UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ESCUELA DE CIENCIAS DE LA COMPUTACIÓN

"Diseño de un modelo para Agentes basados en Redes Neuronales para WebMining"

Tesis de grado previa a la obtención del título de: Ingeniero en Sistemas Informáticos y Computación

LINEA DE INVESTIGACIÓN: Redes Neuronales

AUTORA: Gimena Anavel Moreno Estrada

DIRECTOR: Ing. Nelson Piedra

CODIRECTOR: Ing. Janneth Chicaiza

LOJA – ECUADOR

2008

CERTIFICACIÓN

ING. NELSON PIEDRA
DIRECTOR DE TESIS

CERTIFICA:

Haber dirigido, supervisado y revisado el presente trabajo de investigación, que se ajusta a las normas establecidas por la Escuela de Ciencias de la Computación, previo a la obtención del título de INGENIERO EN SISTEMAS INFORMÁTICOS Y COMPUTACIÓN; por tanto, autorizo su presentación.

…….

ING. NELSON PIEDRA
DIRECTOR DE TESIS

Loja, de Marzo de 2008

AUTORIA

Las ideas y las opiniones expuestas en el presente informe de investigación son de exclusiva responsabilidad de su autor.

.....

GIMENA ANAVEL MORENO ESTRADA

CESIÓN DE DERECHOS

Yo, Gimena Anavel Moreno Estrada, declara conocer y aceptar la disposición del Art. 67 del Estado Orgánico de la Universidad Técnica Particular de Loja, que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos, y tesis de grado que se realicen a través o con el apoyo financiero, académico o institucional (operativo) de la Universidad”.

.....

GIMENA ANAVEL MORENO ESTRADA

DEDICATORIA

Con inmenso amor dedicado este trabajo primeramente a Dios Nuestro Señor, por haberme permitido continuar con mis estudios y completar esta meta.

A mis padres, Rubén y Rosa, a quienes les agradezco por su amor, comprensión, confianza, entrega y por todo su apoyo económico recibido durante esta etapa de mi vida, y sobre todo por haberme dado sus sabios consejos en el momento oportuno.

También dedicó a mis hermanos por haberme brindado su apoyo incondicional y de manera muy especial a dos personas aunque ya no estén presentes físicamente aquí para verme, se que estarían muy orgullosos de mi, gracias por que fueron mi motivo e inspiración para el desarrollo de este trabajo y por ende mi superación personal.

A todas y cada una de las personas que desde que inicié mis estudios me acompañaron y compartieron conmigo alguna etapa, les dedico esta tesis con mucho cariño.

Gimena Anavel

AGRADECIMIENTO

Quiero agradecer a la Universidad Técnica Particular de Loja por haberme brindado la oportunidad de superarme, a la Escuela de Ciencias de la Computación y a los distinguidos maestros por sus enseñanzas.

Mi agradecimiento especial al Ingeniero Nelson Piedra, Director de Tesis, por sus conocimientos y colaboración en esta tesis, por su tiempo y guía para la finalización exitosa de este presente trabajo.

A la Ingeniera Janneth Chicaiza, Codirector-Tesis, por su apoyo permanente, quien contribuyó con su valiosa colaboración y comentarios en beneficio a la realización de mi tesis.

En fin a todas y cada una de las personas que me ayudaron en el desarrollo y culminación de este trabajo investigativo, mis más sinceros agradecimientos para ellos.

La autora

INDICE GENERAL

Páginas

CERTIFICACIÓN II
AUTORIA. III
CESIÓN DE DERECHOS IV
DEDICATORIA V
AGRADECIMIENTO VI
INDICE GENERAL VII
INDICE DE TABLAS X
INDICE DE FIGURAS XI
INTRODUCCIÓN XII
OBJETIVOS. XIII

FASE I: ESTADO DEL ARTE 14

1.1. INTRODUCCIÓN 15
1.2. PROPÓSITO 15
1.3. RESULTADO ESPERADO 15
1.4. WEB MINING 16
1.4.1. EXPLOTACIÓN DE DATOS DEL WEB MINIG: CONTENIDOS, ESTRUCTURA Y USO. 16
1.4.1.1. MINERÍA DE CONTENIDO WEB 17
1.4.1.2. MINERÍA DE ESTRUCTURA WEB 17
1.4.1.3. MINERÍA DE USO WEB 18
1.4.2. PROCESO DE EXPLOTACIÓN DE DATOS DEL WEB 19
1.4.3. TÉCNICAS MÁS USADAS EN LA MINERIA DE DATOS 20
1.5. REDES NEURONALES 21
1.5.1. CONCEPTO 21
1.5.2. ESTRUCTURA Y FUNCIONAMIENTO 21
1.5.2.1. ESTRUCTURA 21
1.5.2.2. FUNCIONAMIENTO 22
1.5.3. APRENDIZAJE 23
1.5.3.1. APRENDIZAJE SUPERVISADO 23
1.5.3.2. APRENDIZAJE NO-SUPERVISADO 24
1.5.4. TRABAJOS RELACIONADOS DE LAS RNA CON WEB MINING 24
1.6. AGENTES 25
1.6.1. CONCEPTOS EXISTENTES 25
1.6.2. CARACTERÍSTICAS 25
1.6.3. ARQUITECTURA DE LOS AGENTES 26
1.6.4. APLICACIONES DE LOS AGENTES 26

FASE 2: ANÁLISIS Y DISEÑO 28

2.1. INTRODUCCIÓN 29
2.2. PRÓSITOS 29
2.3. RESUSTADO ESPERADO 29

2.4. ANÁLISIS Y SELECCIÓN DE LAS TÉCNICAS PARA LA REALIZACIÓN DEL MODELO 30
2.4.1. ANÁLISIS DE LAS REDES NEURONALES 30
2.4.2. SELECCIÓN DE LA RNA 31
2.4.3. SELECCIÓN DEL AGENTE 31
2.5. MODELO DEL AGENTE BASADO EN REDES NEURONALES PARA WEB MINING 32
2.5.1. RECOLECCIÓN DE LOG 33
2.5.1.1. FORMATO DE REGISTROS LOG 33
2.5.2. PROCESAMIENTO DE LOG 36
2.5.3. DESCUBRIMIENTO DE PATRONES 36
2.5.3.1. DISEÑO DE LA RED NEURONAL PARA WEB MINING 37
2.5.4. ANÁLISIS DE PATRONES 38

FASE 3: IMPLEMENTACIÓN DE SOLUCIÓN 39

3.1. INTRODUCCION 40
3.2. PRÓPOSITO 40
3.3. RESULTADO ESPERADO 40
3.4. HERRAMIENTA MATLAB VER 7.0.1 41
3.5. IMPLEMENTACIÓN EN MATLAB DEL DISEÑO DEL AGENTE BASADO EN RED NEURONAL PARA WEB MINING 41
3.6. DESCRIPCIÓN DE LAS FUNCIONES UTILIZADAS DEL TOOLBOX PARA EL ENTRENAMIENTO DE LA RNA 45
3.7. LABORATORIO 45
3.7.1. R EQUISITOS PARA INSTALACIÓN DE HERRAMIENTA MATLAB 45
3.7.2. DESARROLLO DE LA PRÁCTICA 46
3.7.3. PRUEBAS 48
3.7.4. VALIDACIÓN 49

FASE 4: DISCUSIÓN, CONCLUSIONES Y RECOMENDACIONES 51
4.1. DISCUSIÓN 52
4.2. CONCLUSIONES 52
4.3. RECOMENDACIONES 53

BIBLIOGRAFIA 54

ANEXOS…… 58

ANEXO A…. 59
ELEMENTOS DE UNA RED NEURONAL 59
TOPOLOGÍAS 61

ANEXO A-1.. 63
TIPOS DE REDES NEURONALES 63

ANEXO B…. 80
TIPOS DE AGENTES 80

ANEXO C….. 82
LOGS - RECOLECTADOS A NIVEL DE SERVIDOR WEB 82

ANEXO D….. 83
TABLAS DE CÓDIGOS DE PETICIÓN HTTP..... 83

ANEXO E…..... 85
LABORATORIO DE PROCESAMIENTO DE DATOS..... 85

ANEXO E-1.. 92
RESULTADOS DEL LABORATORIO 92

ANEXO F…..... 104
CÓDIGO DE LA IMPLEMENTACIÓN EN MATLAB 104

INDICE DE TABLAS

Páginas

Tabla 1. Ventajas y desventajas de las RNAs 30
Tabla 2. Características de las RNAs 31
Tabla 3. Aprendizaje de la RBF con error de 0.1 49
Tabla 4. Aprendizaje de la RBF con error de 0.6 49
Tabla 5. % de error de aprendizaje de la RBF 50

INDICE DE FIGURAS

Páginas

Figura 1: Categorías de la Minería Web 16
Figura 2: Proceso del Web Mining 19
Figura 3: Estructura de una red neuronal artificial 21
Figura 4: Funcionamiento de una neurona artificial 22
Figura 5: Aprendizaje por corrección de error 23
Figura 6: Modelo de Agente basado en RN para WM 32
Figura 7: Fragmento de un fichero log correspondiente a dos usuarios 34
Figura 8: Diseño de la RNA 37
Figura 9: Menú 46
Figura 10: Ventana para cargar archivo 47
Figura 11: Aprendizaje de la red neuronal de Rase Radial 47
Figura 12: Presentación de los resultados 48
Figura 13: Diseño de RBF en Simulink 49
Figura 14: Resultados de la Tabla 5 50

INTRODUCCIÓN

La Web presenta una fuerza impulsora clave para muchas aplicaciones en las que un usuario interactúa con una sociedad sin fines de lucro. Considerando esto, en los últimos años la tasa de crecimiento de Internet es tan alta, que a diario se desarrollan nuevos sitios con datos que pueden ser relevantes o no a nuestros intereses, como ser en el ambiente educativo. En función de esto surge el Web Mining o exploración Web.

El Web Mining, aborda el estudio de varios aspectos esenciales de un sitio y ayuda a descubrir tendencias y relaciones en el comportamiento de los usuarios, y por ende nos proporciona información útil para el presente y el futuro, por lo que se convertido en un tema de vital importancia para todos aquellos que emplean el dominio virtual.

Para el desarrollo de la presente tesis, se ha organizado en cuatro fases y cada una de ellas incluye los siguientes puntos: Introducción, Propósitos, Resultados Esperados, Desarrollo y Bibliografía. A continuación se ofrece un pequeño resumen de cada fase.

Fase 1: Describe el Estado del Arte, la misma que se refiere al estudio relativo de todos los conceptos y temas que intervienen en el desarrollo de la tesis.

Fase 2: Diseño y Análisis, toma como base la fase 1 y se adiciona más información en cuanto a RNA, parar tener una mejor comprensión en cuanto a rendimiento y por ende poder seleccionar la RNA que mejor se adapte a este tipo de problemas, así como también la selección del agente. Realizados estos análisis, se procede a crear el modelo de la simulación, la misma que describe en forma detallada cada componente del modelo.

Fase 3: Implementación de la Solución, aquí se define la herramienta a utilizar para el modelo de la simulación realizado en la fase 2.
Fase 4: Discusión, Conclusiones y Recomendaciones, la última de este proyecto. Se refiere a todo lo que ha ocurrido durante el desarrollo de esta tesis, las mismas que sirven como sustentación para garantizar los resultados obtenidos después de llevar a cavo esta tesis.

Además esta tesis incluye una notación para las RNA, figuras, anexos e información relevante, para concluir con la bibliografía utilizada.

OBJETIVOS

Objetivo General

- Diseñar el modelo para un Agente basados en Redes Neuronales para WebMining.

Objetivos Específicos

- Analizar e investigar el estado del arte de la gestión del conocimiento.
- Aplicar las técnicas de clasificación de agrupamiento: Web Content Mining (Minería de Contenido Web), Web Structure Mining (Minería de Estructura de Web), Web Usage Mining (Minería de Uso de Web).
- Utilizar una herramienta de cálculo para procesar los datos y obtener información estadística, que nos proporcionen criterios de mejora de un sitio, cuya información debe considerar los siguientes puntos:
  - ✓ Cantidad de visitas por día.
  - ✓ Horas pico y horas de baja audiencia.
  - ✓ Páginas más visitadas.
  - ✓ Páginas de entrada y salida más frecuentes del sitio.
  - ✓ Desde que regiones o puntos de la red están accediendo al sitio web.
- Desarrollar el algoritmo para que el Agente inteligente pueda reconocer mediante las redes neuronales el comportamiento de los usuarios de Internet.
- Construir la red neuronal para dicho modelo.

FASE I

ESTADO DEL ARTE

1.1. INTRODUCCIÓN

La Web es el fenómeno primordial del Internet que se caracteriza por estar definida por un conjunto heterogéneo de elementos, los cuales hacen un importantísimo medio de difusión y comunicación para la sociedad. Actualmente la tasa de crecimiento de Internet es tan alta, esto se debe a que las organizaciones e instituciones de toda índole van creciendo y por ende su volumen de los datos va en aumento, siendo más dificultoso su manejo y posterior uso como información de apoyo a decisiones. Es por ello que surge la Minería de Datos (DM – Data Mining) como una tecnología de extracción de conocimiento a partir de grandes cantidades de datos, "con objeto de describir de forma automatizada modelos previamente desconocidos, predecir de forma automatizada tendencias y comportamientos" [MARTINELLI D., 2006].

La aplicación de la MD "ha diversos tipos de información generalmente provenientes de bases de datos textuales o documentales y de internet ha propiciado la aparición de nuevas áreas de estudio específicas para la explotación de este tipo de datos" [ESCOBAR V., 2007]. Una de estas áreas es la Minería de Texto y la otra área es la Minería Web llamada en inglés Web Mining.

En está fase se estudiará la Minería Web junto con sus dominios y proceso de explotación de datos. También hablaremos de una de sus técnicas siendo está las Redes Neuronales Artificiales, al igual que los Agentes ya que forman parte de este trabajo.

1.2. PROPÓSITO

- Investigar, analizar y comprender la minería web, así como también las redes neuronales y los agentes.

1.3. RESULTADO ESPERADO

- Conocer los trabajos relacionados de minería web con redes neuronales y agentes.

1.4. WEB MINING

Web Mining (Minería Web) es un término que fue acuñado por Oren Etzioni en 1996 y actualmente es un área de investigación extensa. "Algunos autores definen a la web mining como el uso de técnicas para descubrir y extraer de forma automática información de los documentos y servicios de la web. Según M. Scotto, la web mining es el proceso de descubrir y analizar información "útil" de los documentos de la Web. Sin embargo y tomando en cuenta lo expuesto en la introducción la minería web se puede definir como el descubrimiento y análisis de información relevante que involucra el uso de técnicas y acercamientos basados en la minería de datos (Data Mining) orientados al descubrimiento y extracción automática de información de documentos y servicios de la Web, teniendo en consideración el comportamiento y preferencias del usuario" [GYVES C.]. Entre los objetivos principales del web mining se tiene:

- "Descubrir recursos, extraer información, analizar datos e inferir generalidades.
- Obtener nuevos conocimientos provenientes de la información disponible en la W3" [GARCÍA L.].
- Encontrar información relevante.
- Optimizar el diseño y estructura del sitio web.

1.4.1. EXPLOTACIÓN DE DATOS DEL WEB MINIG: CONTENIDOS, ESTRUCTURA Y USO

La minería web se divide en tres dominios que comprenden el contenido del sitio, la estructura de navegación y el comportamiento de los usuarios:

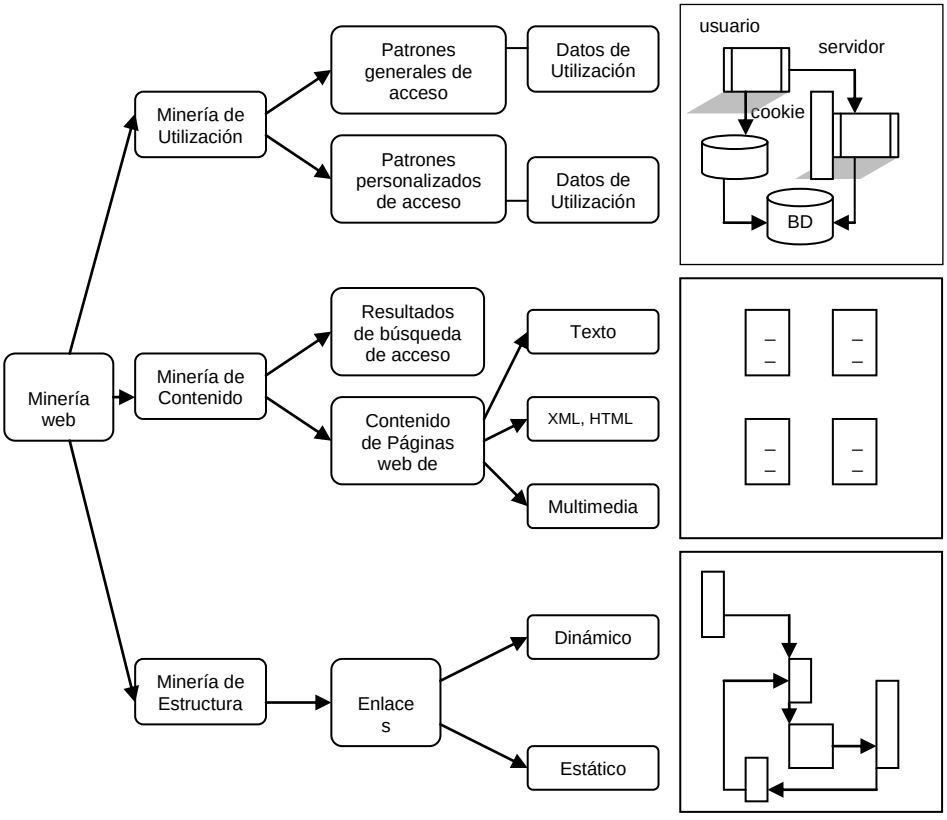

Fig. 1: Categorías de la Minería Web
Basado: [ROMÁN et al., 2005]

1.4.1.1. Minería de Contenido Web

La minería de contenido también conocida por sus siglas en el idioma inglés WCM (Web Content Mining), se centra en la "recogida de datos e identificación de patrones relativos a los contenidos de la web y a las búsquedas que se realizan sobre los mismos" [DÜRSTELER J., 2005]. Existen dos estrategias para la extracción del conocimiento:

- **Minería de páginas web**

Extrae patrones directamente de los contenidos existentes en las páginas, ya sea, texto libre, información procedente de bases de datos generadas en páginas con formato html, páginas xml**,** elementos multimedia y cualquier otro tipo de contenido presente en la web. Las técnicas que se ocupan de este tipo de minería son muchas pero cada una de ellas se aplica dependiendo del contenido de los datos, siendo la principal las técnicas de recuperación de información (information retrieval - IR).

- **Minería de resultados de búsqueda**

Consiste en identificar patrones de comportamiento y características comunes en los archivos de sucesos de los servidores Web.

Según SANABRIA J., la WCM tiene dos aproximaciones: basada en agentes y la de base de datos. La aproximación basada en agentes comprende el desarrollo de sistemas de inteligencia artificial sofisticados que pueden actuar autónomamente o semi-autónomamente en nombre de un usuario particular, descubrir y organizar la información Web; además esta aproximación se organiza en tres categorías:

- Los agentes de búsqueda inteligentes.
- Los de filtrado y/o categorización de información.
- Los de personalización.

La aproximación basada en base de datos, tiene como objetivo integrar y organizar los datos heterogéneos y semi-estructurados de la web para cambiarlos en conjuntos de recursos alto nivel más estructurado, mediante la utilización de técnicas estándar de consulta a bases de datos y técnicas de minería de datos. Está aproximación contiene dos categorías:

- Bases de datos multinivel
  - Nivel más bajo, contiene información primitiva semi-estructurada como documentos hipertextos.
  - Nivel más alto, aquí se extraen meta datos o generalizaciones de niveles más bajos, los cuales son organizados en conjuntos estructurados como las bases de datos relacionales.

- Sistemas de consulta web
  Estos sistemas utilizan lenguajes estándar de consulta de bases de datos como SQL y W3SQL.

1.4.1.2. Minería de Estructura Web

Esta minería web (Web Structure Mining - WSM), trata de revelar como están relacionados los hipervínculos entre las distintas páginas para generar un informe estructural sobre la página y el sitio web. La minería de estructura web, además nos proporciona información acerca de si los usuarios encuentran la información deseada, si la estructura del sitio es demasiado ancha o profunda, si los elementos están ubicados en los lugares adecuados dentro de la página, si la navegación se entiende, cuáles son las secciones menos visitadas y su relación con el lugar que ocupan en la página principal.

Típicamente tiene en cuenta dos tipos de enlaces: estáticos y dinámicos. La herramienta para realizar la WSM es la utilización de grafos, la cual nos permite reflejar el movimiento entre enlaces al navegar de una página a otra y así tener una mejor visión del conocimiento obtenido.

1.4.1.3. Minería de Uso Web

La minería de uso web conocida por sus siglas WUM (Web Usage Mining), utiliza los archivos de sucesos de los servidores Web para indagar cómo la gente accede y usa los sitios web, con el fin de descubrir patrones de comportamiento navegacional comunes entre los mismos. "WUM incluye datos provenientes de accesos a servidores web registrados en archivos del tipo logs, logs de motores de búsqueda, perfiles de usuario, archivos de enrolamiento o registro, sesiones de usuario y transacciones, consultas de usuario, carpetas marcadas, clic del ratón, desplazamientos (scroll) y otros tipos de información" [MERLINO H., 2005].

- **Análisis de la secuencia de navegación**

Este proceso empieza con los ficheros log de usuarios que "se almacenan como ficheros de texto en un directorio determinado por el servidor web" [MARTÍN J., 2004]. Estos ficheros se generan mediante un estándar específico del protocolo HTTP que numerosos servidores web utilizan y está formado por los siguientes campos, aunque algunos de ellos varían dependiendo del servidor web:

- Número IP o nombre del host remoto.
- Nombre del usuario que accede remotamente.
- Nombre de usuario bajo el cual se ha autentificado.
- Fecha y hora en que se realiza la solicitud del servicio.
- La solicitud como se realizó exactamente por el cliente.
- El código de estado HTTP.
- La cantidad de información (en bytes) que se transfiere.

Una vez recolectados los datos del fichero log se procede a realizar una tarea de limpieza que incluye los siguientes pasos: filtrado, identificación de usuarios y determinación de sesiones.

- **Filtrado**

Consiste en eliminar registros que no son necesarios, como los log de solicitudes de imágenes que son solamente parte de la página HTML que las contiene, por lo tanto los accesos a ficheros cuyas extensiones son: '.gif', '.jpeg', '.png', también son eliminadas. En una primera instancia el filtrado da como resultado datos a nivel de página web.

- **Identificación de usuarios**

En este paso se lleva a cabo dos niveles de identificación, en el primer nivel se identifican las peticiones de páginas realizadas por el mismo usuario durante una visita. El segundo nivel radica en reconocer a un usuario dentro de sus múltiples visitas a un determinado sitio web, con la finalidad de poder analizar el comportamiento del usuario a lo largo de días, meses o años. Una estrategia óptima de solución para la identificación de usuario sería mediante un "nombre de usuario" y "contraseña", pero como se conoce la navegación web se lleva de forma anónima, por lo que resulta bastante complicado reconocer a un mismo usuario entre los diferentes servicios a los que accede dentro de una misma sesión y mucho más compleja resulta cuando se tiene en cuenta la evolución temporal.

- **Determinación de sesiones**

Considera una serie de servicios solicitados por un mismo usuario a una única visita al sitio o portal web. Las sesiones son un factor importante ya a través de ellas se puede conocer la percepción del usuario con respecto a su visita al portal. La mejor solución para realizar esta

actividad es mediante una aplicación que cree un identificador de sesión la primera vez que un determinado usuario acceda al portal.

- **Técnicas de extracción de datos en el WUM**

Las técnicas más usadas para esto son: redes de neuronas artificiales (ANN), algoritmos genéticos y lógica difusa.

- Redes de neuronas artificiales: son modelos predecibles, no lineales que aprenden a través del entrenamiento y semejan la estructura de una red neuronal biológica.

- Algoritmos genéticos: son técnicas de optimización que usan procesos tales como combinaciones genéticas, mutaciones y selección natural en un diseño basado en los conceptos de evolución.

- Lógica difusa: es utilizada como soporte de otra técnica en función de lo poco estructurado de la información, la utilización de rangos difusos nos facilita a descubrir comportamientos comunes en forma más rápida.

Además este tipo de minería web tiene muchas aplicaciones, que van desde mejorar el diseño del sitio web hasta optimizar las relaciones entre usuario y los responsables del sitio web.

1.4.2. PROCESO DE EXPLOTACIÓN DE DATOS DEL WEB

La exploración de datos Web está conformada por las siguientes fases: recolección de los datos, procesamiento, generalización y análisis.

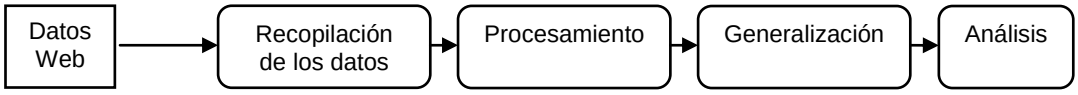

Fig. 2: Proceso del Web Mining
Basado y modificado: [SANABRIA J., 2006]

- **Recopilación de los datos**

Se encarga de detectar los orígenes de los datos para lograr conseguir de la forma más automatizada su captura para su posterior procesamiento. Este proceso incluye la representación de documentos, indexación y búsqueda de documentos. En la actualidad existen cuatro tipos de indexación de documentos Web: indexación humana o manual, indexación automática, indexación inteligente o basada en agentes y la indexación basada en meta datos.

- **Procesamiento**

En esta fase se filtran y limpian los datos recogidos, para luego ser ordenados y categorizados. Una vez realizado este proceso los datos quedan listos para su transformación por medios automáticos.

- **Generalización**

Utiliza diversas técnicas extraídas de diferentes ramas de la computación para obtener o reconocer un patrón común de comportamiento. Las técnicas más comúnmente utilizadas son: series de tiempo (Modelo Arima), redes de neuronas artificiales, algoritmos genéticos, lógica

difusa, teoría de conjuntos incompletos, reglas de decisión, aprendizaje automático y el análisis estadístico.

- **Análisis**

En esta fase se requiere la intervención del humano, debido a que este juega un papel muy elemental en el proceso de descubrimiento de conocimiento de la información de la Web. Su criterio es especialmente importante para la aprobación y/o interpretación de los patrones encontrados.

1.4.3. TÉCNICAS MÁS USADAS EN LA MINERIA DE DATOS

Como se conoce la minería web es una de las extensiones de la minería de datos (MD - Data Mining), por lo tanto está usa sus técnicas. Estás técnicas de la minería de datos emanan de la Inteligencia artificial y de la estadística, las cuales no son más que algoritmos, que se aplican sobre un conjunto de datos para obtener unos resultados.

Dentro de las técnicas más usadas de la MD, se tiene:

- **Redes neuronales artificiales (RNA)**

Esta técnica de inteligencia artificial, en los últimos años se ha convertido en uno de los instrumentos de uso frecuente para detectar categorías comunes en los datos, debido a que son capaces de detectar y aprender complejos patrones, y características de los datos. “Pueden ser utilizadas en problemas de clasificación (la variable de salida es cualitativa) o en predicción (la variable de salida es cuantitativa)” [PALMER et al.]. A profundidad se hablará más delante de este tema, por ser uno de los enfoques de la tesis.

- **Árboles de decisión**

Está técnica se encuentra dentro de una metodología de aprendizaje supervisado. Su representación es en forma de árbol en donde cada nodo es una decisión, los cuales a su vez generan reglas para la clasificación de un conjunto de datos.

Los árboles de decisión son fáciles de usar, admiten atributos discretos y continuos, tratan bien los atributos no significativos y los valores faltantes. Su principal ventaja es la facilidad de interpretación.

- **Algoritmos genéticos**

Los algoritmos genéticos (GAs) fueron inventados John Holland a principio de los 70, para imitar la evolución de las especies mediante la mutación, reproducción y selección, como también proporcionan programas y optimizaciones que pueden ser usadas en la construcción y entrenamiento de otras estructuras como es el caso de las redes neuronales. Además los algoritmos genéticos son inspirados en el principio de la supervivencia de los más aptos.

- **Clustering (Agrupamiento)**

Agrupan datos dentro de un número de clases preestablecidas o no, partiendo de criterios de distancia o similitud, de manera que las clases sean similares entre sí y distintas con las otras clases. Su utilización ha proporcionado significativos resultados en lo que respecta a los clasificadores o reconocedores de patrones, como en el modelado de sistemas. Este método debido a su naturaleza flexible se puede combinar fácilmente con otro tipo de técnica de minería de datos, dando como resultado un sistema híbrido.

Un problema relacionado con el análisis de clúster es la selección de factores en tareas de clasificación, debido a que no todas las variables tienen la misma importancia a la hora de agrupar los objetos. Otro problema de gran importancia y que actualmente despierta un gran interés es la fusión de conocimiento, ya que existen múltiples fuentes de información sobre un mismo tema, los cuales no utilizan una categorización homogénea de los objetos. Para poder

solucionar estos inconvenientes es necesario fusionar la información a la hora de recopilar, comparar o resumir los datos.

- **Aprendizaje automático**

El Aprendizaje Automático es una rama de la Inteligencia Artificial en la cual su principal objetivo es "desarrollar métodos computacionales para los procesos de aprendizaje y a la aplicación de los sistemas informáticos de aprendizaje en problemas prácticos" [SERVENTE, et al].

Dentro del aprendizaje automático se puede generar tres tipos de conocimiento, cada tipo dependerá del tema que se desee aprender: crecimiento, reestructuración y ajuste. Además existen algoritmos que son utilizados en el aprendizaje automático para la generación de conocimiento y el mejoramiento en el rendimiento de los sistemas computacionales, siendo estos: aprendizaje supervisado, aprendizaje no supervisado, aprendizaje por refuerzo, transducción y aprendizaje multi-tarea.

1.5. REDES NEURONALES

1.5.1. CONCEPTO

Las redes neuronales artificiales son dispositivos de calculó inspirados en las redes de neuronas biológicas, que intentan representar el comportamiento del cerebro humano y están constituidas por elementos simples denominados nodos o neuronas, las mismas que se encuentran organizadas en capas e interconectadas mediante enlaces directos llamados conexiones, por lo que son capaces de aprender de la experiencia, de generalizar casos anteriores a nuevos, de extraer características fundamentales a partir de entradas que representan información útil, de realizar aprendizaje adaptativo, de tener tolerancia a fallos y de operar en tiempo real.

1.5.2. ESTRUCTURA Y FUNCIONAMIENTO

1.5.2.1. Estructura

Una red neuronal artificial está básicamente compuesta por un: nodo de entrada, un nodo interno y un nodo de salida.

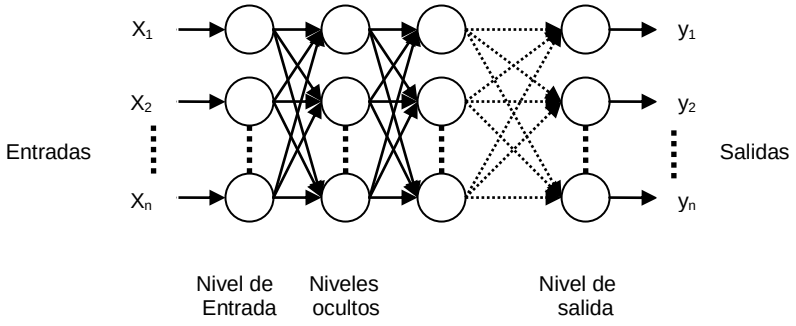

Fig. 3: Estructura de una red neuronal artificial
Basado y modificado: [AYESTARÁN et al.]

- **Nodos de entrada:** Los datos de entrada provienen de fuentes externas a la red de neuronas que conforman un vector de n-dimensiones de elementos numéricos.

- **Nodo interno (intermedios):** Son internas a la red y no tienen contacto directo con el entorno exterior. Aquí se realiza el trabajo de la red.

- **Nivel de salida:** Transfieren información de la red hacia el exterior, el cual es generado a través de la aplicación de una función llamada función de activación.

En el **Anexo A** se muestra los elementos y los diferentes tipos de topologías de la RNAs.

1.5.2.2. Funcionamiento

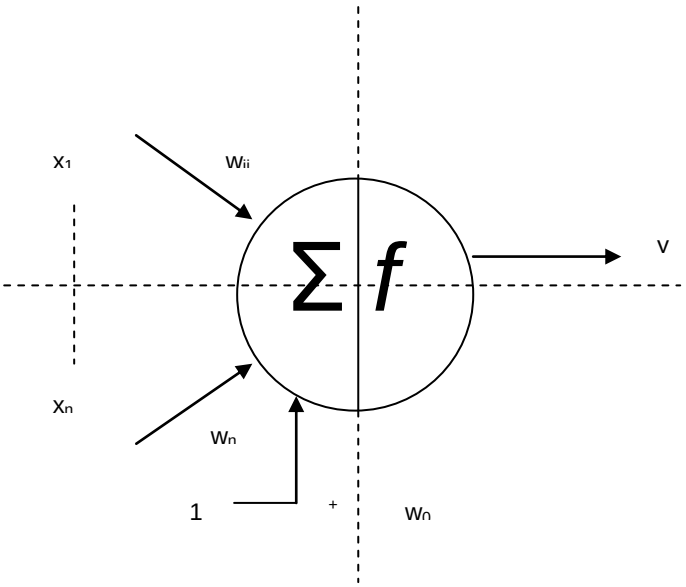

Fig. 4: Funcionamiento de una neurona artificial
Basado y modificado: [SALAS R.]

Una neurona recibe y emite información de otras neuronas y del mundo exterior. Esta información que recibe la neurona artificial la procesa mediante la obtención de dos componentes:

- El primero es un componente lineal denominado función de entrada *x* que están asociadas a pesos *w*, los cuales determinan el nivel de influencia de la neurona i para la neurona j. Internamente se calcula la suma de cada entrada multiplicada por su peso:

$$y_j = \sum_{i=1}^{n} w_{ij} x_i \tag{1}$$

- El segundo es un componente no-lineal conocido como función de activación o función de transferencia *f*, la cual se encarga de transformar la suma ponderada en el valor de salida de la neurona.

$$y = f(y_{j-})$$

(2)

1.5.3. APRENDIZAJE

Una propiedad importante de las redes neuronales es su capacidad de aprender interactuando con su entorno o con alguna fuente de información. El aprendizaje de la red es un proceso adaptativo mediante el cual se van cambiando los pesos sinápticos de la red para mejorar el comportamiento de ésta. Se distinguen dos tipos de aprendizaje: aprendizaje supervisado y el aprendizaje no-supervisado (auto-organización).

1.5.3.1. Aprendizaje Supervisado

El aprendizaje supervisado necesitan de un conjunto de datos de entrada previamente clasificado o cuya respuesta objetivo se conoce. "En este tipo de entrenamiento, la salida de la RNA es comparada con el valor deseado de salida. Los pesos, que normalmente han sido establecidos de manera aleatoria en un principio, son ajustados por la red de manera que en la siguiente iteración, también denominado ciclo, producirá un resultado más cercano entre el valor esperado y la salida real" [MARTINEZ I.]. Una vez que el entrenamiento termina los pesos se fijan, aunque algunas redes permiten el entrenamiento continuo pero con una tasa de aprendizaje baja. Esto ayuda a la red a adaptarse de manera gradual a situaciones de cambio.

La mayor parte de arquitecturas de RNA son entrenadas mediante métodos supervisados, entre ellas tenemos: realimentadas (State in a Box y Fuzzy) y las redes unidireccionales (Perceptrón, Adelina, Madalina, Perceptrón Multicapa, BackPropagation, General Regression Neural Network, Learning Vector Quantizer, Máquina de Boltzmann y Correlación en cascada). Algunas de estas redes neuronales se encuentran descritas en el **Anexo A-1**.

En el aprendizaje supervisado se suele considerar tres formas de aprendizaje: aprendizaje por corrección de error, aprendizaje con refuerzo y aprendizaje estocástico.

- **Aprendizaje por corrección de error**

El entrenamiento consiste en presentar a la red un conjunto de pares de datos, representados por la entrada y la salida deseada para dicha entrada. A este conjunto se lo denomina conjunto de entrenamiento. El objetivo de este método es minimizar el error entre la salida deseada y la actual.

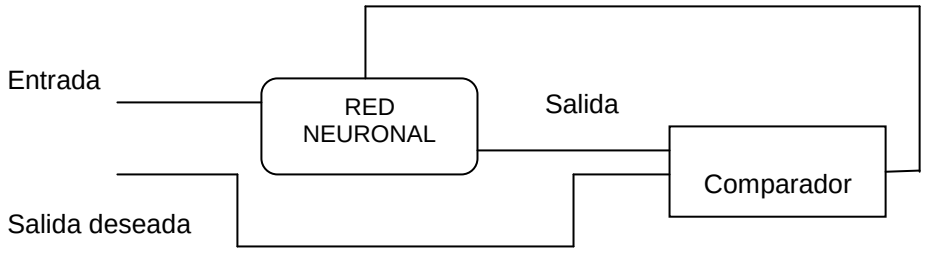

Fig. 5: Aprendizaje por corrección de error

Basado: [GONZÁLEZ M.]

- **Aprendizaje con refuerzo**

Barto, Sutton y Anderson formularon el "aprendizaje por refuerzo". Este aprendizaje es más lento que el aprendizaje por corrección de error, se basa en no disponer de un ejemplo completo del comportamiento deseado; es decir, no conoce exactamente la salida desea para cada entrada, pero si se conoce como debería de ser el comportamiento de manera general ante diferentes entradas.

- **Aprendizaje estocástico**

"Este tipo de aprendizaje consiste básicamente en realizar cambios aleatorios en los valores de los pesos y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad" [GONZÁLEZ M.].

1.5.3.2. Aprendizaje No-Supervisado

El aprendizaje no-supervisado no requieren de influencia externa para ajustar los pesos de las conexiones entre sus neuronas, es decir no existe un agente externo indicando la respuesta deseada para los patrones de entrada. Está limitado a las siguientes redes neuronales: realimentados (ART1, ART2, ART3, Hopfield y Bidirectional Associative Memory) y unidireccionales (Memoria Asociativa Lineal y Asociador Óptimo de Memoria Lineal, Mapas de Kohonen y Neocognitrón), ver **Anexo A-1**.

Dentro de este aprendizaje se suelen considerar dos tipos: aprendizaje hebbiano y aprendizaje competitivo o cooperativo.

1.5.4. TRABAJOS RELACIONADOS DE LAS RNA CON WEB MINING

A continuación mencionaremos algunos trabajos realizados de RNA con el web mining:

- **Identificación de Hábitos de Uso de Sitios Web Utilizando SOM**

Según MARTINELLI D., la identificación de hábitos de uso de sitios Web ha sido obtenida tradicionalmente utilizando métodos estadísticos, con la consiguiente limitación de estos para obtener resultados novedosos e inesperados. Debido a este problema plantea la siguiente la solución, la cual consiste en la utilización de una red neuronal mapa auto-organizativo (SOM) para la identificación de hábitos de usuarios. Esta red neuronal agrupa a los usuarios de un sitio web sobre la base de las páginas accedidas por los mismos. Para ello, se debe procesar el archivo de log del sitio a analizar, para identificar a los usuarios y a las sesiones de los mismos. Luego, con estas sesiones de usuarios es entrenada la red, para agrupar a los usuarios de forma automática. La elección de la red neuronal SOM es debido a la característica de la misma de posee un entrenamiento no supervisado, permitiendo realizar el agrupamiento de los usuarios en forma automática, sin intervención del usuario más que para configurar algunos pocos parámetros para el análisis.

- **Computación flexible aplicada al Web Mining**

Este trabajo emplea la Lógica Difusa (Fuzzy Logic), Redes Neuronales Artificiales, Algoritmos Genéticos, Conjuntos de Aproximación (Rough Set) y los Sistemas Neurodifusos. Todas estás herramientas permitirán "establecer un método para la obtención de conocimiento a partir de la información generada por programas que registran la actividad y los eventos ocurridos en la navegación de un usuario sobre un sitio web; información que es almacenada en archivos de registro de eventos conocidos como "archivos de logs". Dado que los datos a utilizar son del tipo numérico, se emplearan los conjuntos difusos para establecer variables lingüísticas que permitan crear un índice de comportamiento, por medio del cual se puedan obtener aquellas páginas más relevantes empleadas por los usuarios en su navegación a través del sitio. Se utilizara una red neuronal artificial con la finalidad de establecer un motor de procesamiento de patrones automatizado, motor que ordene la información de interés eliminando aquellas referencias sin importancia" [BENÍTEZ et al].

1.6. AGENTES

Hoy en día el paradigma agente constituye uno de los campos de mayor progreso, esto se debe a que permite el desarrollo de aplicaciones complejas. De acuerdo a las aportaciones de múltiples disciplinas, el término agente no mantiene el mismo significado para todos los investigadores del campo.

1.6.1. CONCEPTOS EXISTENTES

Según HERNANSÁEZ, describe tres diferentes enfoques de los agentes propuestos por Bradshaw, Jennings/Wooldridge y Nwana.

- **Bradshaw**

Define el término agente en base a la combinación de dos enfoques diferentes: por un lado una visión de los agentes basada en sus atribuciones, mediante el cual un agente software es aquel elemento del que se espera que actúe en representación de otro elemento/persona con el objetivo de cumplir con las responsabilidades que en él se han delegado. Por otro lado un agente puede ser definido de forma descriptiva, indicando que es una entidad software que opera, de forma continua y autónoma en un entorno particular, generalmente habitado por otros agentes (concepto de cooperación).

- **Jennings y Wooldridge**

Describen el agente como el sistema que situado dentro de un entorno es capaz de actuar de forma autónoma para alcanzar los objetivos para los que se ha diseñado. Esta definición resalta como principal característica de los agentes el concepto de autonomía. Además Jennings y Wooldridge definen como agentes inteligentes, los cuales, además de esta característica cumplen las siguientes:

- "Reactivo: el agente es capaz de responder a cambios en el entorno en que se encuentra situado.
- Pro-activo: a su vez el agente debe ser capaz de intentar cumplir sus propios planes u objetivos.
- Social: debe de poder comunicarse con otros agentes mediante algún tipo de lenguaje de comunicación de agentes" [BOTTI et al., 2000].

- **Nwana**

Considera que el término agente debe comprender todos los componentes software y/o hardware que presentan un comportamiento activo que les permite realizar las tareas que les han sido encomendadas.

1.6.2. CARACTERÍSTICAS

La mayoría de los agentes según SERRANO, poseen tres características: comunicación, inteligencia y autonomía.

- Comunicación. El agente puede comunicarse con el usuario, con otros agentes y con otros programas. Con el usuario se comunica con un interfaz amigable, mediante el que personaliza sus preferencias. Algunos agentes permiten comunicarse en lenguaje natural, algo típico de los chatbots.

- El grado de inteligencia varía mucho de unos agentes a otros, que suelen incorporar módulos con tecnologías procedentes de la Inteligencia Artificial. Los más sencillos se limitan a recoger las preferencias del usuario, quien debe personalizarlos. Un ejemplo son los agentes inteligentes basados en tecnología de redes neuronales especializados en

identificar mensajes de correo electrónico sospechosos de contener spam -mensajes no deseados-. En una primera fase el usuario debe marcarlos como spam, el agente va aprendiendo a identificar los rasgos que caracterizan a estos mensajes y posteriormente los filtra.

- Autonomía: Un agente no sólo debe ser capaz de hacer sugerencias al usuario sino de actuar. En el ejemplo anterior, el agente que filtra el spam no puede estar continuamente alertando al usuario en cada mensaje de correo que llega sobre la posibilidad de que sea un mensaje no deseado y su verdadera utilidad surge cuando elimina de forma autónoma dichos mensajes.

Con estas características y con lo que expone Nwana acerca de los agentes, se puede decir que estos se clasifican en seis clases (**Ver Anexo C**).

1.6.3. ARQUITECTURA DE LOS AGENTES

Componentes de la arquitectura genérica de los agentes:

1. Interfaz con el usuario
Se encarga de recibir requerimientos del usuario, enviarlos al módulo de razonamiento, y de presentar los resultados al usuario.

2. Módulo de razonamiento
Este componente se encarga de evaluar diferentes alternativas de solución además de negociar y seleccionar la mejor opción, basándose en el conocimiento y las metas del agente. Es capaz de tomar en cuenta los mensajes provenientes de otros agentes, o bien, de negociar con otros agentes.

3. Metas
Corresponden a los estados meta en la búsqueda de soluciones del agente.

4. Base de Conocimientos
Se refiere a la información que tenga disponible el agente sobre la realidad que le rodea.

5. Módulo de codificación y decodificación de mensajes
Este módulo es el responsable de codificar los mensajes del agente en el formato de algún lenguaje de comunicación de agentes.

6. Módulo de percepción
Se refiere a los medios con los que cuente el agente, para monitorear variables del medio ambiente que le rodea.

7. Módulo de comunicación
"Se encarga de enviar y recibir mensajes de otros agentes mediante protocolos de transporte (p.ej. en Internet, vía tcp/ip y http)" [MUÑIZ E, 1999].

1.6.4. APLICACIONES DE LOS AGENTES

Las aplicaciones basadas en este nuevo paradigma de programación orientada a agentes permiten abordar nuevos proyectos o mejorar el diseño y las prestaciones de aplicaciones actuales. Hoy en día las aplicaciones son muy numerosas, siendo empleadas en infinidad de áreas.

Nwana augura que uno de los campos en los cuales la teoría de agentes puede cuajar con mayor impacto es el de la Minería de Datos, esto se debe a que tres de las materias en las cuales los agentes tienen una aplicación concreta y validada con éxito, se corresponden con funciones o tareas presentes en los sistemas de Minería de Datos: (a) asistentes inteligentes de usuario, (b) recopiladores de información y (c) elementos de control automático.

- **Asistentes Inteligentes de Usuario**

El proceso de consulta, o más concretamente de construcción de una consulta, de Minería de Datos es muy complejo, esto se debe a que en última instancia, lo que se está solicitando en la consulta es la ejecución de uno o varios algoritmos de Inteligencia Artificial, Machine Learning o estadística que dependen de una multitud de parámetros diferentes. Por lo general, el proceso de resolución de una consulta de Minería de Datos implica la aplicación de varios de estos algoritmos. Esto se traduce en que, para el usuario, el proceso de consulta se convierte en un trabajo de selección de algoritmos. Una vez seleccionados, se tienen que fijar sus correspondientes parámetros. Estos parámetros representan umbrales de selección, límites de iteraciones, factores de corrección, etc. Por lo general, sólo los usuarios con un conocimiento muy profundo de los algoritmos de análisis de datos son capaces de manejar todos esos factores. Estos usuarios distan mucho de otro tipo de usuarios: son aquellos que tienen un menor conocimiento de los fundamentos matemáticos de las operaciones asociadas a la consulta, pero por otro lado poseen una gran experiencia en el dominio del problema al cual pertenecen los datos. La única forma de hacer que usuarios con diferentes perfiles accedan al sistema y aprovechen sus capacidades es por medio de asistentes personales que colaboren con ellos en la definición de la consulta. Estos agentes permitirán que un usuario inexperto pueda realizar una consulta únicamente indicando que desea modelos que definan una determinada circunstancia; por otra parte también permitirán que un analista experto de datos seleccione las fuentes de datos más adecuadas a un determinado dominio, aun sin tener gran conocimiento del mismo.

Un claro ejemplo de este tipo de aportaciones es el sistema Letizia es un **agente de interfaz** de usuario que asiste al usuario en la navegación a través de la Web. El sistema estudia el patrón de navegación del usuario realizando consultas en paralelo cuyo resultado presenta posteriormente al usuario.

- **Recopiladores de Información**

Uno de los requisitos del proceso de extracción de conocimiento a partir de datos es que los datos fuente deben contener suficiente información como para satisfacer la consulta solicitada. Esta idea admite acceder a la información contenida en sistemas de información dispersos, en concreto los autores la aplican a los escenarios de bases de datos federadas y a la red WWW.

- **Elementos de Control Automático**

Es una de las terceras línea en la que se puede aplicar la tecnología de agentes al proceso de Minería de Datos. “Los elementos de control de un sistema complejo son los encargados de gestionar todos los parámetros internos de un sistema necesarios para que dicho sistema sea operativo. La lógica de control para un sistema cualquiera crece exponencialmente según se añade nuevas funcionalidades y componentes al sistema, debido principalmente a la aparición de múltiples interdependencias entre dichos componentes. Un segundo factor que añade mayor complejidad al proceso de control de un sistema es la concurrencia. En el momento en el que el sistema tiene más de un flujo de ejecución (sistemas distribuidos o servicios multiusuario) la complejidad se multiplica. Una solución para controlar esta complejidad y proporcionar un elevado grado de flexibilidad y adaptabilidad es delegar el control. El control puede ser delegado localmente al componente en agentes suficientemente cercanos al componente como para poder sondear su estado y tomar acciones correctivas. Estos agentes pueden cooperar con otros agentes para resolver problemas que afectan a varios componentes o incluso al sistema completo” [HERNANSÁEZ et al, 2005].

ANÁLISIS Y DISEÑO

2.1. INTRODUCCIÓN

En esta fase se abordará el diseño del modelo del Agente basado en una red neuronal para minar los datos de un sitio web, siendo éste el sitio web UTPL. Para obtener una base sólida para el diseño del modelo se tomo como referencia la Fase 1, que presenta un estudio de los aspectos fundamentales de la Minería Web, Redes Neuronales y Agentes, así como de su relación con estas dos últimas técnicas en cuanto a trabajos dentro del área web. Además se adiciono más información sobre las ventajas, desventajas y características de las redes neuronales para la selección de está, igualmente se elegirá un agente que se relacione con la red neural y que se acople a este modelo.

Así mismo se describirá cada componente del modelo, como también se hablará sobre los ficheros Log de los servidores Web, siendo este la fuente principal de información para nuestro trabajo en el proceso de minería web.

2.2. PRÓSITOS

- Procesar el archivo “utpl-access_log” para obtener información estadística del sitio web de la UTPL.

- Construir la red neuronal para conocer el comportamiento de los usuarios en el sitio.

2.3. RESUSTADO ESPERADO

- El modelo diseñado de la red neuronal, debe ser adaptable a una herramienta de programación para que permita la visualización gráfica de resultados y por ende su respectivo análisis por parte del usuario.

2.4. ANÁLISIS Y SELECCIÓN DE LAS TÉCNICAS PARA LA REALIZACIÓN DEL MODELO

2.4.1. ANÁLISIS DE LAS REDES NEURONALES

Si bien es cierto todas las redes neurales se diferencian una de otras por que cada una de ellas posee una serie de ventajas y desventajas, a pesar de que todas ellas se basaron en el Perceptrón por ser la primera red neurona artificial. A continuación se detallan algunas ventajas y desventajas en la tabla siguiente:

Tabla 1. Ventajas y desventajas de las RNAs

RNAs	VENTAJAS	DESVENTAJAS
Perceptrón	- Capaz de resolver operaciones lógicas como: AND, OR y NOT.	- Su incapacidad para clasificar conjuntos de patrones que no son linealmente independientes. - Tiene única neurona de salida.
Adaline	- Trabaja con patrones de entrada y salida reales. - Su gráfica de error es un hiperparaboloide que posee o bien un único mínimo global, o bien una recta de infinitos mínimos, todos ellos globales. Esto evita la gran cantidad de problemas que da el perceptrón a la hora del entrenamiento. - Tiene varias neuronas de salida.	- No generaliza bien con datos que no se han utilizado en el proceso de aprendizaje. - Es más engorrosa desde la óptica computacional que el modelo de regresión lineal. - Posee las limitaciones del propio Perceptrón" [Torra S., 2004].
Backpropagation	- Eficaz para resolver el problema XOR. - Aprovecha el paralelismo de las redes neuronales con el fin reducir el tiempo para encontrar similitud entre patrones dados. - Se puede aplicar a muchos problemas diferentes proporcionando buenas soluciones en poco tiempo de desarrollo. - No requiere información apriorística. - Gran rapidez de procesamiento. - Es la "más aplicada en la práctica" [VILLASEÑOR C., 2003]. - La "capacidad de Generalización (facilidad de dar salidas satisfactorias a entradas que el sistema no ha visto nunca en su fase de entrenamiento)" [CATÁLFAMO A., 2006].	- Lentitud de convergencia. - No garantiza alcanzar el mínimo global, sólo un mínimo local. - El problema a la hora de entrenarlas estriba en que sólo conocemos la salida de la red y la entrada, de forma que no se pueden ajustar los pesos sinápticos asociados a las neuronas de las capas ocultas, ya que no podemos inferir a partir del estado de la capa de salida como tiene que ser el estado de las capas ocultas. - Tiempo de aprendizaje elevado para estructuras complejas.

Tabla 1. Ventajas y desventajas de las RNAs (… continuación)

Red de Base Radial	- RB construyen sus modelos con funciones de activación que son diferente tanto en la capa oculta como la de salida. - Aprendizaje rápido. - Es menos sensible al orden de presentación de patrones.	- Para la construcción de una red RBF se requiere de una mayor cantidad de neuronas en los nodos ocultos que en las redes que usan backpropagation. - El número de neuronas ocultas aumenta exponencialmente con la dimensión del espacio de entradas.

Así mismo se puede decir que algunos tipos de RNAs tienen algunas características en común y deferencias a la vez, debido a su arquitectura y método de aprendizaje, como se puede ver en la siguiente tabla:

Tabla 2. Características de las RNAs

RNA	CARACTERÍSTICAS
Perceptrón	- Aprendizaje supervisado. - Conexiones hacia delante (feedforward).
Adaline	- Aprendizaje supervisado.
Backpropagation	- Aprendizaje supervisado. - p capas ocultas.
Red de Base Radial	- Aprendizaje híbrido. - Arquitectura simple (capa de entrada, una única capa oculta y capa de salida).

2.4.2. SELECCIÓN DE LA RNA

Para seleccionar la red neuronal que forma parte del modelo a realizarse se llevo a cabo un estudio de algunos tipos de modelos de redes neuronales en la Fase 1 (**Ver Anexo A-1**), igualmente se hizo un análisis de estás redes en cuanto a sus ventajas y desventajas, así como también de sus características, con lo que se concluyó que la red neuronal que mejor se adapta a este tipo de problema como de MW es la red de Base Radial (RBF - Radial Basis Function), debido a que tiene un mejor desempeño con un mayor número de datos de entrenamiento como es el de nuestro caso, lo que hace que tenga una alta eficiencia en la fase de entrenamiento y por ende hace que su aprendizaje sea rápido. Otro punto importante que posee la RBF es su tiempo de entrenamiento, el cuál es substancialmente inferior, lo que ayudará a obtener resultados confiables en menor tiempo.

2.4.3. SELECCIÓN DEL AGENTE

Con lo estudiado acerca de los diferentes tipos de agentes (**Ver Anexo B**) y de acuerdo a sus características, se determinó que el Agente de Interfaz es el que mejor se ajusta a este modelo, porque es capaz de mostrarle al usuario los resultados del procesamiento de los datos obtenidos a través de la RNA y permite buscar las páginas mostradas en los resultados, estableciendo de este modo un diálogo entre el usuario (administrador del sitio web) y la máquina, es decir el usuario dirige el funcionamiento de la máquina a través de instrucciones denominadas entradas que se introducen mediante diversos dispositivos (teclado), y se convierten en señales electrónicas que son procesadas por la computadora. “Estas señales se

transmiten a través de circuitos conocidos como bus, y son coordinadas y controladas por la unidad de proceso central y por un soporte lógico conocido como sistema operativo. Una vez que la UPC ha ejecutado las instrucciones indicadas por el usuario, puede comunicar los resultados mediante señales electrónicas, o salidas, que se transmiten por el bus a uno o más dispositivos de salida, por ejemplo una impresora o un monitor" [FLORES C., 2006].

Este agente presentará cuatro módulos: importación de datos, red neuronal, visualización y salida, que serán explicados detalladamente en la Fase 3. Además el Agente de Interfaz ayuda a que el usuario analice los resultados de las páginas en cuanto a la relevancia que tienen las páginas para los visitantes (clientes web) y por ende tome la mejor decisión para el mejoramiento de las páginas y del sitio.

2.5. MODELO DEL AGENTE BASADO EN REDES NEURONALES PARA WEB MINING

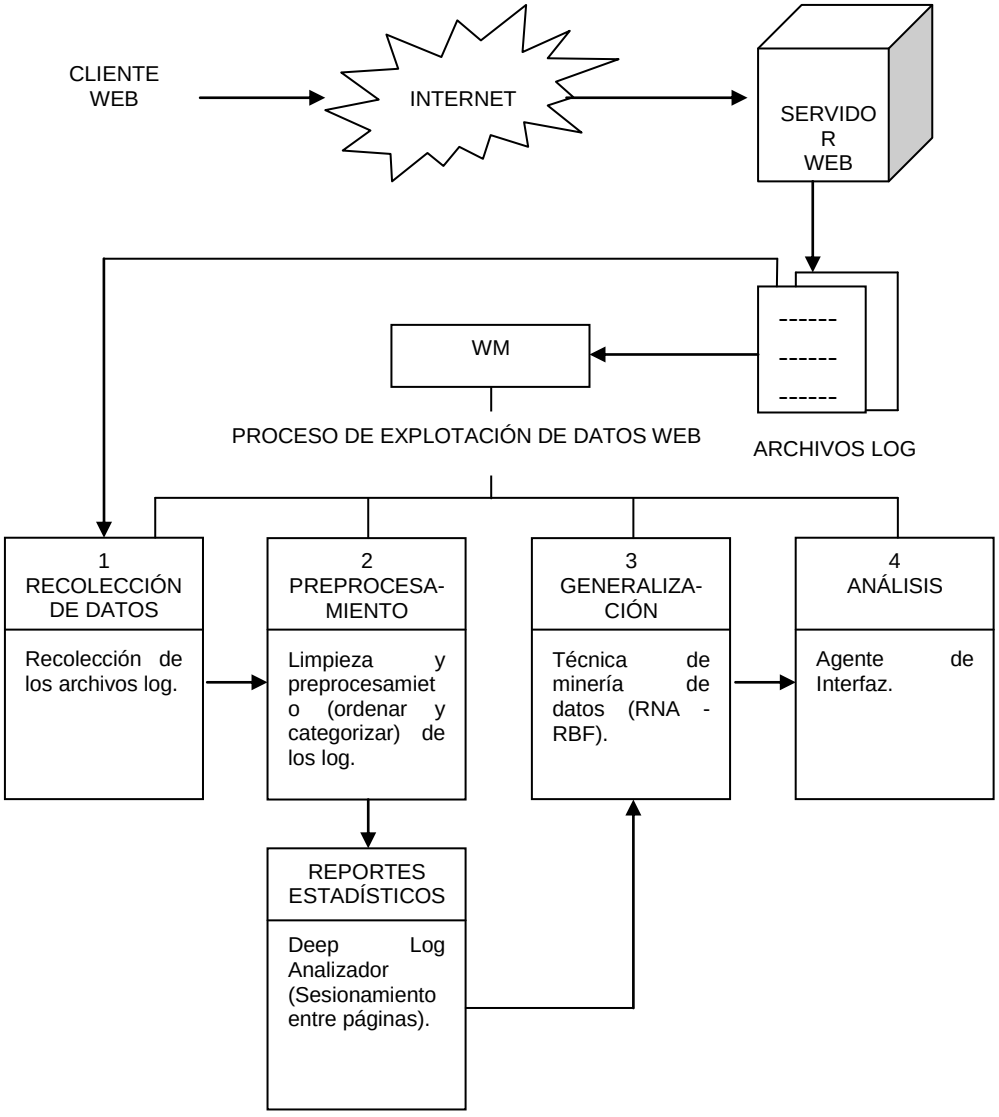

Fig. 6: Modelo de Agente basado en RN para WM

El modelo del Agente basado en RNA de la Fig.6 muestra una vista general de la utilización de la técnica de WM, dentro del esquema de funcionamiento de un sitio web.

Este modelo consta de 3 pasos para transformar los datos en bruto del sitio Web en información útil y analizar la actividad de los usuarios dentro del sitio para conocer su comportamiento.

A continuación se describen cada uno de los pasos del modelo:
1. Los visitantes, son aquellos que realizan accesos a Internet a través de su red local para ingresar al Sitio Web (www.utpl.edu.ec - UTPL).
2. La información solicitada es almacenada en el Servidor Web en el registro de actividades (Archivos Log).
3. Una vez que se tiene los Archivos log, se procede aplicar el proceso de explotación de datos de la WM, la cual consta de 5 tareas: recolección de datos, preprocesamiento, generalización y análisis. Todas estás tareas se las puede ver en la Fase 1 en el apartado 1.2.2, donde se manifiesta en que consiste cada una de ellas.

En la siguiente sección se describirá de forma más detallada como se realizo el proceso para procesar los logs.

2.5.1. RECOLECCIÓN DE LOG

Los servidores de páginas generan numerosos datos provenientes del registro de los pedidos que reciben de parte de las aplicaciones clientes de los usuarios. Estos requerimientos son registrados en el denominado **log del servidor** de manera constante e inmediata durante todo el tiempo en el que el servicio está activo.

Todo ese volumen de datos contiene información que es usado para el análisis del tráfico y su uso con respecto al usuario. En el log se almacena ficheros de texto en diferentes formatos, aunque algunos servidores pueden almacenarlo directamente en una base de datos. La información que se almacena depende también del servidor y su configuración particular. Los datos de los logs pueden ser recolectados a nivel de servidor web, del cliente o del servidor proxy.

Para el presente trabajo los datos se los recolecto a nivel de servidor web, siendo este el servidor ydr1.utpl.edu.ec (**Ver Anexo B**).

2.5.1.1. Formato de Registros Log

Dentro de los log existen tres tipos de formatos: registro log de acceso, registro log de error y registro log de referencia.

- Registro log de acceso

La principal fuente de información sobre los visitantes es el fichero log de accesos (Access Log o Transfer Log). En este fichero se registran todas y cada una de las transacciones entre el navegador del visitante y el servidor.

Los registros de log de acceso pueden ser guardados en un formato de fichero log común (Common Log File CLF) que siguen la mayor parte de los servidores http para mantener un registro de accesos o en un formato de fichero log extendido (Extended Log File ELF).

Los datos recolectados del servidor web (UTPL) tienen un formato CLF, el cual esta formado por los siguientes campos: remotehost, nombre del usuario que accede remotamente, authuser, solicitud, código de estado y el volumen transferido.

```
90.11.17.161 - - [05/Jun/2007:11:28:23 -0500] "GET / HTTP/1.1" 200 12326

172.16.5.68 - - [05/Jun/2007:11:28:31 -0500] "GET / HTTP/1.1" 200 12326
```

Fig. 7: Fragmento de un fichero log correspondiente a dos usuarios (Formato CLF)

- Remotehost: es la dirección IP (host remoto) del cliente desde la que se ha accedido al servidor.

  Ej: 90.11.17.161

- Nombre del usuario que accede remotamente (kRfc931): originalmente "fue pensado para autentificar a las máquinas UNIX conectadas a Internet mediante los antiguos protocolos FTP, Archie, Telnet, etc." [SÁNCHEZ S., 1997-2004]. Apareció en el año 1984 y se trata de un "guión", que significa que la información que debería ir en ese lugar no está disponible.

- Autentificación del usuario (Authuser): es el nombre con el que el usuario se ha identificado previamente en el servidor web, quedando en este almacenado su nombre; en otro caso ese dato se deja vacío.

- Fecha y hora: la hora a la que el servidor recibió la petición.
  El formato es:

  [día/mes/año:hora:minuto:segundo zona_horaria]

  Ej: [05/Jun/2007:11:28:23 -0500]

- Petición del cliente o Solicitud: la línea de la petición del cliente se muestra entre dobles comillas. Esta línea de petición contiene mucha información de utilidad y consta de 3 partes:

  [método de la transacción /recurso solicitado/versión del protocolo]

  Primero, el método usado por el cliente es GET. Segundo, el cliente ha hecho una petición al recurso /apache_pb.gif, y tercero, el cliente uso el protocolo HTTP/1.0.

  Ej: "GET / HTTP/1.1"

  "También es posible registrar una o más partes de la línea de petición independientemente. Por ejemplo, el formato "%m %U%q %H" registrará el método, ruta, cadena de consulta y protocolo, teniendo exactamente el mismo resultado que %r" [APACHE, 2008].

- Código de estado HTTP: es el estado del HTTP devuelto al cliente. Esta información es muy valiosa, porque revela si la petición fue respondida con éxito por el servidor (los códigos que empiezan por 2), una redirección (los códigos que empiezan por 3), un error provocado por el cliente (los códigos que empiezan por 4), o un error en el servidor (los códigos que empiezan por 5).
  Ej: 200 → pedido exitoso.

La lista completa de códigos de estado HTTP con su respectivo significado se puede ver en el **Anexo D**.

- Volumen Transferido o Número de bits enviados: este campo proporciona el tamaño de la transacción (fichero transferido) del objeto en bytes, "retornado por el cliente, no incluidas las cabeceras de respuesta. Si no se respondió con ningún contenido al cliente, este valor mostrará un valor –" [APACHE, 2008].

  Ej: 12326 → cantidad de información (en bytes) que se transfiere.

- Registro log de error

El registro log de error "contiene información de diagnóstico y en él se registra cualquier error que se produzca al procesar peticiones. En él se suele encontrar información detallada de las cuestiones que no han resultado satisfactorias y cómo solucionar los problemas" [GRANDE L., 2005-2007].

El formato del registro log de errores es relativamente libre y descriptivo. No obstante, hay cierta información que se incluye en casi todas las entradas de un registro log de errores. Por ejemplo, este es un mensaje típico:

[Wed Oct 11 14:32:52 2000] [error] [client 127.0.0.1] client denied by server configuration: /export/home/live/ap/htdocs/test

El primer elemento de la entrada es la fecha y la hora del mensaje. El segundo elemento indica la gravedad del error que se ha producido. La directiva LogLevel se usa para controlar los tipos de errores que se envían al registro de errores según su gravedad. La tercera parte contiene la dirección IP del cliente que generó el error. Después de la dirección IP está el mensaje de error propiamente dicho, que en este caso indica que el servidor ha sido configurado para denegar el acceso a ese cliente. El servidor reporta también la ruta en el sistema de ficheros (en vez de la ruta en el servidor web) del documento solicitado.

- Registro log de referencia

Es un fichero opcional que contiene información de las páginas webs desde donde una página concreta es accedida.

Este fichero log contiene las URLs desde las que se originaron las peticiones de las páginas del website. Es decir, este fichero log captura las direcciones de las páginas desde las que han venido los visitantes. El campo referente no siempre contiene información, de hecho, se pueden diferenciar los siguientes 3 tipos de valores:

- Vacío: el campo referente vacío significa que el visitante ha llegado a la página del website escribiendo su URL a mano directamente en la barra de dirección o dispone de un enlace (con la URL del website) en favoritos y ha pulsado en él.

- URL interno: la solicitud de la página procede de otra página del mismo website. Por ejemplo, si en el campo referente figura:

  http://www.doxmatic.com/EE/index.mv

  significa que el visitante, estando en la página index.mv, ha pulsado en un enlace que le llevó a la página cuyo campo referente se esta estudiando. Una información muy valiosa que permite seguir la ruta del visitante por el website.

- URL externo: la solicitud de la página procede de una página que no pertenece al website. Puede ser la página de otra empresa que ha puesto un enlace al website en cuestión. Pero, en mayoría de los casos, contiene la URL de un buscador:

  http://www.google.com/search?q=spss&hl=es

  "Este referente indica que el visitante estuvo buscando la palabra spss en el buscador Google (www.google.com). El URL externo indica cómo los visitantes localizan el website" [SÁNCHEZ S., 1997-2004].

2.5.2. PROCESAMIENTO DE LOG

Radica en convertir la información bruta proveniente de ficheros de log, en información tratable, utilizando las abstracciones necesarias y filtrando o descartando la que sea innecesaria o redundante.

Esta fase se encuentra dividida en tres pasos: limpieza de datos, identificación de visitantes e identificación de sesiones. Para ello se busco una herramienta en internet que sea capaz de realizar estas tareas, lo que se determinó trabajar la herramienta Deep Log Analyzer (**Ver Anexo D**).

Está herramienta es asequible y avanzada solución de análisis web de pequeño y mediano tamaño de los sitios web. Analiza los visitantes del sitio web y obtiene estadísticas de uso en varios pasos fáciles, siendo un objetivo primordial de este trabajo de tesis (**Ver Anexo D-1**).

Entre los principales beneficios Deep Log Analyzer tenemos:
- Genera informes de análisis web con presentación interactiva de navegación y jerárquica.
- Exporta informes a Excel o HTML.
- Permite ver los datos en los informes avanzados.
- Fácil y familiar interfaz de usuario de MS Office. Incluso se puede personalizar la interfaz de usuario para su comodidad.

- **Limpieza de los datos**

En el **Anexo D** (paso 6), se eliminan los datos duplicados, erróneos y accesos irrelevantes, que existen en el archivo "utpl-access_log".

- **identificación de visitantes**

Luego de la limpieza de los logs, se identifica a los visitantes a través de la dirección IP que ingresaron al sitio web UTPL.

- **identificación de sesiones**

Una sesión de usuario corresponde a una secuencia de vínculos (links) seguidos por el usuario. Para ello se necesita dividir las distintas peticiones realizadas por un mismo usuario en una o más sesiones. Para determinar el sesionamiento se ha seleccionado un tiempo máximo de 30 min. entre peticiones siendo está la más recomendable (**Ver Anexo D, paso 5**).

2.5.3. DESCUBRIMIENTO DE PATRONES

La investigación sobre el uso del sitio web de la UTPL se ha centrado en el descubrimiento de patrones de acceso a partir del fichero log, debido a que un patrón de acceso es un patrón secuencial que se repite dentro de los logs. Para procesar este log se utilizó la herramienta Deep Log Analyzer, mencionada anteriormente, la misma que nos proporciona un reporte de Sesionamiento entre Páginas (**Ver Anexo E**).

A partir de este reporte se puede conocer cuales son las páginas que tiene mayor relevancia y cuales no para el visitante, y para conocer estos resultados se utilizará la RNA de Base Radial seleccionada en el apartado 2.4.2. La información obtenida servirá para tomar decisiones efectivas en cuanto a diseño o mejora de algunos aspectos del sitio web de la UTPL.

2.5.3.1. Diseño de la Red Neuronal para Web Mining

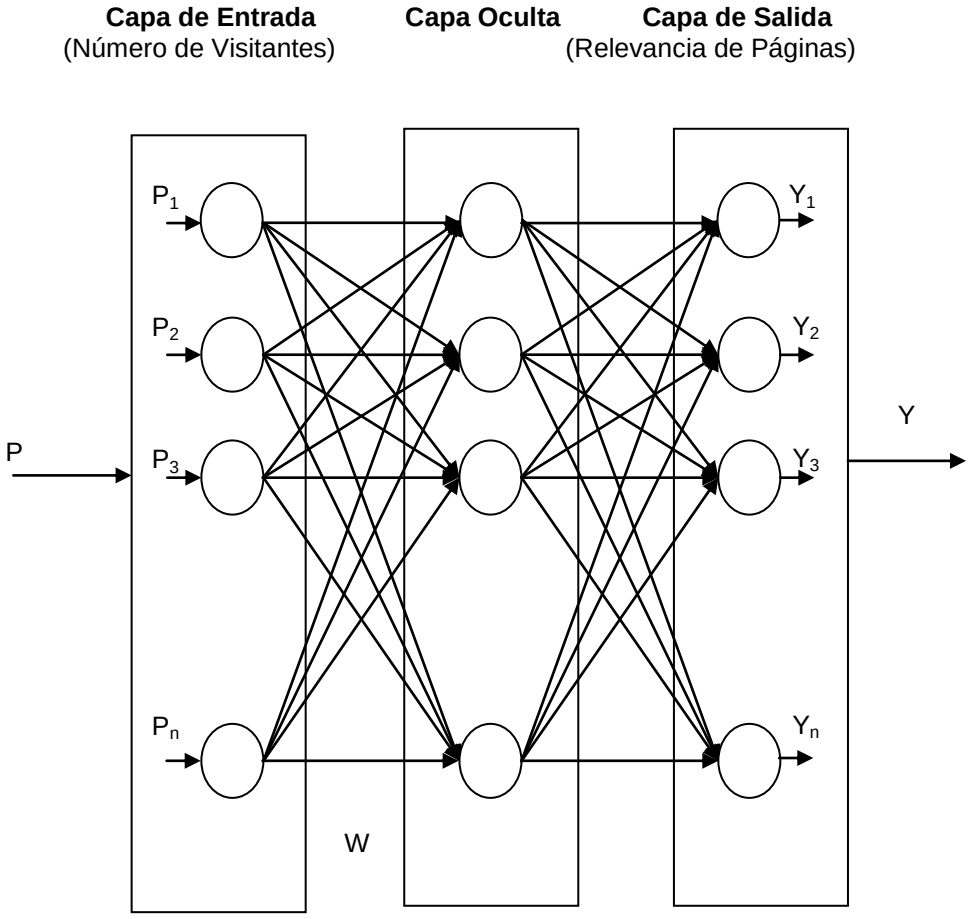

Fig. 8: Diseño de la RNA

La fig. 18 muestra el diseño de RNA de Base Radial para determinar que páginas tienen mayor o menor relevancia, para esto se designo un valor de 1 a las páginas que son de mayor relevancia para los usuarios y un valor de 0 a las páginas de menor relevancia.

Como se puede observar, este diseño de la RBF (Red Neuronal de Base Radial) consta de tres capas. Cada capa esta compuesta de neuronas con las mismas características, que transfieren los patrones de entrada a patrones de salida ejecutando una función de transferencia.

En lo que se refiere a las neuronas de entradas, en este modelo no se puede indicar la cantidad de neuronas (cantidad de sesiones) que va tener, debido a que los reportes de

Sesionamiento entre Páginas no van a contener la misma cantidad de registros, esto se debe a que en algunas fechas las visitas al sitio son más concurrentes. Además esta red posee solo una capa oculta.

2.5.3.2. Funcionamiento de la Red Neural

El objetivo de nuestra red neuronal de Base Radial es procesar un archivo para obtener la relevancia de las páginas. Para llevar acabo este fin, la red neuronal de BR se encargara de simular las propiedades observadas en el archivo de sesionamiento entre páginas a través de un modelo matemático creado mediante mecanismos artificiales (un ordenador).

Para procesar el archivo primeramente debemos ingresarlo, para esto se usará una interfaz gráfica inteligente la cual nos permite buscar el archivo de sesionamiento dentro del disco, una vez identificado se procede a analizar sus datos obteniendo su resultado en dos vectores.

Estos vectores obtenidos anteriormente con el agente inteligente son: los datos de entrada representados por P = (P1, P2,..., Pn) y el objetivo T.

2.5.4. ANÁLISIS DE PATRONES

La última etapa y uno de los procesos más importantes en todo el proceso de la explotación de datos de la minería web es el análisis de los patrones. Una vez que los patrones han sido identificados a través de la RNA en la etapa anterior, este proceso de análisis de patrones es muy eficaz para descubrir la información que no es visible a simple vista, para ello se utiliza un agente de interfaz el cual mostrará los resultados de la RNA.

Con los resultados proporcionados por el agente de interfaz el administrar del sitio puede hacer su propio análisis en cuanto este, ya sea por popularidad del sitio, páginas mayor o menor vitadas, tráfico u otros aspectos correspondientes al sitio. El resultado de dicho análisis debe ayudarle a determinar que factores son o no importantes para su sitio y como consecuencia de ello poder tener una mejor perspectiva general sobre su sitio y por ende tomar una decisión clave en cuanto diseño u otros aspectos del sitio.

IMPLEMENTACIÓN

DE SOLUCIÓN

3.1. INTRODUCCION

Esta fase se enfocada a la implementación del modelo del agente basado en redes neuronales para web mining realizado en la Fase 2, para ello se expone un breve resumen de la herramienta MatLab. Así mismo se describirá cada una de las funciones de toolbox de redes neuronales utilizadas para el entrenamiento de la BR y los resultados obtenidos en esta fase

Este simulador para RNA está en capacidad de procesar cualquier archivo de Sesionamiento entre Páginas almacenado en diferente directorio y mostrar la Prioridad de las Páginas que se obtiene luego del entrenamiento de la red neuronal.

3.2. PRÓPOSITO

- Implementar la red neuronal de base radial y el agente de interfaz en la Herramienta de programación Matlab.

3.3. RESULTADO ESPERADO

- La implementación de la Simulación en la Herramienta Matlab, debe permitir visualizar un entorno gráfico para el agente y para el aprendizaje de la red neuronal.

3.4. HERRAMIENTA MATLAB VER 7.0.1

MATLAB (es el nombre abreviado de “MATrix LABoratory”) es un programa técnico y científico para realizar cálculos numéricos con vectores y matrices. “Como caso particular puede también trabajar con números escalares −tanto reales como complejos−, con cadenas de caracteres y con otras estructuras de información más complejas. Una de las capacidades más atractivas es la de realizar una amplia variedad de gráficos en dos y tres dimensiones. MATLAB tiene también un lenguaje de programación propio” [GARCÍA et al., 2005].

El lenguaje de programación de MATLAB es una herramienta de alto nivel para desarrollar aplicaciones técnicas que son fáciles de utilizar. Además dispone de un código básico y de varias librerías especializadas (toolboxes - orientadas a la optimización). En este caso se hará referencia exclusivamente a la librería Neural Network Toolbox.

MATLAB opera bajo sistemas operativos UNIX, Macintosh y Windows.

3.4.1. NEURAL NETWORK TOOLBOX

El Neural Network Toolbox “proporciona funciones para el diseño, inicialización, simulación y entrenamiento de los modelos neuronales de uso más extendido en la actualidad: Perceptrón, redes lineales, redes de retropropagación, redes de base radial, aprendizaje asociativo y competitivo, aplicaciones autoorganizativas, aprendizaje de cuantización vectorial, redes de Elman y redes de Hopfield” [MONTBRUN et al., 1997]. Así mismo ésta Librería nos permite diseñar nuestras propias arquitecturas y definir las funciones de transferencia e inicialización, regla de aprendizaje, función de entrenamiento y estimación de error.

Otra característica importante del toolbox, aporta las facilidades y prestaciones gráficas para el estudio del comportamiento de las redes: visualización gráfica de la matriz de pesos y vector de desplazamiento mediante diagramas de Hinton, representación de errores a lo largo del entrenamiento, mapas de superficie de error en función de pesos y vector de desplazamiento, etc.

3.5. IMPLEMENTACIÓN EN MATLAB DEL DISEÑO DEL AGENTE BASADO EN RED NEURONAL PARA WEB MINING

3.5.1. CONSTRUCCIÓN E IMPLEMENTACIÓN DE LA RB EN MATLAB

Para la construcción de la RB, se realizo el siguiente algoritmo:

1. Se importa los datos de Excel a MATLAB (Archivo de Sesionamiento entre Páginas), para esto se hace uso del siguiente código:

   Sintaxis

```
pathname='C:\'
[filename,pathname] = uigetfile(...
{'*.xls';'*.mdl';'*.mat';'*.*'},...
 'Open');
        together = [pathname '' filename]
```

   La variable pathname, es una ruta que nos sirve para referenciar un archivo. Una vez que se selecciona el archivo, el nombre de ruta y el nombre de archivo se almacenan en sus respectivas variables. Con el comando uigetfile se puede abrir cualquier tipo de archivo, por que es muy versátil y ofrece muchas opciones.

1.1 Identificado el archivo, se procede a leer este:

Sintaxis
[N, T, rawdata] = xlsread(together)

La función xlsread devuelve texto (T) y datos numéricos (N), además combina estas dos en la variable rawdata.

Para nuestro archivo, estas variables se especifican a continuación:
- N: columna número de visitas.
- T: columna dirección de la página.
- Rawdata: contiene toda la matriz (N y T).

1.2 Se requiere solo la columna N, para almacenar está se utilizó la variable A.

Sintaxis
A=[N];

1.3 Se obtiene la media de la matriz A con la función **mean.**

Sintaxis
X=mean(A);

El resultado de la media de la matriz A, es almacenado en la variable X.

1.4 Se calcula el número de elementos de la matriz A con la función **length**.

Sintaxis
a = length(A)

1.5 Proceso para almacenar la matriz T.

1.5.1 Asignar a la variable E una matriz vacía.

Sintaxis
E=[];

1.5.2 Recorrido de la matriz T y almacenar en la matriz E, para luego sacar la inversa de esta matriz y almacenarla en la variable L, como se puede ver en las siguientes líneas de código:

Sintaxis
```
for k=2:a+1
    E= [E T(k,1)]
    L=E'
 end
```

todo este proceso nos servirá para guardar estos resultados en un archivo csv.

1.6 Obtención de la relevancia de las páginas.

Sintaxis

```
for i=1:a
 M=A(i,1)

        for k=1:m-1
        end

        B(1,m)=A(i,1);
        for j=i:a
            if X >=  M
                'M es de mayor relevancia'
                s = 0
            else
                'M es de menor relevancia'
                s = 1
            end
        end

        B1(1,m)=s
          m=m+1
  end
```

Con este procedimiento se va recorriendo la matriz A, tomando su primer elemento y almacenando en la matriz M. Luego se compara este elemento con la media (X), en el caso de ser mayor e igual se asigna al elemento mayor relevancia, caso contrario se le asigna menor relevancia. Estos resultados son almacenados en la variable B1.

Todo este proceso se repite hasta que se recorre toda la matriz A.

1.7 Almacenamiento de archivos en formato csv.

Sintaxis

```
diary Direcciones.csv
      L
diary off

diary Direcciones.csv
      B1.'
diary off
```

2. **DISEÑO:** Se usa la función newbr para crear una red neuronal de Base Radial.

3. **ENTRENAMIENTO**
El siguiente paso, consiste en la adquisición de datos para entrenamiento:

3.1 Se ingresa el vector de entrada (P).

Sintaxis

```
P = B
```

3.2 Se ingresa el vector deseado (T)**.**

Sintaxis

```
T= B1.
```

3.3 En este punto del procedimiento, se está en condiciones de crear la red para luego simularla. Para lograr el entrenamiento de la red se empleó la entrada P:

Sintaxis
net = newrb(P,T,eg,sc,MIN,1)
lo que garantiza obtener las clasificaciones correctas.

4. **SIMULACIÓN:** Se simula la red, con la siguiente función:

Sintaxis
Y = sim(net,P)

3.5.2. CONSTRUCIÓN E IMPLEMTACIÓN DEL AGENTE DE INTERFAZ

El agente de interfaz consta de tres opciones, que usan para poder interactuar entre el usuario y la maquina.

1. Declaración de variables generales:
   - c → contador para las opciones del menú (c = 1).
   - prew → variable condicional para cerrar el menú (preview = 0).
   - clusterRBF = 1;
   - scluster = 'Visitas de Páginas: RBF ';

   1.1 Creación del agente de interfaz (MENU), el cual consta de tres botones: Importar datos, RBF, visualizacón y salida.

   Sintaxis
   while c < 3
   c = menú('ELIJA LA OPCIÓN', 'Importar Datos', scluster, 'Analisis de Páginas', 'Salir');

2. La opción 1 (Importar Datos)
   2.1 Toma el paso 1 del algoritmo de construcción de la red neuronal descrito anteriormente.

3. La opción 2 (Visitas de Páginas: RBF)
   3.1 Llama el paso 2, 3 y 4, del apartado 3.4.2.1 para procesar los datos.
   3.2 Una vez procesados los datos, enseguida visualiza el aprendizaje de la red neuronal.

4. La opción 3 (Visualización de Datos)
   Muestra los resultados procesados con la red neuronal.

   Sintaxis
   stat = web('http://localhost/Pagina_Principal.php', '-browser')

5. La opción 3 (Salir)
   Permite cerrar la ventana de la simulación de la RN.

   Sintaxis
   if prew
   'show';
   end

   El codigo completo de esta simulación se puede ver en el **ANEXO E (**Código en Matlab de la Simulación**)**.

3.6. DESCRIPCIÓN DE LAS FUNCIONES UTILIZADAS DEL TOOLBOX PARA EL ENTRENAMIENTO DE LA RNA

- **Newbr**

Crea y entrena la red de BR, además añade neuronas a la capa oculta hasta que está obtenga un error mínimo que el especificado en el parámetro *goal*.

Sintaxis
net = newbr(,P,T, EG, SPREAD, MN)

Descripción
- P: es la matriz de entrada de tamaño RxQ, con R entradas y Q muestras.
- T: es la matriz de salidas deseadas de tamaño SxQ, con S salidas y Q muestras.
- EG(GOAL): error mínimo cuadrado a alcanzar (por defecto 0.0) .
- SC(SPREAD): función de base radial, por defecto es 1.0 y retorna una red de base radial exacta.
- MN: máximo número de neuronas en la primera capa, por defecto Q.

Está red presenta una serie de propiedades configurables que definen las características básicas de la red.

- NET.IW: matrices de pesos de las capas de la red. Es un cell array de tamaño N_l x N_i, donde N_l representa el número de capas mientras que N_i es el número de entradas de la red.
- NET.b: define los vectores de bias para cada capa con bias. Es un cell array de tamaño N_l x1.
- NET.layers: define las propiedades de cada una de las capas de la red.

- **Net**

Es lo que devuelve la función, que es una estructura que contiene los ajustes de la red para los datos de entrenamiento dados.

- **Sim**

El comando sim, simula la salida de la red neuronal.

Sintaxis
Y= sim (net, P)

Descripción
- Net: red
- P: patrón de entrada
- Y: salida de la red

3.7. LABORATORIO

3.7.1. R EQUISITOS PARA INSTALACIÓN DE HERRAMIENTA MATLAB.

Para instalar Matlab 7.0.1 se requiere los siguientes requisitos de l sistema:

- **Requisitos generales**

- CD-ROM (para la instalación del CD).

- Adobe y Acrobat Reader 3.0 para ver e imprimir la documentación de MATLAB en formato PDF expuesta en el CD.
- Se requiere TCP/IP (protocolo) en todas las plataformas en donde se instale una licencia para servidor.

- **Requisitos de plataforma específicos.**

Tabla 5. Requisitos para Matlab 7.01

SISTEMAS OPERATIVOS	PROCESADORES	ESPACIO EN DISCO	RAM
Windows Vista Home Edition	Core 2 Duo, Centrino, Pentium III, IV, AMD Athon, Athlon XP, Athlon MP	400 MB	256 MB (Mínimo). 512 MB (Recomendado)
Windows XP o XP Service Pack 2, 3			
Windows 2000 (Service Pack 3 o 4)			
Windows NT 4.0 (Service Pack 5 o 6			

3.7.2. DESARROLLO DE LA PRÁCTICA

Para la práctica se utilizaron dos archivos de sesionamiento entre páginas de los días 6-7/06/2007 y 8-9/06/2007. De cada de estos archivos se obtendrán las páginas de mayor relevancia las mismas que tendrán un valor de cero, como también se obtendrán las páginas de menor relevancia que se las designa con un valor de 1, a continuación se detalla cada paso del proceso:

Se ejecuta aplicación, y como resultado de esto nos muestra el agente de interfaz, como se muestra en la figura:

Fig. 9: Menú

La opción Importar Datos nos permite encontrar el archivo de seseinamiento, para ubicarlo en matrices, ya que es un requerimiento para que la red neuronal pueda procesar éstos datos.

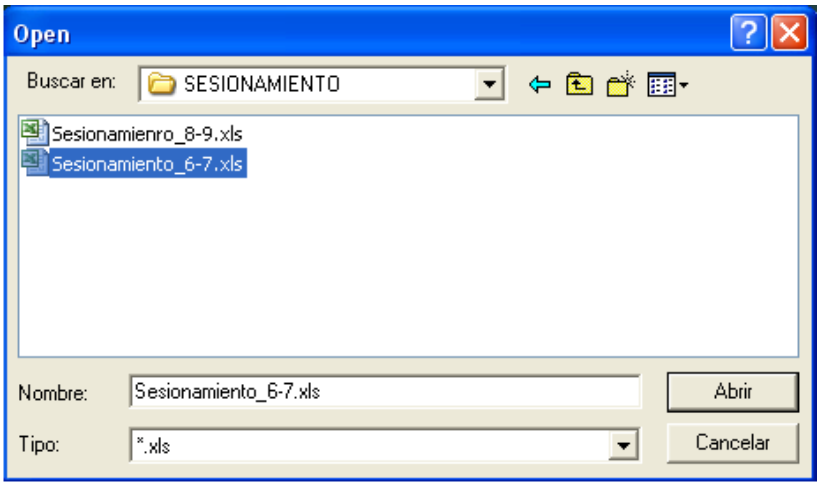

Fig. 10: Ventana para cargar archivo

Con la opción RBF donde se encuentra la red neuronal, se procede a procesar los datos y es aquí donde se obtienen las páginas con mayor y menor relevancia. Además esta opción nos muestra la gráfica de aprendizaje de la red neunal.

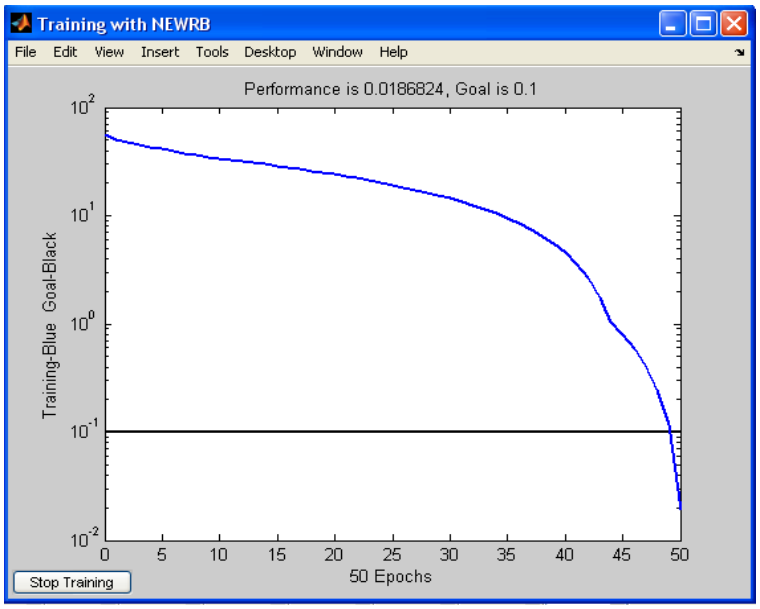

Fig. 11: Aprendizaje de la red neuronal de Rase Radial

Para visualizar los resultados de los datos que procesa la red neuronal en la interfaz del programa Matlab, se utiliza la opción 3 en donde se puede observar de manera más entendible este procesamiento.

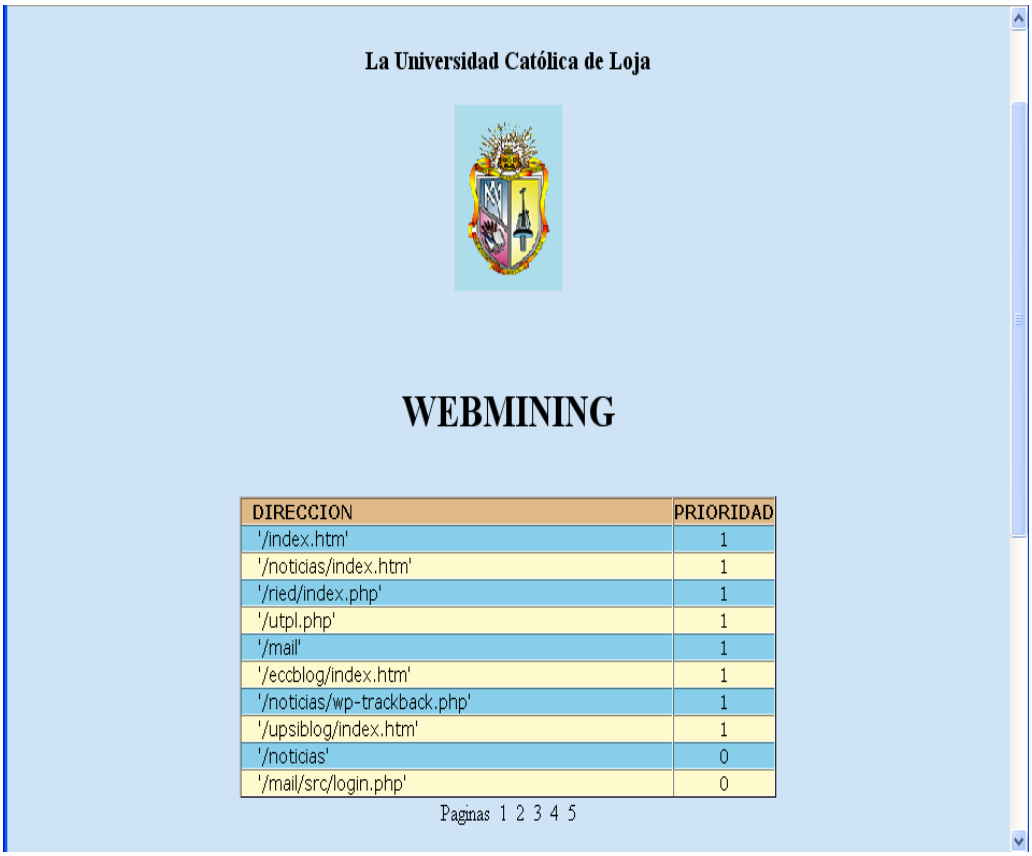

Fig. 12: Presentación de los resultados

Se sigue el mismo procedimiento para procesar el archivo de sesionamiento de los días 8-9/06/2007.

3.7.3. PRUEBAS

Las pruebas se las realizo en cuanto a diseño y entrenamiento de la red.

DISEÑO

- Se comprobó que el diseño de la red neuronal de base radial realizado en la Fase 2 es igual al realizado en el Simulink de Matlab, como se puede observar en la siguiente figura, ésta tiene una solo capa de entrada, una capa oculta y una capa de salida:

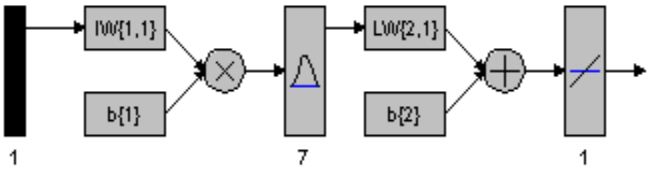

Fig. 13: Diseño de RBF en Simulink

ENTRENAMIENTO
Para el entrenamiento de la RNA, se usó dos reportes de Sesionamiento entre Páginas, con los cuales se obtuvo los siguientes resultados:

- N. P → número de práctica.
- MN → máximo número de neuronas en la capa de entrada.
- EG → error mínimo cuadrado a alcanzar.
- PERF → error de la red

Tabla 3. Aprendizaje de la RBF con error de 0.1

N. P	REPORTE DE SESONAMIENTO	MN	EPOCAS	EG	TIEMPO DE RESPUESTA	PERF.
1	6-7/06/2007	100	25	0.1	0.6 seg.	0.443267
1	6-7/06/2007	1000	25	0.1	0.6 seg.	0.443267
2	8-9/06/2007	100	25	0.1	0.6 seg.	1.60587
2	8-9/06/2007	1000	25	0.1	0.6 seg.	1.60587

Tabla 4. Aprendizaje de la RBF con error de 0.6

N. P	REPORTE DE SESONAMIENTO	MN	EPOCAS	EG	TIEMPO DE RESPUESTA	PERF.
1	6-7/06/2007	100	25	0.6	0.6 seg.	0.443267
1	6-7/06/2007	1000	25	0.6	0.6 seg.	0.443267
2	8-9/06/2007	100	25	0.6	0.6 seg.	1.60587
2	8-9/06/2007	1000	25	0.6	0.6 seg.	1.60587

3.7.4. VALIDACIÓN

De acuerdo con los datos obtenidos de las tablas, se dice que la RNA aprende en la época 25, con un tiempo de máximo de 0.06 seg. y sin tomar en cuenta el MN (número de neuronas

máximo). Además ésta RNA esta en capacidad de procesar cualquier archivo de procesamiento entre Páginas.

De la tabla 3 y 4, se toma sólo el archivo de sesionamiento 6-7/06/2007 para conocer el % de error en aprendizaje de la red neuronal.

Tabla 5. % de error de aprendizaje de la RBF

N. P	REPORTE DE SESONAMIENTO	MN	EPOCAS	EG	TIEMPO DE RESPUESTA	% ERROR
1	6-7/06/2007	100	25	0.1	0.6 seg.	14.28
1	6-7/06/2007	100	25	0.6	0.6 seg.	85.71
		100		0.7	.	99.99

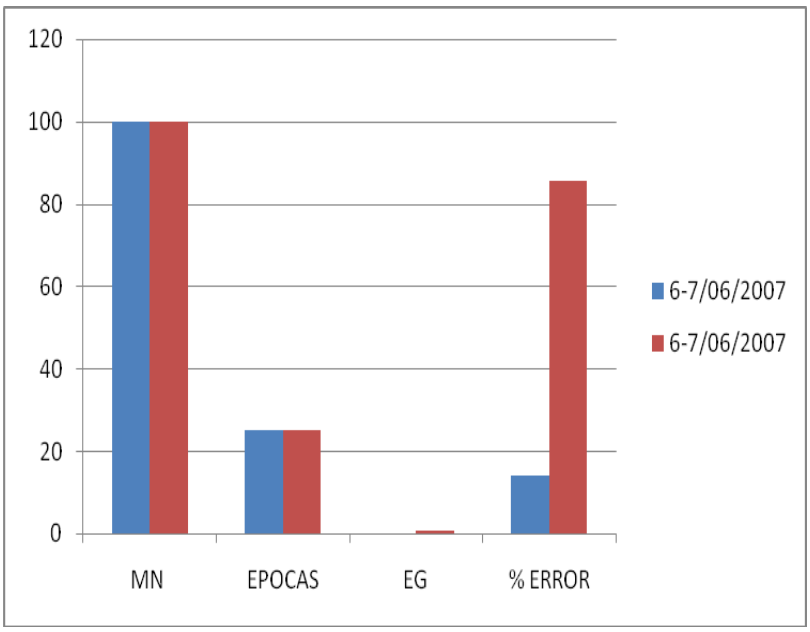

Fig. 14: Resultados de la Tabla 5

DISCUSIÓN, CONCLUSIONES Y RECOMENDACIONES

4.1. DISCUSIÓN

En la actualidad se puede decir que el Internet pasó de ser un lujo académico a una necesidad informativa, siendo sus sitios fuentes de información de todo tipo por lo que se constituye en un enorme repositorio, con datos irrelevantes y redundantes.

En este trabajo se pudo conocer que hoy en día, la abundancia de datos en Internet hace que la UTPL se enfrente a un entorno caracterizado por niveles crecientes de complejidad, globalidad, y cambios rápidos y profundos como resultado del acelerado avance tecnológico. Para contrarrestar y adaptarse a estos cambios debe prestar atención a la tecnología Web Mining (minería de web), que es una de las extensiones del data mining. Éste tipo de minería consiste en aplicar las técnicas de minería de datos (Redes Neuronales, Arboles de Decisión, Algoritmos Genéticos, Clustering, Aprendizaje Supervisado, etc.) a documentos y servicios para extraer información de la Web. Pero si bien esta tecnología tiene mucha importancia son muy pocas las herramientas fabricadas con la técnica de Redes Neuronales que sólo se desarrollan con aprendizaje no- supervisado, es por esta razón que surgió la necesidad de realizar un modelo de Agente basado en una Red Neuronal con aprendizaje híbrido para minar únicamente las páginas hospedadas en el servidor web de la UTPL, el mismo que será implementado en la herramienta Matlab generándose un pequeño simulador de minería web.

La fuente principal de información corresponde a conjuntos de datos provenientes del reporte de Sesionamiento entre Páginas generadas por la Deep Log Analyzer, del cual se obtiene a través del simulador de manera automática la prioridad que tienen las páginas para los usuarios.

Con ésta información se contribuye manera eficaz al administrador o persona encargada del sitio a tomar decisiones puntuales e útiles, para mejor la calidad del sitio, en cuanto a su diseño, estructura u otros aspectos que crea beneficioso para el sitio web.

4.2. CONCLUSIONES

- La tecnología web mining es muy beneficiosa para todos aquellos que están involucrados con el mundo web como instituciones educativas siendo una de ellas la UTPL, permitiéndole conocer aspectos esenciales de su sitio web, al igual que las tendencias y el comportamiento de los usuarios, las mismas que le sirven como puntos de apoyo para la toma de decisiones en cuanto a mejoramiento y estructura del sitio.

- Dentro de la minería web existen tres técnicas o dominios, siendo estas la minería de contenido, la minería de estructura y la minería de uso, cada una de ellas se aplica dependiendo de la naturaleza de los datos. En la actualidad la técnica más aplicada es la minería de uso debido a que utiliza los archivos de sucesos de los servidores Web (log), los cuales pueden analizarse de forma estática o dinámica y la más difícil de procesar es la técnica de minería de contenido debido a la variedad de datos que contienen las páginas web.

- Para la explotación de los datos en la web, se usa una de las cinco técnicas más importantes de la minería de datos siendo esta las redes neuronales artificiales, misma que permiten el reconocimiento de patrones de uso de la web por medio de los archivos log.

- La tecnología de agentes es muy efectividad en el desarrollo de interfaces gráficas por sus características y las prestaciones que ofrecen, permiten augurar una fuerte relación con la tecnología de redes neurales, ya que a través de está puede actuar como un agente más para brindarnos la información en cuanto a la relevancia que tienen las páginas cada día del sitio de la UTPL.

- Para el procesamiento de logs se uso la herramienta Deep Log Analyzer, la cual nos proporcionando una variedad de reportes siendo uno de ellos el sesionamiento entre páginas, diseñado para mostrarnos el uso que hace el visitante al ingresar al sitio web de la universidad.

- El diseño realizado en la Fase 2 de la red neuronal de Base Radial, se la implemento en la herramienta Matlab por ser muy efectiva para realizar cálculos numéricos con RNA y además nos permite comprobar nuestra arquitectura diseñada de la RBF con el simulink.

- La red neuronal de Base Radial se constituye en un pilar fundamental dentro de está simulación ya que nos permite obtener las páginas de mayor y menor relevancia del sitio web en un mínimo tiempo 0.06 segundos, lo que afirma que su aprendizaje es bastante rápido en comparación a las otras redes neuronales.

- En cuanto al entrenamiento de la red neuronal, se realizaron corridas de dos archivos de sesionamiento entre páginas, lo que se determino que esta red aprende en la época 25 siendo este valor casi constate para este tipo archivo sin tomar en cuenta la cantidad de neuronas (datos) en la capa de entrada que tenga que procesar.

- De acuerdo a los datos estadísticos realizados de la Tabla 4, la red neuronal aprende con un error mínimo de 0.1 aproximándose al error mínimo cuadrado de la red neuronal original.

4.3. RECOMENDACIONES

- Primeramente se debe investigar data mining, ya que de aquí parte lo que es la técnica de la minería web o web mining, con sus respectivos dominios.

- Para entender el funcionamiento de las redes neuronales, se debe tomar un libro base debido a que estas poseen una nomenclatura compleja.

- Para el procesamiento de los logs existen diversas herramientas, pero la herramienta más eficaz para este es Deep Log Analyzer por que genera archivos con formato .exe, lo cual facilita la extracción de estos datos en la herramienta Matlba.

- Para implementar una red neuronal es mejor utilizar la herramienta Matlab, debido a que esta posee librerías que nos permite diseñar nuestras propias arquitecturas y definir las funciones de transferencia, regla de aprendizaje, función de entrenamiento y estimación de error.

BIBLIOGRAFIA

[ACOSTA et al, 2000] Acosta, M.; Zuluaga, C. **"TUTORIAL SOBRE REDES NEURONALES APLICADAS EN INGENERIA ELECTRÓNICA Y SU IMPLEMENTACIÓN EN SU WEBN"**, Universidad Tecnológica de Pereira, Consultada: 18 de junio del 2007, [online]. Disponible en Internet: http://ohm.utp.edu.co/neuronales/maindown.htm

[APACHE, 2008] **"ARCHIVOS DE REGISTRO (Log Files)"**, Consultada: 29 de enero del 2008, [online]. Disponible en Internet: http://httpd.apache.org/docs/2.0/es/logs.html

[ARILOG et al, 2002] Arilog, P.; Arango, V. **"MAPAS AUTOORGANIZATIVOS DE KOHONEN PARA LA REPRESENTACIÓN DEL CONOCIMIENTO"**, Universidad de Antioquia Medellín-Colombia, 2002, Consultada: 06 de julio del 2007, [online] Disponible en Internet: http://www.riterm.net/actes/8simposio/vanessaAirlog.htm

[AYESTARÁN et al.] Ayestarán, R.; Heras, F. **"REDES NEURONALES Y RECONSTRUCCIÓN DE FUENTES EQUIVALENTES PARA EL CÁLCULO DE VOLTAJES EN ARRAYS"**, Universidad de Oviedo, Consultada: 5 de mayo del 2007, [online]. Disponible en Internet: http://w3.iec.csic.es/ursi/articulos_coruna_2003/actas_pdf/SESION%206/S6.%20Aula%202.0/1297%20-%20REDES%20NEURONALES.pdf

[BENÍTEZ et al] Benítez, J.; Castro. J.; Valenzuela R. "**COMPUTACIÓN FLEXIBLE APLICADA AL WEB MINING**", Universidad de Granada, Consultada: 5 de junio del 2007, [online]. Disponible en Internet: http://decsai.ugr.es/~castro/docto-csi/Ricardo%20Valenzuela/p80.pdf

[BOTTI et al., 2000] V. Botti, V. Julian **"AGENTES INTELIGENTES: EL SIGUIENTE PASO EN LA INTELIGENCIA ARTIFICIAL"**, Universidad Politécnica de Valencia, Consultada: 23 de julio del 2007, [online]. Disponible en Internet: http://www.ati.es/novatica/2000/145/vjulia-145.pdf

[BOZA G., 2005] **"EFECTO DE LA TOPOLOGIA DE REDES NEURONALES DE BACKPROPAGATION EN LA OPTIMIZACION DE PROCESOS QUÍMICOS VIA MODELOS MATEMATICOS NEURONALES EMPÍRICOS"**, Universidad Nacional del Altiplano de Puno, [online]. Consultada: 18 de junio del 2007, [online]. Disponible en Internet: http://www.ciiq.org/varios/peru_2005/Trabajos/VI/6.01.pdf

[CATÁLFAMO A., 2006] **"REDES NEURONALES APLICADAS A SEÑALES SONORAS OBTENIDAS DE UN SONAR, ESPECIALIDAD EN INGENIERÍA EN SISTEMAS EXPERTOS"**, Instituto Tecnológico de Buenos Aires, Consultada: 28 de enero del 2008, [online]. Disponible en Internet: http://www.centros.itba.edu.ar/capis/webcapis/trabajosfinalesdeespecialidad/catalfamo-trabajofinaldeespecialidad.pdf

[DOMINGUEZ E] **"MAPAS DE KOHONEN"**, Universidad de Malaga, Consultada: 07 de julio del 2007, [online]. Disponible en Internet: http://www.redes-neuronales.netfirms.com/tutorial-redes-neuronales/redes-neuronales-autoorganizadas-mapas-de-kohonen.htm

[DÜRSTELER J., 2005] "**MINERÍA WEB**", InfoVist, Consultada: 07 de julio del 2007, [online]. Disponible en Internet: http://www.infovis.net/printMag.php?num=172&lang=1

[ESCOBAR V., 2007] "**MINERÍA DE USO WEB Y PERFILES DE USUARIO: APLICACIONES CON LÓGICA DIFUSA**", Universidad de Granada, Consultada: 08 de julio del 2007, [online]. Disponible en Internet: http://hera.ugr.es/tesisugr/17296651.pdf

[FLORES C., 2006] **"LA INTERFAZ"**, Consultada: 28 de enero del 2008, [online]. Disponible en Internet:http://comunicacionunapuno.blogspot.com/2006/01/la-interfaz-otro-beneficio-de-la.html

[GARCÍA L.] "**EL WEB MINING: UNA TÉCNOLOGIA PARA LA INDAGACIÓN EN LA WORD WIDE WEB**" Consultada: 07 de julio del 2007, [online]. Disponible en Internet: http://www.scribd.com/doc/2982183/EL-WEB-MINING

[GARCÍA et al., 2005] García, J.; Rodríguez, J.; Vidal J. **"APRENDA MATLAB 7.0"**, Universidad Politécnica de Madrid, Consultada: 28 de enero del 2008, [online]. Disponible en Internet: http://mat21.etsii.upm.es/ayudainf/aprendainf/Matlab70/matlab70primero.pdf

[GONZÁLEZ M.] **"REDES NEURONALES"**, Consultada: 5 de mayo del 2007, [online]. Disponible en Internet: http://carpanta.dc.fi.udc.es/~cipenedo/cursos/scx/scx.html

[GRANDE L., 2005-2007] **"VISUALIZACIÓN DE DATOS"**, Consultada: 29 de enero del 2008, [online]. Disponible en Internet: http://carpe.usal.es/~roberto/SPIP/IMG/pdf/LauraGrande.pdf

[GYVES C.] "**WEB MINING: FUNDAMENTOS BÁSICOS**", Universidad de Salamanca, Consultada: 23 de julio del 2007, [online]. http://zarza.usal.es/~fgarcia/doctorado/iweb/05-07/Trabajos/WMINING.pdf

[HERNANSÁEZ et al, 2005] Hernansáez, J.; Botía, J.; Gómez A. **"ASISTENCIA PERSONALIZADA A LA MINERÍA DE DATOS MEDIANTE AGENTES INTELIGENTE"**, Universidad de Murcia, Consultada: 23 de julio del 2007, [online]. http://www.tecn.upf.es/~vlopez/alfa/alpha-umu.pdf -

[KROSE et al., 1996] Krose, B.; Smagt, P. **"AN INTRODUCTION TO NEURAL NETWORKS"**, University of Amsterdam, Consultada: 5 de mayo del 2007, [online]. Disponible en Internet: http://www.avaye.com/files/articles/nnintro/nn_intro.pdf

[MARTINELLI D., 2006] "**IDENTIFICACIÓN DE HÁBITOS DE USO DE SITIOS WEB UTILIZANDO REDES NEURONALES**", Universidad de Buenos Aires, Consultada: 05 de junio del 2007, [online]. Disponible en Internet: http://materias.fi.uba.ar/7500/martinelli-tesisingenieriainformatica.pdf

[MARTINEZ I.] **"INTRODUCCIÓN A LAS REDES NEURONALES"**, Universidad Complutense de Madrid, Facultad de Informática, Consultada: 05 de mayo del 2007 [online]. Disponible en Internet: http://www.guru-games.org/people/pedro/aad/ivan_martinez.pdf

[MARTÍN J., 2004] **"DETERMINACIÓN DE TENDENCIAS EN UN PORTAL WEB UTILIZANDO TÉCNICAS NO SUPERVISADAS"**, Consultada: 21 de mayo del 2007, [online]. Disponible en Internet: http://www.uv.es/jdmg/tesis_jdmartin.pdf

[MERLINO H., 2005] **"AMBIENTE DE INTEGRACIÓN DE HERRAMIENTAS PARA EXPLORACIÓN DE DATOS CENTRADOS EN LA WEB"**, Consultada: 21 de mayo del 2007, [online]. Disponible en Internet: http://www.itba.edu.ar/capis/webcapis/tesisdemagister/merlino-tesisdemagister.pdf

[MONTBRUN et al., 1997] Montbrun, O.; Montbrun, A. **"LIBRERÍA DE APLICACIONES DE MATLAB"**, Consultada: 28 de enero del 2008, [online]. Disponible en Internet: http://www.eldish.net/hp/automat/toolb_mb.htm#neural

[MUÑIZ E, 1999] "**SISTEMA MULTIAGENTE PARA ENLACE DE INTERESES**", Instituto Tecnológico y de Estudios Superiores de Monterrey, Consultada: 23 de julio del 2007, [online]. Disponible en Internet: http://www.munizfam.com/blog/docs/tesisDocFinal.pdf

[MUÑOZ J., 2005-2006] **"DOCENCIA EN INGENIERÍA INFORMÁTICA – CURSO"**, Universidad de Málaga, 2005-2006, Consultada: 03 de julio del 2007, [online]. Disponible en Internet: http://www.lcc.uma.es/%7Emunozp/

[PALMER et al.] Palmer, A.; Beltrán, M.; Montaño, J.; Jiménez, R.; Sesé, A.; Franconetti F. "**MINERÍA DE DATOS EN ECONOMÍA**", Universitat de les Illes Balears, Consultada: 04 de julio del 2007, [online]. Disponible en Internet: http://conecoib.caib.es/comunicacions/76_mineria_de_datos.pdf

[ROMÁN et al., 2005] Román, U.; Alarcón, L. **"CATEGORÍAS DE LA MINERÍA WEB"**, Universidad Nacional Mayor de San Marcos, Consultada: 21 de mayo del 2007, [online]. Disponible en Internet: http://sisbib.unmsm.edu.pe/BibVirtualData/publicaciones/risi/n3_2005/a01.pdf

[ROMERO L., 2007] **"APPLETS DE REDES NEURONALES"**, Consultada: 5 de mayo del 2007, [online]. Disponible en Internet: http://avellano.usal.es/~lalonso/

[SALAS R.] **"REDES NEURONALES ARTIFICIALES"**, Universidad de Valparaíso, Consultada: 5 de mayo del 2007, [online]. Disponible en Internet: http://www.inf.utfsm.cl/~rsalas/Pagina_Investigacion/docs/Apuntes/Redes%20Neuronales%20Artificiales.pdf

[SÁNCHEZ S., 1997-2004] **"COMPONENTES DE DATOS"**, Data Mining Institute, S.L., Consultada: 29 de enero del 2008, [online]. Disponible en Internet: http://www.estadistico.com/arts.html?20010618

[SANABRIA J., 2006] "**SISTEMA DE PERSONALIZACIÓN WEB PARA EL PROCESO DE APRENDIZAJE EN UNA PLATAFORMA DE EDUCACIÓN VIRTUAL**", Universidad Nacional De Colombia, Consultada: 23 de julio del 2007, [online]. http://dis.unal.edu.co/profesores/ypinzon/2013326-206/docs/Tesis0Sanabria.pdf

[SERRANO C, 2007] **"AGENTES INTELIGENTES"**, Consultada: 23 de julio del 2007, [online]. Disponible en Internet: http://ciberconta.unizar.es/LECCION/INTRODUC/482.HTM

[SERVENTE et al.] Servente, M.; García, R. "**ALGORITMOS TDIDT**", Consultada: 23 de julio del 2007, [online]. Disponible en Internet: http://laboratorios.fi.uba.ar/lsi/R-ITBA-26-datamining.pdf.

[SORIA E.] **"REDES NEURONALES"**, Consultada: 05 de mayo del 2007, [online]. Disponible en Internet: www.acta.es/articulos_mf/19023.PDF

[TORRA S., 2004] **"SINIESTRALIDAD EN SEGUROS DE CONSUMO ANUAL DE LAS ENTIDADES DE PREVISIÓN SOCIAL, LA. PERSPECTIVA PROBABILÍSTICA Y ECONOMÉTRICA. PROPUESTA DE UN MODELO ECONOMÉTRICO NEURONAL PARA CATALUÑA"**, Universidad de Barcelona, Consultada: 28 de junio del 2008, [online]. Disponible en Internet: http://www.tesisenxarxa.net/TESIS_UB/AVAILABLE/TDX-0929104-094807//

[TREC, 1999-2000] **"REDES NEURONALES ARTIFICIALES"**, Consultada: 18 de junio del 2007 [online]. Disponible en Internet: http://electronica.com.mx/neural/informacion/perceptron.html

[VILLALOBOS et al., 2002] Villalobos, I.; Murillo, S.; Álvarez, J.; Garmendia , A.; Martínez , E.; Juárez, U. **"REDES NEURONALES"**, Instituto Tecnológico Autónomo De México, Consultada: 18 de junio del 2007, [online]. Disponible en Internet: http://cursos.itam.mx /akuri /2002 /S22002/RNS /Presentaciones /Adaline/ADALINEYMADALINE2.doc

[VILLASEÑOR C., 2003] **"MODELADO DIFUSO NEURONAL CON ALGORITMO DE APRENDIZAJE ESTABLE"**, Instituto Politécnico Nacional - México, Consultada: 28 de enero del 2008, [online]. Disponible en Internet: http://www.ctrl.cinvestav.mx/~yuw/pdf/MaTesCA.pdf

[VALLS J.] **"REDES DE NEURONAS DE BASE RADIAL"**, Consultada: 18 de junio del 2007, [online]. Disponible en Internet: http://et.evannai.inf.uc3m.es/docencia/rn-inf/documentacion/rnbr.pdf

ANEXOS

ANEXO A

ELEMENTOS DE UNA RED NEURONAL

Una neurona artificial está compuesta de varios elementos, los cuales son válidos aún cuando la neurona sea usada en la entrada, salida o capa oculta.

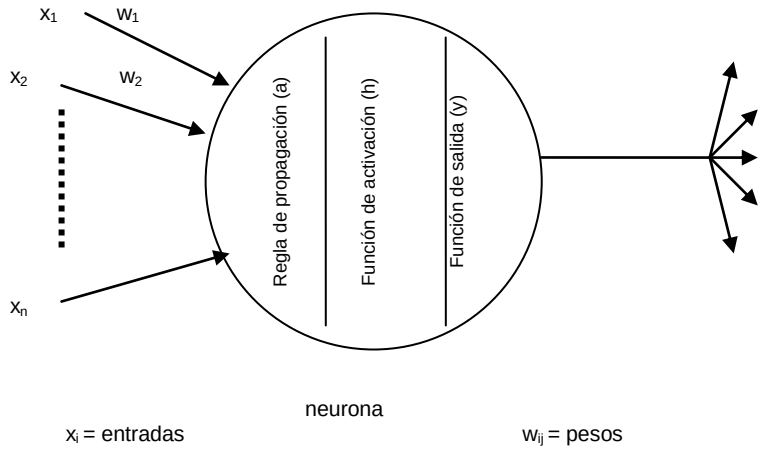

Fig. 1: Elementos básicos de una neurona artificial
Basado: [MARTINEZ I.]

- **Entradas**

Entradas x_i a la neurona.

- **Pesos**

Una neurona recibe un gran número de entradas las cuales tienen su propio peso relativo, el mismo que proporciona la importancia de la entrada dentro de la función de agregación de la neurona. Los pesos de la neurona artificial ejecutan la misma tarea que realizan las fuerzas sinápticas de las neuronas biológicas; en ambas neuronas algunas entradas son más significativas que otras, lo que hace que tengan mayor efecto sobre el procesamiento de la neurona al mezclarse para producir la respuesta neuronal.

Los pesos representan la intensidad de interacción entre cada neurona presináptica i y la neurona postsináptica j. Estas fuerzas pueden ser modificadas en respuesta de los ejemplos de entrenamiento de acuerdo a la topología o las reglas de entrenamiento.

- **Función de propagación**

Esta función permite obtener el valor del potencial postsináptico hj de la neurona a partir de las entradas y los pesos:

$$h_j = \sigma_j(w_{ij}, x_i) \quad (1)$$

"La función más habitual es la suma ponderada de todas las entradas. Podemos agrupar las entradas y pesos en dos vectores (x1, x2, ... , xn) y (w1, w2, ... , wnj) para calcular esta suma se realiza el producto escalar sobre estos dos vectores" [MARTÍNEZ I.].

$$h_j(t) = \sum_i w_{ij} \cdot x_i \tag{2}$$

Si el peso es positivo tenderá a excitar a la neurona postsináptica lo que se denomina sinapsis excitadoras, pero si el peso es negativo tendremos una sinapsis inhibidora.

- **Función de activación o transferencia**

El resultado de la función de propagación es transformada en la salida real de la neurona mediante un proceso algorítmico conocido como función de activación.

$$a_j = f_j(a_j(t-1), h_j(t)) \tag{3}$$

En este caso la función de activación depende del potencial postsináptico $h_j(t)$ y del propio estado de activación anterior. Sin embargo, en varios tipos de redes neuronales se considera que el estado actual de la neurona no depende de su estado anterior $a_j = (t - 1)$, sino del actual:

$$a_j(t) = f_j(h_j(t)) \tag{4}$$

La función de activación el valor de la salida puede ser comparada con algún valor umbral para determinar la salida de la neurona. Si la suma es mayor que el valor umbral la neurona generará una señal, caso contrario no genera señal. Las funciones de activación mas comunes son: escalón, lineal y sigmoidal.

- **Función escalón**: se utiliza cuando las salidas de la red son binarias. La salida de una neurona produce un 1 cuando el estado de activación es mayor o igual que cierto valor umbral, caso contrario produce 0. Las redes creadas por este tipo de neuronas son fáciles de implementar en hardware pero sus capacidades están limitadas.

- **Función lineal o mixta**: se representa con la expresión f(x) = x. Se define un límite inferior y otro superior. Si la suma de las señales de entrada es menor que el límite inferior la activación se define como [0 o -1], caso contrario la activación es 1. Para valores de entrada ubicados entre ambos límites, la activación seria una función lineal de la suma de las señales de entrada.

- **Función sigmoidal**: es la más apropiada cuando se quiere como salida información analógica. Con esta función, la mayoría de los valores del estímulo de entrada, el valor dado por la función es cercano a uno de los valores asintóticos. La importancia de esta función es que su derivada es siempre positiva y cercana a cero para los valores grandes positivos o negativos; además toma su valor máximo cuando x es 0.

- **Función de salida**

Proporciona la salida $y_j(t)$ en función del estado de activación. Esta salida es directamente equivalente al valor resultante de la función de activación.

$$y_j = F_j(a_j) = a_j \quad (5)$$

Algunas topologías de redes neuronales, pueden modificar el valor de la función de transferencia para agregar un factor de competitividad entre neuronas vecinas.

- **Función de error y el valor propagado hacia atrás**

La mayor parte de los algoritmos de entrenamiento necesita calcular la diferencia entre la salida actual y la esperada, esta diferencia es transformada por la función de error que en algunas veces es denominado **error actual**.

El error actual es propagado hacia atrás a la capa anterior, siendo este valor o bien el valor actual de error de esa capa obtenido al escalarlo de alguna manera o bien es tomado como la salida esperada.

TOPOLOGÍAS

De acuerdo al número de capas y a su conectividad las redes neuronales se pueden clasificar en dos grupos:

TOPOLOGÍA DE ACUERDO AL NÚMERO DE CAPAS

- **Redes monocapa:** están compuestas por una única capa de neuronas.
- **Redes multicapa (layered networks):** son aquellas cuyas neuronas se organizan en varias capas.

TOPOLOGÍA DE ACUERDO AL FLUJO DE DATOS EN LA RED NEURONAL

Las neuronas se suelen organizar en columnas paralelas denominadas capas, las cuales pueden estar conectadas en dos formas: FeedForward y red recurrentes.

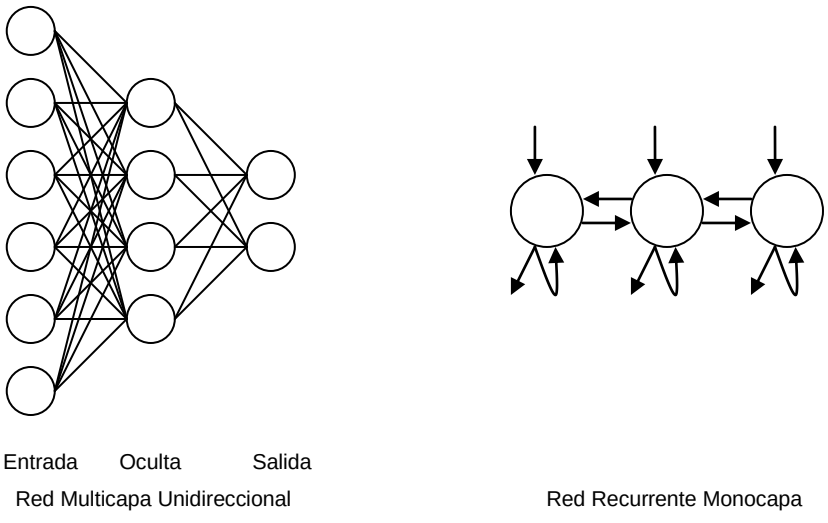

Fig. 2: Arquitectura de redes de neuronas artificiales
Fuente: [ROMERO L., 2007]

- **Red feedforward (Unidireccionales)**: consiste en capas de neuronas donde la salida de una neurona de una capa, alimenta todas las neuronas de la capa siguiente, es decir la información circula en un único sentido. Su arquitectura típica es de una red multicapa.

- **Redes Recurrentes:** la información puede circular entre las capas en cualquier sentido, debido a que poseen conexiones de realimentación que proporcionan un comportamiento dinámico. Su arquitectura típica es la de una red monocapa con una gran realimentación.

ANEXO A-1

TIPOS DE REDES NEURONALES

Las redes neuronales artificiales están diseñadas para solucionar problemas muy específicos y su arquitectura depende del problema a tratar. A continuación describiremos varias Redes Neuronales y para comprenderlas se realizó una notación, la misma que se la puede ver al finalizar los tipos de redes.

.

- **Perceptrón**

Es la primera red neuronal desarrollada en 1943 por Warren McCulloch y Walter Pitts, que consiste en una suma de las señales de entrada, multiplicadas por pesos escogidos aleatoriamente. En definitiva, se trata de una unidad de proceso con dos únicos valores de salida posibles (0 ó 1) y constantes. Sin embargo, esta red neuronal cada vez fue mejorando de acuerdo a algunos estudios hechos por psicólogos y matemáticos, uno de ellos es Donald Hebb quién contribuyo con la mejora de los aspectos biológicos de la neurona artificial.

En el año 1957, el psicólogo Frank Rosenblatt aplica las ideas de aprendizaje de Hebb a la neurona McCulloch y Pitts, dando origen a la red tipo Perceptrón denominados máquinas de aprendizaje. "El primer modelo de Perceptrón fue desarrollado en un ambiente biológico" [ACOSTA et al, 2000], por lo tanto su estructura es inspira en las primeras etapas de procesamiento de los sistemas sensoriales de los animales (visión). De acuerdo a esto, el modelo que se conoce como perceptrón de McCulloch-Pitts fue la base de la mayor parte de las arquitecturas de las RNA, aunque se diferencian en su activación, modelo de neurona, implementación y modo de funcionamiento, todos ellos tienen rasgos comunes. Actualmente son usadas en varios campos debido a su capacidad aprender y reconocer patrones.

Estructura
La red perceptrón es un modelo no lineal y unidireccional que esta compuesta por dos capas de neuronas, una sensorial o de entradas y otra de salida.

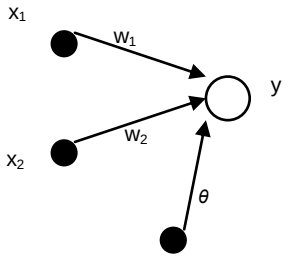

Fig. 3: Red de una sola capa con una salida y dos entradas
Basado: [KROSE et al., 1996]

Su modo de funcionamiento es simple, la función de proceso de las entradas *x* es una suma ponderada de los valores de las mismas, lo que significa que no todos los valores de entrada tienen igual aportación para la suma, sino que cada uno de ellos se modifica multiplicándolo por un valor de ponderación o peso *w*, antes de proceder a sumar.

Expresado de manera formal la salida de la neurona quedaría:

$$y = f\left[\sum_{i=1}^{n} w_i x_i + \theta\right] \tag{1}$$

Habitualmente la función de activación de las neuronas de la capa de salida es de tipo escalón. Esto permite que la salida pueda asumir bien valores discretos, bien valores continuos dentro de un determinado intervalo.

$$f(x_1,\ldots,x_n) = \begin{cases} 1 & si \quad w_o + w_1 x_1 + \ldots + w_n x_n + \theta > 0 \\ -1 & si \quad w_o + w_1 x_1 + \ldots + w_n x_n + \theta < 0 \end{cases} \tag{2}$$

Por lo anteriormente expresado, la salida del perceptrón es utilizado para tareas de clasificación. Por ejemplo, sea una neurona tipo perceptrón de dos entradas (x_1) y (x_2) con salida (y), la separación de estas dos categorías es mediante una recta de ecuación:

$$w_1 x_1 + w_2 x_2 + \theta = 0 \tag{3}$$

Una representación geométrica de la ecuación anterior se puede escribirse como:

$$x_2 = -\frac{w_1}{w_2} x_1 - \frac{\theta}{w_2} \tag{4}$$

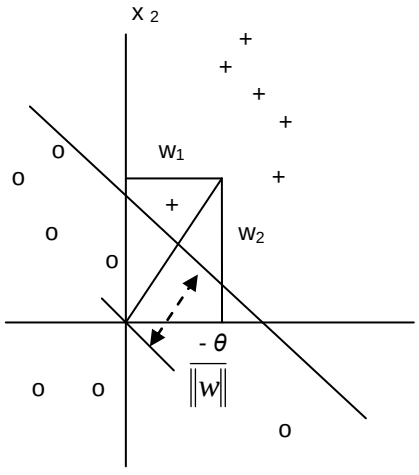

Fig. 4: Representación geométrica de la función del discriminante y los pesos
Basado: [KROSE et al., 1996]

Regla de aprendizaje
Para el aprendizaje del perceptrón se usa diferentes métodos, siendo uno de estos la regla de aprendizaje por corrección de error. La cual consiste en ajustar los pesos en base a la

diferencia entre los valores de salida deseados *d* y los obtenidos en la salida de la red *y* es decir, en función del error cometido. Formalmente viene dada por la expresión:

$$\Delta w_{ij} = \gamma x_i (d_j - y_j)$$

(5)

donde $d_j - y_j$ es el error originado en la neurona *j*.

Así, el peso (w_{ij}) se puede ajustar de tal forma:

$$w_{ij}(+1) = w_{ij}() + \Delta w_{ij}()$$

(6)

Limitación de la red Perceptrón
El perceptrón sólo puede resolver funciones definidas por un hiperplano (objeto de dimensión N-1 contenida en un espacio de dimensión N) que corte un espacio de dimensión N. Un ejemplo de una función que no puede ser resuelta es el operador lógico XOR.

Tipos de perceptrón

- El Perceptrón de dos capas (entrada con neuronas lineales, analógicas, y la de salida con función de activación de tipo escalón, digital) establece dos regiones separadas por una frontera lineal en el espacio de patrones de entrada, donde se tendría un hiperplano.

- Perceptrón con tres niveles origina regiones convexas, que se forman mediante la intelección entre las regiones formadas por cada neurona de la segunda capa y cada uno de estos elementos se comporta como un Perceptrón simple.

- "Un Perceptrón con cuatro capas puede generar regiones de decisión arbitrariamente complejas. El proceso de separación en clases, consiste en la partición de la región deseada en pequeños hipercubos. Cada hipercubo requiere 2n neuronas en la segunda (siendo n el número de entradas a la red), una por cada lado del hipercubo, y otra en la tercera capa, que lleva a cabo el AND lógico de la salida de los nodos del nivel anterior. La salida de los nodos de este tercer nivel se activaran solo para las entradas de cada hipercubo. Los hipercubos se asignan a la región de decisión adecuada mediante la conexión de la salida de cada nodo del tercer nivel solo con la neurona de salida (cuarta capa) correspondiente a la región de decisión en la que este comprendido el hipercubo llevándose a cabo una operación lógica Or en cada nodo de salida. Este procedimiento se pude generalizar de manera que la forma de las regiones convexas sea arbitraria, en lugar de hipercubos.

- El Perceptrón de 4 capas puede solucionar una gran variedad de problemas cuyas entradas sean analógicas, la salida sea digital y sea linealmente separable" [TREC, 2000].

- **Perceptrón multicapa**

Fue creado para resolver el problema de la función XOR. Este tipo de red se basa en otra red más simple llamada perceptrón simple solo que el número de capas ocultas puede ser mayor o igual que una. Además el perceptrón multicapa es una red unidireccional (feedforward), con alimentación hacia delante.

Su arquitectura es la siguiente:

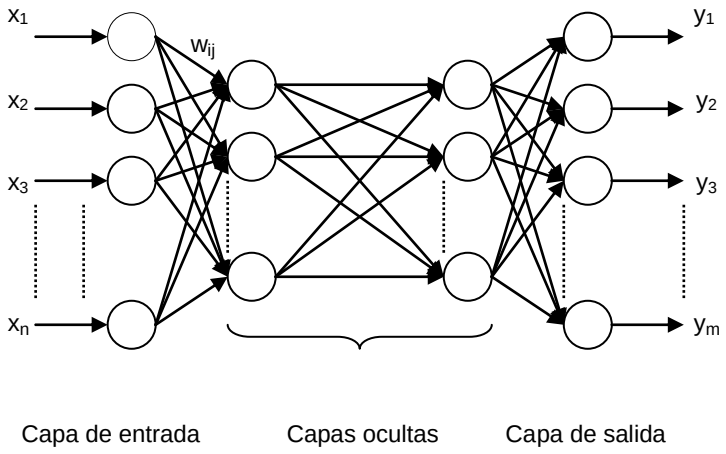

Fig. 5: Topología de un Perceptrón Multicapa
Basado: [MUÑOZ J., 2005-2006]

La capa oculta usa como regla de propagación la suma ponderada de las entradas con los pesos sinápticos wij y sobre ella se aplica una función de transferencia de tipo sigmoide.

- **Adaline**

Bernard Widrow y su estudiante Ted Hoff de la universidad de Stanford en 1959, presentan un modelo llamado "ADALINE" basado en la neurona de McCulloch-Pitts y su regla de aprendizaje llamada algoritmo LMS (Least Mean Square). De acuerdo a esto, Adaline se la considera una red similar al Perceptrón excepto en su función de transferencia (tipo lineal) y su modo de trabajo, mientras que este último sólo trabaja con entradas y salidas binarias o bipolares, Adaline trabaja con patrones de entrada y salida reales.

Adaline es una sigla que proviene de ADAptive LInear NEuron (Neurona Lineal Adaptiva) para pasar después a ser Adaptive LInear Element (Elemento Lineal Adaptivo), este cambio se debió a que la Adaline es un dispositivo que está formado de un único elemento de procesamiento.

La utilización principal de la red Adaline ha sido en el campo del procesamiento de señales (el filtro de ruido y los filtros adaptativos) y que en la actualidad sigue ocupando un lugar muy importante en la industria.

Estructura
El Adaline está compuesto de un combinador adaptativo lineal (ALC) y un cuantificador (función bipolar de salida). Se alimenta de un vector de entrada denotado por X_i^p , i = 0,1,…,n y con una entrada constante igual a 1 denominada sesgo (bias). La salida viene dada por la suma ponderada de los valores de las señales de entrada con sus pesos asociados, si el resultado es positivo la salida es 1, en caso contrario es 0 (o -1).

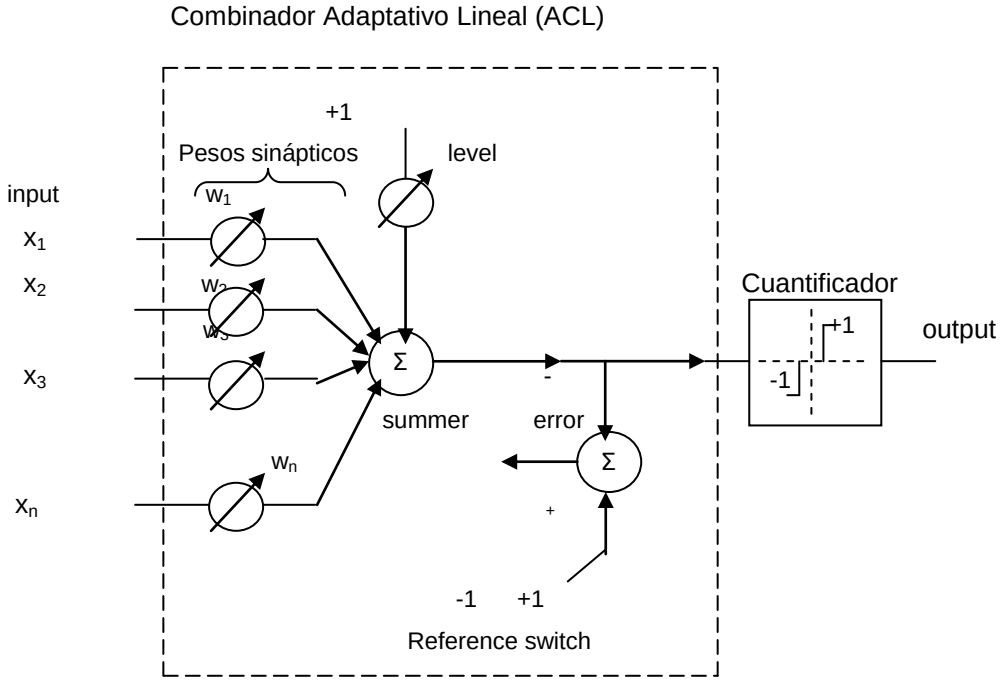

Fig. 6: El Adaline
Basado y modificado: [KROSE et al., 1996]

Formalmente expresado quedaría:

- El ALC realiza la suma ponderada de las entradas, la misma que produce una salida definida en la siguiente ecuación:

$$y = \sum_{i=1}^{n} w_i x_i + \theta \tag{1}$$

Regla de aprendizaje
La red ADALINE utiliza la regla LMS (Least Mean Squared) o regla del mínimo cuadrado medio. También conocida como regla delta, porque trata de de modificar los pesos para tratar de reducir la diferencia entre la salida deseada y la salida producida para patrón ($d^p - y^p$). El error cuadrático medio (E) cometido por el ADALINE para todo el conjunto de patrones es:

$$E = \sum_{p=1}^{n} E^p = \frac{1}{2}\sum_{p=1}^{n} \left(d^p - y^p \right)^2 \tag{2}$$

Realizada esta operación, la regla Delta busca minimizar dicho error mediante la modificación del vector de pesos ($w_0, \ldots, w_n$), para esto utiliza el método de Descenso de Gradiente, que se fundamenta en los siguientes puntos:

- Primeramente realiza un cambio en cada peso proporcional a la derivada del error, medida en el patrón actual, respecto del peso:

$$\Delta w_j = -\gamma \frac{\partial E^p}{\partial w_j}$$

(3)

- En segundo lugar, aplica la regla de la cadena dándonos como resultado la siguiente derivada:

$$\frac{\partial E^p}{\partial w_j} = \frac{\partial E^p}{\partial y^p} \frac{\partial y^p}{\partial w_j}$$

(4)

- Tercero, como son unidades lineales, sin función de activación se cumple:

$$\frac{\partial y^p}{\partial w_j} = x_j \qquad (5)$$

$$\frac{\partial E^p}{\partial y^p} = -\left(d^p - y^p\right) \qquad (6)$$

- Y finalmente sustituyendo en la ecuación (3) queda:

$$\Delta w_j = \gamma\left(d^p - y^p\right) x_j$$

(7)

Donde: $\delta^p = d^p - y^p$ → es la gradiente local del patrón *p*.

$$\Delta w_j = \gamma \delta^p x_j$$

(8)

"Siguiendo este método se garantiza, que para un conjunto de entrenamiento adecuado, después de un número finito de iteraciones el error se reduce a niveles aceptables. El número de iteraciones necesarias y el nivel de error deseado dependen de cada problema particular" [VILLALOBOS et al, 2005].

- **Backpropagation**

En 1986, Rumelhart, Hinton y Williams, establecieron un método para que una red neuronal aprendiera la asociación que existe entre los patrones de entrada y las clases correspondientes, utilizando varios niveles de neuronas. El método backpropagation (propagación del error hacia atrás) basado en la generalización de la regla delta, a pesar de sus limitaciones ha ampliado de forma considerable el rango de las aplicaciones de las redes neuronales.

Uno de los grandes avances logrados con la red Backpropagation, es el aprovechamiento de la naturaleza paralela de las redes neuronales para minimizar el tiempo requerido por un procesador secuencial para determinar la correspondencia entre unos patrones dados.

La Backpropagation es un tipo de red de aprendizaje supervisado, que emplea un ciclo propagación – adaptación de dos fases. Su funcionamiento consiste en el aprendizaje de un conjunto predefinido de pares de entradas-salidas dados como ejemplo: primero se aplica un patrón de entrada como estímulo para la primera capa de las neuronas de la red, se va propagando a través de todas las capas superiores hasta generar una salida, se compara el resultado en las neuronas de salida con la salida que se desea obtener y se calcula un valor de error para cada neurona de salida. Las salidas de error se transmiten hacia atrás, partiendo de la capa de salida hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida, recibiendo de error aproximado a la neurona intermedia a la salida original. "Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total. Basándose en el valor de la señal de error recibido, se reajustan los pesos de conexión de cada neurona, para hacer que la red converja hacia un estado que permita clasificar correctamente todos los patrones de entrenamiento" [BOZA G., 2003].

La importancia de este proceso, reside en su capacidad de autoadaptar los pesos de las neuronas de las capas intermedias para aprender a reconocer distintas características que existe en un conjunto de patrones de entrada y sus salidas correspondientes. Además es importante la capacidad de generalización, facilidad de dar salidas satisfactorias a entradas que el sistema no ha visto nunca en su fase de entrenamiento. La red debe encontrar una representación interna que le permita generar las salidas deseadas cuando se le dan entradas de entrenamiento, y que pueda aplicar, además, a entradas no presentadas durante la etapa de aprendizaje para clasificarlas.

Estructura
Una red backpropagation o multicapa tiene una capa de entrada, una capa de salida, y al menos una capa oculta de neuronas internas. No hay ningún límite teórico en el número de capas ocultas pero normalmente es una o dos. En la siguiente figura se puede observar la estructura típica de este tipo de redes neuronales:

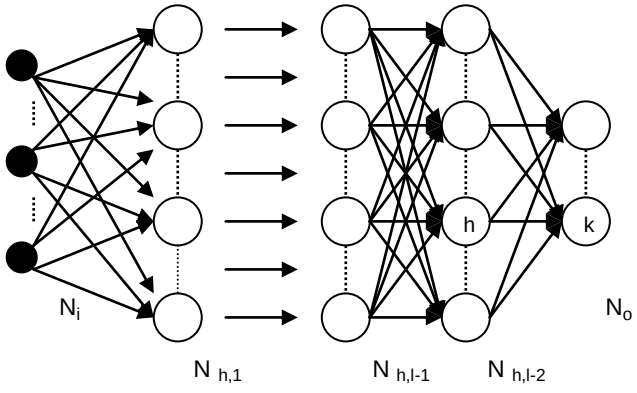

Fig. 7: Red multicapa
Fuente: [KROSE et al., 1996]

En esta red se interconectan varias unidades de procesamiento en capas, en la cual las neuronas de cada capa no se interconectan entre sí, pero cada neurona de una capa

proporciona una entrada a cada una de las neuronas de la siguiente capa, con la finalidad de que cada neurona transmita su señal de salida a cada neurona de la capa siguiente.

La regla delta generalizada

La regla Delta propuesta por Widrow, ha sido extendida a redes con capas intermedias con conexiones hacia delante (*feedforward)* y cuyas células tienen funciones de activación continuas (lineales o sigmoidales), dando lugar al algoritmo Backpropagation. Ambos algoritmos realizan la tarea de actualización de pesos y ganancias con base en el error medio cuadrático.

Para iniciar el entrenamiento de la red se presenta un patrón de entrada x_i (i = 0,1,…,n) y un patrón de entrenamiento *p*, este último se propaga a través de las conexiones existentes originando una entrada neta *S* en cada neurona de la siguiente capa. Por lo tanto "la entrada neta a la neurona j de la siguiente capa debido a la presencia de un patrón de entrenamiento en la entrada esta dada por la ecuación" [ACOSTA et al, 2000]:

$$s_j^h = \sum_i w_{ji} x_i + \theta_j \tag{1}$$

Una vez obtenida la entrada neta, se calcula la salida de cada una de las neuronas de la capa oculta:

$$y_j^h = f\left(\sum_i w_{ji} x_i + \theta_j \right) \tag{2}$$

donde las salidas y_j^h de las neuronas de la capa oculta son las entradas a los pesos de conexión de la capa de salida, este comportamiento esta descrito por la ecuación:

$$s_k^p = \sum_j w_{jk} y_j^h + \theta_k \tag{3}$$

Por último, se obtiene la salida final que produce la red:

$$y_k^p = f(s_k^p) \tag{4}$$

Para calcular los términos de error de las neuronas, se usa el método de gradiente descendiente. En este caso, el incremento de los parámetros se expresa como:

$$\Delta w_{jk} = -\gamma \frac{\partial E^p}{\partial w_{jk}} \tag{5}$$

En cuanto a la medida del error E^p, el error cuadrático total para cada patrón *p* propagado, esta dado por:

$$E^p = \frac{1}{2}\sum_{k=1}^{No}\left(d_k^p - y_k^p\right)^2 \tag{4}$$

donde d_k^p es la salida deseada para la unidad *o* ante la presentación del patrón *p*, y y_k^p es la salida de la red.

A partir de esta expresión se puede obtener una medida general de error $E = \sum_p E^p$, escrita en la siguiente ecuación:

$$\frac{\partial E^p}{\partial w_{jk}} = \frac{\partial E^p}{\partial s_k^p}\frac{\partial s_k^p}{\partial w_{jk}} \tag{5}$$

Para la ecuación (1), el segundo factor es:

$$\frac{\partial s_k^p}{\partial w_{jk}} = y_j^p \tag{6}$$

De la ecuación (5) del primer factor, se obtiene la siguiente ecuación:

$$s_k^p = -\frac{\partial E^p}{\partial s_k^p} \tag{7}$$

Finalmente, tenemos las actualizaciones de los pesos:

$$\Delta w_{jk} = \gamma \delta_k^p y_j^p \tag{8}$$

- **Base Radial**

Creadas por M.J.D. Powell, D.S. Broomhead y D. Lowe a mediados de los 80. Las redes de Base Radial son redes multicapa con conexiones hacia delante, su salida depende de la distancia a un punto denominado Centro.

Su filosofía de neuronas es muy diferente a las del resto de arquitecturas de red, por lo que requieren más neuronas que las redes feed-foward, pero generalmente se diseñan en una fracción de tiempo menor que las redes backpropagation. Se obtienen excelentes resultados si se entrenan con los vectores apropiados.

Además estas redes tienen una única capa oculta. Las neuronas ocultas poseen carácter local, esto se debe al uso de las funciones de base radial como funciones de activación, generalmente la función gaussiana.

Dentro de las características, más sobresaliente de la red de base radial tenemos:

- Son funciones que poseen un carácter local debido que alcanzan un nivel cercano al máximo de su recorrido cuando el patrón de entrada *x* está próximo al centro de la neurona, es decir son simétricas respecto de x=0.

- Definen dos parámetros:
  Centro: punto donde la función posee un extremo.
  Anchura: magnitud de la variación de la función según se aleja del centro.

Estructura
Las redes de Base Radial o redes BR son arquitecturas que disponen únicamente una capa oculta, con lo que en total, suman 3 capas: entrada, oculta y salida, como se puede observar en la siguiente figura:

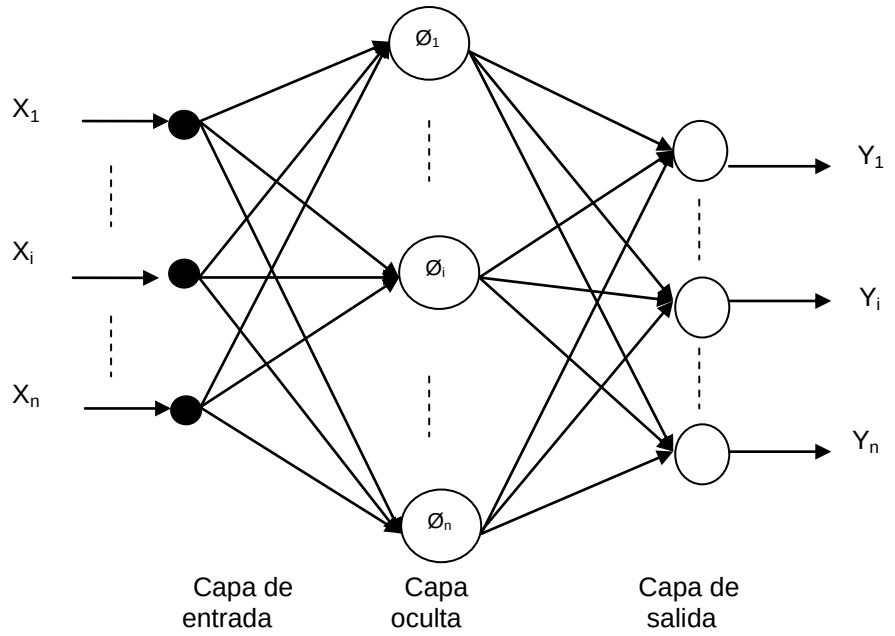

Fig. 8: Red de Base Radial
Basado y modificado: [VALLS J.]

- Capa de entrada: reciben las señales del exterior, no realizan ningún preprocesado.

- Capa Oculta: reciben las señales de la capa de entrada y realizan una transformación local y no lineal sobre dichas señales.

- Capa de Salida: Se realiza una combinación lineal de las activaciones de las neuronas de la capa oculta y actúa como salida de la red.

Las funciones de transferencia de la capa oculta son similares a una función de densidad Gaussiana, es decir:

$$\phi_i(\vec{x}) = \phi\left(\frac{\|X(\vec{x}) - C_i\|}{\sigma_i}\right)$$

(1)

donde ϕ es una función de base radial, $C_i(c_{i1}, c_{12}, ..., c_{ip})$ son vectores centros de las funciones de base radial y σ_i representa la desviación, anchura o dilatación de la función de base radial asociada a dicho elemento.

Quedando la salida de la red, de la siguiente manera:

$$y = \sum_{i=1}^{n} w_i \phi_i(\vec{x}) + \theta$$

(2)

Regla de aprendizaje
El entrenamiento de este tipo de redes, se puede dar por medio de los siguientes métodos de aprendizaje:

- ✓ **Método híbrido**

Se da en dos fases: no supervisada y la supervisada.

- Fase no-supervisada: Determina los parámetros de la capa oculta, puesto que las neuronas ocultas se caracterizan porque representan zonas diferentes del espacio de entradas, los centros y las desviaciones deben de ser calculados con este objetivo (clasificar el espacio de entradas en diferentes clases).

  Determinación de Centros
  - Algoritmo K-medias.
  - Mapas de Kohonen.

  Determinación de Desviaciones
  "Se deben calcular de manera que cada neurona de la capa oculta se active en una región del espacio de entradas y de manera que el solapamiento de las zonas de activación de una neurona sea lo más ligero posible, para suavizar así la interpolación" [GONZÁLEZ M.].

  Una opción bastante efectiva es determinar la amplitud de la función de base radial como la media geométrica de la distancia del centro a sus dos vecinos más cercanos:

$$\sigma_i = \sqrt{\|C_i - C_t\| \ \|C_i - C_s\|}$$

(4)

C_t y C_s → los dos centros más cercanos al centro C_i.

- Fase supervisada: Determina de forma supervisada los pesos y umbrales de la capa de salida, para minimizar las diferencias entre las salidas de la red y las salidas deseadas. Con este proceso se obtiene la minimización de una función error computada en la salida de la red. El aprendizaje a utilizar es el de corrección de error.

✓ **Método totalmente supervisado**

Este tipo de aprendizaje no conserva, en principio las propiedades o características locales de las redes de base radial. En este caso, todos los parámetros de la red, centros, amplitudes, pesos y umbrales, se determinan de manera completamente supervisada y con el objetivo de minimizar el error cuadrático medio. Por lo tanto, las salidas de la red solo dependen de los pesos.

- **Recurrentes (Feedback Networks)**

Estas redes se caracterizan porque los valores de activación experimentan un proceso de relajación, pero los pesos permanecen sin cambios. Las redes recurrentes poseen lazos de realimentación, los mismos que pueden ser entre neuronas de diferentes capas, neuronas de la misma capa o entre una misma neurona. Cuando la red es inherentemente estable, la red evoluciona hacia un estado en el cual no hay cambios en los valores de las neuronas y las salidas se mantienen constantes, en otro caso, la red cambiaría sus valores de activación hasta el infinito.

El modelo de Elman, Hopfield, máquina de Bolzman son ejemplos de red neuronal recurrente.

Estructura

Su estructura típica es la de una red monocapa con realimentación. Esta estructura posibilita el estudio dinámico de sistemas no lineales. La siguiente figura representa el esquema de una red recurrente:

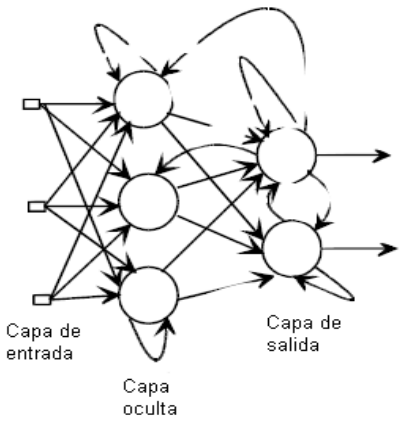

Fig. 9: Red neuronal recurrente
Basado: [SORIA E.]

- **Red de Elman**

En 1990, Elman modifica la arquitectura de la red de Jordan considerando realimentaciones desde las capas ocultas hacia la capa de contexto, sin considerar realimentaciones locales.

Estructura

La red de Elman está formada por una capa de entrada, dos capas intermedias (una capa oculta y otra capa de contexto) y una capa de salida. Los sensores de entrada y las unidades

de salida recogen la información del entorno, es decir la proveída por el conjunto de patrones de entrenamiento. Las unidades de salida toman la señal de la salida de las unidades de la capa oculta ponderadas por los correspondientes pesos sinápticos y suelen usar como función de transferencia la función tipo lineal.

Lo interesante de este modelo es "la introducción de las unidades de contexto, las mismas que se utilizan para memorizar las salidas de las unidades ocultas en la etapa anterior, de manera que cada unidad de contexto tiene como salida la salida de la unidad oculta correspondiente en la etapa anterior. Por lo tanto, esta red es sólo parcialmente recurrente y cada unidad de proceso oculta recibe como entrada las salidas de las unidades de contexto y de los sensores de entrada ponderadas por sus pesos sinápticos. De esta manera la salida de la red depende no sólo del patrón de entrada actual sino también de los patrones anteriores a través de las unidades de contexto" [MUÑOZ J., 2005-2006]. Las unidades de proceso de la capa oculta tienen como función de transferencia la función sigmoidal.

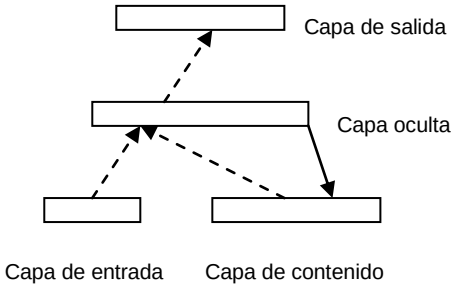

Fig. 10: Red de Elman
Fuente: [KROSE et al., 1996]

Entrenamiento de la red
Debido a la estructura similar de la red de Elman con la red Backpropagation, está red puede entrenarse con cualquier algoritmo de propagación:

El entrenamiento, se resume en los siguientes pasos:
- Presentar a la red, los patrones de entrenamiento y calcular la salida de la red con los pesos iníciales, comparar la salida de la red con los patrones objetivo y generar la secuencia de error.
- Propagar inversamente el error para obtener el gradiente del error para cada conjunto de pesos y ganancias.
- Actualizar cada uno de los pesos y ganancias con el gradiente encontrado con base al algoritmo de propagación inversa.

La red de Elman no es tan confiable, ya que el gradiente se calcula en base a una aproximación del error. Además para solucionar un problema con este tipo de red se necesitan muchas neuronas en la capa oculta.

- **Red de Hopfield**

En 1982, John Hopfield presenta un trabajo en el cuál desarrolla la idea del uso de una función de energía para comprender la dinámica de una red neuronal recurrente con uniones sinápticas simétricas. Pero este trabajo sólo permite salidas bipolares 0 ó 1, por lo que posteriormente Hopfield amplía la función energía planteada para estos sistemas permitiendo la salida continua de las neuronas. Hopfield con estos trabajos realizados, crea la primera red neuronal recurrente de naturaleza dinámica denomina Red de Hopfield.

Estas redes son muy similares a la red tipo Perceptrón, pero presentan una característica primordial: no son auto-recurrentes, ya que las salidas de las neuronas se comunican con todas las demás pero no consigo misma.

La red de Hopfield funciona como una memoria asociativa no lineal que almacena internamente patrones presentados de forma incompleta o con ruido. Por lo que el principal uso de estas redes ha sido como memorias autoasociativas y como instrumento para resolver problemas de optimización (por ej. el problema del viajante).

Estructura
Las redes Hopfield tienen una estructura de una red monocapa con N neuronas interconectadas totalmente, que actualizan sus valores de activación de forma asíncrona e independiente del resto de las neuronas de procesado. En esta red todas las neuronas son a la vez de entrada y salida.

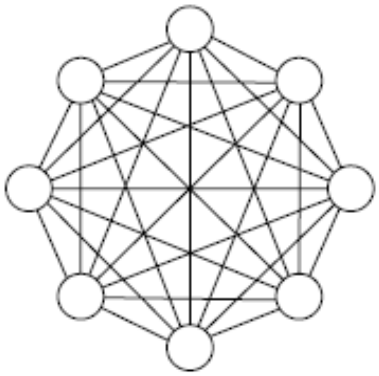

Fig. 11: Red auto-asociativa
Fuente: [KROSE et al., 1996]

Sus valores de salida son binarios: 0/1 ó -1/1, siendo las funciones de activación de las neuronas del tipo escalón. Esto quiere decir que si la suma de las entradas de las neuronas es mayor o igual que el umbral la activación es 1, caso contrario la activación es 0 (o -1). Expresado matemáticamente el estado del sistema, tenemos:

$$s_j(t+1) = \sum_{i \neq j} x_i(t) w_{ij} + \theta_j \tag{1}$$

$$y_j(t+1) = \begin{cases} +1 & if\ s_j(t+1) > 0 \\ -1 & if\ s_j(t+1) > 0 \\ y_j(t) & otherwise \end{cases} \tag{2}$$

donde $y_j(t+1) = \mathrm{sgn}(s_j(t+1))$

Una neurona *j* en la red de Holpfield, se llama estable cuando mantiene su valor de activación en el tiempo *t*. De acuerdo con las ecuaciones (1) y (2), tenemos:

$$y_j = \operatorname{sgn}\left(\xi_j\right)$$

(3)

Por lo tanto, se dice que tenemos un estado estable cuando todas las neuronas de procesado son estables.

Con la restricción extra de simetría en los pesos de conexión, $w_{ij}=w_{ji}$, el sistema puede ser descrito mediante una función energía de la forma:

$$\varepsilon = -\frac{1}{2}\sum_{i \neq j}\sum x_i y_j w_{ij} - \sum_k \theta_j y_j$$

(4)

Regla de aprendizaje
La elección de la regla de aprendizaje no es trivial, ya que depende de la interrelación de los patrones que se desea memorizar. Si estos patrones están poco correlacionados se aplica la regla de Hebb o regla del producto. La cuál establece que: si la salida deseada y la entrada son ambas activas o inactivas, el peso de conexión se incrementa usando la tasa de aprendizaje, en otro caso se decrementa el peso usando la tasa de aprendizaje.

- **Competitivas**

En las redes con aprendizaje competitivo, suele decirse que las neuronas compiten unas con otras con el único propósito de llevar a cabo una tarea dada. Con este tipo de aprendizaje se pretende que cuando se presente a la red cierta información de entrada, sólo una de las neuronas de salida de la red, o una por cierto grupo de neuronas, se active (alcance su valor de respuesta máximo). Por tanto las neuronas compiten para activarse quedando finalmente una, o una por grupo, como neurona vencedora y el resto quedan anulados y siendo forzadas a sus valores de respuesta mínimos.

La competición entre neuronas se realiza en todas las capas de la red, existiendo en estas redes neuronas con conexiones de autoexcitación (signo positivo) y conexiones de inhibición (signo negativo) por parte de neuronas vecinas.

El objetivo de este aprendizaje es categorizar (clusterizar) los datos que se introducen en la red, de esta forma las informaciones similares son clasificadas formando parte de la misma categoría y por tanto deben activar la misma neurona de salida.

Estructura
Una red competitiva está compuesta por *X* sensores de entrada, *Y* unidades de proceso (neuronas artificiales), y conexiones entre cada sensor y cada unidad de proceso, de manera que la conexión entre el sensor *i* y la unidad de proceso *j* tiene asociado un valor *wij*.

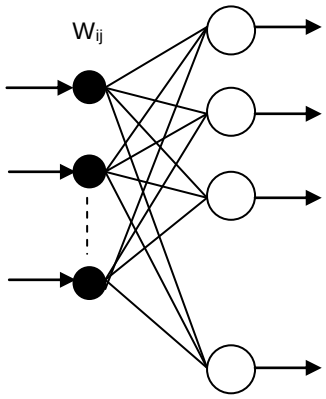

Fig. 12: Arquitectura de la red
Basado y modificado: [ACOSTA et al, 2000]

Para cada neurona de entrada recogida por los sensores solamente una unidad de proceso se activa, es decir, aquella que tiene el mayor potencial sináptico por lo que también se le considera como la unidad ganadora.

NOTACIÓN

Se describe a continuación la notación empleada en las redes neuronales, algunos significados pueden cambiar localmente dependiendo del contexto.

$x = [\chi_1, \chi_2 \cdots \chi_n]$	Vector n-dimensiones
χ_i	Entradas i-ésima de una neurona.
w_i	Peso i-ésima (tomando el conjunto de pesos como un vector).
w_{ij}	Peso de la conexión entre la neurona *i* y la *j*.
y_j	Estado interno de la neurona *j*.
f	Función de transferencia (o función de activación).
θ	Umbral activación.
d_j	Valor esperado de la neurona *j*.
y_j	Valor de la salida de la neurona *j*.
E^p	Error del patrón *p*
γ	Factor de aprendizaje (0 < γ <= 1), controla la velocidad de aprendizaje.
h	Representa la capa oculta.

Δw Variación del peso.

γ Factor de aprendizaje (0 < γ <= 1), controla la velocidad de aprendizaje.

ε Energía de la red.

ANEXO B

TIPOS DE AGENTES

De acuerdo a la definición de Nwana, muestra los diferentes tipos de elementos que él considera agentes. "Esta clasificación representa una definición del término general agente como la unión de todos los tipos de agentes, los cuales se alinean en base a tres atributos ideales que son: autonomía, capacidad de aprendizaje y cooperación. De las combinaciones de estas características, aparecen otra serie de clases de agentes identificadas por otra serie de cualidades (movilidad, uso de información, etc.). En conjunto los agentes se clasifican en [HERNANSÁEZ et al, 2005]:

- **Agentes colaborativos**

Los agentes colaborativos ponen un mayor énfasis en los aspectos de autonomía y cooperación. Este tipo de agente dispone de los mecanismos necesarios para negociar unas condiciones de cooperación satisfactorias que les permitan resolver un problema general.

Según MUÑIZ, los agentes colaborativos pueden usarse para:
- Resolver problemas que son demasiado grandes para sistemas centralizados (debido a limitaciones de recursos o en los que se necesita tolerancia a fallas).
- Permitir la interconexión y operación de sistemas existentes.
- Dar solución a problemas inherentemente distribuidos.
- Dar solución a problemas en los que existen varias fuentes de información.
- Dar solución a problemas en donde la experiencia se encuentra distribuida.

- **Agentes de interfaz**

El modelo de "agente de este tipo es el asistente personal que colabora directamente con el usuario en su propio entorno de trabajo; su aplicación más habitual es la de guiar procesos de aprendizaje en un nuevo entorno o como herramientas que aprenden de las acciones del usuario con la intención de conseguir un mayor aprovechamiento del entorno" [HERNANSÁEZ et al, 2005].

"Los agentes de interfaz aprenden para mejorar su ayuda al usuario en cuatro formas:
- Al observar e imitar al usuario.
- Al recibir retroalimentación del usuario.
- Al recibir instrucciones explícitas del usuario.
- Al pedir consejo a otros agentes" [MUÑIZ E., 1999].

- **Agentes móviles**

Este grupo de agentes, son programas de software capaces de viajar por redes de computadoras como por Internet, de interactuar con hosts, pedir información a nombre de su usuario y regresar a su lugar de origen una vez que ha realizado las tareas especificadas por su usuario.

Algunos de los beneficios de la implementación de los agentes móviles son:
- Reducen los costos de comunicación.
- Proporcionan a un desarrollo natural de ambientes al implementar servicios de libre comercio.
- Proveen una única arquitectura de computación distribuida.

- **Agentes de internet/ información**

Este tipo de agentes se centran en gestionar, manipular y registrar fuentes de información que se encuentran distribuidas. Su principal aplicación ha sido en la exploración de la información contenida en los documentos WWW. Los agentes desarrollados con la intención de obtener

esta información se suelen denominar softbots (software robot). Contrariamente de lo que pueda parecer, este tipo de agentes no suelen ser agentes móviles, más bien suele tratarse de sistemas integrados dentro de la herramienta de navegación (browser) [HERNANSÁEZ et al, 2005].

- **Agentes reactivos**

Los agentes reactivos también son conocidos como agentes situacionales, por que responden a cambios en el entorno en que se encuentra situado. Son aplicados en situaciones donde se toma la decisión de llevar a cabo una acción, ya que estas situaciones no involucran un proceso deliberativo ni hace referencia a la experiencia del agente, sino a la situación actual del agente, la cual implicará que se tome una acción determinada.

- **Agentes híbridos**

Estrictamente, este elemento de clasificación no se corresponde con un tipo concreto de agente, por lo que se define como una combinación de las características de dos tipos de agentes, dentro de un agente simple (móvil, interfaz, colaborativo, etc.). Los beneficios obtenidos de tener una combinación de filosofías dentro de un agente simple son mayores que los de un agente basado en una filosofía determinada.

ANEXO C

LOGS - RECOLECTADOS A NIVEL DE SERVIDOR WEB

90.11.17.161 - - [05/Jun/2007:11:28:23 -0500] "GET / HTTP/1.1" 200 12326

172.16.5.68 - - [05/Jun/2007:11:28:31 -0500] "GET / HTTP/1.1" 200 12326

172.16.5.68 - - [05/Jun/2007:11:28:32 -0500] "GET /verificar.js HTTP/1.1" 304 -

39.135.uio.satnet.net - - [05/Jun/2007:11:28:22 -0500] "GET /mail/src/webmail.php HTTP/1.1" 200 330

172.16.5.68 - - [05/Jun/2007:11:28:32 -0500] "GET /utpl.css HTTP/1.1" 304 -

gdr4.utpl.edu.ec - - [05/Jun/2007:11:28:33 -0500] "GET /mail HTTP/1.0" 301 236

gdr4.utpl.edu.ec - - [05/Jun/2007:11:28:33 -0500] "GET /mail/ HTTP/1.0" 302 14

172.16.5.68 - - [05/Jun/2007:11:28:33 -0500] "GET /images/buscador-lupa.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:33 -0500] "GET /images/contactos.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:33 -0500] "GET /images/cabecera.jpg HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:33 -0500] "GET /images/btn-i-0act.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:33 -0500] "GET /images/textura2.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:33 -0500] "GET /images/btn-lu-0.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:33 -0500] "GET /images/btn-oa-0.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:33 -0500] "GET /images/btn-ed-0.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:33 -0500] "GET /images/btn-cittes-0.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:34 -0500] "GET /images/btn-ri-0.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:34 -0500] "GET /images/cuadro-gris.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:34 -0500] "GET /images/utpl-diners.gif HTTP/1.1" 304 -

gdr4.utpl.edu.ec - - [05/Jun/2007:11:28:33 -0500] "GET /mail/src/login.php HTTP/1.0" 200 3411

172.16.5.68 - - [05/Jun/2007:11:28:34 -0500] "GET /images/servicios-al-estudiante.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:34 -0500] "GET /images/acceso-usuario.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:34 -0500] "GET /images/acceso-clave.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:34 -0500] "GET /images/ico-fondo.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:34 -0500] "GET /images/ico-izq.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:34 -0500] "GET /images/ico-misiones.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:34 -0500] "GET /images/ico-ex-alumnos.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:34 -0500] "GET /images/ico-mail.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:34 -0500] "GET /images/ico-edu.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:34 -0500] "GET /images/direc.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:34 -0500] "GET /images/ico_servicios.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:34 -0500] "GET /images/autoevaluacion.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:34 -0500] "GET /images/calendario.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:34 -0500] "GET /images/ico-biblioteca.gif HTTP/1.1" 304 -

172.16.5.68 - - [05/Jun/2007:11:28:34 -0500] "GET /images/ico-der.gif HTTP/1.1" 304 -

ANEXO D

TABLAS DE CÓDIGOS DE PETICIÓN HTTP

Tabla 1: Lista de código de Petición HTTP

LISTA DE CÓDIGOS DE PETICIÓN HTTP		
CÓDIGO	**SIGNIFICADO**	**DESCRIPCIÓN**
200	Ok	El pedido fue exitoso.
201	Creado	
202	Aceptado	
204	Sin contenido	Es una página intencionalmente en blanco.
301	Movido permanentemente	El recurso ha sido movido definitivamente hacia otra ubicación.
302	Movido temporalmente	Similar al 301, pero es una situación temporal.
303	Ver otro	Redirección automática a una dirección como resultado de un POST.
304	No modificado	Indica q la copia encontrada en la caché del cliente (o proxy) se encuentra actualizada.
400	Petición errónea	El pedido no pudo ser comprendido por el servidor.
401	Sin autorización	El objeto solicitado no puede ser recuperado hasta que no se realice la autorización correspondiente.
403	Perdido	El servidor ha negado el acceso a un determinado recurso.
404	No encontrado	El documento solicitado no existe en el servidor.
408	Tiempo de espera excedido	El cliente no envió una respuesta dentro del tiempo en el que el servidor estuvo esperando.
500	Error interno del servidor	Un error genérico que indica un fallo inesperado en el servidor.
501	No implementado	Se ha realizado un pedido no válido y no fue posible procesarlo.
503	Servicio no disponible	El servicio se halla temporalmente fuera de servicio.
508	Versión de HTTP no soportada	La versión de http solicitada por el cliente no está soportada por el servidor.

Tabla 2: Código de Petición HTTP

CÓDIGOS DE PETICIÓN HTTP	
CÓDIGO	**SIGNIFICADO**
1xx	De información
2xx	Éxito
3xx	Redirección: se requiere una nueva acción para completar la petición.
4xx	Error en el cliente.
5xx	Error de servidor

ANEXO E

LABORATORIO DE PROCESAMIENTO DE DATOS

HERRAMIENTA DEEP LOG ANALYZER

Para analizar el archivo “utpl-access_log” del sitio web UTPL, es necesario configurarla como un proyecto Deep Log Analyzer. Para crear este proyecto, se utilizo el Asistente de Configuración del proyecto.

Para esto se debe realizar los siguientes pasos:

- **Paso 1**

Hacer clic en Crear Nuevo Proyecto en el panel de tareas o en el menú Archivo, luego se elije la opción Nuevo Proyecto. El Proyecto Asistente de configuración se abrirá.

- **Paso 2**

Se asigna un nombre al proyecto, para esta práctica al proyecto se lo designo con el nombre de “UTPL_LOG” el mismo que servirá para almacenar todos los datos procesados con sus respectivos informes, a continuación se hace clic en Siguiente.

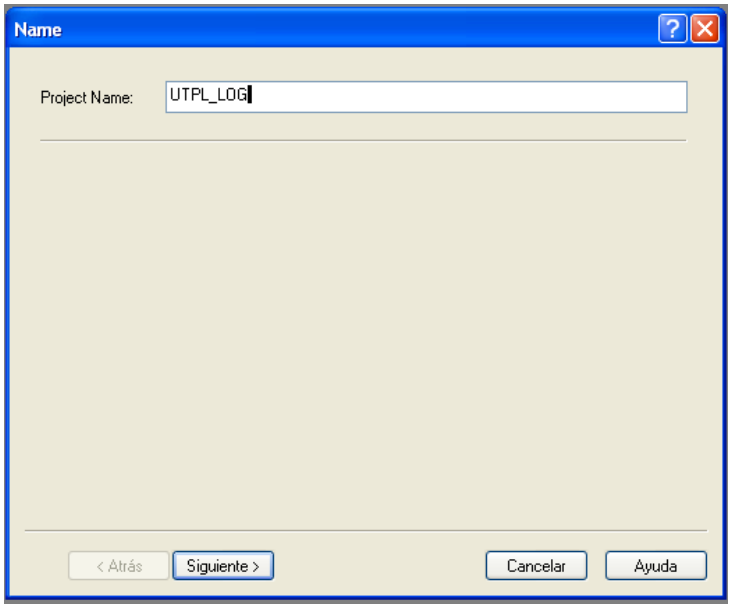

Fig. 14: Pantalla para asignar nombre de Proyecto

- **Paso 3**

Se define la ubicación del archivo, para esto se debe hacer click en “add log files” para agregar el archivo que se desea procesar, a continuación se presentan las siguientes pantallas en donde se detalla cada paso para seleccionar el archivo:

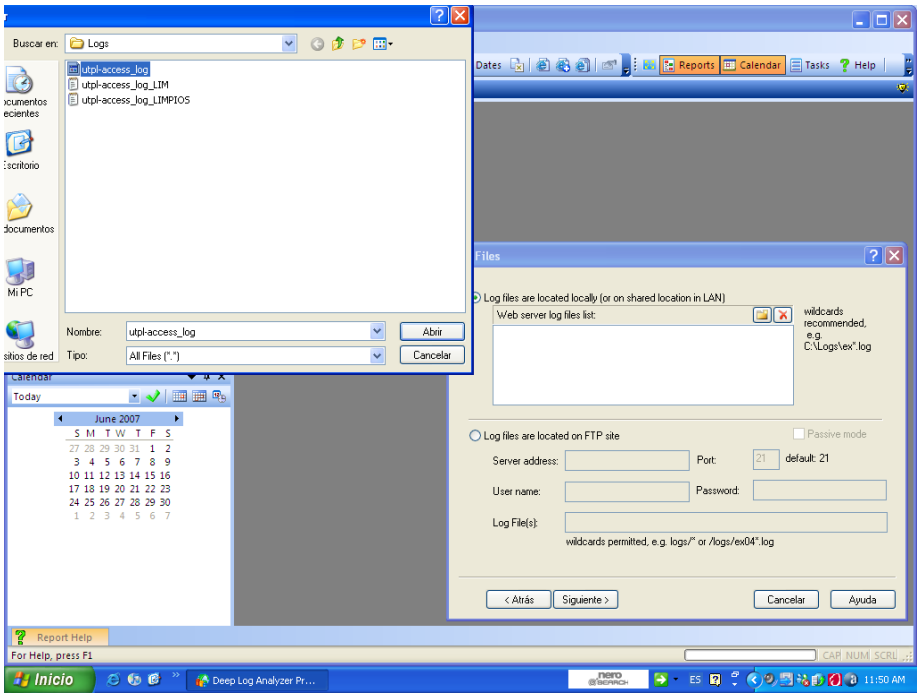

Fig. 15: Pantalla ubicar el archivo log

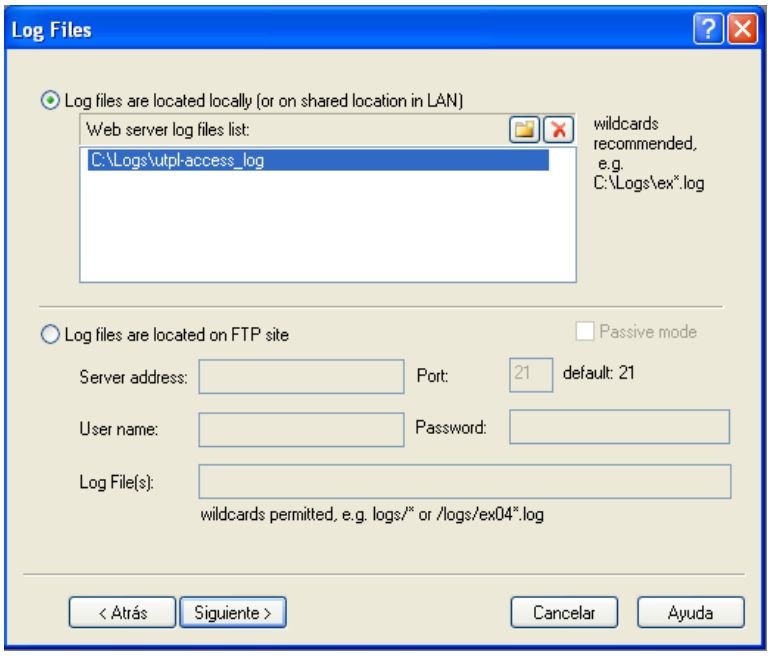

Fig. 16: Pantalla para agregar archivo log

- **Paso 4**

La pestaña del sitio es la tercera pestaña en la ventana de Configuración del proyecto, la cual contiene los siguientes campos:

- URL del sitio web: URL del sitio Web de raíz (www.utpl.edu.ec)

- Todos los Nombres de Dominio (s): aquí se escribe el nombre del dominio del sitio, incluidos los alias y las direcciones IP. Esta información se utiliza principalmente para filtrar nuestras libres referencias si la próxima opción está activada. Si a pesar de ver las referencias de su sitio web, la dirección IP o nombre de host en los referidos informes, añadir que el nombre y / o dirección IP a esta lista.

- Eliminar Self Referencias: marque la casilla para realizar un seguimiento de las referencias a su sitio web desde otros sitios web y hacer caso omiso de las referencias dentro de su propio sitio. Se recomienda mantener esta opción pues no existe mucha información útil en las libres referencias.

Nota: En la mayoría de los casos se puede dejar los valores por defecto para el resto de los parámetros de esta página.

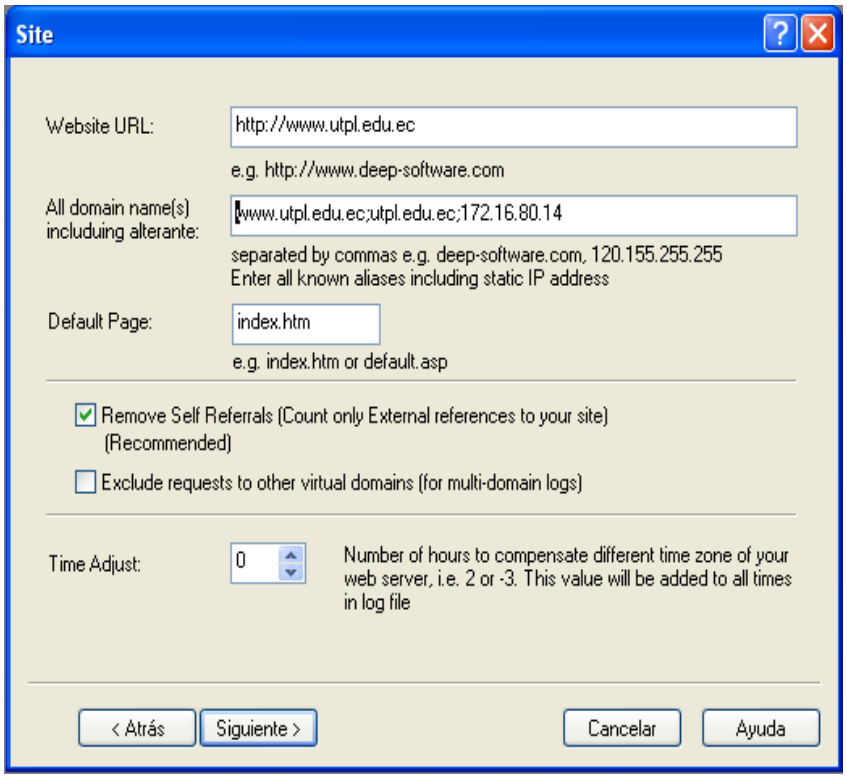

Fig. 17: Pantalla para escribir el del dominio del sitio

- **Paso 5**

Análisis de la ficha es la cuarta pestaña en la ventana de Configuración del proyecto. En este paso se realiza el Sesionamiento para identificar visitantes únicos, para esto se eligió trabajar con la opción Deep Tracker Method y con un tiempo de visita de 30 min, ya que es el tiempo más recomendable en cuanto al sesionamiento. También se escogió el formato DD/MM/YY, para fecha.

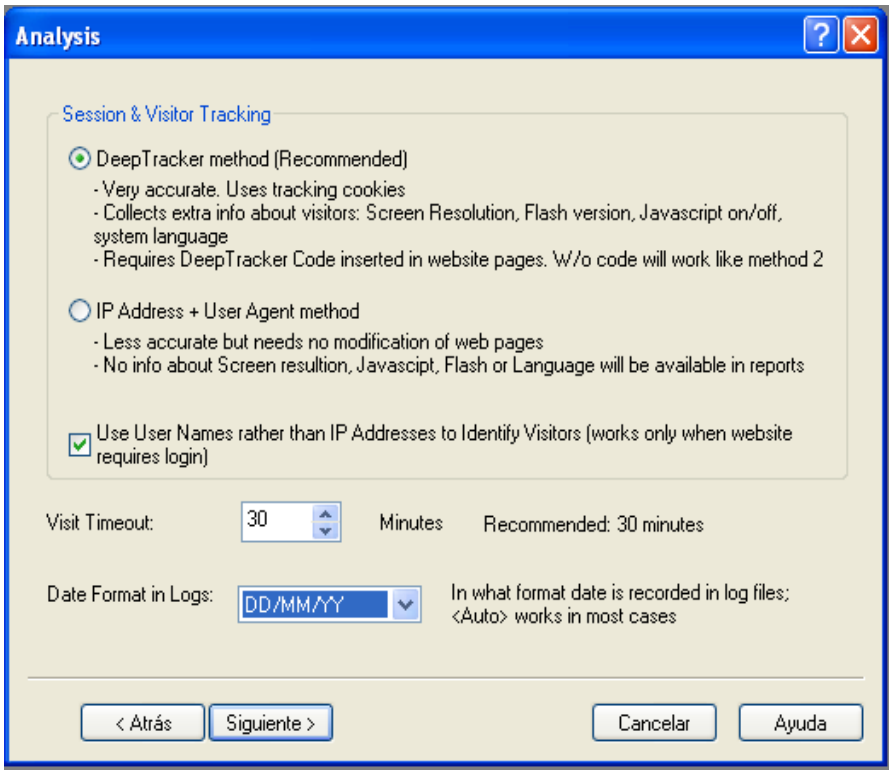

Fig. 18: Pantalla para el sesionamiento de visitantes

- **Paso 6**

Dynamic content, contiene los siguientes valores:

1. Conservar Parámetros URL: marca esta casilla de control si el sitio utiliza CGI, ASP u otros parámetros de URL y que desea realizar un seguimiento de las páginas solicitadas con diferentes parámetros, tales como páginas distintas.

2. Conservar el único parámetro de la lista: marca la casilla de verificación e introduzca los parámetros separados por coma. Se deben poner nombres de parámetros que identifican el contenido de la página web.

3. Eliminar URL Parámetros de Referencia de Páginas Informes: establezca esta opción no mostrar URL de referencia en dichos informes como referencia Páginas Informe sin parámetros de URL. Ajustar esta opción permitirá mejorar el rendimiento DLA.

En este caso utilice la 3 opción, debido a que permitirá mejorar el rendimiento DLA y requiere menos espacio en la base de datos del proyecto.

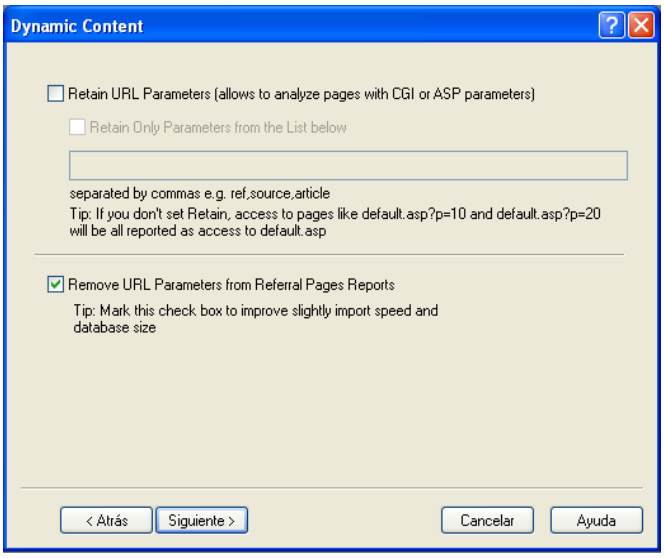

Fig. 19: Pantalla para Dynamic content

- **Paso 7**

En el Exclude se comienza a filtrar los datos, para esto se da 2 opciones:

1. No importar hits en archivos gráficos: esta casilla de verificación sirve para filtrar las visitas a los archivos gráficos, permitiendo mejorar la velocidad de importación de alrededor de 3 veces o más y tener mucho más pequeña la base de datos dependiendo de la cantidad de gráficos en su sitio.

2. No importar error hits: esta casilla de verificación filtra las visitas con la condición de error.

   Para este ensayo se marco la primera opción, ya que nos permite eliminar archivos jpg y gif, los cuales no son objetivo de nuestro proyecto de WEB MINING.

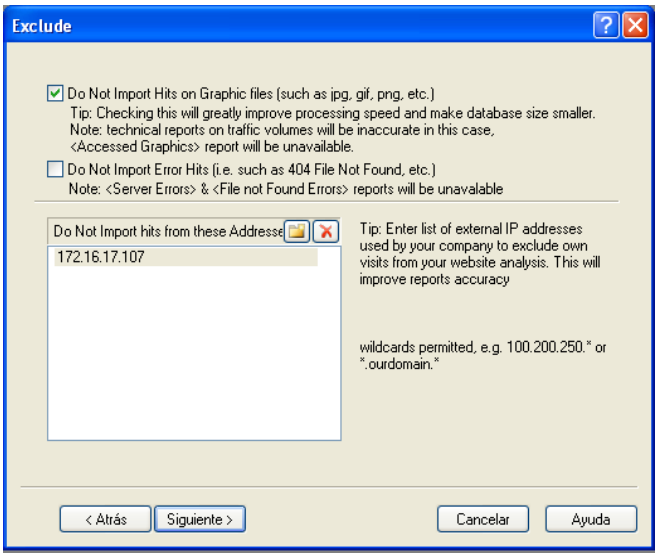

Fig. 20: Pantalla Exclude

- **Paso 8**

Se selecciona como se quiere que se guarden los reportes en la base de datos. Como se puede observar en la Fig. 21, se eligió tener un reporte por día.

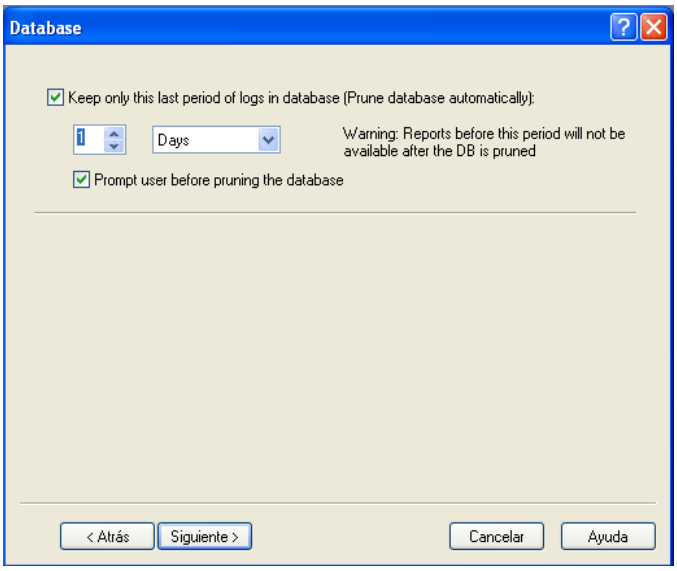

Fig. 21: Pantalla para guardar los reportes en la Base de Datos

- **Paso 9**

Esta pantalla facilita la configuración de los reportes para su exportación a través de su botón Change.

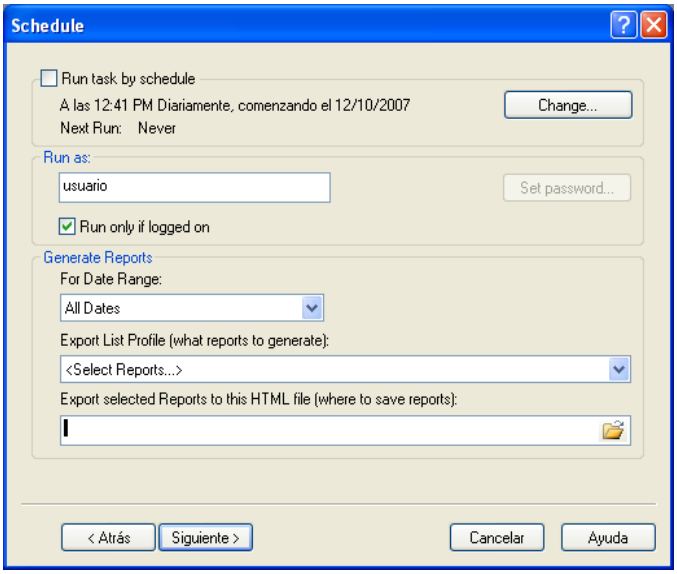

Fig. 22: Pantalla Schedule

- **Paso 10**

La ventana de Deep Tracker Code, como se puede observar en ella muestra 2 opciones, en esta práctica la hemos tomado los valores por defecto. Una vez realizado todo este proceso procedemos a analizar los datos haciendo clic en Analyse Now, como se puede ver en el paso 10 los resultados de este procesamiento.

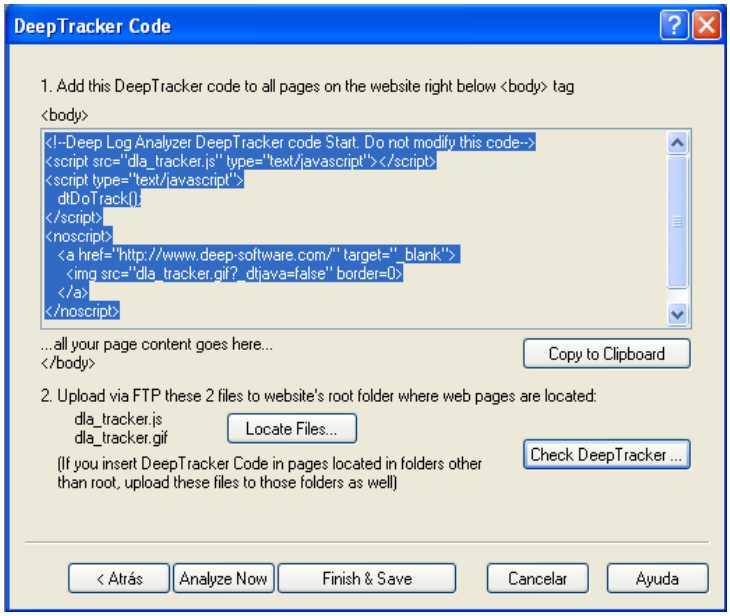

Fig. 23: Pantalla Deep Tracker Code

Se realiza el proceso de análisis del archivo “utpl-access_log”.

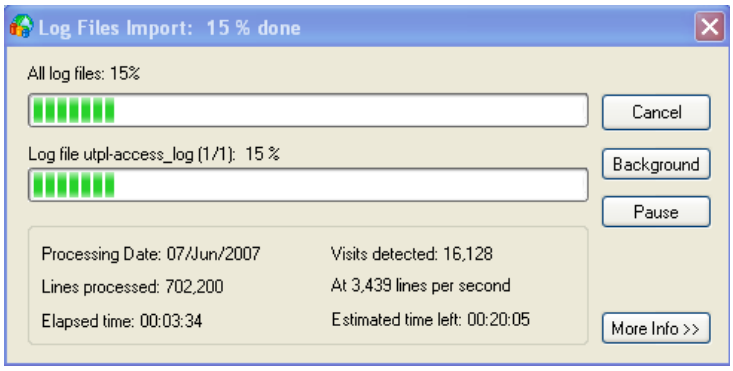

Fig. 24: Pantalla de procesamiento de los log

ANEXO E-1

RESULTADOS DEL LABORATORIO

Informe General de Estadística

Este informe muestra la clave del sitio web, sobre los recursos del sitio de acceso, de visitas y de la actividad de navegación, los sitios web de tráfico, las consultas de búsqueda, errores de servidor web y mucho más. Todas estas estadísticas proporcionan una instantánea e inmediata actividad del sitio, convirtiéndose en el punto de partida de análisis de la página web.

Como se puede observar en la pantalla principal se muestra un reporte general de los datos procesados del archivo “utpl-access_log”.

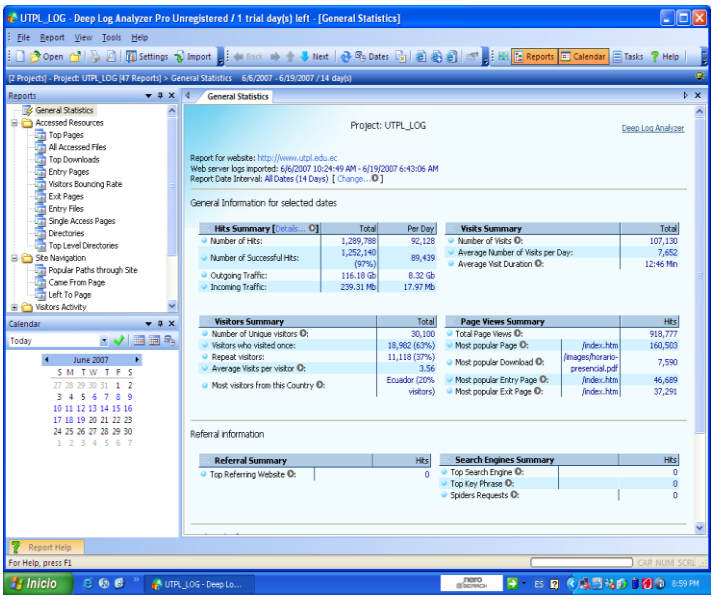

Fig. 25: Reporte General

- **TOP PAGES**

Muestra la popularidad de las páginas web del sitio. Las páginas se ordenan por el número de veces en que la página fue solicitada por los visitantes. Deep Log Analyzer considera sólo a petición de archivos con extensiones específicas como visitas de páginas. Estas extensiones incluyen html, htm, asp, aspx, php, y otros. Las hojas de estilo CSS, archivos javascript, gráficos (gif, jpg, etc), de sonido (wav, mp3 ,...) y video (avi, mpg ,...), no son considerados como páginas y no figuran en el presente informe .

El Top Pages, esta formado por tres columnas: la primera se refiere al FileName, la segunda Page Views y la tercera columna Datos Transferidos, esta señala el número total de Kb que fueron transferidas por el servidor web a los visitantes de cada página visualizada.

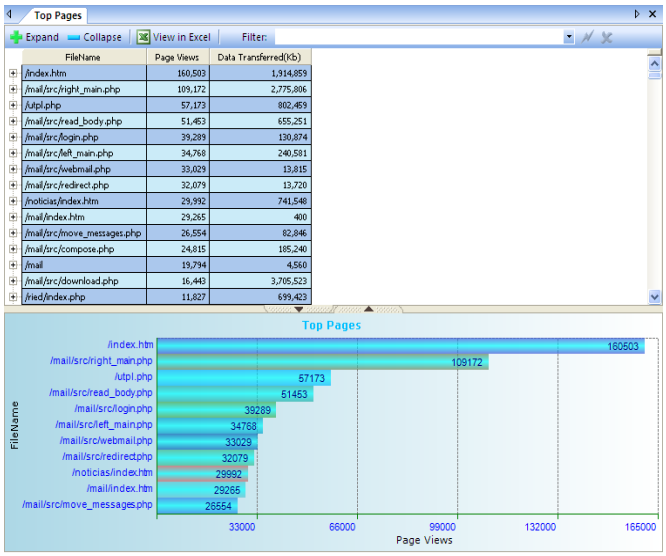

Fig. 26: Reporte Top Pages

- **ALL ACCESSED FILE**

Este reporte muestra la popularidad de todos los archivos ubicados en el sitio web, incluyendo las páginas web, los gráficos, los medios de comunicación y otros archivos. Los archivos se clasifican por el número de veces que fueron solicitados por los visitantes.

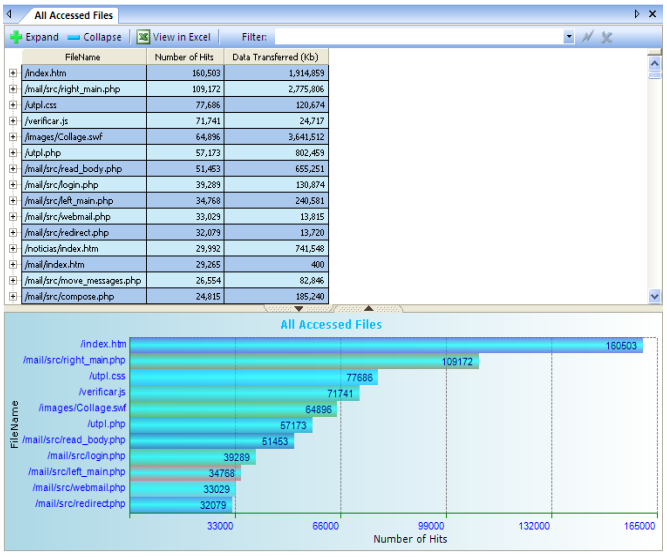

Fig. 27: Reporte All Accessed File

- **TOP DOWNLOADS**

Muestra la popularidad de los archivos descargados del sitio web. Deep Log Analyzer considera sólo a petición de archivos con extensiones específicas para su descarga, estos incluyen extensiones zip, exe, rar, tar, etc, páginas web (html, asp, php, etc), gráficos (gif, jpg, etc), de sonido (wav, mp3 ,...) y video (avi , Mpg ,...) no son considerados como descargas y no figuran en el presente informe.

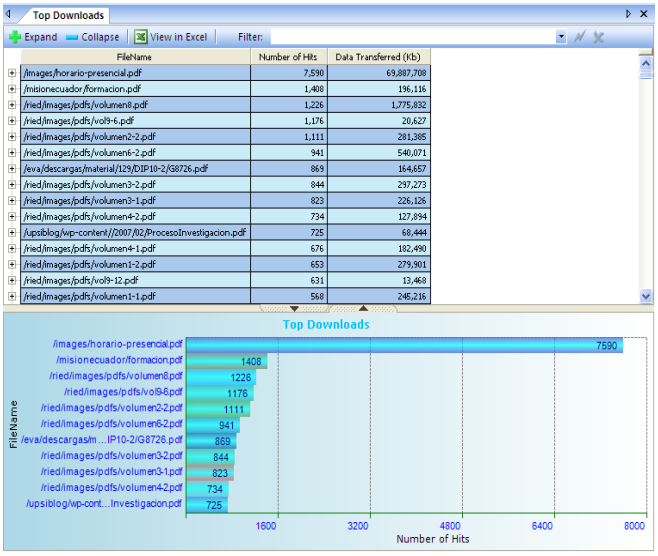

Fig. 28: Reporte Top Downloads

- **ENTRY PAGES**

El informe de Entrada a Páginas indica la popularidad de las páginas del sitio web que los visitantes utilizan para entrar en la página. La entrada es la primera página web de la página solicitada por el visitante durante su visita. La columna Número de Votos muestra el número de veces que los usuarios han accedido a la web a través de la página en particular. Deep Log Analyzer considera sólo los archivos con extensiones html, htm, asp, aspx, php, y otros, pero menos las hojas de estilo CSS, archivos javascript, gráficos y video.

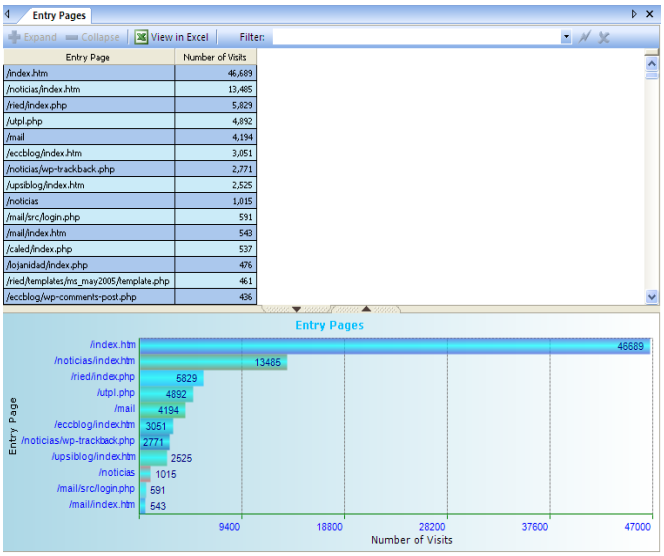

Fig. 29: Reporte Entry Pages

- **VISITORS BOUNCING RATE**

Este informe es importante ya que a partir de él se puede diseñar estrategias para reducir la tasa de rebote de las principales páginas web.

Entre los beneficios que presenta este informe tenemos:

- Ayuda a determinar la eficiencia de diseño, maquetación y redacción de las diferentes páginas importantes en el sitio web.
- Consulta si sus visitantes están interesados en el contenido del sitio web

Como se puede observar, la fig.5 muestra cuatro columnas:

1. Entrada página: nombre de la página utilizada por los visitantes para entrar en el sitio.
2. Las entradas: número de veces que los usuarios han accedido a la web a través de la página.
3. Devueltos: número de veces que los usuarios han abandonado el sitio web sin ver más páginas, es decir, no solicitan más páginas del servidor web.
4. Devueltos%: es el porcentaje de visitas devueltas.

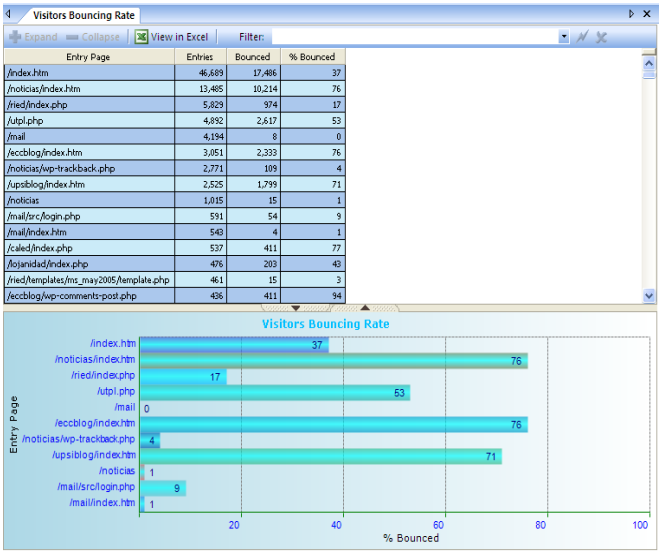

Fig. 30: Reporte Visitors Bouncing Rate

- **EXIT PAGES**

El reporte Exit Pages refleja la popularidad de las páginas web en la que los visitantes abandonan la página web. La columna Salir de la Página web es la última página solicitada por el visitante durante su visita. La salida del sitio web del usuario se da cuando cierra la ventana del navegador, o directamente entra en el URL en el campo de la dirección y va a otro sitio. La columna Número de Votos muestra el número de veces que los usuarios han abandonado el sitio web a través de la página en particular. Deep Log Analyzer considera como salidas sólo los archivos con extensiones (html, htm, asp, aspx, php, y otros) específicas como las páginas.

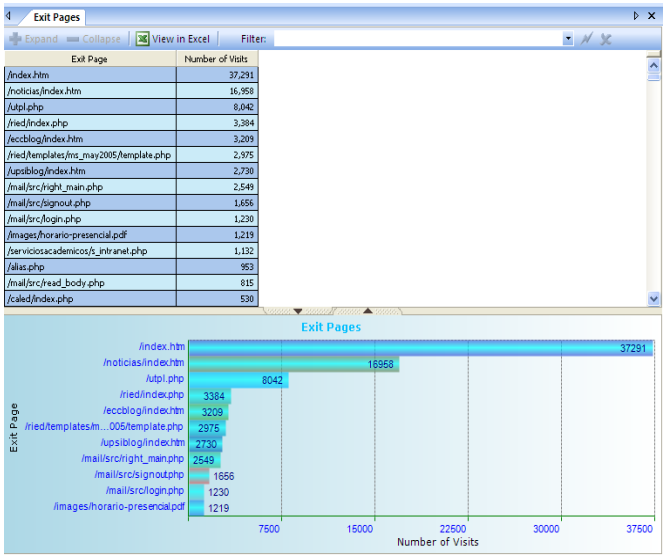

Fig. 31: Reporte Exit Pages

- **ENTRY FILE**

Este informe muestra la popularidad de los archivos de los visitantes utilizan para entrar a la página web, incluye todos los archivos como los gráficos y las descargas y no sólo páginas web.

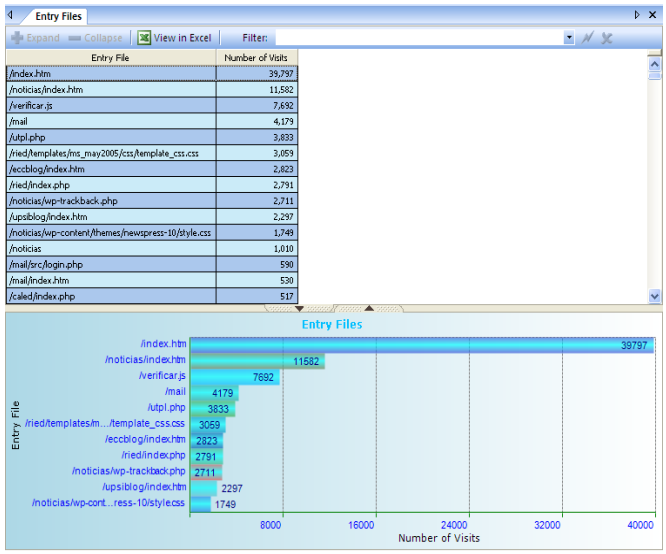

Fig. 32: Reporte Entry File

- **SINGLE ACCESS PAGES**

Clasifica el número de veces que esta página fue el único pedido durante una visita. Este informe incluye el éxito de visitas y no tiene en cuenta las solicitudes de páginas de error. Además este informe sólo toma en cuenta el número de visitas y no en el número de visitantes únicos.

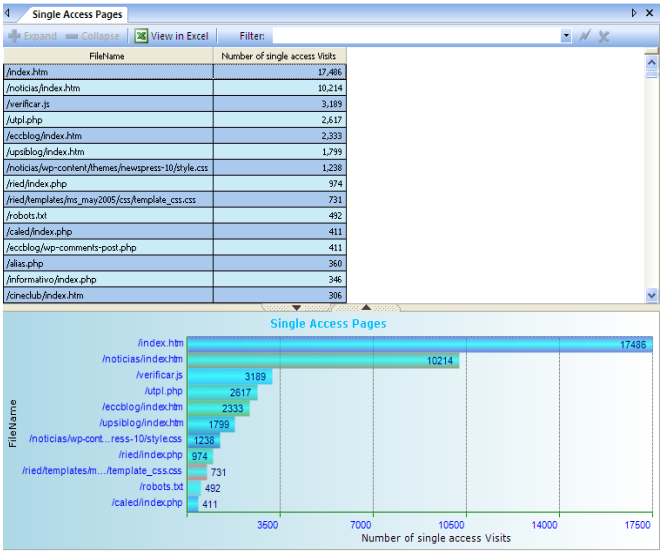

Fig. 33: Reporte Single Access Pages

- **DIRECTORIES**

Este informe muestra la popularidad del sitio Web de nivel superior de los directorios (carpetas). Este informe se clasifica por el número de veces (Hits) que pidió a los visitantes de páginas web o de cualquier tipo que se encuentren en ese nivel superior de su directorio y todos los subdirectorios.

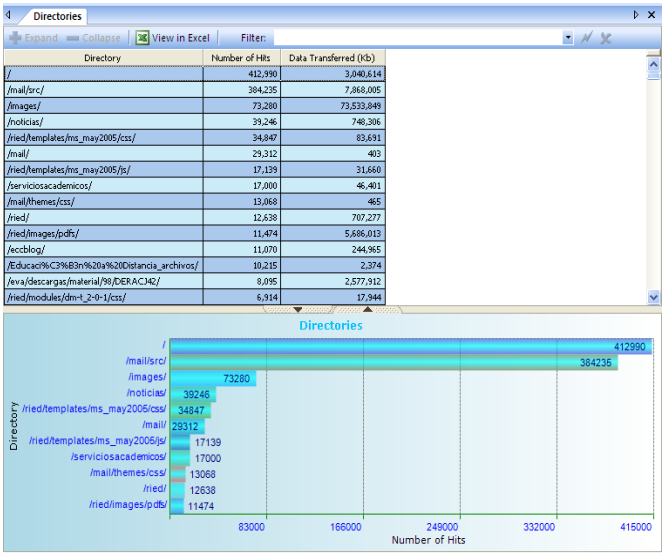

Fig. 34: Reporte Directories

SITE NAVIGATION

- **POPULAR PATHS THROUGH SITE**

El siguiente informe muestra la popularidad de caminos por los que el visitante recorrió dentro del sitio web a partir de la página de acceso y de su salida a sus páginas. Cada fila de las listas indica el orden en que fueron vistas una o más páginas, separados por guión símbolos.

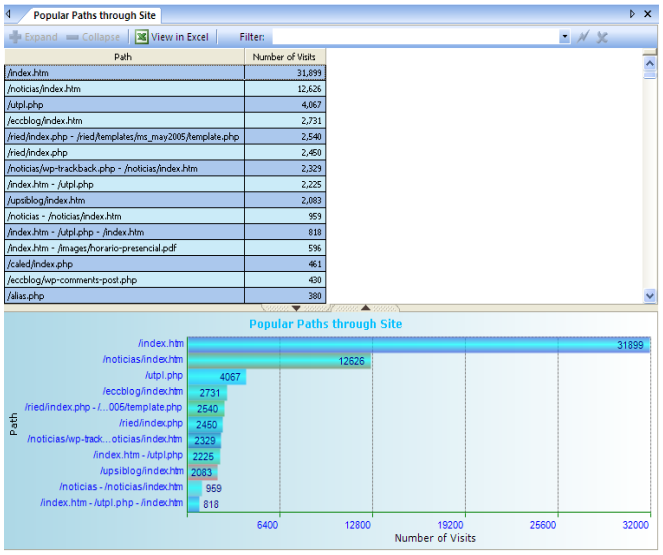

Fig. 35: Reporte Popular Paths Through Site

- **CAME FROM PAGE**

El Came from Page muestra la navegación de los visitantes para llegar a una página en particular del sitio web.

Beneficio

- Analizar el comportamiento de los usuarios en la web.

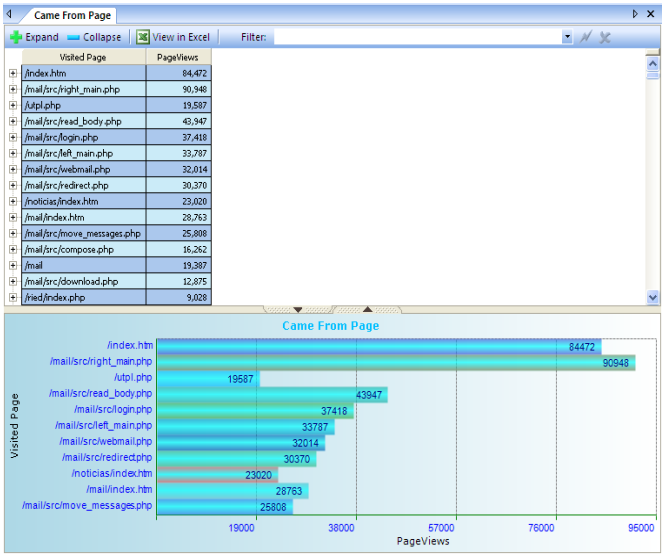

Fig. 36: Reporte Came From Page

VISITORS ACTIVITY

- **TOP VISITORS**

Este informe muestra los todos usuarios que han accedido al sitio web, los cuales son clasificados por el número de visitas. Como se pude observar en la fig. 14, este reporte esta formado por tres columnas: la primera columna corresponde a Visitantes (muestra la dirección IP del usuario o nombre de host, lo que se presenta en el servidor web del archivo de registro), la segunda columna se refiere al País del usuario y la tercera columna corresponde al Número de Visitas.

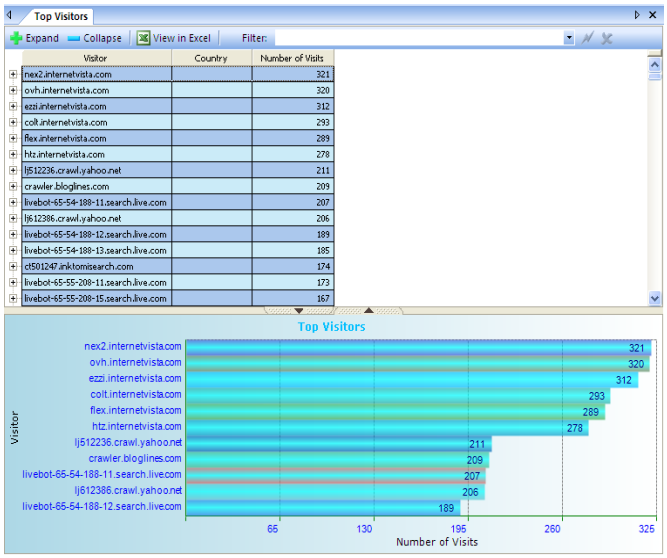

Fig. 37: Reporte Top Visitors

- **VISITS HISTORY**

Este informe muestra el número de visitas por día.

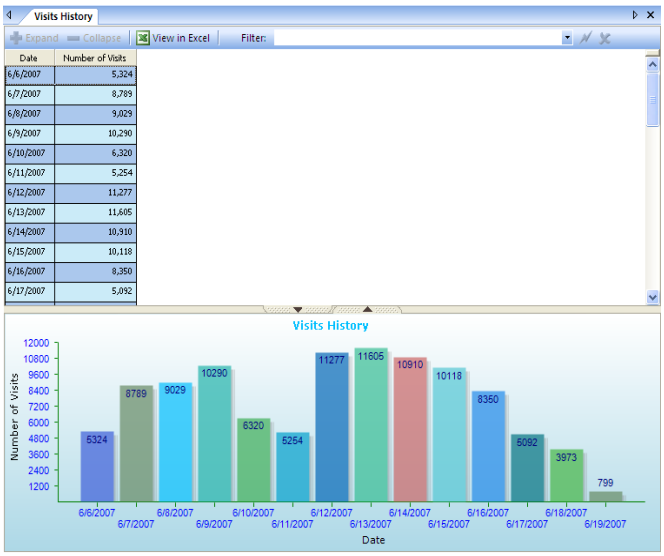

Fig. 38: Reporte Visits History

- **HITS HISTORY**

El Hits History muestra el número de visitas por día. En la fig. 40 se puede observar la columna de Datos Transferido, la cual indica la cantidad total de datos en Kb transferidos por el servidor web a los visitantes del sitio durante cada día.

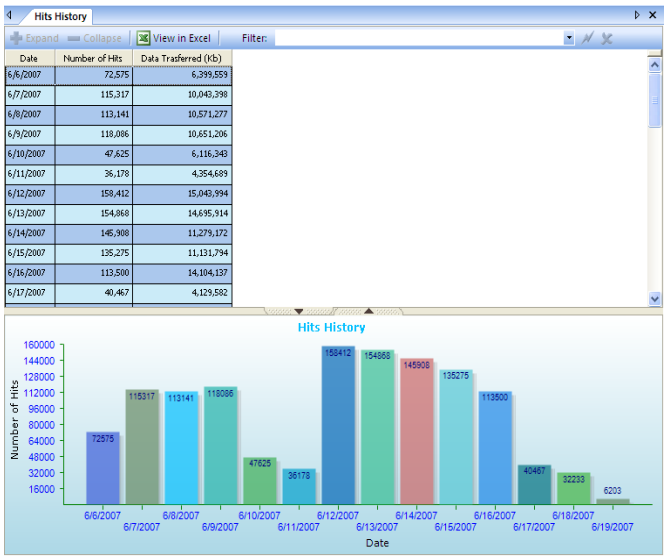

Fig. 39: Reporte Hits History

- **VISITORS STAY LENGTH**

Este informe muestra cuánto tiempo permanecen los visitantes en el sitio web. La columna Duración de Visita muestra el promedio de duración de la visita y la columna Número de Visitas manifiesta el número total de visitas.

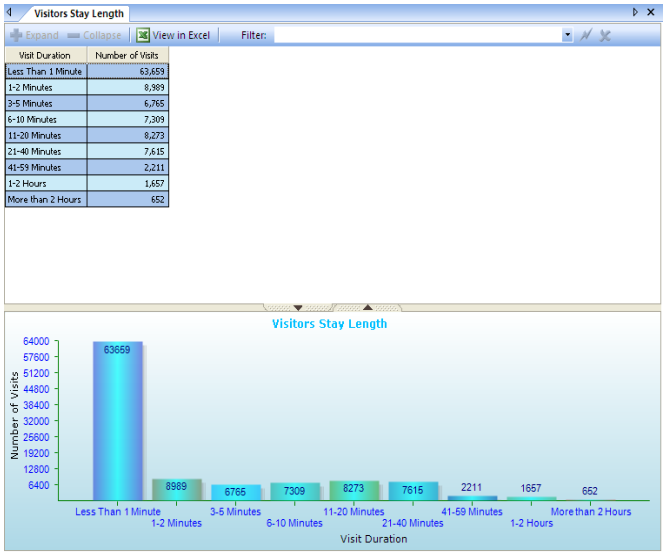

Fig. 40: Reporte Visitors Stay Length

- **POPULAR DAY OF WEEK**

Muestra el tráfico del sitio web en función de los cambios del día de la semana.

Beneficio
- investiga en qué día el sitio web recibe más tráfico.

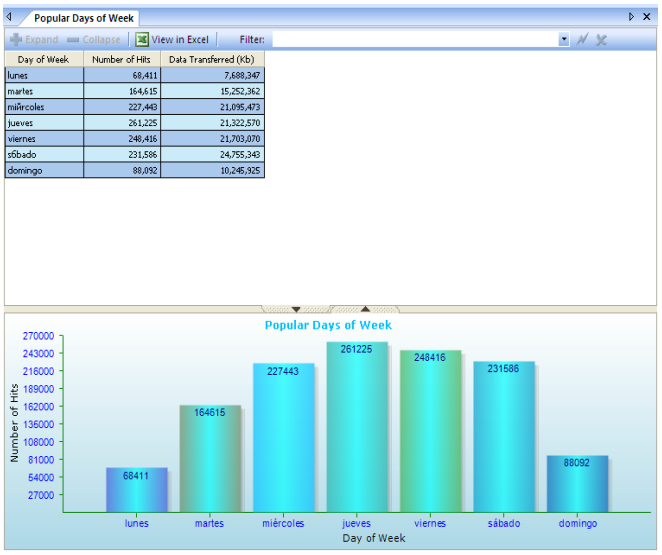

Fig. 41: Reporte Popular Day of Week

- **VISITS BY DAY OF WEEK**

Este reporte indica el total de visitas del sitio web recibidas cada día de la semana.

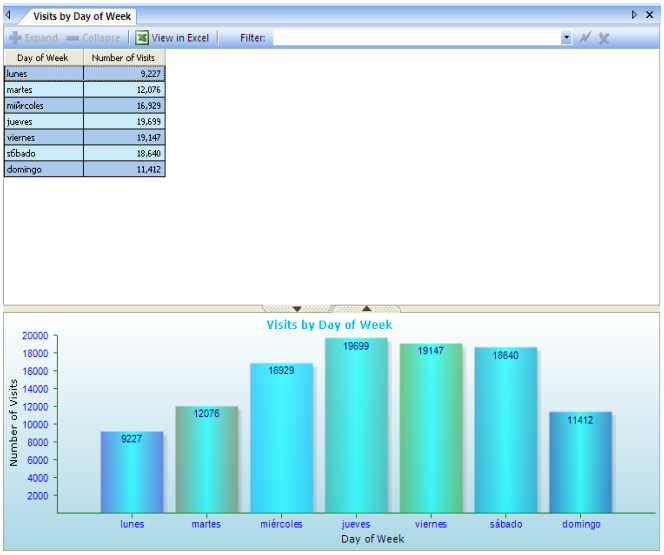

Fig. 42: Reporte Visits by Day of Week

- **POPULAR HOURS OF DAY**

El Popular Hours of Day muestra el tráfico del sitio web puede cambiar dependiendo de la hora del día.

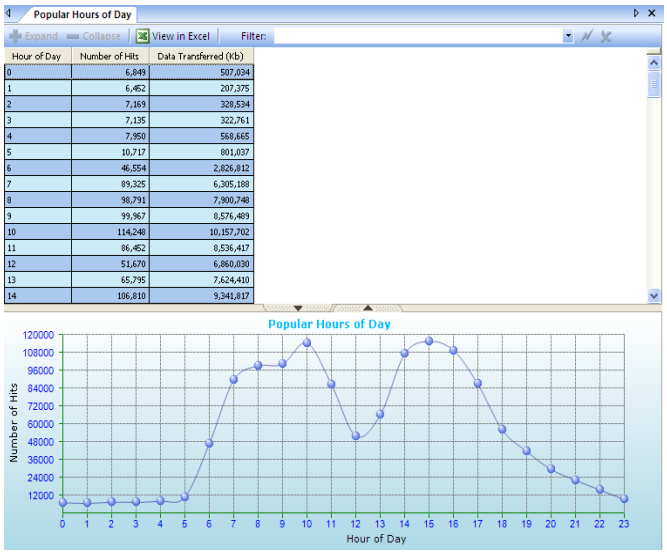

Fig. 43: Reporte Popular Hours of Day

- **VISITS BY HOURS OF DAY**

Este informe refleja el número de visitas del sitio web puede cambiar dependiendo de la hora del día.

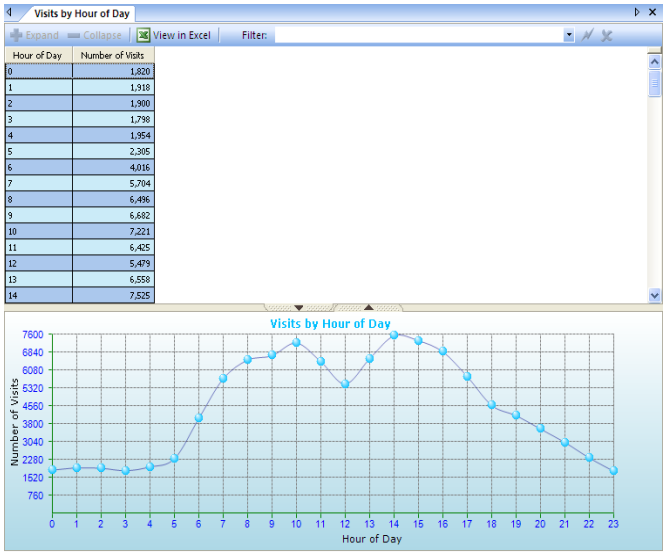

Fig. 44: Reporte Visits Hours of Day

- **AUTHENTICATED USER**

El Authenticated User muestra el nombre de usuario que utiliza durante la autenticación del usuario como lo exige el sitio, clasificadas según el número de visitas. Esta información sólo muestra los informes de seguridad de los sitios que requieren autenticación de usuario.

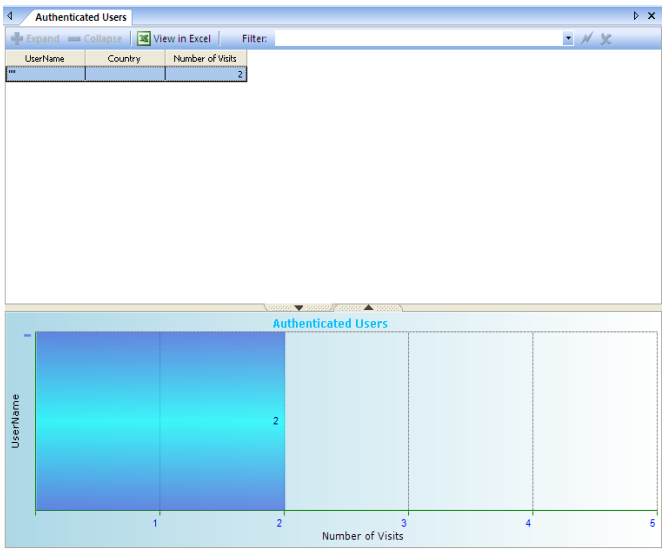

Fig. 45: Reporte Autehenticated User

- **TOP COUNTRIES**

El Top Countries muestra los países de los visitantes que accedieron al sitio web, estos se encuentran clasificados por número de visitantes únicos de ese país.

Beneficio

- Averigua si está atrayendo al sitio web desde la perspectivas adecuadas ubicaciones geográficas.

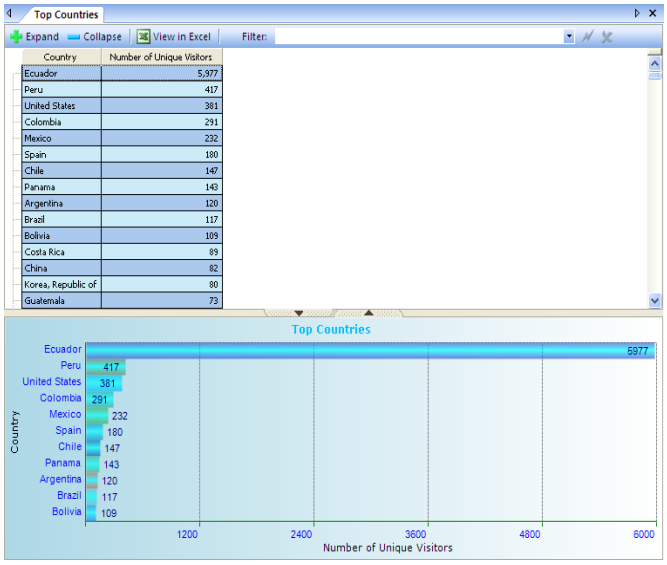

Fig. 46: Reporte Top Countries

ANEXO F

CÓDIGO DE LA IMPLEMENTACIÓN EN MATLAB

```
clear all
home
clc;

warning off

c = 1;
prew = 0;
clusterRBF = 1;
scluster = 'Visitas de Páginas:   RBF  ';


% AGENTE DE INTERFAZ
%
================================================================================
while c < 4
    c = menu('ELIGA LA OPCIÓN', 'Importar Datos',
scluster,'Visualización de Datos', 'Salir');

    switch c

        % Seleccionar el archivo de datos a procesar
        case 1
            %Selección del archivo
            pathname='C:\'                    %Pathame, ruta para
                                               identificar el archivo
            [filename,pathname] = uigetfile(...  %
            [filename,pathname]= [Nombre de archivo, ruta]
            {'*.xls';'*.mdl';'*.mat';'*.*'},...  % uigetfile (), los
             'Open');                                diferentes

            together = [pathname '' filename]


            [N, T, rawdata] = xlsread(together)
            A=[N];
            X=mean(A);

            together
            a = length(A)                      % a ? longitud del
                                               archivo


            m=1;
            E=[];

            %Recorrido de la matriz T, para almacenarla en la matriz E
             para luego sacar la inversa de esta y almacenarla en la
             variable L, para su posterior almacenamiento en el
```

```
 archivo
%csv
for k=2:a+1
    E= [E T(k,1)]
    L=E'
end

%Procedimiento para comparar conocer la relevancia
for i=1:a                                %para cada fila
    M=A(i,1)

    for k=1:m-1
    end

    B(1,m)=A(i,1);                       %Matriz B,almacena el
                                          número de visitantes
    for j=i:a                            %Condición para
                          obtener la relevancia de las páginas
        if X >=  M
            'M es de mayor relevancia'
             s = 0
        else
            'M es de menor relevancia'
             s = 1
        end
    end
    B1(1,m)=s                             %Matriz B1,almacena
                                  la prioridad de las páginas
    m=m+1
end


% Almacenamientos archivos csv
diary Direcciones.csv
    L
diary off

diary Relevancia.csv
    B1.'
diary off


% Procesamiento del archivo utilizando la RNA
case 2
    if clusterRBF;
        clusterRBF = 0;
        scluster = 'Visitas de Páginas:   RBF';
    else
        clusterRBF = 1;
        scluster = 'Visitas de Páginas:   RBF  ';
    end

% Visualizar los datos de la RN
case 3
```

```
            web http://localhost/webmining/Pagina_Principal.php


        % Visualizador en modo gráfico para el entrenamiento de la RNA
        case 4
        if prew
              'show';
        end
       return;
    end


 % DISEÑO Y ENTRAMIENTO DE LA RNA
 %
=========================================================================
 if c == 2
     if clusterRBF == 1
         P = B              % P ? patrón de entrada
         Q = B1             % T ? objetivo

         df=10;             % Frecuencia de display
         eg=0.1;            % Umbral de error cuádratico medio
         sc= 1;             % Constante Spread de la redes RBF (defecto
1.0)
         mn=100;            % Maximo número de neuronas

         net=newrb(P,Q,eg,sc,mn); % Creación de la red neuronal

         IW= net.iw{1};
         C1=net.b{1}         % Bias
         C2=net.b{2};
         W=net.lw{2,1};
         Y = sim(net,P);     % Y ? salidad de la red neuronal RBF
         Y=Y'
     end
 end
end
```