Resultado de las pruebas - Detalles de sesión



Mediante la ejecución del proceso de sesionalización aplicado a los datos de prueba se obtuvo un total de 1984 sesiones de usuario. En la **Figura 45** se observa el registro del host remoto y la fecha y hora de inicio y finalización de cada sesión identificada. En la **Figura 46,** se observa la información asociada a cada sesión, donde se encuentra información importante como los archivos

descargados en cada sesión, las extensiones de los archivos y los códigos de estado HTTP registrados.

6.3.2 Pruebas de resumen general. - El objetivo de esta prueba es determinar si los datos generados en la ejecución del nodo Summary, permite obtener información general correcta y confiable sobre estadísticas obtenidas a partir de peticiones y visitas registradas en los archivos log. En la Figura 47, se observa la información obtenida.

Figura 47. Resultado de las pruebas - estadísticas generales

OVERVIEW	
HITS	
Total Hits	16134.0
Successful Hits	14295.0
Redirected Hits	1592.0
Failed Hits (Client Error)	244.0
Failed Hits (Server Error)	3.0
Average Hits per Day	2304.857
VISITAS	
Page View	3457.0
Average Page Views per Day	493.857
Total Visitors	1984.0
TRANSFERENCIA DE DATOS	
Total Transfer (Kilobytes)	694651.5
Average Transfer per Day (Kilobytes)	99235.929
Average Transfer per Hit (Kilobytes)	0.023

Como resultado de la prueba de estadísticas generales (**Figura 47**), se obtiene información general organizada en tres bloques: Hits, Visitas y Trasferencia de datos. En el bloque **Hits** se observa información sobre la cantidad total de solicitudes, que es la suma de las solicitudes exitosas, redireccionadas y fallidas las cuales se identifican por el código de estado de cada petición. En el bloque **Visitas** se observa el número de visitantes y páginas vistas. Y en el bloque **Transferencia de datos** se tiene información sobre la cantidad de bytes trasferidos.

6.3.3 Pruebas de actividad estadística. El objetivo de esta prueba es obtener estadísticas de tráfico web por cada día de la semana, por cada hora del día y por cada fecha diferente registrada en el archivo log. Para esta prueba se seleccionó el parámetro visitantes como criterio de medición, además de este parámetro es posible seleccionar el parámetro hits o el parámetro transferencia de datos. A continuación se observan los gráficos de barras generados en la ejecución de esta prueba:

Figura 48. Resultado de las pruebas - estadísticas por fecha

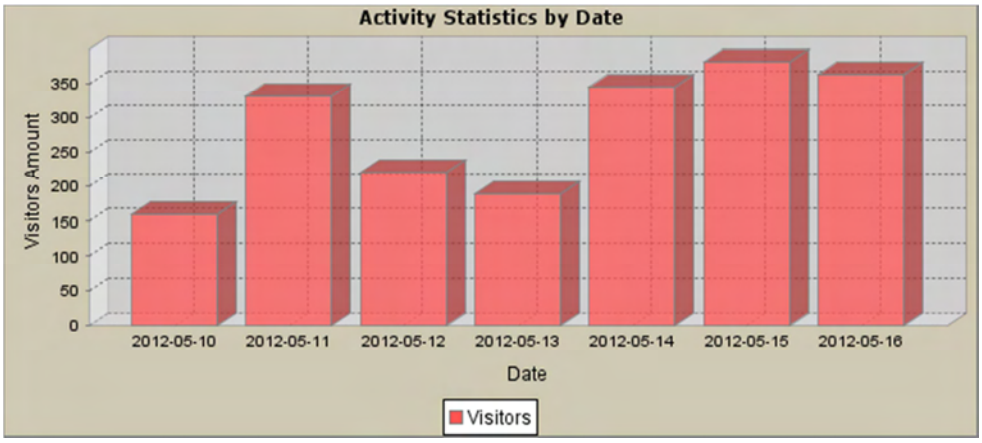


Figura 49. Resultado de las pruebas - estadísticas por día

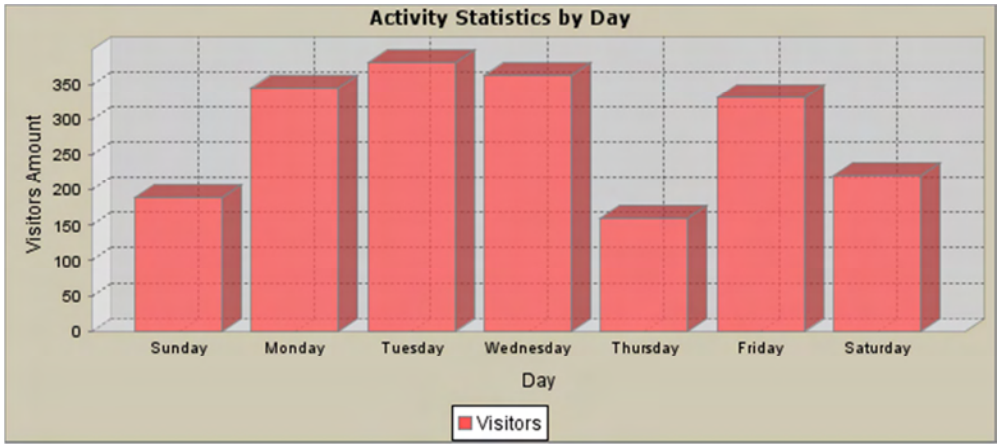
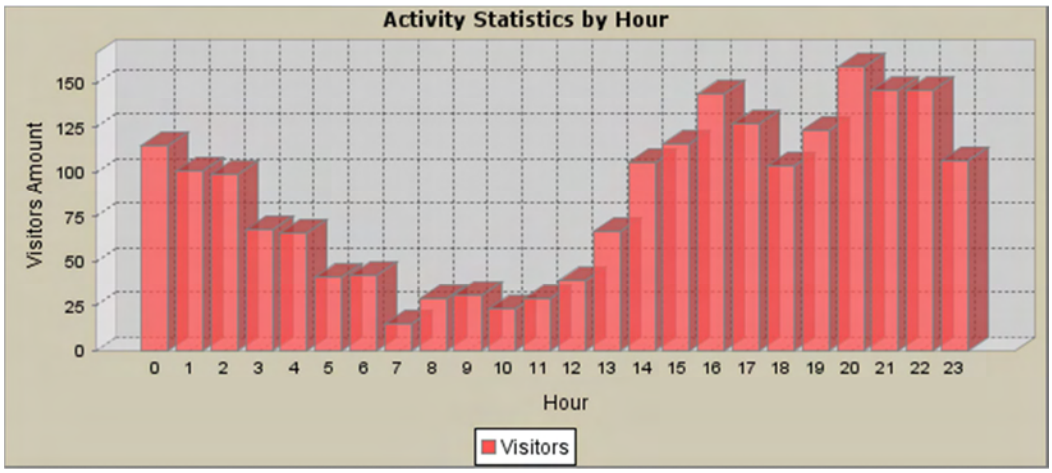


Figura 50. Resultado de las pruebas - estadísticas por hora



En los resultados obtenidos en las pruebas de actividad estadística por fecha (**Figura 48**) y actividad estadística por día (**Figura 49**) se observa movimiento en siete fechas diferentes que corresponden a los siete días de la semana, esto debido a que el archivo log analizado registra accesos realizados en un periodo de una semana. En cuanto a las estadísticas por horas del día (**Figura 50**) se observa que en el intervalo de tiempo que registra al archivo log, la mayor cantidad de visitas se hacen en las horas de la tarde y de la noche y se reducen en las horas de la mañana, siendo las 4 pm la hora de mayor tráfico en las horas de la tarde y las 8 pm la hora de mayor tráfico en las horas de la noche.

6.3.4 Pruebas de estadísticas de accesos. Para realizar las pruebas de estadísticas de accesos se selecciono como criterio de medición el parámetro visitantes. En relación con estadísticas de acceso la herramienta permite configurar el tipo de archivo y la extensión de archivo. En este caso se selecciono el tipo de archivo Web Page y la extensión de archivo aspx. En la Figura 51, se observa un gráfico circular de las estadísticas de tipos de archivos, en la Figura 52 se observa un gráfico circular de les estadísticas de extensiones de archivos y en la Figura 53 se observa un gráfico de barras de las estadísticas de accesos.

Figura 51. Resultado de las pruebas - tipos de archivo

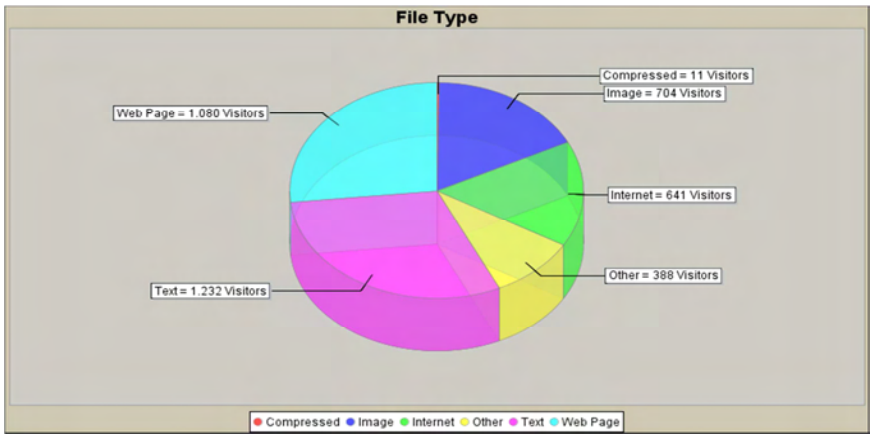


Figura 52. Resultado de las pruebas - extensiones de archivo

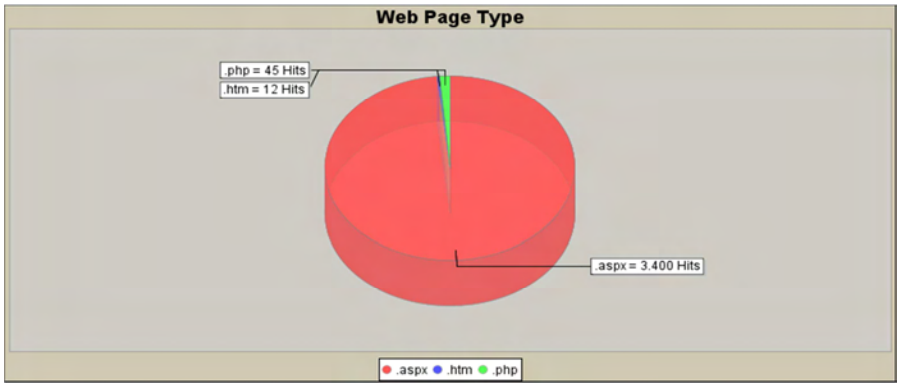
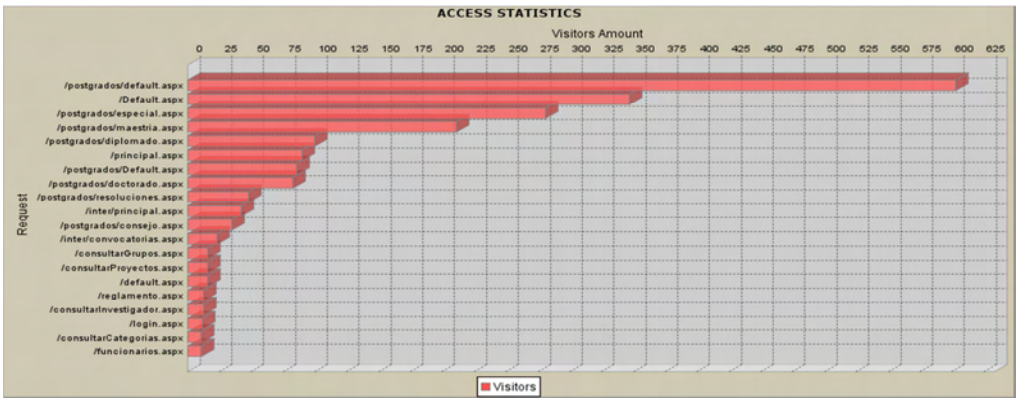


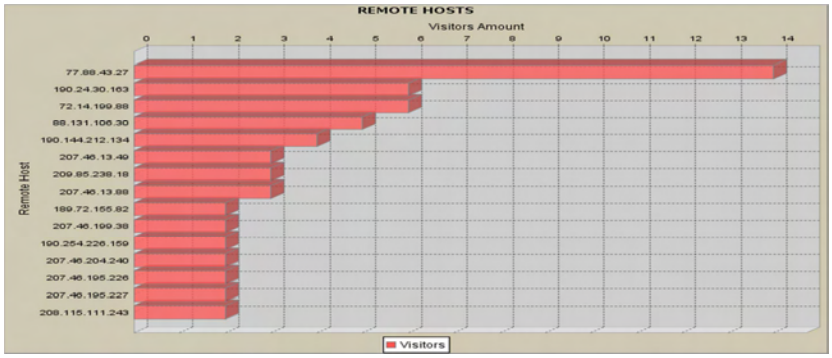
Figura 53. Resultado de las pruebas - estadísticas de acceso



En el resultado de las pruebas de estadísticas de acceso se observa que el tipo de archivo con mayor cantidad de visitantes es el de tipo Texto, seguido de los archivos tipo Pagina Web (**Figura 51**). Al analizar los archivos tipo Página Web, se puede observar mayor número de visitas a archivos con extensión .aspx y en mínima cantidad de archivos .php y .htm (**Figura 52**). En cuanto a los archivos, aspx, al realizar un análisis más especifico se tiene como resultado que el archivo con mayor cantidad de visitas es Default.aspx y en menor grado los archivos especial.aspx y maestria.aspx (**Figura 53**).

6.3.5 Pruebas de estadísticas de IPs únicas. Para realizar las pruebas de estadísticas de IPs únicas se selecciono como criterio de medición el parámetro visitantes. Los resultados obtenidos se observan en la Figura 54:

Figura 54. Resultado de las pruebas - estadísticas de agentes de usuario



La herramienta registra los datos de las 15 direcciones IP con mayor movimiento de usuarios tal como se muestra en la **Figura 54**. Los resultados obtenidos en esta prueba permiten observar que efectivamente puede haber varias visitas relacionadas con una misma dirección IP, de no ser así se hubiera tenido como resultado un visitante por cada Ip. En este caso la mayor cantidad de visitas se asocian a la dirección IP 77.88.43.27

6.3.6 Pruebas de estadísticas de agentes de usuario. Para realizar las pruebas de estadísticas de agentes de usuarios, se selecciono el parámetro hits como criterio de medición ya que el agente de usuario puede represenentar accesos realizados por clientes virtuales como robots o spiders, siendo la cantidad de hits un criterio de medición más confiable que la cantidad de visitantes. En la Figura 55, Figura 56, Figura 57 y Figura 58 se observa las estadísticas generadas en esta prueba.

Figura 55. Resultado de las pruebas - estadísticas de agentes de usuario

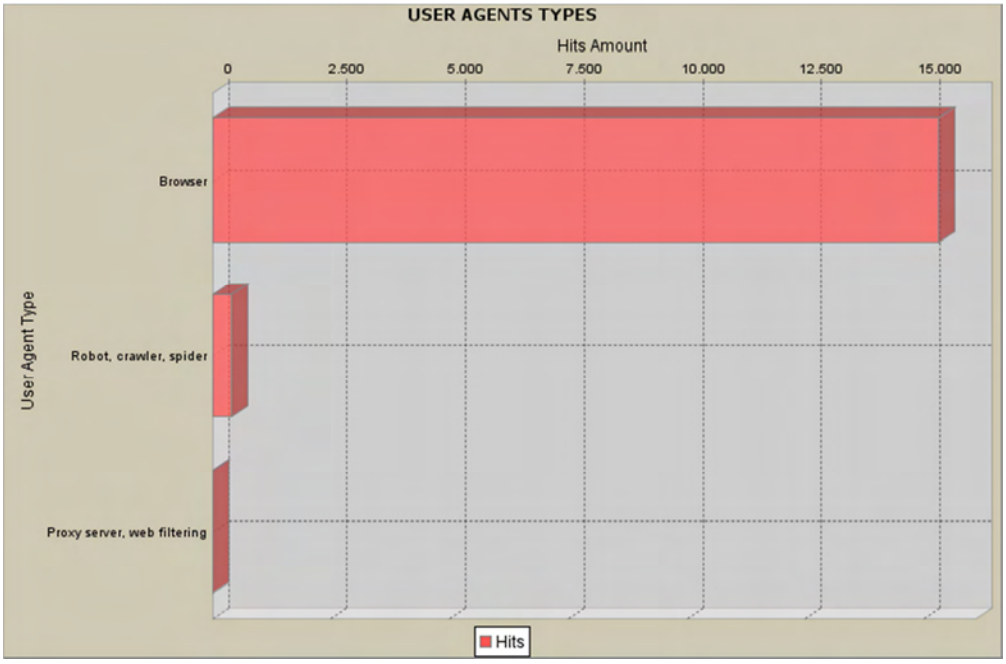


Figura 56. Resultado de las pruebas - estadísticas de sistemas operativos

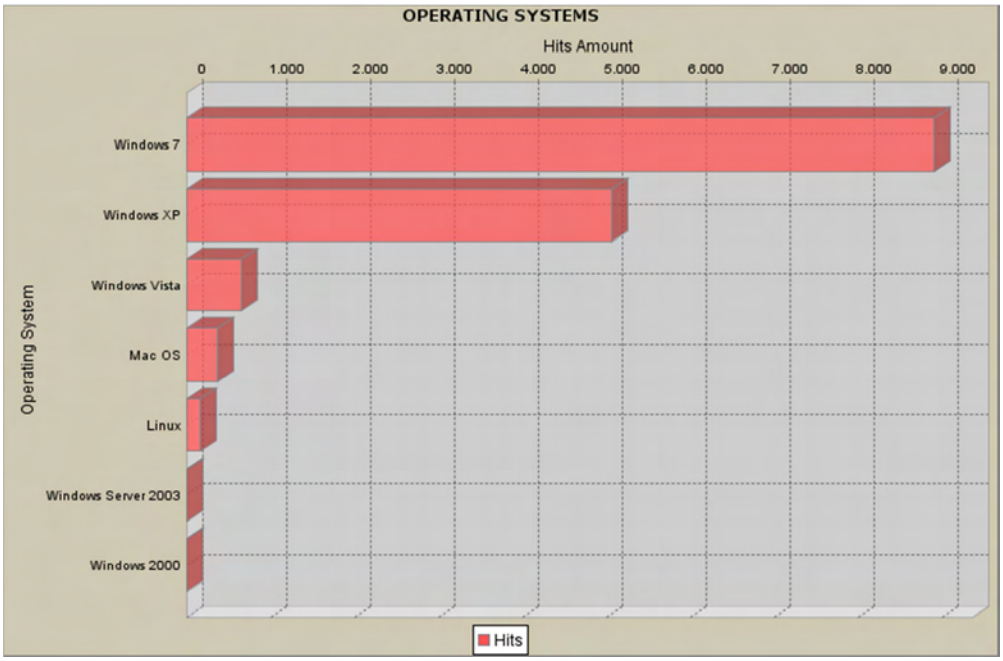


Figura 57. Resultado de las pruebas - estadísticas de browsers

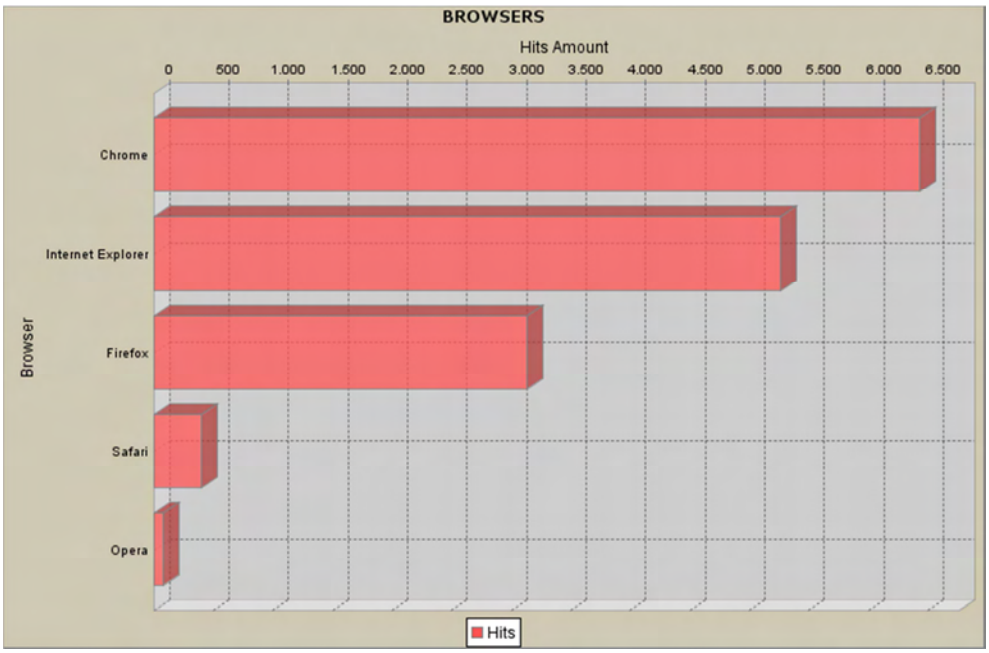
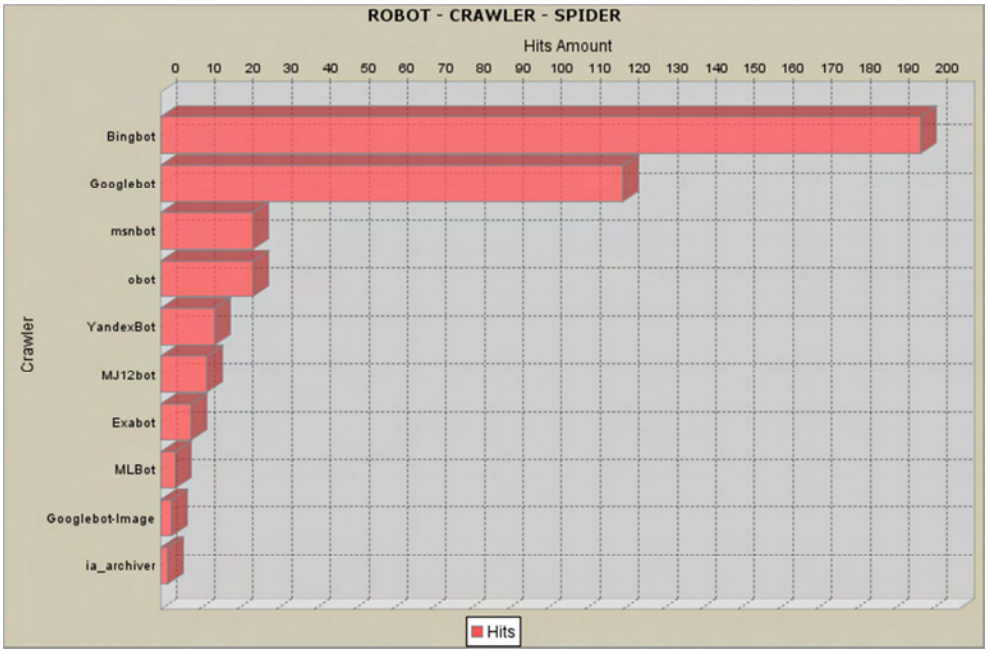


Figura 58. Estadísticas de robots, crawlers y spiders



En los resultados arrojados por esta prueba se observa que la mayor cantidad de accesos al servidor web se realizo por usuarios reales, es decir que accedieron al sitio utilizando un browser o navegador web, también se observa accesos al servidor por clientes virtuales de tipo robot, crawler y/o spider (**Figura 55**). Se observa que el sistema operativo más utilizado por los usuarios del sitio es Windows 7, seguido de Windows Xp y Windows Vista; las visitas realizadas desde sistemas operativos Linux son mínimas (**Figura 56**). En cuanto a browsers o navegadores web, se presenta la mayor cantidad de accesos utilizando Chrome seguido por Internet Explorer y Firefox (**Figura 57**). En cuanto a las estadísticas relacionadas con robots, crawlers y/o spiders, se observa que el robot Bingbot, que es el robot de búsqueda usado por el motor de búsqueda Bing, realizo la mayor cantidad de accesos al servidor, seguido por Googlebot que es el robot rastreador utilizado por Google (**Figura 58**).

6.3.7 Pruebas de estadísticas de referentes. Las estadísticas de referentes muestran los sitios de procedencia de las visitas. Para realizar las pruebas de estadísticas de Referentes se selecciono como criterio de medición el parámetro visitantes. Los resultados obtenidos se observan en la Figura 59, Figura 60, Figura 61 y Figura 62.

Figura 59. Resultado de las pruebas - estadísticas de dominios

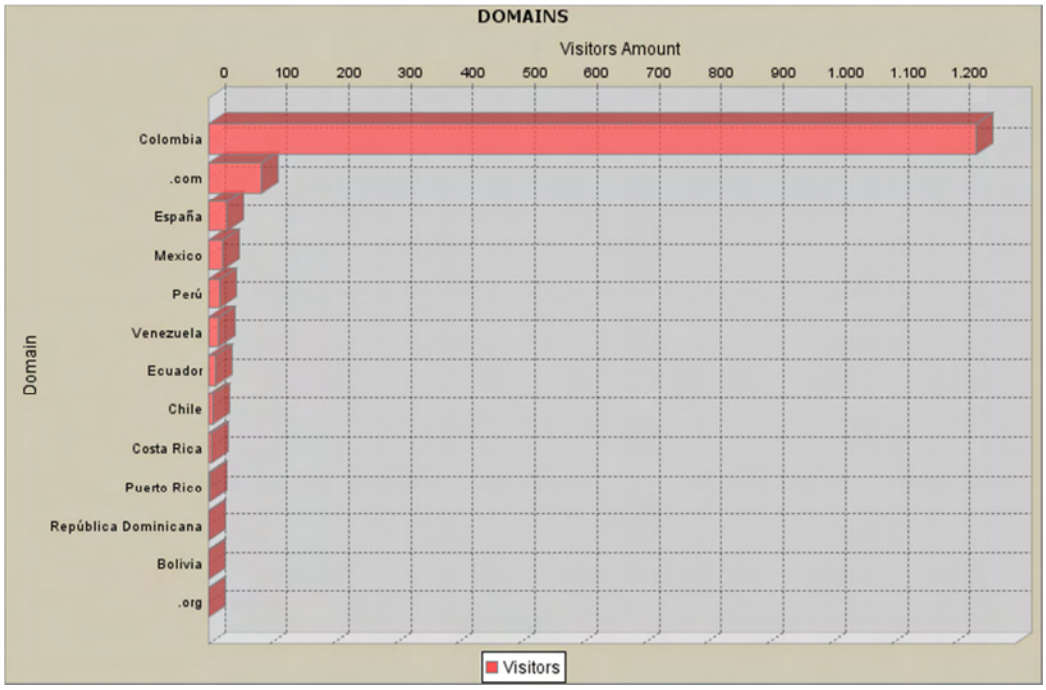


Figura 60. Resultado de las pruebas - estadísticas de motores de búsqueda

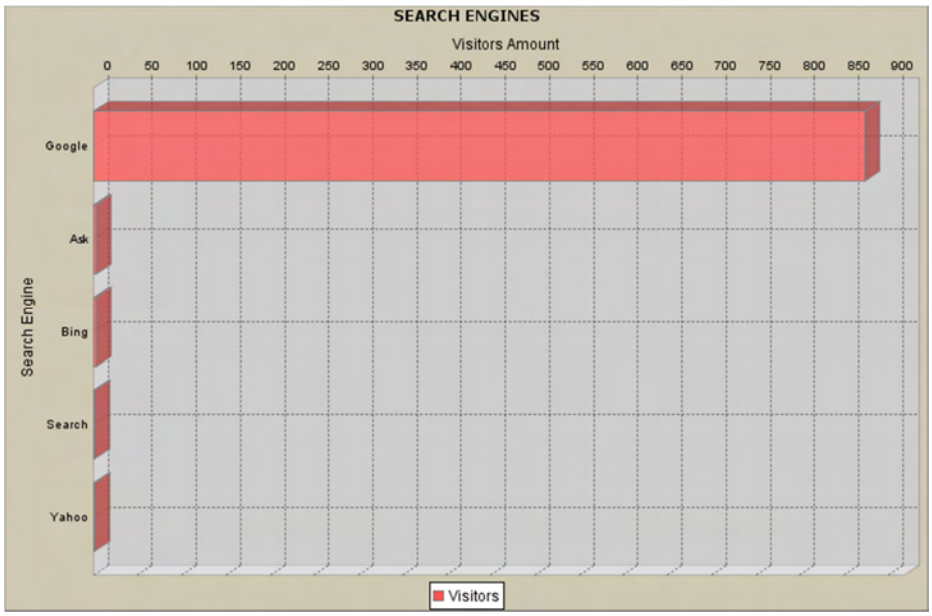


Figura 61. Resultado de las pruebas - estadísticas de sitios referentes

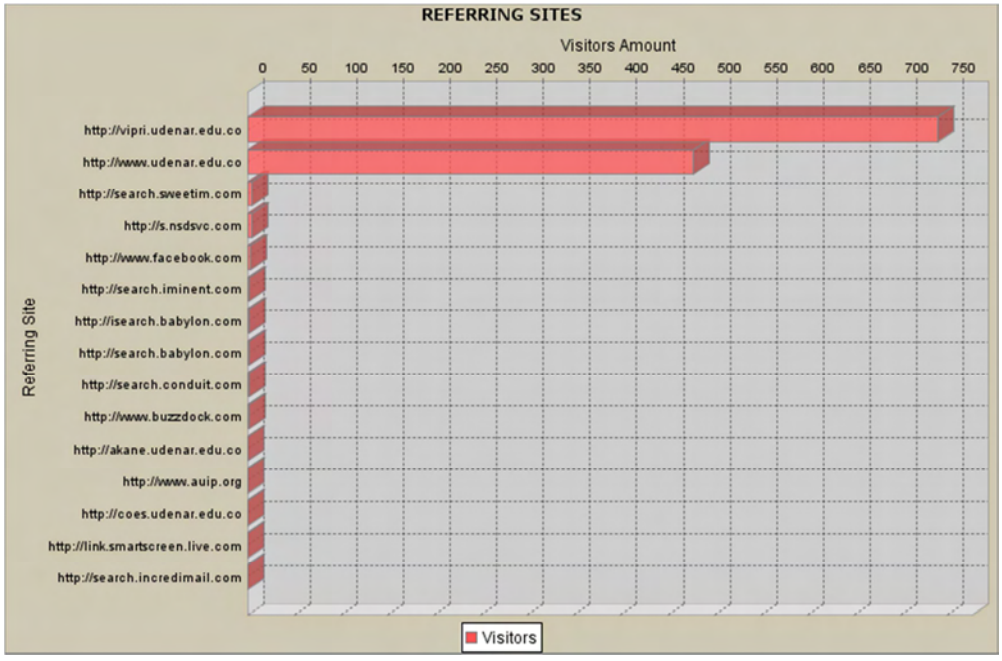
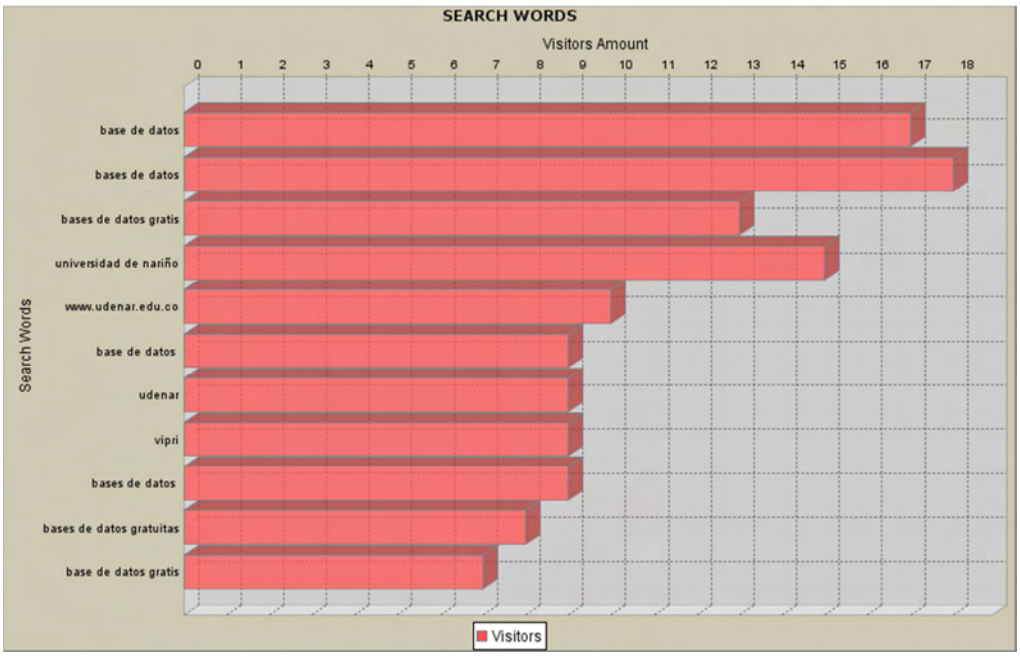


Figura 62. Resultado de las pruebas - estadísticas de palabras de búsqueda



Los resultados de las pruebas de estadísticas de referentes muestran que los motores de búsqueda utilizados para encontrar el sitio web de la Vicerectoria de Investigaciones de la Universidad de Nariño fueron Google, Bing y Search, siendo Google el más utilizado con una diferencia significativa con respecto a los demás (**Figura 59**). En cuanto a los sitios web referentes se observa gran cantidad de acceso desde el mismo sitio, es decir desde vipri.udenar.edu.co, esto representa los enlaces entra paginas del mismo sitio. Teniendo en cuenta esto, se deduce que el sito de mayor procedencia es udenar.edu.co (**Figura 60**). Con respecto a las palabras utilizadas en los motores de búsqueda, se observa que las más utilizadas fueron “bases de datos”, “base de datos” y “universidad de Nariño” (**Figura 61**). En relación con los dominios, como es normal, se observa mayor cantidad de visitantes en Colombia y una cantidad de visitas significativas realizadas desde México y España (**Figura 62**).

7. CONCLUSIONES

En este proyecto de investigación se presentó el proceso de análisis, diseño e implementación del módulo de análisis estadístico de tráfico web en la herramienta de minería web POLARIS, logrando ampliar y potenciar el campo de acción de esta herramienta.

Como resultado de este proyecto se cuenta con una nueva versión de POLARIS, la suite de minería web POLARIS Versión 3.0 con los módulos de minería de uso de la web, estructura web y análisis estadístico de tráfico web.

Tras el análisis diferentes herramientas de software que realizan procesos de análisis de archivos logs de servidores web, se observó que ninguna de ellas permite efectuar tareas de minería web y de análisis de tráfico de manera conjunta, lo cual se logro con la implementación del módulo de análisis estadístico de tráfico web en la herramienta Polaris.

La arquitectura de Polaris Versión 3.0 es modular, compuesta por los módulos de interfaz gráfica, kernel y utilidades, lo que permite la reutilización de sus componentes para incluirlos en otras herramientas de este tipo y facilita su mantenimiento.

El módulo de análisis estadístico de tráfico web de Polaris Versión 3.0, tiene una interfaz gráfica que permite al usuario visualizar gráficos y reportes de las estadísticas de tráfico web generadas por los procesos implementados en esta herramienta. En el kernel de este módulo se encuentran implementadas técnicas y procedimientos que permiten realizar análisis estadístico de tráfico web a partir de los registros almacenados en los archivos logs de servidor web.

Para realizar un análisis completo de tráfico web, se creó una base de datos en la cual se almacena datos relevantes relacionados con motores de búsqueda, navegadores web, crawlers, spiders, robots y tipos de archivos, esta información se usa para realizar comparación de datos según sea necesario.

Los resultados de las pruebas realizadas con el módulo de análisis estadístico de tráfico web la herramienta Polaris Versión 3.0 y utilizando los archivos logs de acceso de la Universidad de Nariño, permiten observar que esta herramienta implementa importantes características de las herramientas analizadas además de importantes mejoras como la selección de diversos tipos de gráficos y criterios de medición.

La herramienta Polaris Versión 3.0 está desarrolla bajo Java, que es un lenguaje de programación libre y multiplataforma, esto la convierte en una herramienta de software libre y portable a cualquier sistema operativo.

8. RECOMENDACIONES

Hacer uso de la herramienta en el desarrollo de investigaciones que estén relacionadas con minería web, con el fin de el fin de evaluar su comportamiento y resultados en casos reales.

Incluir dentro del portal web del grupo de investigación GRIAS un enlace a esta herramienta con el fin de que sea conocida, descargada y utilizada por el mundo entero.

Implementar el módulo de minería de contenido web con el fin de lograr que POLARIS se convierta en una suite de minería de Web.

Realizar futuros estudios a la estructura y organización del portal Web de la Universidad de Nariño.

REFERENCIAS BIBLIOGRÁFICAS

[1] **ALDANA J.F.** Metadata functionality for semantic Web integration. Proceedings of the Seventh International Society of Knowledge Organization (ISkO'02) conference. Granada : 2002. p. 10-13.

[2] **ALTERWIND THE SOFTWARE COMPANY.** AlterWind Log Analyzer. [En línea]. <http://www.alterwind.com/loganalyzer/>. [Citado en 26 de febrero de 2012].

[3] **ANALOG.** The most popular log file analyser in the world. [En línea]. <http://www.analog.cx/>. [Citado en 26 de febrero de 2012].

[4] **AWSTATS OFFICIAL WEB SITE.** What is Awstats. [En línea]. <http://awstats.sourceforge.net/>. [Citado en 26 de febrero de 2012].

[5] **BAEZA YATES, Ricardo.** *Del HTML A La Fidelidad: Un modelo integral para el diseño web. No.13 (Feb.,2004).* [Citado en 28 de febrero de 2012].

[6] **BAEZA YATES, Ricardo.** Sitio Web Personal de Ricardo Baeza-Yates: Excavando la Web <http://www.dcc.uchile.cl/~rbaeza/spanish.html>.

[7] **BAEZA YATES, Ricardo y POBLETE, B.** Una herramienta de minería de consultas para el diseño de contenido y la estructura de un sitio web. Granada : Thomson, 2005. p. 39-48. - ISBN: 84-9732-449-8.

[8] **BAEZA YATES, Ricardo y RIBIERTO, Neto B.** Modern information retrieval. Addison Wesley Longman. England: 1999.

[9] **BERNERS-LEE, T.** HTTP: A protocol for networked information**.** [En línea]. (1993). Disponible en: <http://info.cern.ch/hypertext/WWW/MarkUp/HTTP.html>. [Citado en 11 de enero de 2012].

[10] **BERTHOLD, M. y HAND, D.J.** Intelligent Data Analysis: An Intoduction. Springer, 2003. - 2ndEdition.

[11] **CIBERNETIA.** Fundamentos de la web: El protocolo HTTP. [En línea]. <http://www.cibernetia.com/manuales/introduccion_aplicaciones_web/2_1_fundamentos_web.php>. [Citado en 12 de enero de 2012].

[12] **ETZIONI, O.** The World Wide Web: Quagmire or gold mine? Communications of the ACM . 1996. Vol. 39 : 11. p. 65 - 68.

[13] **ETZIONI, Shakes J. y LANGHEINRICH, M.** Ahoy! The homepage finder. Proceedings of the 6th WWW conference. Santa Clara : 1997.

[14] **FUENTES, Rubén y PAVÓN, Juan.** Agentes para la recuperación de información especializada en Internet. Madrid: Facultad de Informática de la Universidad Complutense de Madrid, 2003.

[15] **GIBSON D., KLEINBERG, Jon y RAGHAVAN P.** Inferring Web Communities from Link Topologies. Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia: Links, objects, time and space structure in hypermedia system. 1998. p. 225-234.

[16] **GONZÁLEZ DÍAZ, Ernesto y PÉREZ HERNÁNDEZ, Zady**. Técnicas de minería de datos. [En línea] . <http://www.monografias.com/trabajos55/mineria-de-datos/mineria-de-datos.shtml>. [Citado en 18 de enero de 2012].

[17] **GONZÁLEZ, G. y LLUÍS, J.** Preprocesamiento de base de datos masivas y multidimensionales en minería de uso web para modelar usuarios: Comparacion de herramientas y técnicas con un caso de estudio. Granada : Thomson, 2005. p. 193 - 202.

[18] **Hernández Orallo J., Ramírez M.J. y Ferri C.** Introducción a la minería de datos. Pearson Educación. Madrid : Prentice Hall, 2004.

[19] **HISPAZONE.** Los Archivos: Tipos, extenciones y programas para su uso. [En línea]. <http://www.hispazone.com/Guia/91/Los-archivos-tipos-extensiones-y-programas-para-su-uso.html>. [Citado en 10 de febrero de 2012].

[20] **INTERNET FAQ ARCHIVES.** Internet RFC/STD/FYI/BCP Archives [En línea]. <http://www.faqs.org/rfcs/>. [Citado en 11 de enero de 2012].

[21] **KOHAAVI, R. y JOHN, G.** Wrappers for the features subset selection. Artificisl Intelligence. 1997. Vol. 97 : 1-2 : p. 273-324.

[22] **LAMARCA LAPUENTE, María Jesús.** Hipertexto: El nuevo concepto de documento en la cultura de la imagen. Tesis Doctoral. Madrid. Universidad Complutense de Madrid. Facultad de Ciencias de la Información. Dpto. de Biblioteconomía y Documentación.

[23] **IAN, H. Witten y EIBE, Frank.** Data Mining: Practical Machine Learning Tools and techniques with Java Implementations. Morgan Kaufmann, 2000.

[24] **MALKIN, G. y LAQUEY PARKER, T.** Internet Users' Glossary. En: User Glossary Working Group of the User Services Area. [En línea]. (2003). Disponible en: < http://www.ietf.org/rfc/rfc1392.txt>. [Citado en 11 de enero de 2012].

[25] **MARBÁN, O y MENASALVAS, E.** Estudio de perfiles de visitantes de un sitio Web a partir de los logs de los servidores Web aplicando técnicas de Data Mining (Web Mining) . - Madrid : 2002.

[26] **MARTIN, Gregorio.** Arquitectura de la Web: Conceptos. En: Tecnologías XML. [En línea]. (12 de noviembre de 2004). Disponible en: <http://www.w3c.es/gira/paradas/presentaciones/Gregorio_XML.pdf >. [Citado en 11 de enero de 2012].

[27] **MICROSOFT TECHNET.** Formatos de registro. [En línea]. <http://technet.microsoft.com/es-es/library/cc780772%28v=ws.10%29.aspx>. [Citado en 16 de enero de 2012].

[28] **NETCRAFT**. January 2012 Web Server Survey. [En línea]. <http://news.netcraft.com/archives/2012/01/03/january-2012-web-server-survey.html#more-5297>. [Citado en 15 de enero de 2012].

[29] PAL, S.K., TALWAR, V. y MITRA, P. Web Mining in Soft Computing Framework: Relevance, state of the art and future directions [Sección del libro] // IEEE Transactions on Neural Netwoks. -2002. Vol. 13 : 5.

[30] **POLO, Luciano.** World Wide Web Technology Architecture. En: World Wide Web Technology Architecture: A conceptual analysis [En línea]. (2003). Disponible en: <http://newdevices.com/publicaciones/www/>. [Citado en 10 de enero de 2012].

[31] **OPENUP**. Introduction to OpenUp: What is OpenUp. [En línea]. <http://epf.eclipse.org/wikis/openup/> [Citado en 18 de febrero de 2012].

[32] **ROBERT Cooley, BAMSHAD Mobasher, JAIDEEP Srivastava**. Data Preparation for Mining World Wide Web Browsing Patterns., Department of Computer Science and Engineering University of Minnesota. 1999.

[33] **SMITH B, WELTY C**. What is Ontology? Ontology: Towards a new synthesis. Proceedings of the Second.

[34] **THE INTERNET ENGINEERING TASK FORCE.** [En línea]. <http://www.ietf.org/>. [Citado en 11 de enero de 2012].

[35] **THE WORLD WIDE WEB CONSORTIUM (W3C).** Architecture of the World Wide Web, Volume One. En: W3C Recommendation. [En línea]. (15 de diciembre de 2004). Disponible en: < http://www.w3.org/TR/webarch/#acks >. [Citado en 10 de enero de 2012].

[36] **THE WORLD WIDE WEB CONSORTIUM (W3C).** Extended Log File Format.

[En línea]. Disponible en <http://www.w3.org/TR/WD-logfile.html >. [Citado en 17 de enero de 2012].

[37] **THE WORLD WIDE WEB CONSORTIUM (W3C).** Hypertext Transfer Protocol: Status Code Definitions. [En línea]. <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>. [Citado en 24 de febrero de 2012].

[38] **THE WORLD WIDE WEB CONSORTIUM (W3C).** Logging Control In W3C httpd - The Common Logfile Format. [en línea]. (Julio 1995). Disponible en <http://www.w3.org/Daemon/User/Config/Logging.html>. [Citado en 17 de enero de 2012].

[39] **THE WORLD WIDE WEB CONSORTIUM (W3C).** Overview of SGML Resources. [En línea]. Disponible en: <httphttp://www.w3.org/MarkUp/SGML/>. [Citado en 11 de enero de 2012].

[40] **TIMARÁN, R., DAZA, J., ZULETA, A., ANGULO, D.** Polaris: Una Herramienta para Minería de Uso de la Web. Trabajo de Grado. San Juan de Pasto. Universidad de Nariño. Facultad de Ingeniería. Departamento de Ingeniería de Sistemas. 2007. 271 p.

[41] **USER-AGENTS.ORG.** List of User-Agents (Spiders, Robots, Crawler, Browser). [En línea]. <http://www.user-agents.org/>. [Citado en 12 de febrero de 2012].

[42] **USER AGENT STRING.** List of User Agent Strings. [En línea]. <http://www.useragentstring.com/pages/useragentstring.php/>. [Citado en 15 de febrero de 2012].

[43] **UTF8-CHARTABLE.** UTF-8 encoding table and Unicode characters. [En línea]. < http://www.utf8-chartable.de>. [Citado en 18 de febrero de 2012].

[44] **VEGAS, Jesus.** La Interacción entre el Browser y el Servidor. En: HyperText Transfer Protocol, HTTP. [En línea]. (2002). Disponible en: <http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node14.html>. [Citado en 12 de enero de 2012].

[45] **WEBLOG EXPERT. WEBLOG** EXPERT: Information. [En línea]. <http://www.weblogexpert.com/>. [Citado en 26 de febrero de 2012].

[46] **WIKIPEDIA LA ENCICLOPEDIA LIBRE.** Tràfico web [En línea]. <http://es.wikipedia.org/wiki/Tr%C3%A1fico_web>. [Citado en 10 de enero de 2012].

[47] **WIKIPEDIA LA ENCICLOPEDIA LIBRE.** Dominio de nivel superior

geográfico: Lista de dominios de nivel superior geográfico. [En línea]. <http://es.wikipedia.org/wiki/Dominio_de_nivel_superior_geogr%C3%A1fico>. [Citado en 20 de febrero de 2012].

[48] **WIKIPEDIA LA ENCICLOPEDIA LIBRE.** Log (registro). [En línea]. <http://es.wikipedia.org/wiki/Log_%28registro%29>. [Citado en 15 de enero de 2012].

[49] **WIKIPEDIA LA ENCICLOPEDIA LIBRE.** World Wide Web [En línea]. <http://es.wikipedia.org/wiki/World_Wide_Web>. [Citado en 10 de enero de 2012].

ERROR: syntaxerror
OFFENDING COMMAND: --nostringval--

STACK:

/Title
()
/Subject
(D:20121203143715-06’00’)
/ModDate
()
/Keywords
(PDFCreator Version 0.9.5)
/Creator
(D:20121203143715-06’00’)
/CreationDate
(Administrador)
/Author
-mark-