UNIVERSIDAD DON BOSCO

FACULTAD DE INGENIERÍA

TRABAJO DE GRADUACIÓN PARA OPTAR AL GRADO DE

INGENIERO EN CIENCIAS DE LA COMPUTACIÓN

ELABORACION DE GUIA TEÓRICA Y PRÁCTICA SOBRE LOS ROBOTS SPIDER

PRESENTADO POR

BLANCO FLORES, VANESSA ESMERALDA

CUBIAS CORTEZ, VICTOR FERNANDO

SORTO ALVARENGA, JOSÉ WILLIAM

ASESOR

ING. RENÉ ANGULO

MARZO 2008

EL SALVADOR, CENTROAMERICA.

UNIVERSIDAD DON BOSCO

FACULTAD DE INGENIERÍA

RECTOR

ING. FEDERICO MIGUEL HUGUET

SECRETARIO GENERAL

LIC. MARIO RAFAEL OLMOS

DECANO FACULTAD DE INGENIERIA

ING. ERNESTO GODOFREDO GIRON

MARZO 2008

EL SALVADOR, CENTROAMERICA.

UNIVERSIDAD DON BOSCO

FACULTAD DE INGENIERÍA

TRABAJO DE GRADUACIÓN PARA OPTAR AL GRADO DE INGENIERO EN CIENCIAS DE LA COMPUTACIÓN

ELABORACION DE GUIA TEÓRICA Y PRÁCTICA SOBRE LOS ROBOTS SPIDER

_____	_____	_____
JURADO	**JURADO**	**JURADO**

_____	_____
ASESOR	**TUTOR**

MARZO 2008

EL SALVADOR, CENTROAMERICA.

PREFACIO

En la actualidad Internet es considerado como la principal fuente de recursos de información, convirtiéndose de hecho en el cuarto medio de comunicación social utilizado a nivel mundial.

En Internet existen básicamente cinco formas de acceder a la información: entrando directamente a un grupo de noticias, suscribiéndose a un boletín de noticias y recibir la información solicitada en su lista de correo, ir directamente a una página Web de la que se conoce de antemano su dirección URL, navegando a través de los enlaces presentes contenidos en una página Web o usando los motores de búsqueda.

Desde el punto de vista del usuario "navegar" en el World Wide Web en busca de información de interés resulto en sus comienzos, un poco dificultoso, lento y muchas veces improductivo.

Con el objetivo de solucionar estos problemas, surgieron distintas soluciones: las anotaciones de URL´S favoritas (bookmarks), buscadores por palabras clave como: Google, Yahoo, AltaVista entre otros que ayudan a los usuarios a buscar información de acuerdo a un criterio especifico. Los buscadores por palabras clave utilizan un programa llamado Robot Spider para llevar a cabo las búsquedas.

Los Robots Spider son agentes inteligentes pues tienen incorporadas técnicas de aprendizaje. A diferencia de los buscadores tradicionales, los robots no indexan el Web, ellos "caminan" sobre el Web, analizando y almacenando información relevante.

Las personas que hacen uso de la mayoría de buscadores comerciales como Google, no saben realmente como estos funcionan, ni mucho menos como están compuestos.

En nuestro país la mayoría de personas se han vuelto consumistas del Internet, todo el mundo utiliza los buscadores y por ende los robots spiders; pero nadie produce esta clase de software para el cual no se necesitan muchos recursos, más que una computadora con conexión a Internet.

Las pocas tesis acerca del tema que se han realizado en el país no detallan con exactitud como implementaron el robot spider en la realización de su buscador y mucho menos se explica como funcionan estos robots, ya que no era el objetivo de estos trabajos de graduación, además que en estos documentos no se utilizaron las nuevas herramientas tecnológicas como por ejemplo MySQL para el almacenamiento de la información obtenida y catalogada por el robot que hacen mas fácil la puesta en marcha de un buscador.

Por tales razones se hace necesario el desarrollo de un documento que detalle el funcionamiento y las partes que componen un robot spider, así como un ejemplo de uno de sus principales usos, en este caso implementarlo dentro de un buscador.

Se espera que las personas que hagan uso de esta guía sean capaces de elaborar su propio buscador, despierten su interés y que desarrollen futuros proyectos.

Basándose en lo expuesto anteriormente el documento esta dividido en seis capítulos que se resumen de la siguiente manera: el capítulo uno es una **Introducción a la robótica** en donde se explica el concepto de "robot" aplicado a programas computacionales que utilizan el protocolo HTTP para explorar grandes porciones del Web.

El capítulo dos lleva como nombre **Conceptos básicos de redes y programación**, contiene información sobre redes y protocolos de Internet para comprender la comunicación entre computadoras en el Web, se abordan temas como modelo cliente/servidor, lenguajes de programación para Web y fundamentos de Internet.

El capítulo tres **Sistemas de búsqueda en Internet (robot Spider)** contiene la información sobre los robots Spider y su papel en los motores de búsqueda: partes o componentes y los diferentes tipos de buscadores, indexación, estandarización de documentos, estructura de la base de datos.

En el capítulo cuatro **Algoritmos y conceptos utilizados en la elaboración de robots Spider** se estudian los algoritmos de ranking y sus técnicas WebQuery, HITS, PageRank y algoritmo de crawling y el recorrido que puede ser Breadth-First (cobertura amplia pero no profunda) Depth-First (cobertura vertical). Y se explica además "el protocolo de exclusión de robots" y escrutinio de sitios.

Como parte practica de la guía se tiene el capítulo cinco llamado **Combine System**, en él se estudian los distintos comandos de configuración de este robot como: CombineINIT, CombineCtrl, CombineUtil, Combine Export y sus funciones y configuración básica: Crawling o búsqueda, administración de base de datos y archivos de configuración.

Y por ultimo el capítulo 6 que lleva por nombre **Implementación del robot Combine System en un buscador Web**, es una guía practica para que el lector de la misma sea capaz de elaborar su propio buscador Web, teniendo como herramienta una computadora conectada a Internet, el software de ingeniería en una caja y conocimientos básicos en programación.

AGRADECIMIENTOS

- En primer lugar a Dios por permitirme por medio de sus bendiciones, su fidelidad y amor poder terminar mis estudios y de la sabiduría y tolerancia que puso en mí cada día.

- A mis padres y abuelita, por su confianza puesta en mí y su apoyo incondicional a lo largo de mi carrera.

- A mis compañeros de tesis y todas las personas que de manera directa o indirectamente han sido un apoyo y han colaborado para la finalización de este trabajo.

Vanessa Flores

AGRADECIMIENTOS

- Primeramente a Dios todopoderoso que me permitió estar con vida para poder culminar mi carrera, gracias por darme las fuerzas para seguir adelante a pesar de las adversidades que se presentaron.

- Le agradezco a mis padres por darme la vida, especialmente a mi padre Napoleón Cubías que siempre estuvo conmigo apoyándome a lo largo de mi carrera, gracias por toda tu ayuda este logro también es tuyo y no hubiese sido posible sin tu gran apoyo.

- Agradezco también a mis segundos padres mis abuelos quienes me han apoyado a lo largo de mi vida, gracias por estar conmigo cuando lo necesitaba, deseo dedicarle este logro también a mi abuela Julia García que estoy seguro me seguirá apoyando desde el cielo gracias por todo.

- Y agradezco también a todas las personas que de una u otra forma me ayudaron para que esto se llevase a cabo familia, amigos y a mis compañeros de tesis con los cuales compartí los problemas para la realización de este trabajo.

Víctor Cubías

AGRADECIMIENTOS

- A Dios y a la Virgen María:

Por su continua presencia en mi vida, por las dádivas incomparables, como lo son, amigos y compañeros de estudio que conocí a lo largo de estos años, también por otorgarme salud, serenidad y concentración.

- A mis padres:

Por haberme brindado su apoyo incondicional y haber confiado en mí, especialmente a mi madre, por su amor, ejemplo y sabiduría, de conocer el tiempo correcto de instruir, aconsejar y corregir; en especial por sus oraciones.

- A mis hermanos:

Por su compañía y nobleza.

- A mis compañeros de tesis:

Por brindarme su amistad, con quienes he compartido momentos de trabajo y alegría, y que convierten este logro en algo especial.

- Y a todas aquellas personas que de manera directa o indirectamente me ayudaron a lo largo de este camino con consejos, muestras de cariño y/o constante apoyo.

William Sorto

ÍNDICE.

CAPITULO I. DEFINICIÓN EL ANPROYECTO.

CAPITULO II. ELABORACIÓN DE GUÍA TEÓRICA Y PRÁCTICA SOBRE LOS ROBOTS SPIDER.

ÍNDICE DE TABLAS Y FIGURAS.

Págs.

CAPITULO I. DEFINICIÓN DEL ANTEPROYECTO.

1.1 ANTECEDENTES[1].

1.1.1 Breve historia de los Robots Spiders.

Los robots Web o robots Spider son potentes programas que recorren la Web de forma automática y buscan diferentes tipos de datos como texto, imágenes o sonido.

Estos surgen a partir del año de 1993 con el desarrollo del World Wide Web Wanderer, robot de búsqueda creado en Perl, que pretendía medir el tamaño de la red.

Luego se creo Wandex, robot que podía leer direcciones URL, se considera que fue el primer buscador de Internet. A partir de ahí se da el inicio del desarrollo de la primeros robots Spider.

El 20 de abril del año de1994 Brian Pinkerton presenta WebCrawler, spider que indexaba las páginas de forma completa y buscaba información de ellas,

Siempre en 1994 apareció Yahoo; con la desventaja que en un principio funciono como un directorio hecho por personas, lo que hacia que, llevara mucho tiempo la búsqueda de información. Para solucionar este problema sus creadores incorporaron un Spider para su directorio.

Luego surge Lycos, creado por Michale Mauldin. Su algoritmo fue muy interesante, ya que incluía el concepto de proximidad entre palabras. Convirtiéndose en el motor de búsqueda mas destacado en el año de 1994.

En diciembre de 1995 surge AltaVista, este tenía un ancho de banda casi ilimitado, consultas avanzadas, añadir o eliminar direcciones Web, permitía hacer búsquedas de imágenes y ficheros multimedia.

[1] La historia acerca de los buscadores Web fue obtenida en http://manuales.ojobuscador.com/historia/

En 1996 se comenzó a desarrollar Google, llamándose en un principio BackRub. Este tenía una interfaz muy clara y resultados relevantes.
En el 2004 MSN Search con la ayuda de Christopher Payne y Oshoma Momoh echan andar la primera fase del motor de Microsoft. Presentándose así en noviembre de 2005 la plataforma de Windows Live que será la nueva interfaz del motor de búsqueda de Microsoft.

Ya para el 2006 nace Exaltad utilizando el motor de búsqueda Quaero que es un buscador Europeo.

1.1.2 Estudios realizados en el país.

La documentación y estudios existentes en el país referente a robots spider se orientan en la implementación sobre un buscador, encontrándose los siguientes registros:

En la Universidad Centroamericana "José Simeón Cañas" (UCA) del año de 1998 a la actualidad, se encuentra la tesis:

Desarrollo e implementación de un buscador de sitios Web para El Salvador. (Desarrollada por: Elio David Vides).
Trabajo de graduación elaborado con el objetivo de ofrecer un buscador en Internet que permita a los usuarios encontrar información solicitada y relacionada a El Salvador en sus diferentes ámbitos: cultural, social, económico, político, comercial y educativo.
Para su implementación se utilizó el robot Web llamado Combine System escrito en Perl.
Se contaba con el respaldo de la empresa SVNet grupo nacional la cual permite a las empresas y particulares tener sus propios dominios en Internet.
El buscador llamado Mirador se encontraba dentro de la dirección de la empresa www.svnet.org.sv/mirador
En la Universidad Don Bosco, se encontró en los registros de la biblioteca la tesis: **Diseño, desarrollo e implementación de un buscador de sitios Web nacionales.**

(Desarrollada por: Rolando Francisco Álvarez, Jorge Alexander Cruz y Guadalupe Mejía Salguero en el año 2000).
Este proyecto se tuvo el apoyo de SALNET; una empresa salvadoreña dedicada a proveer servicios de Internet.

El trabajo de graduación documenta que utilizó el robot llamado BDDBot; la documentación de su funcionamiento se encuentra en Internet y es elaborado en el lenguaje de programación Java, eligiéndolo, por que se adecuaba a los fines que se pretendían alcanzar. El buscador tuvo por nombre Ciber Izalco, el cual se implementó en una red de tres computadoras para su elaboración.

1.2 IMPORTANCIA DE LA INVESTIGACIÓN.

1.2.1 PLANTEAMIENTO DEL PROBLEMA.

En la mayoría de las instituciones educativas del país, actualmente se cuenta con poca información acerca de los buscadores Web; frente a esta situación surge la idea de la creación de un documento que detalle y describa el funcionamiento de un robot spider, ya que éste es la base fundamental para la elaboración de un buscador Web moderno que no este basado en directorios.

1.2.2 DEFINICIÓN DEL TEMA.

Desarrollar un estudio monográfico que explique el funcionamiento de los robots spider, y que exponga como implementar un robot en un motor de búsqueda.

1.2.3 JUSTIFICACIÓN.

La búsqueda de instrumentos que fortalezcan el proceso de enseñanza aprendizaje es uno de los aspectos que mayormente ha interesado a la Universidad Don Bosco en su afán por lograr la formación de profesionales de calidad.

El presente proyecto de investigación acerca de los robots spider o exploradores de la red, es un esfuerzo para facilitar el proceso de enseñanza aprendizaje y de investigación, como parte esencial de la función formadora de profesionales de calidad de la Universidad Don Bosco.

Por otra parte es importante señalar que, el hecho de fomentar la investigación en estudiantes y profesores es una buena justificación, y por lo tanto, la generación de conocimiento que esto implica. También se considera que el producto de esta investigación podrá ser la base para dar paso a su vez, a futuros trabajos de investigación que se relacionen con temas variados, tales como la programación en Internet, la programación Web, la obtención o recuperación de información de colecciones heterogéneas, publicaciones Web y los sistemas de búsqueda por mencionar algunos.

1.3 OBJETIVOS.

1.3.1 GENERAL.

Elaborar una monografía basándose en el modelo instruccional que sirva de base teórica y de guía practica para la configuración de los componentes de un Robot Spider o explorador de la red.

1.3.2 ESPECIFICOS.

1. Elaborar dos documentos en forma de artículos para la revista de la Facultad de Ingeniería acerca de la utilidad y el funcionamiento de los robots spider.

2. Investigar algunos algoritmos que se utilizan en la construcción de robots spiders o exploradores de la red.

3. Realizar una monografía que sirva de base teórica y de guía practica para la configuración de los componentes de un robot spider o explorador de la red.

4. Implementar un robot spider ya existente, con las modificaciones necesarias para su funcionamiento dentro de un motor de búsqueda.

5. Preparar una guía que sirva de herramienta de apoyo para la comunidad académica para poder ejecutar su propio buscador, y del mismo modo incentivar el desarrollo de este tipo de software en el país.

1.4 ALCANCES Y LIMITANTES.

1.4.1 ALCANCES.

1. Dar a conocer el funcionamiento de los robots spiders, componentes, clasificación, características propias y algoritmos utilizados en su creación.

2. Redactar una guía didáctica que contenga la información relevante sobre los robots spider, su papel en los motores de búsqueda y la implementación de un robot spider como el Combine System en un buscador; este escrito contendrá los siguientes capítulos:

Capitulo 1 Introducción a la robótica.

1.1 Concepto de robot aplicado a la Web.

1.2 Agentes.

- 1.2.1 Agente autónomo.
- 1.2.2 Agente inteligente.
- 1.2.3 Agente de usuario.

1.3 Comportamiento de robots.

- 1.3.1 Spider.
- 1.3.2 Web Ants.
- 1.3.3 Web Crawler.
- 1.3.4 Worms.
- 1.3.5 Wanderes.
- 1.3.6 Knowbots.

1.4 Tipos de robots.

- 1.4.1 Robot de análisis estadísticos.
- 1.4.2 Robot de mantenimiento.
- 1.4.3 Robot de copia a espejo.
- 1.4.4 Robot de ordenamiento de recursos.
- 1.4.5 Robot combinado.

Capitulo 2 Conceptos básicos de redes y programación.

2.1 Componentes conceptuales de una red.

2.1.1 Subsistemas de comunicaciones.

2.1.1.1 Modelo OSI.

2.1.1.2 TCP/IP.

2.1.2 Aplicaciones de red.

2.2 Modelo Cliente/Servidor.

2.3 Lenguajes de programación para la Web.

2.3.1 Java.

2.3.2 Perl.

2.3.3 CGI.

2.3.4 ActiveX.

2.3.5 Scripts.

2.3.5.1 JavaScript y Jscript.

2.3.5.2 VBScript, ASP.

2.3.5.3 PHP.

2.4 Fundamentos de Internet.

2.4.1 Protocolos de Internet.

2.4.2 dirección IP.

2.4.3 Sistemas de nombres de dominio.

2.4.3.1 Dominios del nivel más alto.

2.4.3.2 Designaciones de país.

2.4.4 URI y URL.

2.4.4.1 Relaciones entre URL, protocolos y tipos de archivo.

2.4.4.2 URL y HTML.

2.4.4.3 URL absolutos y relativos.

2.4.5 HTTP.

2.4.5.1 Clases de códigos de respuesta de HTTP.

2.4.5.2 Métodos HTTP.

2.4.5.3 Otros métodos de HTTP.

2.4.6 FTP.

2.4.7 HTML.

2.4.7.1 Etiquetas HTML.

[2] Estos módulos son listados en anexos

3. Elaborar la configuración respectiva para que el buscador funcione solo para sitios nacionales con extensión .sv.

1.4.2 LIMITANTES.

Entre los factores que podrían afectar el desarrollo del proyecto, se encuentran los siguientes:

1. No se cuenta con completo acceso a los algoritmos de los buscadores comerciales por cuestiones de seguridad, esto implica que solo se estudiarán algunas de sus partes.

2. El estudio monográfico contendrá únicamente el análisis detallado del robot spider Combine System.

1.5 FACTIBILIDAD[3].

1.5.1 Factibilidad Técnica:

Desde el punto de vista técnico el proyecto es realizable, ya que se dispone con la información necesaria para la realización de la investigación, además del equipo necesario para el diseño y la implementación que se va a desarrollar, las herramientas de software y el recurso humano.

Para la realización del estudio monográfico de los robots spiders se cuenta con información disponible en su mayoría en Internet como lo son libros en formato digital y la información técnica encontrada sobre spiders o motores de búsqueda, además de trabajos de graduación impresos similares al tema de investigación en El Salvador.

Para la implementación del motor de búsqueda, como se dijo anteriormente, se encuentran disponibles en Internet un gran numero de robots spider; para la mayoría de ellos es accesible su documentación de funcionamiento, lenguaje de programación en que está elaborado, código fuente, disponibilidad comercial (Shareware[4], Freeware[5]), foros de discusión, contacto con el o los creadores del robot y la actualización de las ultimas versiones para un robot Web en particular.

1.5.2 Factibilidad Económica:

El costo monetario que se generará en el proceso de desarrollo del presente proyecto en cuanto al uso de tecnología que se empleará para su impulso, es económicamente factible de acuerdo a lo siguiente:

Para el estudio monográfico, la investigación en ésta etapa se avocará mayoritariamente en Internet y los costos incurridos para llevarla a cabo son el pago mensual por conexión residencial a Internet a velocidad de 256 Kbps, el cual es de $28.25.

Para la implementación de motor de búsqueda, se efectuará sobre una red de tres computadores a modo de ensayo antes de su puesta en Internet. La red de

[3] La factibilidad establece si el proyecto es posible de realizarse, a fin de que se cuente con los recursos y capacidades para cumplir con los objetivos y necesidades de los usuarios.
[4] Software protegido por leyes de copyright, que se encuentra disponible gratuitamente durante cierto tiempo.
[5] Aplicaciones de uso gratuito que pueden encontrarse en Internet.

computadoras tiene como objetivo realizar las diferentes pruebas pertinentes necesarias para poder visualizar como se comportará el algoritmo del robot Web en cuanto a peticiones de búsqueda.

HARDWARE			
Cantidad	Articulo	Costo U.	Costo N.
3	Computadoras	$ 307	$ 921
1	Impresora	$ 100	$ 100
1	UPS	$ 50	$ 50
		Total	$ 1071
SOFTWARE			
1	S. O. Linux distribución Debian 3.0	$ 0.00	$ 0.00
1	MySQL	$ 0.00	$ 0.00
1	Zebra	$ 0.00	$ 0.00
1	Tidy	$ 0.00	$ 0.00
1	Perl	$ 0.00	$ 0.00
1	PHP	$ 0.00	$ 0.00
DESARROLLO DEL ESTUDIO MONOGRÁFICO E IMPLEMENTACIÓN DE MOTOR DE BÚSQUEDA			
3	Programadores	$ 500.00 (c/u)	$ 13,500 durante 9 meses
OTROS GASTOS			
	Conexión a Internet Residencial (256K)	$ 28.25	$ 254.25 durante 9 meses
	Energía eléctrica	$ 120.00	$ 1080 durante 9 meses
		Costo Total	**$ 15,905.25**

1.5.3 Factibilidad Operativa:

La factibilidad operativa comprende una determinación sobre el uso de la información que se da a conocer para que se maneje de la mejor forma posible, considerando los siguientes aspectos:

- El estudio monográfico se realiza a fin de explicar las partes que componen un robot Web destinado a estudiantes de universidades y profesores con conocimientos básicos de programación, con el fin de conocer su funcionamiento.
- La implementación del motor de búsqueda expone la utilidad de los robots Web como una de las tecnologías de Internet que hoy en día utilizamos.

1.6 METODOLOGÍA DE LA INVESTIGACIÓN.

Para la investigación del proyecto se utilizarán principalmente 3 herramientas de investigación:

- La investigación documental, se hará uso de fuentes bibliográficas impresas mayores de estudio como: manuales para el uso de las herramientas como el Combine System, libros de texto referentes al Web y distintas tesis tanto nacionales como internacionales, además de información que se encuentra en Internet y en otros medios magnéticos.

- Entrevistas a profesionales y/o profesores del área de informática.

- Evaluación de productos de software, específicamente sistemas de búsqueda, como buscadores comerciales, motores de búsqueda y robots spiders.

Para la realización de cada uno de los capítulos de investigación se ha pensado en poner en práctica el siguiente proceso:

- Investigación previa del capitulo; éste apartado cubre identificar y seleccionar la documentación adecuada al tema del capitulo de estudio.

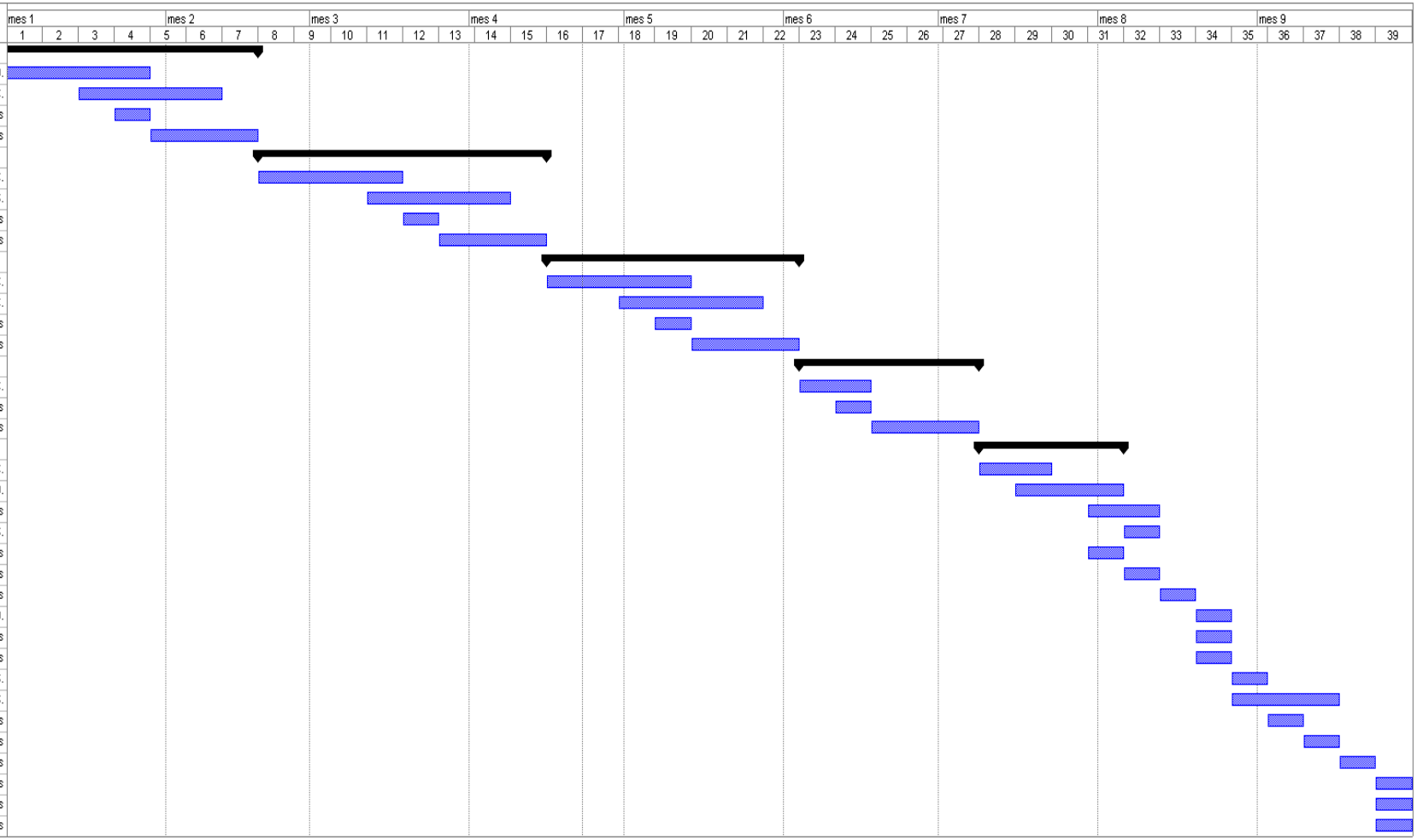
- Identificación de los lugares donde están ubicados los documentos seleccionados y la recuperación de ellos.

- Análisis de los documentos recuperados.

- Sistematización de la investigación obtenida de los documentos estudiados.

1.7 CRONOGRAMA DE ACTIVIDADES.

Objetivos	Nombre de tarea	Responsable
3	Capitulo 1	
	Investigación sobre robótica	Vanesa B.
	Análisis de la información	Víctor C.
	Reunión con asesor	Todos
	Elaboración del capitulo	Todos
3	Capitulo 2	
	Investigación de conceptos	Vanesa B. y Víctor C.
	Análisis de información	William S.
	Reunión con asesor	Todos
	Elaboración del capitulo	Todos
2 y 3	Capitulo 3	
	Investigación robots spider	William S. y Víctor C.
	Análisis de la información	Vanesa B. y Víctor C.
	Reunión con asesor	Todos
	Elaboración del capitulo	Todos
2 y 3	Capitulo 4	
	Análisis de la información	William S. y Víctor C.
	Reunión con asesor	Todos
	Elaboración del capitulo	Todos
2, 3 y 5	Capitulo 5	
	Análisis de la información	Víctor C.
	Elaboración del capitulo	Víctor C. y Vanesa B.
	Depuración del documento	Todos
1	Elaboración del 1° articulo	Víctor C y William S.
	Reunión con asesor	Todos
	Reunión con tutor	Todos
	Primera defensa	Todos
	Corrección de observaciones	William S. y Vanesa B.
	Reunión con asesor	Todos
	Reunión con tutor	Todos
1	Elaboración del 2° articulo	Vanesa B. y William S.
2, 3, 4 y 5	Elaboración capitulo 6	Víctor C. y William S.
	Reunión con asesor	Todos
	Reunión con tutor	Todos
	Segunda defensa	Todos
	Corrección de observaciones	Todos
	Reunión con asesor	Todos
	Reunión con tutor	Todos



Nº	Objetivos.
1	Elaborar dos documentos en forma de artículos para la revista de la Facultad de Ingeniería acerca de la utilidad y el funcionamiento de los robots spider.
2	Investigar algunos algoritmos que se utilizan en la construcción de robots spiders o exploradores de la red.
3	Elaborar una monografía que sirva de base teórica y de guía practica para la configuración de los componentes de un robot spider o explorador de la red.
4	Implementar un robot spider ya existente, con las modificaciones necesarias para su funcionamiento dentro de un motor de búsqueda.
5	Elaborar una guía que sirva de herramienta de apoyo para la comunidad académica para poder ejecutar su propio buscador, y del mismo modo incentivar el desarrollo de este tipo de software en el país.

Nº	Capítulos.
1	Introducción a la robótica.
2	Conceptos básicos generales.
3	Sistemas de búsqueda en Internet (robot spider).
4	Algoritmos y conceptos utilizados en la elaboración de robots spider.
5	Combine System.
6	Implementación del robot Combine System en un buscador Web.

1.8 MARCO TEÓRICO.

Un buscador es la herramienta que permite investigar, averiguar, examinar la red[6] a partir de unas palabras que se tienen que introducir para describir los que se busca.

Detrás de un buscador hay una base de datos que contiene la información y unos sistemas de indexación, compresión y organización de los datos que permiten efectuar búsquedas por palabras rápidamente.

Las búsquedas se hacen con palabras clave o con árboles jerárquicos por temas; el resultado de la búsqueda es un listado de direcciones Web en los que se mencionan temas relacionados con las palabras clave buscadas.

Los buscadores se pueden clasificar en dos tipos:

- Índices temáticos: Son sistemas de búsqueda por temas o categorías jerarquizados (aunque también suelen incluir sistemas de búsqueda por palabras clave). Se trata de bases de datos de direcciones Web elaboradas "manualmente", es decir, hay personas que se encargan de asignar cada página Web a una categoría o tema determinado.

- Motores de búsqueda: Son sistemas de búsqueda por palabras clave. Las bases de datos, en los motores de búsqueda, incorporan automáticamente páginas Web mediante "robots" de búsqueda por la red.

1.8.1 Robots del Web: Un robot es un término aplicado a programas de comunicación que utilizando HTTP como protocolo de comunicación, exploran grandes porciones del Web y de manera recursiva extraen información de ellas. Dicha información puede después ser utilizada para alimentar motores de búsqueda, para efectos estadísticos, para realizar copias de respaldo, etc.

Dentro de la terminología propia del Internet, a estos programas se les dan varios nombres que, en algunos casos, denotan algún comportamiento especial:

- Spider (araña): robot del Web.

[6] World Wide Web

- Crawler (reptil): orientado a dar la impresión de movimiento.
- Worm (lombriz): trabajan de forma distribuida, explorando simultáneamente diferentes porciones del Web.

1.8.2 Robot Spider Combine System: El robot que se utilizara para las prácticas en el último capitulo sobre implementación de un robot spider en un buscador Web será el Combine System. Combine System es un robot para recolección de recursos en el Web, y está diseñado para ser un sistema distribuido, paralelo y flexible.

- Distribuido: Ya que todos los protocolos de comunicación interna están basados en la familia de protocolos TCP/IP y los diferentes componentes del sistema pueden ser distribuidos entre un número ilimitado de computadoras conectadas a Internet.

- Paralelo: Ya que el sistema es totalmente distribuido, es posible ejecutar varios componentes en computadoras individuales, las cuales son más adecuadas a los requerimientos de hardware y comunicación para esos componentes específicos.

- Flexible: El sistema esta organizado como una colección de muchos programas pequeños. Cada uno de ellos está encargado de pocas tareas dedicadas. Esta característica permite que el sistema pueda ser adaptado de una manera fácil a un nuevo tipo de aplicación agregando o reemplazando uno o algunos componentes.

La decisión de utilizar el robot Combine System se debe a que:

- Es parte de la ingeniería en una caja, esto le da cierta ventaja ya que tiene una gran compatibilidad con Zebra para realizar en conjunto un buscador bastante eficiente.

- Una gran posibilidad de configuración, por ser software bajo la licencia GPL, el código fuente de este puede ser manipulado libremente para obtener mejores resultados.

- Posee un sistema avanzado de tópicos, que permite realizar búsquedas avanzadas dándole prioridad a palabras con mayor relevancia en una búsqueda.

- Se puede manipular para que funcione en un dominio determinado, en nuestro caso este será el dominio .sv.

- Posee una enorme compatibilidad para distintos tipos de documentos como text, HTML, PDF, PostScript, MSWord, Power Point, Excel, RTF, TeX y distintos tipos de imágenes.

- Obedece la reglas de los robots, al reconocer el archivo robots.txt.

- Utiliza un sistema de bases de datos de fácil entendimiento y bastante eficiente MySQL.

- Ha sido probado en distintos motores de búsqueda como en los buscadores europeos Alvis Search, Engine New y Alvis Wikipedia Search.

- Posee una buena documentación técnica desde el sitio http://combine.it.lth.se/.

1.8.3 Clases de buscadores.

1. Los motores de búsqueda o arañas:

La mayoría de los grandes buscadores internacionales que todos conocemos son de este tipo.

Recorren las páginas recopilando información sobre los contenidos de las páginas.

Ejemplos de Spiders: Google, Altavista, Hotbot, Lycos.

2. Los Directorios:

Una tecnología que es ampliamente utilizada por la cantidad de programas scripts en el mercado. No se requieren muchos recursos de informática, pero si de soporte humano y mantenimiento.

Los directorios no recorren las Web ni almacenan sus contenidos. Los resultados de la búsqueda estarán determinados por la información que se haya suministrado al directorio cuando se registra la página Web.

3. Sistemas mixtos Buscador - Directorio:

Poseen características de buscadores y además presentan las páginas registradas en catálogos sobre contenidos temáticos que a su vez se dividen en subsecciones.

4. Metabuscadores:

No son buscadores reales; lo que hacen es realizar búsquedas en auténticos buscadores, analizan los resultados de la página y presentan sus propios resultados.

Presentan la ventaja de seleccionar para el usuario, los mejores sitios que presentan los buscadores consultados

CAPITULO II. ELABORACIÓN DE GUÍA TEÓRICA Y PRÁCTICA SOBRE LOS ROBOTS SPIDER.

INTRODUCCION

La siguiente guía esta diseñada para facilitar el proceso de enseñanza y aprendizaje para todos los estudiantes a partir de un nivel de tercer año en la carrera de Ingeniería en Ciencias de la Computación y/o carreras afines, por lo tanto, es necesario que tenga conocimientos de: **sistema operativo LINUX, programación en PHP, HTML, conocimientos básicos de redes;** mas que todos enfocados a Internet.

Por lo tanto esperamos que la siguiente guía teórica-práctica despierte el interés de las personas en conocer sobre los robot spider, así mismo, que ésta inquietud lleve a la creación de nuevos proyectos de investigación.

2.1: INTRODUCCIÓN A LA ROBÓTICA.

Tiempo recomendado de estudio: 1 hora con 30 minutos.

Objetivos:

Al concluir este capítulo, el lector estará capacitado para:

- Definir el concepto de robot aplicado al Web.
- Identificar que es un agente y nombrar sus distintos tipos.
- Comprender como se clasifican los robots en base a su comportamiento.
- Nombrar y describir los distintos tipos de robots Web existentes.

Introducción:

Cuando la mayoría de nosotros escuchamos la palabra *"robot"* inmediatamente se nos viene a la mente un ser con forma semihumana o parecido a algún animal, con brazos, piernas, con capacidad de movilizarse por si mismo y que además posee inteligencia artificial, pero en este capítulo veremos que los robots no necesariamente deben poseer un cuerpo físico; es decir, muchos de los robots que utilizamos hoy en día y de manera cotidiana, únicamente los podemos ver a través de una computadora, esperamos que este capitulo sea de mucha ayuda al lector para que se familiarice con esta clase de robots, en especial con los denominados robots spider.

En este capítulo el lector se familiarizará con algunos términos básicos que le ayudaran a entender los capítulos posteriores, algunas definiciones como robot spider, agentes y la clasificación de los robots según su comportamiento serán vitales para llegar a construir un buscador Web en los últimos capítulos.

2.1.1 CONCEPTO DE ROBOT APLICADO AL WEB.

Un robot es un programa que rastrea recursos a través del Internet siguiendo los vínculos que contienen las páginas Web albergadas en un servidor o en otros servidores de la red.
En general, la búsqueda comienza con una lista inicial de servidores, para luego seguir los vínculos que tengan estos con el resto de los documentos que se encuentran en la *Web.* La arquitectura y funcionamiento de estos sistemas se verá en forma detallada en los siguientes capítulos.

Los principales buscadores de Internet construyen sus bases de datos usando robots comúnmente denominados spiders, crawlers o webcrawlers.

Los sistemas de búsqueda basados en robots (*robots spider)* realizan las siguientes acciones:

- Recuperan y procesan todas las páginas Web que encuentran.
- Extraen información de referencia (índices) sobre las páginas.
- Los índices se almacenan en bases de datos que ofrecen servicios de búsqueda basados en palabras clave.

Los robots Web son potentes programas que recorren la Web de forma automática y buscan diferentes tipos de datos como texto, imágenes o sonido.

Estos datos, junto con las direcciones URL que los contienen, son indexados, clasificados y almacenados en grandes bases de datos para que los usuarios de Internet posteriormente, envíen sus consultas e interroguen a la base de datos buscando alguna frase o palabra clave.

Los robots periódicamente hacen un recorrido de estas páginas para buscar alguna modificación o la incorporación de nuevos datos y/o actualizaciones, de esta forma, la actualización de los datos se realiza de forma automática.

En general, los robots comienzan con un listado de enlaces y URL's preseleccionadas y, recurrentemente, visitan los documentos que se referencian desde las mismas.

Ejemplos de robots son:

- Googlebot (robot de Google).
- Mozilla Compatible Agent (robot de Yahoo).
- Msnbot (robot de MSN).

2.1.2 AGENTES.

En el campo informático, podemos definir un agente como un componente software y/o hardware que es capaz de actuar para realizar tareas en beneficio del usuario.
De forma más específica:

Un agente es un sistema de *hardware y/o software* que interactúa con su entorno éste puede ser otros agentes o humanos, guiado por uno o varios propósitos, es proactivo es decir, reacciona a eventos y a veces se anticipa haciendo propuestas, es adaptable, esto significa que se puede enfrentar a situaciones novedosas, es sociable, se comunica, coopera o negocia y su comportamiento es predecible en cierto contexto[7].

Las aplicaciones de los agentes son muy numerosas, entre las que podemos destacar:

- El uso de agentes en Internet e interfaces de usuarios.
- Utilización en sistemas de información, juegos y animaciones, comercio electrónico, educación, etc.

[7] Olivares, Jesús. Agentes: concepto, aplicaciones y referencias. http://www.jesusolivares.com/

En el ámbito de la Web, podemos destacar los siguientes tipos de agentes:

2.1.2.1 AGENTE AUTÓNOMO: se trata de un programa que *"viaja"* entre los sitios Web, decidiendo por él mismo qué debe hacer y cuándo debe moverse a otros lugares.

2.1.2.2 AGENTE INTELIGENTE*:* se trata de un programa que ayuda al usuario a ciertas acciones. Por ejemplo, a rellenar formularios, elegir productos, encontrar algo en especifico, etc.
Este tipo de agentes también se denominan *softbot* (*software robot)*.
Usa herramientas de *software* y servicios basados en el comportamiento de las personas.

2.1.2.3 AGENTE DE USUARIO*:* es el nombre técnico para denominar a un programa que ejecuta determinadas tareas para un usuario en la red.
Ejemplos son: un navegador como *Internet Explorer*, o un agente de correo del tipo *Email User-agent, Eudora,* etc.

Preguntas de repaso.

1. ¿Qué es un robot aplicado al Web?

_____
_____.

2. ¿Qué es un agente?

_____
_____.

3. Clasifique los siguientes programas según el agente al que pertenece autónomo, inteligente o de usuario. Ejemplo*: Mozilla Fire Fox* *usuario*

Navegador Web Opera _____

Googlebot _____

Formulario de relleno gmail _____

MsnBot _____

Navegador Internet Explorer _____

2.1.3 COMPORTAMIENTO DE ROBOTS.

Los robots spiders adoptan numerosas denominaciones todas ellas tienen que ver con la traducción al español de su significado en inglés, como por ejemplo Web = telaraña, y en la forma que estos robots se mueven a través de Internet.

Sin embargo, lo único que hace un robot es visitar los sitios y extraer los enlaces que están incluidos dentro de estos.

A continuación se detallan los principales tipos y denominaciones de robots:

2.1.3.1 ARAÑAS *(Spiders*): es un programa usado para rastrear la red. Lee la estructura de hipertexto y accede a todos los enlaces referidos en el sitio Web. Se utiliza como sinónimo de robot y crawler. Ejemplos: *Googlebot, Combine System y BDDBot.*

2.1.3.2 HORMIGAS (*Web Ants*): cooperativa de robots. Trabajan de forma distribuida, explorando simultáneamente diferentes porciones de la Web. Son robots que cooperan en un mismo objetivo, por ejemplo, para llevar a cabo una indexación[8] distribuida.

[8] Proceso que consiste en la representación del contenido de un documento o de una parte del mismo, mediante la selección de términos apropiados y se expresa en un lenguaje de búsqueda informativa o natural para facilitar la recuperación.

2.1.3.3 ORUGAS (*Web crawlers*): es un programa que inspecciona las páginas del World Wide Web de forma metódica y automatizada. Los Web crawlers se utilizan para crear una copia de todas las páginas Web visitadas para su procesado posterior por un motor de búsqueda que indexa las páginas proporcionando un sistema de búsquedas rápido. Ejemplos: *WebCrawler y MetaCrawler.*

2.1.3.4 GUSANOS (*Worms*): es lo mismo que un robot, aunque técnicamente un gusano es una réplica de un programa, a diferencia de un robot que es un programa original. Se usan, por ejemplo, para duplicar los directorios de FTP para que puedan acceder más usuarios. Ejemplos: *Bagle, NetSky o Passer.*

2.1.3.5 VIAJERO, VAGABUNDO (*Wanderes*): son una clase de robots que realizan estadísticas sobre la Web, como por ejemplo, número de servidores, servidores conectados, número de Webs.

2.1.3.6 ROBOTS DE CONOCIMIENTO (*Knowbots*): localizan referencias hipertextuales dirigidas hacia un documento o servidor concreto. Permiten evaluar el impacto de las distintas aportaciones que engrosan las distintas áreas de conocimiento de la Web. Ejemplos: *Newstracker*, *Mind-i, Eliza*.

Preguntas de repaso.

1. ¿En que se basan las clasificaciones de los robots spider?

_____

_____.

2. Mencione y explique brevemente tres clasificaciones de los robots spider.

_____

_____.

3. Mencione al menos un robot spider que se comporte como:

Gusano _____.

Oruga _____.

Araña _____.

2.1.4 TIPOS DE ROBOTS.

2.1.4.1 ROBOT DE ANÁLISIS ESTADÍSTICO: Este tipo de robot extrae información y la utiliza para realizar cálculos de tipo estadísticos, tales como controlar el número promedio de documentos existentes en los servidores de un dominio o región y el tamaño de los documentos que existen en los servidores de un dominio. Estos robots son implementados en pocas ocasiones y normalmente se utilizan para cubrir otro propósito además del estadístico. Ejemplo*: The Wanderer* fue el primer robot en Internet y se diseñó para contar el número de servidores *Web* que existían en la red.

2.1.4.2 ROBOT DE MANTENIMIENTO: Orientado a la administración de la estructura hipertexto en servidores Web. Principalmente funciona generando peticiones de tipo HEAD para detectar si un URL es valido, en caso que no lo sea, informa a una base de datos el estado del recurso, y especifica si es posible acceder a él desde una ubicación diferente. Por el tipo de trabajo de verificación que realiza, éste robot debe recorrer una base de datos y compararla contra el estado de los vínculos dentro del Web y desechar los que hayan sido desplazados o eliminados. Ejemplos: *Checkbot* y *MomSpider*, este último también es capaz de funcionar como un robot combinado.

2.1.4.3 ROBOT DE COPIA A ESPEJO: Son utilizados para mantener estructuras de hipertexto redundante con el objetivo de proveer una respuesta rápida y segura a los fallos en los servicios Web. También se utilizan para distribuir carga de servidores remotos que contienen la misma información o información recientemente actualizada. Este tipo de distribución de recursos es normal en sitios FTP, debido a que se tienen limitaciones grandes en el número de conexiones permitidas por un sitio que maneja este protocolo. Así que es preferible distribuir la carga por zonas geográficas o mejor aun, por dominios. Este tipo de robots también realiza, aunque de manera implícita, una labor de mantenimiento del estado de los vínculos. Un ejemplo de estos robots es: *HTML Gobble.*

2.1.4.4 ROBOT DE ORDENAMIENTO DE RECURSOS: Este tipo de robot se caracteriza por explorar y reunir grandes porciones del Web. Además la información extraída, es decir URL's y Meta Tags, es puesta en bases de datos e indexada para después ponerla accesible a usuarios en el Web a través de buscadores de sitios. Otra de las tareas que realizan estos robots es la de actualizar periódicamente la base de datos del buscador. Con esto se lleva a cabo un mantenimiento implícito de la base de datos. Normalmente se requiere de un proceso que ordene la información que se extrae, antes de pasarla a la base de datos que accede al buscador, ya que un proceso de este tipo podría requerir de más tiempo y recursos, lo cual provocaría que el robot tenga que esperar a que el proceso termine sobre la información anterior antes de continuar. Ejemplo: *Webpages Robot, Scanner* y *Combine System.*

2.1.4.5 ROBOT COMBINADO: Como su nombre lo indica entra en esta clasificación todo robot capaz de realizar tareas propias de otros tipos de robot, pero que no cumplen con algunas características propias de alguno de ellos. Entre los que comúnmente se implementan son los robots estadísticos combinados con robots de Mantenimiento, y los robots estadísticos y de copia a espejo. Estos robots surgieron del hecho que un robot de mantenimiento

normalmente extrae datos dentro del cuerpo de la entidad (Entitu-body) de un mensaje de respuesta de http, y luego la desechan debido a que no sirve a sus propósitos, así que es posible utilizar esos datos para otros propósitos como el mantenimiento *del contenido del documento y/o de la dirección URL*. Ejemplos *Pack Rat* (Estadístico-Mantenimiento).

Preguntas de repaso.

1. Marque con X en el enunciado que mejor defina el comportamiento de los robots de mantenimiento:

__ Explora y reúne grandes porciones del Web.

__ Extrae información y la utiliza para fines estadísticos

__ Esta orientado a la administración de la estructura hipertexto en servidores Web.

2. Defina los siguientes tipos de robots:

Robot de copia a espejo: _____
_____
_____.

Robot estadístico: _____
_____
_____.

Robot combinado: _____
_____
_____.

Auto evaluación.

Instrucciones.

En la hoja provista, subraye la respuesta correcta. Coteje las respuestas al final de la auto evaluación.

1. Un agente es:
   a. Un robot que indexa las paginas Web.
   b. Un programa para rastrear la red.
   c. Sistema de hardware y/o software que interactúa con su entorno.

2. Los robots Web son:
   a. Programas para rastrear la red.
   b. Potentes programas que recorren la Web de forma automática y buscan diferentes tipos de datos.
   c. Sistemas de hardware y/o software que interactúan con su entorno.

3. Un robot Spider es:
   a. Un programa que lee la estructura de hipertexto y accede a todos los enlaces referidos en el sitio Web.
   b. Un programa para rastrear la red.
   c. A y B son correctas.

4. Una de las clasificaciones de robot según su comportamiento es:
   a. Robot estadístico.
   b. Robot Spider.
   c. Ninguna de las anteriores.

5. Cual de los siguientes programas es un ejemplo de robot Spider
   a. Combine System.
   b. Mozilla Fire Fox.
   c. A y B son correctas.

Contestaciones de las preguntas de la página anterior.

1. C, 2. B, 3. C, 4. B, 5. A

De 5 a 3 preguntas correctas: Puede continuar con el capitulo 2

Menos de 3 preguntas: Repase el capitulo 1 y compruebe el ejercicio.

Investigación Complementaria.

- Investigue como funciona el Googlebot.
- Investigue 2 ejemplos mas de robots combinados excluyendo el mencionado en éste capitulo.
- Realice un resumen del capitulo describiendo con sus propias palabras ¿Qué es un robot aplicado al Web? ¿Cómo se clasifican los robots? ¿Qué es un agente? ¿Cómo se clasifican los agentes?

2.2: CONCEPTOS BÁSICOS DE REDES Y PROGRAMACIÓN.

Tiempo recomendado de estudio: 4 horas.

Objetivos:

Al concluir este capitulo, el lector estará capacitado para:

- Definir los conceptos básicos de redes.
- Comprender como funciona el modelo OSI y sus distintas capas.
- Mencionar algunos de los lenguajes utilizados para la programación Web.
- Entender de manera básica como funciona el Internet.

Introducción:

En el presente capítulo el lector se familiarizará con los conceptos básicos, necesarios para comprender como funcionan las redes en general y en especial Internet.
En este capítulo se estudiara el modelo de referencia de Interconexión de Sistemas Abiertos (OSI, Open System Interconnection).

Para enfrentar el problema de incompatibilidad de redes, la Organización Internacional para la Estandarización (ISO) investigó modelos de networking como la red Digital Equipment Corporation (DECnet), la Arquitectura de Sistemas de Red (SNA) y TCP/IP a fin de encontrar un conjunto de reglas aplicables de forma general a todas las redes. Con base en ésta investigación, la ISO desarrolló un modelo de red que ayuda a los fabricantes a crear redes que sean compatibles con otras redes.
Por otro lado también se estudiarán los fundamentos básicos de Internet y los distintos lenguajes que se utilizan en la Web como Java, Perl, HTML.
De esta manera el lector comenzara a habituarse a conceptos que se utilizaran en próximos capítulos.

2.2.1 COMPONENTES CONCEPTUALES DE UNA RED.

A continuación se definirán algunos de los términos elementales de una red:

2.2.1.1 MODELO OSI.

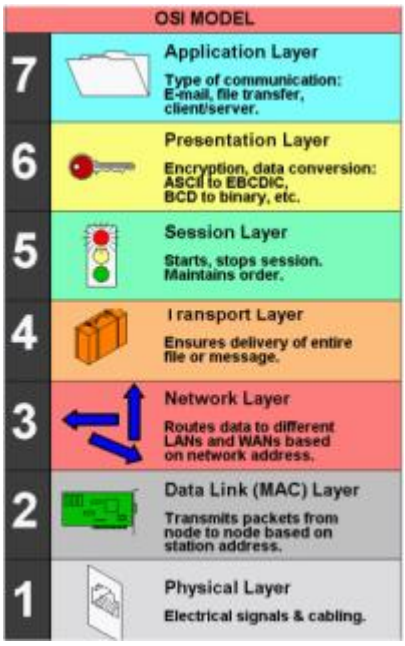


Figura 2.2.1 Capas del modelo OSI.

El Modelo OSI es un lineamiento funcional para tareas de comunicaciones y, por consiguiente, no especifica un estándar de comunicación para dichas tareas. Sin embargo, muchos estándares y protocolos cumplen con los lineamientos del Modelo OSI.

Como se mencionó anteriormente, OSI nace con el objetivo de uniformizar los elementos que participan en la solución del problema de comunicación entre equipos de cómputo de diferentes fabricantes.

Estos equipos presentan diferencias en:

- Procesador central.
- Velocidad.
- Memoria.
- Dispositivos de almacenamiento.
- Interfaces para comunicaciones.
- Códigos de caracteres.
- Sistemas Operativos.

Estas diferencias propician que el problema de comunicación entre computadoras no tenga una solución simple. Dividiendo el problema general de la comunicación, en problemas específicos, facilitando la obtención de una solución a dicho problema.

A continuación se describen las distintas capas del modelo OSI:

2.2.1.1.1 Capa Física.

La capa física, es la capa más baja del modelo OSI, se encarga de las conexiones físicas de la computadora hacia la red. Es la encargada de transmitir los bits de información por el medio utilizado para la transmisión; medios guiados como: cable coaxial, cable de par trenzado, fibra óptica o medios no guiados como: radio, infrarrojos, microondas.

También se ocupa de las propiedades físicas y características eléctricas de los diversos componentes y de la velocidad de transmisión

El nivel físico recibe una trama binaria que debe convertir a una señal eléctrica, electromagnética, óptica u otra dependiendo del medio, de tal forma que a pesar de la degradación que pueda sufrir en el medio de transmisión vuelva a ser interpretable correctamente en el receptor.

Dicho en otras palabras transforma un paquete de información binaria en una sucesión de impulsos adecuados al medio físico utilizado en la transmisión. Estos impulsos pueden ser por transmisión por cable, transmisión inalámbrica *(Wireless)* o transmisión óptica. Cuando actúa en forma de recepción el trabajo es inverso, se encarga de transformar estos impulsos en paquetes de datos binarios que serán entregados a la capa de enlace.

2.2.1.1.2 Capa de Enlace.

La capa de enlace proporciona transferencia de tramas de datos desde un nodo a otro sin error por encima de la capa física. Se ocupa del direccionamiento físico, de la topología de la red, del acceso a la red, de la notificación de errores, de la distribución ordenada de tramas y del control del flujo.

Puede decirse que ésta capa traslada los mensajes hacia y desde la capa física a la capa de red. Especifica como se organizan los datos cuando se transmiten en un medio particular.

Además del direccionamiento local, se ocupa de la detección y control de errores ocurridos en la capa física, del control del acceso a dicha capa, de la integridad de los datos y fiabilidad de la transmisión. Para esto, agrupa la información a transmitir en bloques, e incluye a cada uno una suma de control que permitirá al receptor comprobar su integridad. Los datagramas recibidos son comprobados por el receptor. Si algún datagrama se ha corrompido se envía un mensaje de control al remitente solicitando su reenvío.

La capa de enlace se divide en dos subcapas:

- **LLC (Control lógico de enlace):** define la forma en que los datos son transferidos sobre el medio físico, proporcionando servicio a las capas superiores.

- **MAC (Control de acceso al medio):** Esta subcapa actúa como controladora del hardware. Regula la utilización del medio físico para facilitar que varios equipos puedan competir simultáneamente por el manejo de un mismo medio de transporte.

2.2.1.1.3 Capa de Red.

La capa de red controla el funcionamiento de la subred que decide la ruta física que deben seguir los datos.

Se ocupa de la transmisión de los datagramas o paquetes y de encaminar cada uno en la dirección adecuada.

El rol principal de la capa de red es hacer que los datos lleguen desde el origen al destino, aún cuando ambos no estén conectados directamente. Es decir, que se encarga de encontrar una ruta manteniendo una tabla de enrutamiento y atravesando los equipos que sean necesarios, para hacer llegar los datos al destino. Los equipos encargados de realizar este encaminamiento se denominan encaminadores o routers,

Adicionalmente la capa de red debe gestionar la congestión de red, que es el fenómeno que se produce cuando una saturación de un nodo tira abajo toda la red.

La capa de red proporciona:

- Enrutamiento de marcos entre redes.
- Control de tráfico de subred.
- Traducción de direcciones o nombres lógicos en direcciones físicas.

La capa de red se subdivide en dos:

- **Transporte:** Encargada de encapsular los datos a transmitir. Utiliza los paquetes de datos. En esta categoría se encuentra el protocolo IP.
- **Conmutación:** Esta parte es la encargada de intercambiar información de conectividad específica de la red. Los routers son dispositivos que trabajan en este nivel y se benefician de estos paquetes de actualización de ruta.

Los protocolos más frecuentemente utilizados en esta capa son dos: X.25 e IP.

2.2.1.1.4 Capa de Transporte.

Es la capa encargada de efectuar el transporte de los datos libres de errores de la máquina origen a la máquina destino, independizándolo del tipo de red física que se esté utilizando.

Garantiza la fiabilidad del servicio, describe la calidad y naturaleza del envío de datos. Define cuando y como debe utilizarse la retransmisión para asegurar su llegada. Para ello divide el mensaje recibido de la capa de sesión en trozos o datagramas, los numera correlativamente y los entrega a la capa de red para su envío.

Durante la recepción, si la capa de red utiliza el protocolo IP, la capa de transporte es responsable de reordenar los paquetes recibidos fuera de secuencia. También puede funcionar en sentido inverso multiplexando una conexión de transporte entre diversas conexiones de datos. Este permite que los datos provenientes de diversas aplicaciones compartan el mismo flujo hacia la capa de red. Su función básica es aceptar los datos enviados por las capas superiores, dividirlos en pequeñas partes si es necesario, y pasarlos a la capa de red.

En ésta capa se proveen servicios de conexión para la capa de sesión que serán utilizados finalmente por los usuarios de la red al enviar y recibir paquetes. Estos servicios estarán asociados al tipo de comunicación empleada, la cual puede ser diferente según el requerimiento que se le haga a la capa de transporte.

La comunicación puede ser manejada para que los paquetes sean comunicación punto a punto libre de errores, o sin tener en cuenta el orden de envío. Una de las dos modalidades debe establecerse antes de comenzar la comunicación para que una sesión determinada envíe paquetes, y ése será el tipo de servicio brindado por la capa de transporte hasta que la sesión finalice.

El tamaño y la complejidad de un protocolo de transporte dependen del tipo de servicio que puede obtener de la capa de red. Si la capa de red no es confiable y si sólo admite datagramas, el protocolo de transporte debería incluir detección de errores amplios y recuperación.

Normalmente, el nivel de transporte puede aceptar mensajes relativamente grandes pero se encuentra la capa de tamaño de límites impuestos por la red (o inferiores) de mensaje estricto. Por consiguiente, el nivel de transporte debe dividir los mensajes en unidades o marcos más pequeños que anteponen un encabezado a cada marco.

La información de encabezado de capa de transporte debe incluir entonces información de control como inicio de mensaje y final de mensaje. Si las capas inferiores no mantienen secuencia, el encabezado de transporte además debe contener información de secuencia para permitir que obtenga conjuntamente de nuevo las piezas en el orden correcto antes de entregar el mensaje recibido hasta la capa anterior al nivel de transporte en el extremo receptor.

2.2.1.1.5 Capa de Sesión.

La capa de sesión permite el establecimiento de sesión entre procesos que se ejecutan en estaciones diferentes. Ofrece control de diálogo y sincronización.

Esta capa es la que se encarga de mantener el enlace entre las dos computadoras que estén trasmitiendo archivos.

Esta capa ofrece varios servicios:

- Control de la sesión a establecer entre el emisor y el receptor.
- Control de la concurrencia, es decir, que dos comunicaciones a la misma operación crítica no se efectúen al mismo tiempo.
- Mantener puntos de verificación, para que, ante una interrupción de transmisión por cualquier causa, la misma se

pueda reanudar desde el último punto de verificación en lugar de repetirla desde el principio.

Se asegura que, dada una sesión establecida entre dos máquinas, la misma se pueda efectuar para las operaciones definidas de principio a fin, reanudándolas en caso de interrupción. En muchos casos, los servicios de la capa de sesión son parcialmente, o incluso, totalmente prescindibles.

2.2.1.1.6 Capa de Presentación.

La capa de presentación da formato a los datos presentados a la capa de aplicación. Se ocupa de los aspectos semánticos de la comunicación, estableciendo los arreglos necesarios para que puedan comunicar máquinas que utilicen diversa representación interna para los datos. Describe como pueden transferirse números de coma flotante entre equipos que utilizan distintos formatos matemáticos.

En teoría esta capa presenta los datos a la capa de aplicación tomando los datos recibidos y transformándolos en formatos como texto, imágenes y sonido.

El objetivo principal es encargarse de la representación de la información, de manera que aunque distintos equipos puedan tener diferentes representaciones internas de caracteres, números, sonido o imágenes, los datos lleguen de manera reconocible.
La capa de presentación es la primera en trabajar más el contenido de la comunicación que cómo se establece la misma.
Dicha capa proporciona:

- Carácter de traducción de código.
- Conversión de datos.
- Compresión de datos: Reduce el número de bits que se tienen que transmitir en la red.

- Cifrado de datos: Cifra datos por motivos de seguridad, por ejemplo cifrado de contraseñas.

La capa de presentación es la encargada de manejar las estructuras de datos abstractas y realizar las conversiones de representación de datos necesarios para la correcta interpretación de los mismos.

2.2.1.1.7 Capa de Aplicación.

La capa de aplicación sirve como la ventana para que los usuarios y procesos de aplicación tengan acceso a servicios de red.

Ésta capa describe como hacen su trabajo los programas de aplicación como los navegadores, clientes de correo, terminales remotos, transferencia de ficheros. Esta capa implementa la operación con ficheros del sistema. Por un lado interactúan con la capa de presentación y por otro representan la interfaz con el usuario, entregándole la información y recibiendo los comandos que dirigen la comunicación.

Funciones:

- Redirección de recursos.
- Acceso de archivo remoto.
- Acceso de impresora remota.
- Administración de redes.
- Servicio de directorio.
- Mensajería electrónica.

Preguntas de Repaso.

1. Complete las oraciones de acuerdo a la capa del modelo OSI que corresponde.
   a) Proporciona transferencia de tramas de datos desde un nodo a otro. _____.
   b) Controla el funcionamiento de la subred que decide la ruta física que deben seguir los datos. _____.

c) Sirve como ventana para que los usuarios y procesos de aplicación tengan acceso a servicios de red. _____.
d) Se encarga de las conexiones físicas de la computadora hacia la red. _____.
e) Lleva los datos libres de errores de la maquina origen a la destino. _____,
f) Da formato a los datos presentados a la capa de aplicación. _____.
g) Mantiene el enlace entre las dos computadoras que estén transmitiendo archivos. _____.

2. ¿Capa que transmite la información en Bloques?

_____.

3. ¿Capa que divide el mensaje recibido en trozos o datagramas?

_____.

2.2.1.2 PROTOCOLO.

Es un conjunto estricto de reglas o procedimientos que se requieren para iniciar y mantener las comunicaciones.

Los protocolos de comunicación de datos son los que hacen posible el intercambio de información, después de establecer una llamada a través de un canal informativo.

El sistema de protocolos que fue desarrollado como producto de las primeras investigaciones realizadas por el Departamento de Defensa de los Estados Unidos, llego a conocerse como TCP/IP, después de que los dos protocolos iniciales fueron desarrollados: el protocolo de Control de Transmisión (TCP) y el protocolo Internet (IP).

2.2.1.3 PROTOCOLO TCP/IP.

2.2.1.3.1 IP.

El Protocolo de Internet (IP, de sus siglas en inglés Internet Protocol) es un protocolo no orientado a conexión usado tanto por el origen como por el destino para la comunicación de datos a través de una red de paquetes conmutados.

Los datos en una red basada en IP, son enviados en bloques conocidos como paquetes o datagramas (en el protocolo IP estos términos se suelen usar indistintamente). En particular, en IP no se necesita ninguna configuración antes de que un equipo intente enviar paquetes a otro con el que no se había comunicado antes.

El Protocolo de Internet provee un servicio de datagramas no fiable (también llamado del mejor esfuerzo (best effort), lo hará lo mejor posible pero garantizando poco). IP no provee ningún mecanismo para determinar si un paquete alcanza o no su destino y únicamente proporciona seguridad (mediante checksums o sumas de comprobación) de sus cabeceras y no de los datos transmitidos. Por ejemplo, al no garantizar nada sobre la recepción del paquete, éste podría llegar dañado, en otro orden con respecto a otros paquetes, duplicado o simplemente no llegar. Si se necesita fiabilidad, ésta es proporcionada por los protocolos de la capa de transporte, como TCP.

2.2.1.3.2 TCP.

El Protocolo de Control de Transmisión (TCP en sus siglas en inglés, Transmission Control Protocol que fue creado entre los años 1973 - 1974 por Vint Cerf y Robert Kahn) es uno de los protocolos fundamentales en Internet. Muchos programas dentro de una red de datos compuesta por computadoras pueden usar TCP para crear conexiones entre ellas a través de las cuales se envían datos.

El protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron. También proporciona un mecanismo para distinguir distintas aplicaciones dentro de una misma máquina, a través del concepto de puerto (computación).

TCP soporta muchas de las aplicaciones más populares de Internet, incluidas HTTP, SMTP y SSH.
El Protocolo de Control de Transmisión (TCP) es un protocolo de comunicación orientado a conexión y fiable del nivel de transporte, actualmente documentado por IETF RFC 793.

2.2.1.3.3 Dirección IP.

Una dirección IP es un número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo habitualmente una computadora dentro de una red que utilice el protocolo IP (Internet Protocol), que corresponde al nivel de red o nivel 3 del modelo de referencia OSI. Dicho número no se ha de confundir con la dirección MAC que es un número hexadecimal fijo que es asignado a la tarjeta o dispositivo de red por el fabricante, mientras que la dirección IP se puede cambiar.

Es habitual que un usuario que se conecta desde su hogar a Internet utilice una dirección IP. Esta dirección puede cambiar al reconectar; y a esta forma de asignación de dirección IP se denomina una dirección IP dinámica (normalmente se abrevia como IP dinámica).

Los sitios de Internet que por su naturaleza necesitan estar permanentemente conectados, generalmente tienen una dirección IP fija (se aplica la misma reducción por IP fija o IP estática), es decir, no cambia con el tiempo. Los servidores de correo, DNS, FTP públicos, y servidores de páginas Web necesariamente deben contar con una dirección IP fija o estática, ya que de esta forma se permite su localización en la red.

2.2.2 MODELO CLIENTE - SERVIDOR.

Es una arquitectura computacional que involucra procesos de cliente que se encuentran requiriendo servicios de procesos de un servidor.

Cliente/Servidor es el concepto computacional que viene a ser la extensión lógica de la programación modular, la cual asume fundamentalmente la separación de grandes piezas de software, en partes mas pequeñas llamadas *"módulos"*, creando la posibilidad de obtener un desarrollo más fácil y darle un mejor mantenimiento.

El proceso Cliente/Servidor reconoce que estos módulos no necesitan ser ejecutados dentro del mismo espacio de memoria, de tal manera que al utilizar esta arquitectura, el modulo que realiza la llamada se convierte en el *"cliente"* (que es quien hace la requisición de un servicio), y el modulo que es llamado se convierte en el *"servidor"* (que es el que provee el servicio).

Para aplicar dicho concepto, el siguiente paso será tener a clientes y servidores corriendo en el hardware, y bajo el software de la plataforma, apropiados para realizar sus funciones. Por ejemplo, servidores de manejo de sistemas de bases de datos, ejecutándose en plataformas especialmente diseñadas y configuradas para manejar requisiciones en forma de pregunta, o archivo de servidores corriendo en plataformas con elementos especiales para manejo de archivos

.

2.2.2.1 PROCESO CLIENTE.

El cliente es un proceso (programa) que envía un mensaje a un proceso servidor, requiriéndole a éste la realización de una tarea (servicio). El programa cliente usualmente maneja la parte de la aplicación que hace interfaz con el usuario, validando los datos introducidos por éste, enviando las requisiciones al programa servidor, y a veces ejecutando lógicamente las tareas.

2.2.2.2 PROCESO SERVIDOR.

Un proceso (programa) servidor satisface las requisiciones del cliente realizando la tarea solicitada. El programa servidor general

recibe las solicitudes desde el programa cliente, ejecuta las extracciones de información de las bases de datos, las actualiza, manejando la integridad de los datos, y envía respuestas a las interrogantes del cliente.

2.2.3 LENGUAJES DE PROGRAMACIÓN PARA LA WEB.

2.2.3.1 C/C++.

C es un lenguaje de programación creado en 1969 por Ken Thompson y Dennis M. Ritchie en los Laboratorios Bell como evolución del anterior lenguaje B, a su vez basado en BCPL. Al igual que B, es un lenguaje orientado a la implementación de Sistemas Operativos, concretamente Unix. C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones.

Los compiladores suelen ofrecer extensiones al lenguaje que posibilitan mezclar código en ensamblador con código C o acceder directamente a memoria o dispositivos periféricos.

La primera estandarización del lenguaje C fue en ANSI, con el estándar X3.159-1989. El lenguaje que define este estándar fue conocido vulgarmente como ANSI C. Posteriormente, en 1990, fue ratificado como estándar ISO (ISO/IEC 9899:1990). La adopción de este estándar es muy amplia por lo que, si los programas creados lo siguen, el código es portable entre plataformas y/o arquitecturas.

Los Sockets son una interfaz que permiten que las aplicaciones puedan acceder a los servicios que brinda el software TCP/IP, permitiendo la comunicación entre procesos en el mismo equipo o en equipos diferentes.

La Interfaz Socket proporciona funciones generalizadas que dan soporte a las comunicaciones en red empleando para ello muchos de los protocolos disponibles hoy en día. Los llamados sockets hacen referencia a todos los protocolos TCP/IP como una única

familia. Las llamadas permiten al programador especificar el tipo de servicio requerido, en vez del nombre de un protocolo especifico.
Los sockets pueden ser:

- Basados en la conexión (connection based) o Independientes de la conexión (connectionless).
- Basados en paquetes (packet based) o basados en flujos (streams based).
- Fiable (reliable) o inestable (unreliable).

Los Sockets se caracterizan por un dominio, un tipo y un protocolo de comunicación. El dominio de comunicación nos dice como se va a realizar la comunicación de los procesos que se van a intercomunicar, o sea, en que ambiente.

Si los procesos se comunicarán bajo la forma de un único sistema (tipo *root* de *Unix*), el dominio de comunicación será *AF_UNIX*, si los procesos se comunicarán como sistemas independientes y estos se hallan unidos mediante una red TCP/IP, el dominio de comunicación será *AFP_INET*.

Los sockets no se han diseñado solamente para TCP/IP. La idea original fue que se usase la misma interfaz también para distintas familias de protocolos.
Los tipos de Sockets más comunes son:

- Sockets Stream: hacen uso del protocolo TCP, el cual nos provee un flujo de datos bidireccional, secuenciado, sin duplicación de paquetes y libre de errores.

- Sockets Datagram: hacen uso del protocolo UDP, el cual nos provee un flujo de datos bidireccional, pero los paquetes pueden llegar fuera de secuencia, pueden no llegar o contener errores. Por lo tanto el proceso que recibe los datos debe realizar re-secuenciamiento, eliminar duplicados y asegurar la confiabilidad. Se llaman también sockets sin conexión, porque no hay que mantener una conexión activa, como en el caso de sockets

stream. Son utilizados para transferencia de información paquete por paquete. Ejemplo: dns, tftp, bootp, etc.

- Sockets SeqPacket: Establece una conexión fiable bidireccional con un tamaño de mensaje máximo definido. (Este tipo puede no estar habilitado en algunos sistemas).

- Sockets Raw: no son para el usuario común, son provistos principalmente para aquellos interesados en desarrollar nuevos protocolos de comunicación o para hacer uso de facilidades ocultas de un protocolo existente.

Uso de Sockets:
Los sockets basados en la conexión son cliente-servidor: el servidor espera por una conexión del cliente.
Los sockets Independientes de la conexión son de igual a igual (peer-to-peer): cada proceso es simétrico.

2.2.3.2 JAVA.

La tecnología Java se creó como una herramienta de programación en una pequeña operación secreta y anónima denominada “The Green Project” en Sun Microsystems en el año 1991.
El equipo (“Green Team”), compuesto por trece personas y dirigido por James Gosling, trabajó sin descanso durante 18 meses.
Intentaban anticiparse y prepararse para el futuro de la informática. Su conclusión inicial fue que al menos en parte, se tendría una tendencia hacia la convergencia de los dispositivos digitales y las computadoras.
Su resultado, fue un lenguaje de programación que no dependía de los dispositivos denominado “Oak”.

Tiempo después, Internet estaba listo para la tecnología Java, justo a tiempo para su presentación en público en 1995, el equipo pudo

anunciar que el navegador Netscape Navigator incorporaría la tecnología Java.

Actualmente, con más de 10 años de existencia, la plataforma Java ha atraído a muchos desarrolladores de software, se utiliza en los principales sectores de la industria de todo el mundo y está presente en un gran número de dispositivos, computadoras y redes de cualquier tecnología de programación.

La tecnología Java es una tecnología madura, extremadamente eficaz y versátil, que se ha convertido en un recurso valioso ya que permite a los desarrolladores:

- Desarrollar software en una plataforma y ejecutarlo en prácticamente cualquier otra plataforma.
- Crear programas para que funcionen en un navegador Web y en servicios Web.
- Desarrollar aplicaciones para servidores como foros en línea, tiendas, encuestas, procesamiento de formularios HTML, etc.
- Combinar aplicaciones o servicios basados en la tecnología Java para crear servicios o aplicaciones totalmente personalizados.
- Desarrollar potentes y eficientes aplicaciones para teléfonos móviles, procesadores remotos, productos de consumo de bajo coste y prácticamente cualquier dispositivo digital.

2.2.3.3 Scripts.

Se conoce como Script o lenguaje interpretado, a un lenguaje de programación que fue diseñado para ser ejecutado por medio de un intérprete, en contraste con los lenguajes compilados.

A diferencia de los lenguajes compilados, en los lenguajes interpretados el código no necesita ser preprocesado mediante un compilador, eso significa que la computadora es capaz de ejecutar la sucesión de instrucciones dadas por el programador sin necesidad de leer y traducir exhaustivamente todo el código.

Para que esto sea posible hace falta un intermediario, un programa encargado de traducir cada instrucción escrita con una semántica *"humana"* a código máquina (instrucciones que la computadora entiende), este programa recibe el nombre de intérprete (en inglés parser).
El intérprete se encarga de leer una a una las instrucciones textuales del programa conforme estas necesitan ser ejecutadas y descomponerlas en instrucciones del sistema, además se encarga de automatizar algunas de las tareas típicas de un programador como declaraciones de variables o dependencias, de esta manera el proceso de programar se suele agilizar mucho lo cual repercute en la eficiencia del que tiene que escribir el código.
La principal ventaja de un lenguaje interpretado es que es independiente de la máquina y del sistema operativo ya que no contiene instrucciones propias de un procesador sino que contiene llamadas a funciones que el intérprete deberá reconocer.

Basta que exista un intérprete de un lenguaje para dicho sistema y todos los programas escritos en ese lenguaje funcionaran.

Además un lenguaje interpretado permite modificar en tiempo de ejecución el código que se está ejecutando así como añadirle nuevo, algo que resulta idóneo cuando queremos hacer pequeñas modificaciones en una aplicación y no queremos tener que recompilarla toda cada vez.
Sin embargo, un lenguaje interpretado es mucho más lento que uno compilado ya que constantemente el código tiene que ser analizado y *"traducido"* a lenguaje máquina, sin embargo con el auge de los procesadores cada vez más rápidos esto ya no supone un problema.
Ejemplos de lenguajes interpretados en al actualidad son PHP, ASP, Perl, Python.

2.2.3.4 Perl.

Es un lenguaje programación que toma su nombre de Practical Extraction and Report Language y fue ideado por Larry Wall.
La primera versión se originó en el año de 1987.

Originalmente Larry Wall lo describe como un excelente lenguaje optimizado para leer archivos de texto, extraer información de esos archivos y crear reportes basados en esa información, combinando lo mejor de C, sed, awk y sh.

Perl se considera un lenguaje interpretado, es decir, no es necesaria una previa compilación para poder ejecutarse, lo único que se necesita es darle al interprete, Perl y el código que queremos que ejecute.

Cabe mencionar que Perl hoy en día es usado para una gran variedad de aplicaciones, desde avanzados programas de seguridad hasta sencillos CGI's para administrar formularios.

Uno de los iconos representativos de Perl, es la existencia de CPAN, un directorio de módulos que se pueden integrar a los scripts para facilitar su uso. Perl lleva a CPAN desde su nacimiento.

La filosofía comunitaria de Perl ha hecho de CPAN lo que es ahora, es el centro de distribución comunitario de paquetes que ningún otro lenguaje ha tenido.

Sin duda, CPAN representa todo el trabajo que la comunidad de Perl ofrece, y a su vez, es una excelente manera de contribuir en algo.

2.2.3.5 PHP.

Es un lenguaje de programación usado frecuentemente para la creación de contenido para sitios Web con los cuales se puede programar las páginas HTML y los códigos fuente.

PHP es un acrónimo recursivo que significa “PHP Hypertext Pre-processor” (inicialmente PHP Tools, o, *P*ersonal *H*ome *P*age Tools), y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web. Últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando las librerías Qt o GTK+.

El lenguaje PHP fue originalmente diseñado en Perl, seguidos por la escritura de un grupo de CGI binarios escritos en el lenguaje C por el programador danés-canadiense Rasmus Lerdorf en el año 1994 para mostrar su currículum vitae y guardar ciertos datos, como la cantidad de tráfico que su página Web recibía. El 8 de junio de 1995 fue publicado "Personal Home Page Tools" después de que Lerdorf lo combinara con su propio *Form Interpreter* para crear PHP/FI.

El fácil uso y la similitud con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores experimentados crear aplicaciones complejas, así como también, involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones y prácticas.
Debido al diseño de PHP, también es posible crear aplicaciones con una interfaz gráfica para el usuario (también llamada GUI), utilizando la extensión PHP-Qt o PHP-GTK. También puede ser usado desde la línea de órdenes, de la misma manera como Perl o Python pueden hacerlo, esta versión de PHP se llama PHP CLI (*Command Line Interface*).

Su interpretación y ejecución se da en el servidor Web, en el cual se encuentra almacenado el script, y el cliente sólo recibe el resultado de la ejecución. Cuando el cliente hace una petición al servidor para que le envíe una página Web, generada por un script PHP, el servidor ejecuta el intérprete de PHP, el cual procesa el script solicitado que generará el contenido de manera dinámica, pudiendo modificar el contenido a enviar, y regresa el resultado al servidor, el cual se encarga de regresárselo al cliente.

Además es posible utilizar PHP para generar archivos PDF, Flash, así como imágenes en diferentes formatos, entre otras cosas.
Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de aplicaciones Web muy robustas.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX (y de ese tipo, como Linux o MacOS X) y Windows, y puede interactuar con los servidores de Web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

El modelo PHP puede ser visto como una alternativa al sistema de Microsoft que utiliza ASP.NET/C#/VB.NET, a ColdFusion de la compañía Adobe (antes Macromedia), , a JSP/Java de Sun Microsystems, y a CGI/Perl. Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, existe además un IDE comercial llamado Zend Optimizer.

2.2.3.6 XML.

XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones.

Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información.
Entre las tecnologías XML disponibles se pueden destacar:

- *XSL* funciona como un lenguaje avanzado para crear hojas de estilos. Es capaz de transformar, ordenar y filtrar datos XML, y darles formato basándolo en sus valores.

- *XPath*: Lenguaje de Rutas XML, es un lenguaje para acceder a partes de un documento XML, como pueden ser sus atributos, elementos, etc.

- *XLinK*: Lenguaje de Enlace XML, es un lenguaje que permite insertar elementos en documentos XML para crear enlaces (hipervínculos) entre recursos XML.

- *XPointer:* Lenguaje de Direccionamiento XML, es un lenguaje que permite el acceso a la estructura interna de un documento XML, esto es, a sus elementos, atributos y contenido.

- *XQL*: Lenguaje de Consulta XML, es un lenguaje que facilita la extracción de datos desde documentos XML. Ofrece la posibilidad de realizar consultas flexibles para extraer datos de documentos XML en la Web.

2.2.3.7 ActiveX.

ActiveX es una tecnología de Microsoft para el desarrollo de páginas dinámicas. Tiene presencia en la programación del lado del servidor y del lado del cliente, aunque existan diferencias en el uso en cada uno de esos dos casos.

En el cliente:
Son pequeños programas que se pueden incluir dentro de páginas Web y sirven para realizar acciones de diversa índole. Por ejemplo hay controles ActiveX para mostrar un calendario, para implementar un sistema de FTP, etc.
Son un poco parecidos a los Applets de Java en su funcionamiento, aunque una diferencia fundamental es la seguridad, pues un Applet de Java no podrá tomar privilegios para realizar acciones malignas (como borrarnos el disco duro) y los controles ActiveX sí que pueden otorgarse permisos para hacer cualquier cosa.
Los controles ActiveX son particulares de Internet Explorer.

En el servidor:
También existen controles ActiveX del servidor y la gente que utiliza ASP los utiliza ya, aunque sea sin darse cuenta. Por ejemplo, cuando realizamos una conexión con una base de datos, estamos utilizando un control ActiveX del servidor.

Desarrollo de ActiveX.

Los controles ActiveX se desarrollan con entornos de Microsoft para la creación de aplicaciones Windows, como pueden ser Visual Basic Script o Visual C.

2.2.3.8 CGI.

Es un método que implementa la interfaz entre el Web y el manejador de la base de datos, consiste en que el proceso que es invocado por el cliente es ejecutado por el Servidor Web ya que existe, como un programa dentro de uno de sus directorios.

Una de las ventajas de esta tecnología de acceso es que permite escribir programas sencillos que accedan a la base de datos, en lenguajes populares como el C++ o Perl. De estos dos Perl, es el que está siendo más utilizado por su facilidad de aprendizaje y por que permite escribir “scripts” sencillos que se basan en variables de entorno para cambiar su funcionamiento de acuerdo al momento en que se ejecutan. Uno de los aspectos por los cuales esta tecnología está siendo utilizada cada vez menos es debido a que, teniendo que cargar un programa cada vez que el cliente lo demande, implica que el servidor pierda cierto tiempo teniendo que cargarlo y después de ejecutado el proceso debe descargarlo, además sería posible que un momento dado no existan suficientes recursos en la máquina que funcione como servidor debido a las condiciones cambiantes de la demanda dentro de la red, y que el programa simplemente no pueda ser ejecutado provocando esto un fallo de parte del servidor Web para poder responder a la petición con la información adecuada.

2.2.3.9 VBScript.

Visual Basic Script Edition es un lenguaje interpretado por el Windows Scripting Host de Microsoft. Su sintaxis refleja su origen como variación del lenguaje de programación Visual Basic. Ha logrado un apoyo significativo por parte de los administradores de Windows como herramienta de automatización, ya que, conjunta y paralelamente a las mejoras introducidas en los sistemas operativos Windows donde opera fundamentalmente, permite más margen de actuación y flexibilidad que el lenguaje *batch* (o de proceso por lotes) desarrollado a finales de los años 1970 para el MS-DOS.

El crecimiento del uso de las tecnologías de Internet ha supuesto un significativo avance para este lenguaje, dado que es parte fundamental de la ejecución de aplicaciones de servidor programadas en ASP (*Active Server Pages*), las cuales estuvieron en auge en el período 1997-2003, declinando actualmente en favor de tecnologías de código gestionado y máquinas virtuales, más seguras en la ejecución de procesos, y por tanto, más adaptadas para ejecuciones en entornos públicamente accesibles y distribuidos. Microsoft ha intentado competir mediante esta tecnología también en entornos de cliente, donde el lenguaje más utilizado es Javascript o su versión estandarizada ECMAScript, sin éxito. Actualmente Microsoft no ha puesto a disposición pública nuevas versiones del lenguaje, en favor de la tecnología .NET en la que se incluye el lenguaje hermano Visual Basic, dentro del entorno de ejecución de la plataforma .NET (CLR, o *Common Language Runtime*). Sin embargo sigue siendo muy útil en gestión de estaciones de trabajo y servidores en Windows.

VBScript es interpretado por el motor de *scripting* vbscript.dll, que puede ser invocado por el motor ASP asp.dll en un entorno Web, por wscript.exe en un entorno Windows de interfase gráfica y por cscript.exe es un entorno de línea de comandos. Cuando el código

fuente VBScript se guarda en ficheros independientes, éstos tienen típicamente la extensión *.vbs.*

Cuando se emplea en Internet Explorer, VBScript funciona de forma muy similar a JavaScript, procesando código contenido en el documento HTML. VBScript también puede usarse para crear aplicaciones HTML independientes (extensión *.hta*), que necesitan Internet Explorer 5.0 o superior para poder ser ejecutados. Los desarrolladores de aplicaciones en Web suelen preferir JavaScript debido a su mayor compatibilidad con otros navegadores de Internet, ya que VBScript sólo está disponible para el navegador de Microsoft Internet Explorer.

VBScript es el lenguaje usado para escribir algunos gusanos de red, como "*I Love You"*. La programación de éste tipo de robots Web, se debe a varias razones. Primero, el icono parecido a un pergamino que representa a los ficheros *.vbs* puede llevar a pensar a los usuarios inexpertos que se trata de un fichero de texto. Segundo, es fácil escribir un gusano informático en VBScript que se propague por correo electrónico (se necesitan pocas líneas de código).

2.2.3.10 JavaScript.

JavaScript permite crear aplicaciones específicamente orientadas a su funcionamiento en la red Internet. Usando JavaScript, se pueden crear páginas HTML dinámicas que procesen la entrada del usuario y que sean capaces de gestionar datos persistentes usando objetos especiales, archivos y bases de datos relacionales.
Con JavaScript se pueden construir aplicaciones que varían desde la gestión de la información corporativa interna y su publicación en Intranets hasta la gestión masiva de transacciones de comercio electrónico.
Aunque Javascript de cliente y de servidor comparten el mismo conjunto base de funciones y características, en algunos casos se

utilizan de distinta forma. Los componentes de JavaScript son los siguientes:

- Núcleo de JavaScript (Core JavaScript).
- JavaScript para Cliente.
- JavaScript para Servidor.

JavaScript para cliente engloba el núcleo del lenguaje y algunos elementos adicionales como, por ejemplo, una serie de objetos predefinidos que sólo son relevantes para la ejecución de JavaScript en el contexto de un cliente Web.

Así mismo, JavaScript para servidor incluye también el núcleo de lenguaje y los objetos predefinidos y funciones necesarias para el correcto funcionamiento en el marco de un servidor.

El código JavaScript para cliente se integra directamente en páginas HTML y es interpretado en su totalidad por el cliente Web en tiempo de ejecución.
Puesto que con frecuencia es necesario ofrecer el mayor rendimiento posible, las aplicaciones JavaScript desarrolladas para servidores se pueden compilar antes de instalarlas en dichos servidores.

2.2.3.11 JScript.

JScript es la implementación de Microsoft de ECMAScript.

Está disponible mediante Internet Explorer y el Windows Scripting Host. La versión más reciente es JScript .NET, que está basado en la versión 4 del estándar ECMAScript, y puede ser compilado para la plataforma Microsoft .NET.

JScript no es lo mismo que JavaScript. Este último es el estándar (también se le llamó ECMAScript), mientras que el primero es propiedad de Microsoft.

Script.NET es un lenguaje de secuencias de comandos moderno con una gran variedad de aplicaciones. Es un auténtico lenguaje orientado a objetos y aún mantiene su espíritu de *"secuencias de comandos".* JScript .NET mantiene compatibilidad total con las versiones anteriores de JScript.

Preguntas de repaso.

1. ¿Que es protocolo?

_____

_____.

2. ¿Que es dirección IP?

_____

_____.

3. Dentro del modelo Cliente/Servidor defina:

Proceso cliente: _____

_____

_____.

Proceso servidor: _____

_____

_____.

4. ¿Nombre del lenguaje de programación que se describe como optimizado para leer archivos de texto, extraer información y crear reportes de los mismos?

_____

5. Defina el concepto de script.

_____

_____.

6. ¿Mencione uno de los dos tipos de aplicaciones, la cuales, PHP es frecuentemente usado en su creación?

   _____

   _____.

2.2.4 FUNDAMENTOS DE INTERNET.

2.2.4.1 SISTEMAS DE NOMBRES DE DOMINIOS (DNS).

El sistema DNS es en esencia una base de datos distribuida. Esta base de datos es jerárquica, al estilo de como lo son los sistemas de ficheros de UNIX. La raíz de la base de datos está representada por el nodo *"."* y cada uno de los nodos que descienden de ella reciben el nombre de dominios.

Un dominio es una colección de nodos relacionados de alguna forma porque estén en la misma red, tal como los nodos de una universidad. Por ejemplo, las universidades americanas se agrupan en el dominio *edu*. Cada universidad tiene allí un subdominio, tal como la Universidad Groucho Marx (*groucho.edu*).

A su vez, podemos encontrar nuevos subdominios dentro, como el Departamento de Matemáticas *(maths).* Finalmente, un nodo de ese departamento llamado *"erdos"* tendrá un nombre completo (conocido como totalmente cualificado) tal como *erdos.maths.groucho.edu.* Este nombre totalmente cualificado también se conoce por las siglas FQDN.

En la Figura 2.2.2 vemos una parte del espacio de nombres. La raíz del árbol, que se identifica con un punto sencillo, es lo que se denomina dominio raíz y es el origen de todos los dominios. Para indicar que un nombre es FQDN, a veces se termina su escritura en un punto. Este punto significa que el último componente del nombre es el dominio raíz.

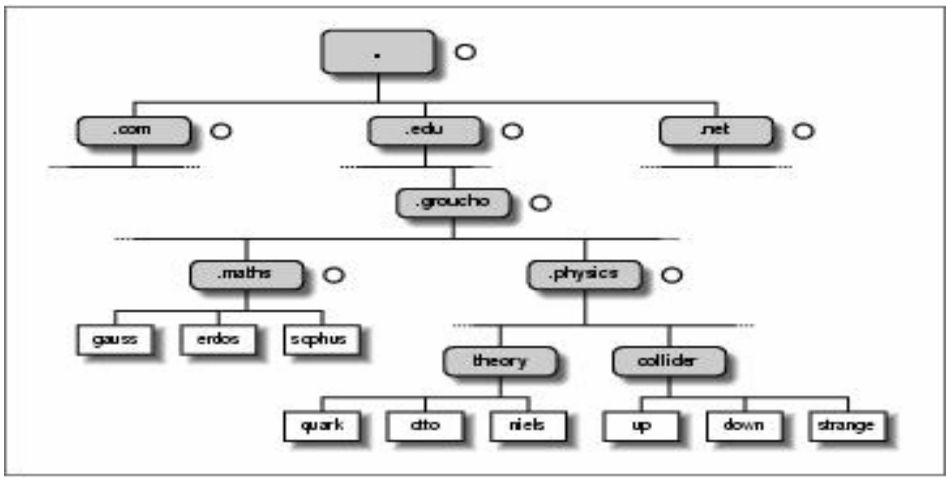


Figura 2.2.2 Una parte del espacio de nombres de dominios.

Dependiendo de su localización en la jerarquía, un dominio puede ser de primer nivel (top-level), segundo nivel o tercer nivel.

Los que siguen son los dominios de primer nivel que veremos con frecuencia:

Dominio	Descripción
Edu	Instituciones universitarias
Com	Organizaciones comerciales.
Org	Organizaciones no comerciales. Las redes privadas UUCP suelen estar en este dominio.
Net	Pasarelas y otras redes administrativas.
Mil	El ejército norteamericano.
Gov	El gobierno norteamericano.
Uucp	Dominio para redes UUCP.

Tabla 2.2.1 Dominios de primer nivel.

Inicialmente los cuatro primeros dominios de la lista anterior pertenecían solo a los Estados Unidos, sin embargo, los cambios de política posteriores han hecho que estos dominios, llamados de dominios globales primer nivel (gTLD) sean realmente globales. Además se están negociando nuevos dominios de primer nivel.

Fuera de los Estados Unidos, cada país suele tener su propio dominio de primer nivel codificado con las dos letras del país

definidas en la tabla ISO-3166. Finlandia, por ejemplo, usa el dominio *fi*; en España se usa el dominio *es*; en México se usa *mx*; en Argentina, *ar*, etc. Por debajo de cada dominio de primer nivel, cada país los organiza según se estime conveniente. Algunos crean a segundo nivel una serie de dominios similares a los gTLD.
Por ejemplo, en Argentina encontramos los dominios *com.ar* para las empresas, y *org.ar* para las organizaciones sin ánimo de lucro.
Otros países, como España, ponen directamente como nombres de segundo nivel las instituciones o empresas que los solicitan.
Por ejemplo, tenemos *hispalinux.es.*

Por supuesto, el hecho de que un nombre esté en uno de estos dominios nacionales, no implica que la máquina esté realmente en ese país; significa simplemente que ha sido registrada en el NIC de ese país. Un fabricante sueco puede tener oficinas en Australia y tener sus computadoras de ése país registrados en el dominio sueco, que es "*se"*.

2.2.4.2 URI ó URL.

URL significa *Uniform Resource Locator*, es decir, localizador uniforme de recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

Las URL fueron una innovación fundamental en la historia del *Internet*. Fueron usadas por primera vez por Tim Berners-Lee en 1991, para permitir a los autores de documentos establecer hiperenlaces en la World Wide Web. Desde 1994, en los estándares del *Internet*, el concepto de URL ha sido incorporado dentro del más general de URI (*Uniform Resource Identifier* - Identificador Uniforme de Recurso), pero el término URL aún se utiliza ampliamente.
Aunque nunca fueron mencionadas como tal en ningún estándar, mucha gente cree que las iniciales URL significan Universal Resource Locator (Localizador Universal de Recurso). Esta

interpretación puede ser debida al hecho de que, aunque la U en URL siempre ha significado Uniforme, la U de URI significó en un principio Universal, antes de la publicación del RFC 2396.

El URL es la cadena de caracteres con la cual se asigna una dirección única a cada uno de los recursos de información disponibles en el *Internet*. Existe un URL único para cada página de cada uno de los documentos de la *World Wide Web*, para todos los elementos de *Gopher* y todos los grupos de debate *USENET*, y así sucesivamente.

El URL de un recurso de información es su dirección en Internet, la cual permite que el navegador la encuentre y la muestre de forma adecuada. Por ello el URL combina el nombre de la computadora que proporciona la información, el directorio donde se encuentra, el nombre del fichero y el protocolo a usar para recuperar los datos.
El formato general de un URL es:
Protocolo://máquina/directorio/fichero
También pueden añadirse otros datos:
protocolo://usuario:contraseña@máquina:puerto/directorio/fichero

Por ejemplo: *http://google.com.sv/*
La especificación detallada se encuentra en la RFC 1738, titulada *Uniform Resource Locators*.

Una URL se clasifica por su esquema, que generalmente indica el protocolo de red que se usa para recuperar, a través de la red, la información del recurso identificado. Una URL comienza con el nombre de su esquema, seguida por dos puntos, seguido por una *parte específica del esquema*.

Algunos ejemplos de esquemas URL:

Esquema.	Descripción de Esquema.
http	Recursos HTTP.
HTTPS	http sobre SSL.
FTP	File Transfer Protocol.
Malito	Direcciones e-mail.
Idap	Búsquedas LDAP (Lightweigth Directory Access Protocol).
File	Recursos disponibles en la computadora local o en una red local.
News	Grupo de noticias Usenet (newsgroup).
Gopher	Protocolo Ghoper (ya en desuso).
Telnet	Protocolo Telnet.
Data	El esquema para insertar pequeños trozos de contenido en los documentos

Tabla 2.2.2 Esquemas URL.

Algunos de los esquemas URL, como los populares “malito”, “http”, “ftp” y “file”, junto a los de sintaxis general URL, se detallaron por primera vez en 1994, en el Request for Comments RFC 1630, sustituido un año después por los más específicos RFC 1738 y RFC 1808.

Algunos de los esquemas definidos en el primer RFC aun son válidos, mientras que otros son debatidos o han sido refinados por estándares posteriores. Mientras tanto, la definición de la sintaxis general de las URL’s se ha dividido en dos líneas separadas de especificación de URI: RFC 2396 (1998) y RFC 2732 (1999), ambos ya obsoletos pero todavía ampliamente referidos en las definiciones de esquemas URL.

El estándar actual es STD 66 / RFC 3986 (2005).

2.2.4.2.1 Referencias URI.

El término referencia URI se refiere a una caso particular de una URI, o una porción de éste, tal como es usada en un documento HTML, por ejemplo, para referirse a un recurso particular. Una referencia URI habitualmente se parece a una URL o a la parte final de una URL. Las referencias URI introducen dos nuevos conceptos: la distinción entre referencias *absolutas* y *relativas*, y el concepto de un identificador de fragmento.

Un URL absoluto es una referencia URI que es parecida a las URL's definidas arriba; empieza por un esquema seguido de dos puntos (*":"*) y de una parte específica del esquema.

Una URL relativa es una referencia URI que comprende sólo la parte específica del esquema de una URL, o de algún componente de seguimiento de aquella parte. El esquema y componentes principales se infieren del contexto en el cual aparece la referencia URL: el URI base (o URL base) del documento que contiene la referencia.
Una referencia URI también puede estar seguida de un carácter de numeral (*"#"*) y un puntero dentro del recurso referenciado por el URI en conjunto. Esto no es parte de la URI como tal, sino que es pensado para que el *"agente usuario"* (el navegador) lo interprete después que una representación del recurso ha sido recuperada. Por tanto, no se supone que sean enviadas al servidor en forma de solicitudes HTTP.
Ejemplos de URL's absolutos:
http://es.wikipedia.org/w/wiki.phtml?title=URL&action=history
http://es.wikipedia.org/wiki/URL#Esquemas_en_URL

Ejemplos de URL's relativos:
//en.wikipedia.org/wiki/Uniform_Resource_Locator
/wiki/URL
URL#Referencias_URI

Diferenciación entre mayúsculas/minúsculas.

De acuerdo al estándar actual, en los componentes esquema y anfitrión no se diferencian mayúsculas y minúsculas, y cuando se normalizan durante el procesamiento, deben estar en minúsculas.
Se debe asumir que en otros componentes sí hay diferenciación.

Sin embargo, en la práctica, en otros componentes aparte de los de protocolo y anfitrión, esta diferenciación es dependiente del servidor Web y del sistema operativo del sistema que albergue al servidor.

2.2.4.3 HTML.

Son las siglas de HyperText Markup Language, Lenguaje marcador de hipertexto. Está basado en el SGML (Standard Generalizaed Markup Language que significa, Lenguaje marcador estándar generalizado), mismo que se utiliza para delinear la estructura general de varios tipos de documentos. La atención del HTML se concentra en el contenido del documento, no en su apariencia.
Los archivos que utiliza como fuente son simples archivos de texto ASCII (American Standar Code for Information Interchange), de tal manera que para crearlos se utilizará cualquier editor de texto. Dichos archivos podrán funcionar adecuadamente en todos los sistemas computacionales.

Para crear archivos fuente de HTML, se usa cualquier editor de texto, con las siguientes indicaciones:

- Colocar las extensiones *".html"* o *".htm"*
- Emplear un editor de texto simple como: VI, Edit, Notepad.

Para poder visualizar el archivo HTML, pueden utilizarse los navegadores: FireFox Mozilla, Opera, Internet Explorer, entre otros.
HTML posee las siguientes características:

- Los documentos en HTML, son simples archivos de texto plano.

- No es necesario incluir información referente al formato ni a las fuentes, ya que esto disminuiría la velocidad y aumentaría, en consecuencia, el tiempo para que el documento fuera cargado y desplegado en pantalla, este trabajo es realizado por el navegador.

- Los documentos en HTML son independientes de los dispositivos. Es decir, que se despliegan en cualquier plataforma; todo lo que se necesita es un navegador capaz de interpretar el HTML, no importando el sistema operativo en que se trabajé.

2.2.4.4 HTTP (Hypertext Transfer Protocol).

Este protocolo ha estado en uso en el World Wide Web desde 1990, presentándose como un protocolo genérico orientado a objetos, que puede ser usado para distintas tareas tales como servidores de aplicaciones y sistemas de control de distribución de información, a través de sus listas de extensión. Una característica de HTTP es que permite al sistema cargarse independientemente de que los datos se estén transfiriendo.

El propósito del HTTP es que los sistemas de información sean más funcionales que simplemente dar una respuesta a un requerimiento hecho por el usuario, sino que también deberán incluir búsquedas, anotaciones y actualizaciones continuas.

En el Internet, la comunicación se lleva a cabo gracias a la conexión que realiza el TCP/IP, pero esto no le debe permitir dirigir a cualquier otro protocolo en el Internet o en otras redes, de tal manera que la estructura de búsqueda establecida por HTTP para analizar y responder una solicitud, y luego transportar las unidades de datos, no puede ser dominada por el TCP/IP.

El HTTP es básicamente estable, y la transmisión que realiza se divide en los pasos siguientes:

- Conexión: que es establecida desde el cliente hacia el servidor.
- Solicitud: que es enviada por el cliente y consiste en un mensaje de solicitud al servidor.
- Respuesta: enviada por el servidor hacia el cliente, y es una respuesta a la solicitud de éste.
- Cierre: es el cierre o finalización de la conexión, tanto por parte del cliente como del servidor.

El formato de las partes de requisición y respuesta es definido por el HTTP, mientras que la información de cabecera definida en especificación es enviada en caracteres latinos ISO (International Standards Organization), y la transmisión de objetos es realizada, si es posible, en forma binaria.

2.2.4.5 FTP.

FTP (*File Transfer Protocol*) es un protocolo de transferencia de archivos entre sistemas conectados a una red TCP basado en la arquitectura cliente-servidor, de manera que desde un equipo cliente nos podemos conectar a un servidor para descargar archivos desde él o para enviarle nuestros propios archivos independientemente del sistema operativo utilizado en cada equipo.
El servicio FTP es ofrecido por la capa de aplicación del modelo OSI al usuario, utilizando normalmente el puerto de red 20 y el 21.

Un problema básico de FTP es que está pensado para ofrecer la máxima velocidad en la conexión, pero no la máxima seguridad, ya que todo el intercambio de información, desde el *login* y *password* del usuario en el servidor hasta la transferencia de cualquier archivo, se realiza en texto plano sin ningún tipo de cifrado, con lo que un

posible atacante lo tiene muy fácil para capturar este tráfico, acceder al servidor, o apropiarse de los archivos transferidos.
Para solucionar este problema son de gran utilidad aplicaciones como scp y sftp, incluidas en el paquete SSH, que permiten transferir archivos pero cifrando todo el tráfico.

El siguiente modelo representa el diagrama de un servicio FTP.

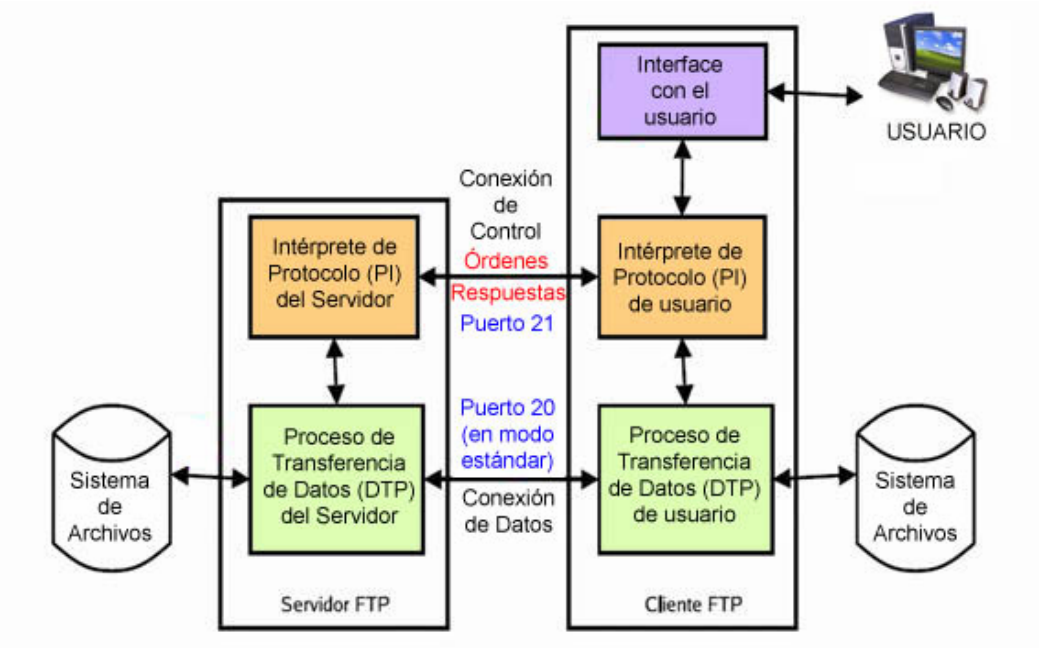


Figura 2.2.3 Esquema FTP.

En el modelo, el intérprete de protocolo (PI) de usuario, inicia la conexión de control en el puerto 21. Las órdenes FTP estándar las genera el PI de usuario y se transmiten al proceso servidor a través de la conexión de control. Las respuestas estándar se envían desde el PI del servidor al PI de usuario por la conexión de control como respuesta a las órdenes.

Estas órdenes FTP especifican parámetros para la conexión de datos (puerto de datos, modo de transferencia, tipo de representación y estructura) y la naturaleza de la operación sobre el sistema de archivos (almacenar, recuperar, añadir, borrar, etc.).
El proceso de transferencia de datos (DTP) de usuario u otro proceso en su lugar, debe esperar a que el servidor inicie la

conexión al puerto de datos especificado (puerto 20 en modo activo o estándar) y transferir los datos en función de los parámetros que se hayan especificado.

Vemos también en el diagrama que la comunicación entre cliente y servidor es independiente del sistema de archivos utilizado en cada computadora, de manera que no importa que sus sistemas operativos sean distintos, porque las entidades que se comunican entre sí son los PI y los DTP, que usan el mismo protocolo estandarizado: el FTP.
También hay que destacar que la conexión de datos es bidireccional, es decir, se puede usar simultáneamente para enviar y para recibir, y no tiene por qué existir todo el tiempo que dura la conexión FTP.

2.2.4.5.1 Servidor FTP.

Un servidor FTP es un programa especial que se ejecuta en un equipo servidor normalmente conectado a Internet (aunque puede estar conectado a otros tipos de redes, LAN, MAN, etc.). Su función es permitir el intercambio de datos entre diferentes servidores/computadoras.

Por lo general, los programas servidores FTP no suelen encontrarse en computadoras personales, por lo que un usuario normalmente utilizará el FTP para conectarse remotamente a uno y así intercambiar información con él.

Las aplicaciones más comunes de los servidores FTP suelen ser el alojamiento Web, en el que sus clientes utilizan el servicio para subir sus páginas Web y sus archivos correspondientes; o como servidor de backup (copia de seguridad) de los archivos importantes que pueda tener una empresa. Para ello, existen protocolos de comunicación FTP para que los datos se transmitan cifrados, como el SFTP (*Secure File Transfer Protocol*).

2.2.4.5.2 Cliente FTP.

Cuando un navegador no está equipado con la función FTP, o si se quiere cargar archivos en una computadora remota, se necesitará utilizar un programa cliente FTP. Un cliente FTP es un programa que se instala en la computadora del usuario, y que emplea el protocolo FTP para conectarse a un servidor FTP y transferir archivos, ya sea para descargarlos o para subirlos.

Para utilizar un cliente FTP, se necesita conocer el nombre del archivo, la computadora en que reside (servidor, en el caso de descarga de archivos), el terminal al que se quiere transferir el archivo (en caso de querer subirlo nosotros al servidor), y la carpeta en la que se encuentra.

Algunos clientes de FTP básicos en modo consola vienen integrados en los sistemas operativos, incluyendo Windows, DOS, Linux y Unix. Sin embargo, hay disponibles clientes con opciones añadidas e interfaz gráfica. Aunque muchos navegadores tienen ya integrado FTP, es más confiable a la hora de conectarse con servidores FTP no anónimos utilizar un programa cliente.

2.2.4.5.3 Acceso Anónimo.

Los servidores FTP anónimos ofrecen sus servicios libremente a todos los usuarios, permiten acceder a sus archivos sin necesidad de tener un *"USERID"* o una cuenta de usuario. Es la manera más cómoda fuera del servicio Web de permitir que todo el mundo tenga acceso a cierta información sin que para ello el administrador de un sistema tenga que crear una cuenta para cada usuario.

Si un servidor posee servicio *"FTP anonymous*" solamente con teclear la palabra *"anonymous",,* cuando pregunte por tu usuario tendrás acceso a ese sistema. No se necesita ninguna contraseña preestablecida, aunque tendrás que introducir una sólo para ese momento, normalmente se suele utilizar la dirección de correo electrónico propia.

Solamente con eso se consigue acceso a los archivos del FTP, aunque con menos privilegios que un usuario normal. Normalmente

solo podrás leer y copiar los archivos existentes, pero no modificarlos ni crear otros nuevos.
Normalmente, se utiliza un servidor FTP anónimo para depositar grandes archivos que no tienen utilidad si no son transferidos a la máquina del usuario, como por ejemplo programas, y se reservan los servidores de páginas Web (HTTP) para almacenar información textual destinada a la lectura en línea.

2.2.4.5.4 Acceso de Usuario.

Si se desean tener privilegios de acceso a cualquier parte del sistema de archivos del servidor FTP, de modificación de archivos existentes, y de posibilidad de subir nuestros propios archivos, generalmente se suele realizar mediante una cuenta de usuario.

En el servidor se guarda la información de las distintas cuentas de usuario que pueden acceder a él, de manera que para iniciar una sesión FTP debemos introducir un *login* y un *password* que nos identifica unívocamente.

2.2.4.5.5 Acceso de Invitado.

El acceso sin restricciones al servidor que proporcionan las cuentas de usuario implica problemas de seguridad, lo que ha dado lugar a un tercer tipo de acceso FTP denominado invitado *(guest),* que se puede contemplar como una mezcla de los dos anteriores.
La idea de este mecanismo es la siguiente: se trata de permitir que cada usuario conecte a la máquina mediante su *login* y su *password,* pero evitando que tenga acceso a partes del sistema de archivos que no necesita para realizar su trabajo, de esta forma accederá a un entorno restringido, algo muy similar a lo que sucede en los accesos anónimos, pero con más privilegios.

Preguntas de repaso.

1. ¿Mencione 3 tipos de dominios de primer nivel?

   _____

2. ¿Que es URL?

   _____

   _____

   _____

3. ¿Protocolo orientado a objetos usado para distintas tareas como servidores de aplicaciones y sistemas de control de distribución de información?

   _____

4. ¿Que es FTP?

   _____

   _____

   _____

5. ¿Cual es el nombre de la capa de modelo OSI que ofrece el servicio FTP?

   _____

Auto evaluación.

Instrucciones.

En la hoja provista, subraye la respuesta correcta. Coteje las respuestas al final de la auto evaluación.

1. Se considera un lenguaje interpretado, es decir, no es necesario una previa compilación para poder ejecutarse.
    a. C++.
    b. Java.
    c. Perl.

2. Es uno de los iconos representativos de Perl.
    a. CPAN.
    b. CGI.
    c. VBScript.

3. Una ventaja de lenguaje interpretado es.
    a. Se necesita intérprete.
    b. Es independiente de la maquina y del sistema operativo.
    c. Se traduce la instrucción de semántica humana a código maquina.

4. Servidor de base de datos al cual PHP permite realizar una conexión
    a. MySQL.
    b. Access.
    c. DBMS.

5. Cual es la diferencia entre HTML y XML
    a. HTML escribe etiquetas y XML escribe código.
    b. HTML es simple y XML es complejo.
    c. XML describe datos y HTML se concentra en mostrar datos.

6. ¿Cual es la función de un servidor FTP?
    a. Conectarse a una computadora.
    b. Permitir el intercambio de datos entre diferentes servidores/computadoras.
    c. Escribir datos en una computadora.

7. Mencione una de las características de los documentos HTML.
    a. Los documentos en HTML son independientes de los dispositivos, es decir, que se despliegan en cualquier plataforma.
    b. Los documentos en HTML, son simples archivos de texto plano.
    c. A y B son correctas.

Contestaciones de las preguntas de las páginas anteriores.

1. C, 2. A, 3. B, 4. A, 5. C, 6. B, 7. C

De 7 a 5 preguntas correctas: Puede continuar con el capitulo 3.
Menos de 5 preguntas: Repase el capitulo 2 y compruebe el ejercicio.

Investigación Complementaria.

- Explique con sus propias palabras el modelo OSI.
- Investigue más acerca de las instrucciones Perl y PHP.
- Investigue sobre la instalación de módulos Perl a través de CPAN.

2.3: SISTEMAS DE BÚSQUEDA EN INTERNET (ROBOT SPIDER).

Tiempo recomendado de estudio: 5 horas.

Objetivos:

Al concluir este capitulo, el lector estará capacitado para:

- Definir lo que es un Sistema de Búsqueda en la Web.
- Diferenciar entre un Directorio en la Web y un Motor de Búsqueda.
- Clasificar los sistemas de búsqueda de acuerdo a su funcionamiento.
- Comprender el funcionamiento de los Motores de Búsqueda.

Introducción:

Para realizar búsquedas en Internet, se utiliza el uso de buscadores para efectuar investigaciones de algún tema específico, pero desconocemos como realmente, estas “paginas de Internet” realizan las búsquedas que solicitamos.

Este capítulo contiene una descripción de cómo están estructurados y clasificados estos sistemas de búsqueda, ya que estos pueden ser Directorios, Motores de Búsqueda o Metabuscadores.

Se abordaran temas como el papel que tienen los robots en los motores de búsqueda, como esta estructurada la base de datos de estos sistemas de búsqueda.

2.3.1 DEFINICIÓN DE SISTEMAS DE BÚSQUEDA.

Un sistema de búsqueda es un conjunto de programas en Internet, cuya finalidad es la de proporcionar información y/o referencias de información sobre uno o varios tópicos a usuarios de la Web a petición de éstos. Se debe poseer una base de datos y procurar mantenerla actualizada.

2.3.2 ESTRUCTURA DE LOS SISTEMAS DE BÚSQUEDA.

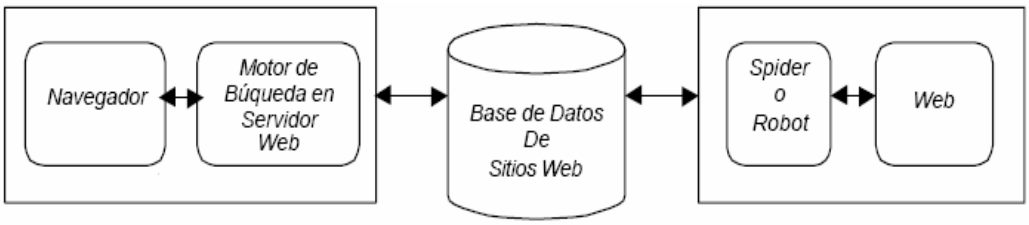


Figura 2.3.1 Estructura de un motor de búsqueda.

Un sistema de búsqueda en la Web se compone de 3 partes principales:

- **Robot (o crawler):** Es el encargado de recorrer la estructura de vínculos de la *Web* a través de listas de *url's* que se usan como punto de partida para el recorrido recursivo de los documentos.

- **Base de datos:** Almacena los datos que recolecta el robot dentro de una estructura ordenada para que sea posible accederla rápidamente para generar la salida de información necesaria para responder las consultas. La estructura, datos y forma que maneja éste componente varía en gran medida de acuerdo a la implementación específica del motor de búsqueda.

- **Aplicación Cliente/Servidor:** Se utilizan para manejar para la atención de las solicitudes de búsqueda, las cuales son responsabilidad del servidor Web.
    - **Servidor Web**: Este componente es el encargado de establecer la comunicación con el usuario a través del protocolo HTTP, recibiendo consultas de éste y enviándole el resultado de las mismas a través del mismo protocolo. En cuanto a la comunicación con los módulos internos del sistema, el Servidor Web envía los datos de la consulta

del usuario hacia el generador de consultas, y por otro lado recibirá los datos desde el generador de presentación.

- **Generador de consultas**: Recibe la consulta realizada en forma de una cadena de texto ingresada por el usuario conteniendo las palabras clave, frases u otros atributos. Una vez que obtuvo estos datos a través del servidor *Web*, los interpreta y genera una consulta nueva en formato nativo para que pueda ser efectuada la búsqueda dentro del la base de datos.

- **Buscador**: Será el encargado de acceder a la estructura interna de datos (repositorios índices, etc.), para satisfacer la petición de datos.

- **Generador de Presentación**: Construye la vista a ser incluida en el documento HTML que recibirá el usuario a través del servidor Web, a partir de los datos que generó el buscador.

2.3.3 ARQUITECTURA DE LOS SISTEMAS DE BÚSQUEDA.

2.3.3.1 ARQUITECTURA CENTRALIZADA.

La mayoría de los motores de búsqueda (como es el caso de Altavista y Google) utiliza una arquitectura centralizada (también llamada crawler-indexer) esta basada en los siguientes elementos:

- **Crawler (robot, *spider, wanderer, walker, knowbot*):** Son programas *(agentes software)* que recorren la Web mediante peticiones a servidores trayendo nuevas (o modificadas) páginas a indexar. Se ejecutan en un sistema local y envían las peticiones a servidores remotos.
- **Indexador:** Mantiene un índice con las paginas que trae el crawler. Responde a consultas realizadas desde diferentes ubicaciones en la Web.
- **Maquina de búsqueda:** Realiza las búsquedas en el índice devolviendo las URL's ordenadas.

- **Interfaz:** Recibe la consulta y muestra los resultados. Permite retroalimentación.

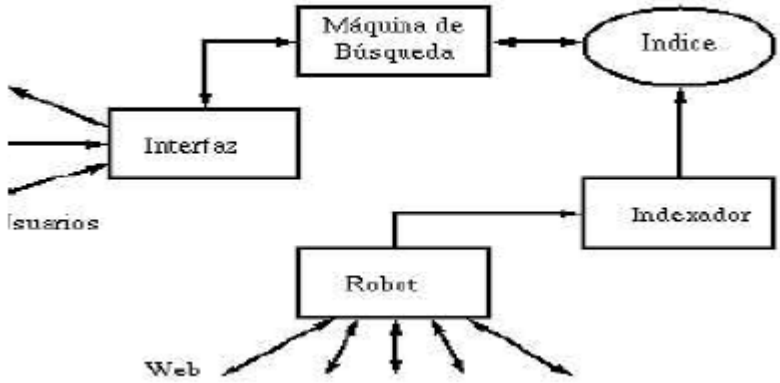


Figura 2.3.2 Componentes de una arquitectura centralizada.

2.3.3.2 ARQUITECTURA DISTRIBUIDA.

Existen algunas variantes de la arquitectura anterior. Entre ellas, la más importante es la arquitectura *Harvest.*

Harvest utiliza una arquitectura distribuida para unir y distribuir datos y resulta más eficiente que la arquitectura centralizada. El principal problema esta en que *Harvest* precisa la coordinación de varios servidores Web.

La aproximación distribuida *Harvest* pretende evitar algunos de los problemas de la arquitectura crawler-indexer anterior, como son:

- Los servidores Web se saturan con las peticiones recibidas desde diferentes crawlers.
- El tráfico Web aumenta debido a que los crawlers recuperan objetos enteros, para luego descartar casi todo su contenido.
- Cada crawler recolecta información de un modo independiente, sin que exista cooperación entre los diferentes motores de búsqueda.

Para resolver estos problemas, *Harvest* introduce dos elementos principales: *gatherers* y *brokers:*

Un *gatherer* recopila páginas de determinados servidores Web. Extrae la información a indexar periódicamente.

Los *brokers* proporcionan el mecanismo de indexación y la interfaz de consulta de los datos recuperados. Recuperan información de determinados *gatherers* y otros *brokers*, actualizando incrementalmente sus índices.
Dependiendo de la configuración de los *gatherers* y *brokers*, se pueden tratar diferentes aspectos relacionados con la carga del servidor y el tráfico de la red.
Por ejemplo, un *gatherer* puede ejecutarse en un servidor Web y no generar trafico externo. También podría enviar la misma información a varios *brokers*, evitando repetir la recolección. Por otro lado, los *brokers* pueden construir índices y enviarlos a otros *brokers*, que mezclarían varios. Este diseño permite compartir el trabajo y evita transmitir tanta información.

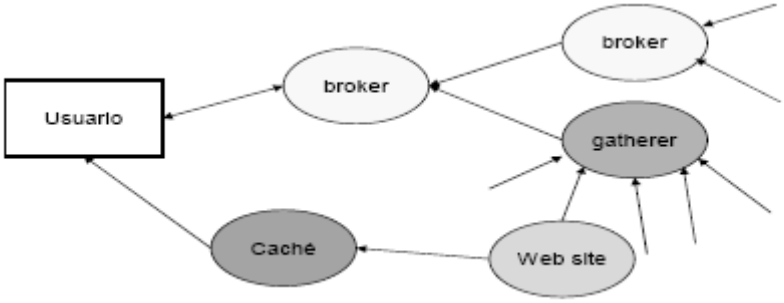


Figura 2.3.3. **Componentes de una arquitectura distribuida.**

Uno de los objetivos de *Harvest* es permitir la creación de *brokers* específicos por temas, de modo que se evite la creación de índices genéricos complejos. *Harvest* incluye un broker *"de registro"* con información sobre los *gatherers* y *brokers* restantes. Esto permite localizarlos cuando se construye un nuevo sistema. La arquitectura también incluye replicadores y caches de objetos. Los *replicadores* permiten duplicar servidores, mejorando la escalabilidad.

Por ejemplo, puede replicarse el broker de registro en diferentes regiones geográficas para permitir un acceso más rápido. La replicación permite también dividir el proceso de fusión de datos entre varios servidores Web.

Por otro lado, la caché de objetos reduce la carga del servidor y de la red, así como el tiempo de respuesta cuando se accede a las páginas Web.

2.3.4 APLICACIONES DE LOS ROBOTS WEB.

Las posibilidades de utilización de un robot son muchas y variadas, la función asociada a un motor de búsqueda Web es la más común, pero, es sólo una de ellas.

- **Creación de Offline Browsers *(navegadores sin conexión):*** es decir programas que se conectan a un determinado URL y se descargan *(download)* todos los archivos que encuentran dentro del mismo. La ventaja de los Offline Browsers es, una vez *"bajados"* los archivos se da la posibilidad de consultarlos sin estar conectado a Internet.
- **Creación de agentes de verificación de enlaces:** aquí los Robots cumplen la función de mantener al día los enlaces de un sitio Web e informar de los que han cambiado o los que ya no hacen referencia a ninguna página. Son muy útiles como herramienta administrativa para un servidor Web propio.
- **Creación de Buscadores:** debido al incremento de sitios Web, los buscadores se han especializado para encontrar datos concretos.

  Así tenemos entre otros:

  - **Buscadores de software:** para buscar la última utilidad en Internet.
  - **Buscadores de personas:** permiten encontrar la dirección e-mail de una persona a partir de su nombre y apellidos.

2.3.5 ENTENDIENDO EL FUNCIONAMIENTO DE UN ROBOT.

Un robot es una aplicación que de manera automática se mueve por la Web, ya sea en un sitio local o en un dominio más amplio. La forma normal de comportamiento de un robot es acceder a documentos y seguir los enlaces encontrados y almacenarlos en una base de datos para su uso posterior dentro de un motor de búsqueda.

El robot puede ser útil para diversas tareas, tales como, probar enlaces obtenidos de los documentos HTML de un sitio específico e imprimir un reporte de todos los enlaces que ya no son validos.

Un detalle a considerar en el diseño y contracción de un robot spider es que éste bombardeara al servidor con solicitudes rápidas y frecuentes que pueden demandar los recursos que posee un servidor, dado que, un robot no consumirá tiempo en cargar archivos multimedia para preséntalos del lado del cliente como lo hace un navegador.

2.3.5.1 EL PAPEL DE LOS ROBOTS EN LOS MOTORES DE BÚSQUEDA.

Los robots crean y actualizan base de datos y el motor de búsqueda realiza consultas en ésta como respuesta a petición de los usuarios. Los robots existentes tienen distintos criterios y métodos para la obtención de información. Además algunos robots tratan descubrir cada sitio en Internet, mientras que otros solo incluyen aquellas paginas cuyo propietarios hayan solicitado su inclusión en la base de datos.

2.3.5.2 ASPECTOS DE DISEÑO DE LOS ROBOTS.

Los robots pueden ser escritos en lenguaje de programación de tal forma que cumpla con lo siguiente:

- El lenguaje de programación a seleccionar, debe tener funciones para conectividad y comunicación en redes.
- Habilidades para el análisis gramatical.

Entonces si se cumple con lo anterior, es un candidato para crear robots Web. La mayoría esta escrito en lenguajes como C/C++, Perl, etc. Para mas referencias sobre lenguajes de programación ver apartado 2.2 sección 2.2.3 Lenguajes de programación Web.

Preguntas de repaso.

1. ¿Que es un Sistema de Búsqueda?

   _____

   _____.

2. Mencione las partes por las que se compone un Sistema de Búsqueda:

   _____

   _____

   _____

3. Complemente cada idea agregando si corresponde a un significado de: Generador de consulta, Buscador, Generador de presentación.

   Encargado de acceder a la estructura interna de datos para satisfacer la petición de datos.

   _____.

   Construye la vista a ser incluida en el documento HTML que recibirá el usuario a través del servidor Web.

   _____.

   Recibe la consulta realizada en forma de una cadena de texto ingresada por el usuario.

   _____.

4. Mencione los elementos de la Arquitectura Centralizada:

   _____

   _____

   _____

5. Mencione los dos elementos que la arquitectura *Harvest* introduce para resolver los problemas de la arquitectura crawler:

   _____

   _____

   _____

2.3.6 CLASIFICACIÓN DE LOS SISTEMAS DE BÚSQUEDA.

Los sistemas de búsqueda pueden clasificarse en: directorios, motores de búsqueda, metabuscadores y buscadores híbridos. Estos son los más consultados y conocidos por todos en la Web.

Todos estos sistemas de búsqueda poseen características que los hace buenos candidatos en el momento de presentarse la necesidad de encontrar información. La capacidad para hacer de estos sistemas eficaz y eficiente depende de muchos factores entre los cuales se destacan por importancia: el tamaño de la base de datos con la información sobre la Web y la efectividad del sistema de índices que permiten realizar la búsqueda solicitada.

2.3.6.1 DIRECTORIOS.

Un directorio es un sitio web que organiza una parte de los contenidos de la red en categorías y temas cuando se consulta informacion. Dichas categorías forman una estructura en forma de árbol previamente diseñada, lo cual quiere decir, que cuando se crea un directorio, se crea una clasificación de la información.

La interfaz con el usuario para este tipo de sistemas de búsqueda es discriminatoria, es decir, que de la lista principal de categorías, el usuario elige una y discrimina el resto, el directorio envía al usuario a esa rama del árbol y le muestra otra lista ahora de subcategorías y se repite el proceso hasta llegar al nivel mas profundo de subcategorías que es el que muestra las referencias a las paginas o sitios.

Figura 2.3.4 Ejemplo de un Directorio.

2.3.6.2 MOTORES DE BÚSQUEDA.

Tal y como sucede con muchos de los conceptos ligados con el Internet, el termino de motor de búsqueda o *"Search Engin"* no tiene una definición universalmente aceptada, debido a que se refiere a los sistemas de búsqueda en general como lo son los directorios como motores de búsqueda.

La mayor parte de la bibliografía consultada define un motor de búsqueda, como "una página Web o programa para conseguir información en Internet"; en esta definición no se especifica si dicha información se encuentra en un solo punto en Internet o si el motor de búsqueda *"navega"* por la Web para localizar la información solicitada, incluso no se especifica si se consigue la información en sí, o referencia en forma de enlaces para llegar a lo que se busca. Sin embargo, esta definición abarca a todos los sistemas de búsqueda (incluyendo directorios).

Por lo tanto, se define como motor de búsqueda o buscador como:

El sistema de búsqueda que utiliza robots y permite a los usuarios buscar y recuperar información específica de la World Wide Web. El motor de búsqueda puede buscar en todo el texto de los documentos Web, una lista de palabras clave, o usar técnicas de revisión de documentos, indexando la información recopilada en su base de datos.

2.3.6.3 BUSCADORES HÍBRIDOS.

Un buscador híbrido posee las características de los directorios y de los motores de búsqueda.

Es decir, la información está clasificada por categorías pero también se pueden realizar búsquedas por palabras clave sin necesidad de especificar categorías.

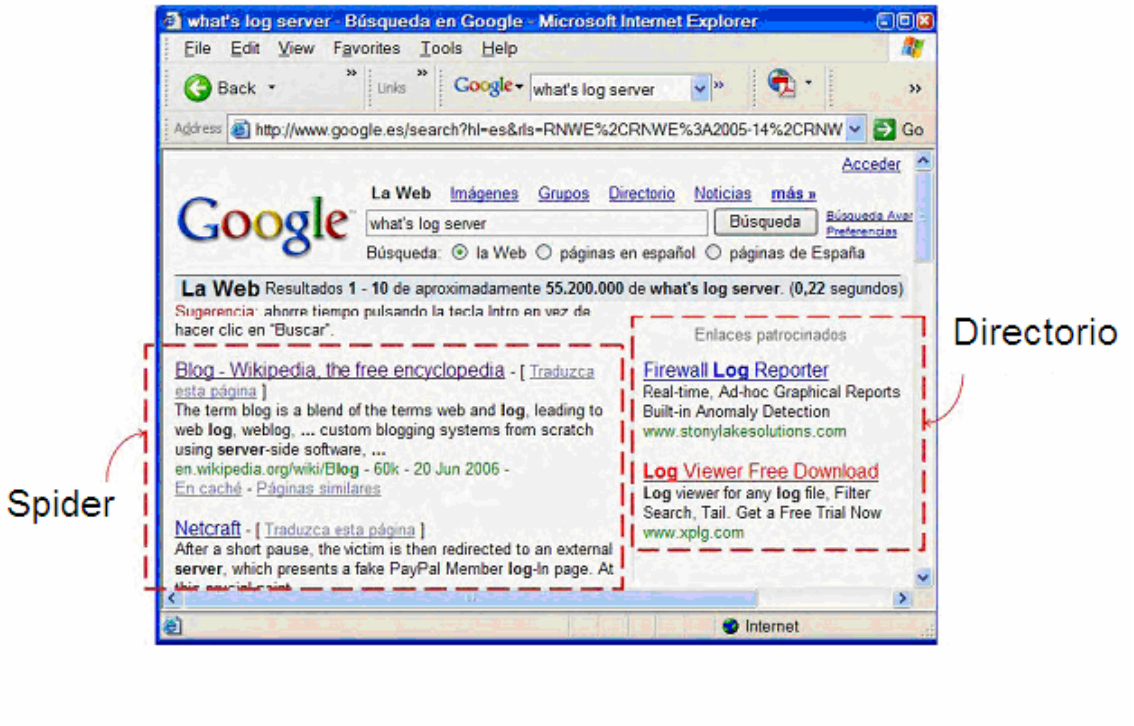


Figura 2.3.5 Ejemplo de buscador híbrido.

2.3.6.4 METABUSCADORES.

Los metabuscadores son sistemas de búsqueda que no son motores de búsqueda completos, ni directorios clasificados; son en realidad un tipo de buscador que aprovecha el trabajo de recopilación que realizan otros sistemas de búsqueda.

Este tipo de sistema de búsqueda carecen de una base de datos propia, por lo que al admitir una consulta, su trabajo consiste en redirigir dicha consulta a diferentes sistemas de búsqueda (motores y directorios) y recibir los resultados con la finalidad de hacerlos disponibles a los usuarios.

Este tipo de buscadores no obtienen toda la potencia de aquellos sistemas de búsqueda de los cuales se auxilia, debido principalmente a que los formatos de inserción de consulta no son universales, pero son generalmente un buen punto de partida para iniciar una investigación profunda sobre el tema de interés.

Figura 2.3.6 Ejemplo de un Metabuscador.

2.3.7 CONCEPTOS DE RECOLECCIÓN Y ROBOTS.

Como ya se menciono anteriormente hay dos tipos de sistemas de búsqueda mayormente utilizados: *los directorios y motores de búsqueda*. En la etapa de la recolección de la información es desde luego distinta para cada tipo.

Para el caso de los directorios la recolección se lleva acabo a solicitud del propietario del sitio Web, es decir, se llena una solicitud en la cual indica el URL, el titulo, una descripción y una o dos categorías en las cuales considera que debe estar clasificada su pagina o sitio Web, los catalogadores del grupo de trabajo del sistema de búsqueda se encargan de incluirla en la categoría adecuada.
Los motores de búsqueda se ven apoyados para la recolección de datos por un programa robot que normalmente tiene autonomía parcial o total de operación.

Esto quiere decir que en algunas o todas las fases de arranque, duración y terminación pudieran ser automáticas. Los robots Web o simplemente robots, conocidos también como *spiders, worms, crawlers* entre otros nombres son una clase especial de *Bots* y son usados para desglosar la estructura de un sitio Web o incluso World Wide Web completo. La aplicación más común es crear una base de datos usada por un motor de búsqueda. Sin embargo ésta, no es la única aplicación para estos programas, también por ejemplo para la localización de recursos multimedia en la red, para el análisis de contenidos de archivos gráficos, como lo son mapas y fotografías.

Debido a que los robots se enfrentan con enormes cantidades de sitios Web, típicamente registran sus resultados en una base de datos y la mayoría obtiene su información efectuando una o más de las siguientes funciones:

- **Descubrimiento:** determinar que sitios existen en Internet probando con cada dirección IP posible.
- **Escrutinio de sitios:** Analizar un sitio para determinar las paginas que lo componen.
- **Indexación:** recolectar información descriptiva acerca de las páginas Web.

2.3.8 REPRESENTACIÓN DEL DOCUMENTO.

Un factor importante que pasa inadvertido, es que los documentos (páginas o sitios Web completos) son clasificados de acuerdo al contenido. Esto significa que es necesario identificar aquellos elementos dentro de estos.

Algunos robots no se dan a la tarea de elegir que parte dentro del documento deberán almacenar como representación del contenido, sino que en lugar de esto almacenan todo el documento (texto HTML), sin embargo esto no los libera de tener que almacenarlo dentro de una estructura de índices en base a palabras o frases que permita mas tarde encontrarlo en una búsqueda, el problema es identificar esas palabras o frases que representen el contenido.

2.3.8.1 PALABRAS CLAVE (KEYWORDS).

Para profundizar en el tema de la representación del documento, se define palabras clave a aquellas palabras o frases que representan el contenido de un documento, pagina o sitio Web. El concepto *"palabras clave destino" (target keyword)* es usado para referirse a las palabras o frases que el usuario de un sistema de búsqueda digitará para realizar su indagación sobre un tema en especifico.

¿Como se localizan estas palabras clave para incluirlas en la base de datos y utilizarlas en el sistema de búsqueda?

Existen varias formas y dependen mucho de qué tanta información incluye el desarrollador o programador de la pagina Web. La más sencilla se da cuando el robot localiza las etiquetas <META> con el atributo *keyword.*

Sin embargo debido a que no es indispensable que los diseñadores incluyan esta etiqueta, existen otras formas para determinar las palabras clave cuando no se definen expresamente. Una de ellas es basarse en el titulo, es decir a lo escrito entre las etiquetas <TITLE> y </TITLE>. Otra forma menos concreta pero frecuentemente efectiva es basarse en las palabras del primer o primeros párrafos, bajo la premisa de que los diseñadores usaran de alguna forma las palabras importantes para el contenido del documento en los primeros párrafos.

Por lo anterior, es recomendable para los diseñadores de páginas Web que usan palabras clave en la parte alta del documento Web.

2.3.8.2 USO DE META DATOS Y ETIQUETAS <META>.

Los metadatos o metainformación, es información acerca de los recursos de la Web entendible por las maquinas. La clave de ésta definición es la frase *"entendible por las maquinas",* que refleja que los programas agentes que navegan por Internet en búsqueda de recursos deberán entender la información sobre los documentos Web para facilitar a los usuarios todo el proceso de búsqueda y recuperación de información.

Hay dos tipos de metadatos: los implícitos en las propiedades de los recursos (como la fecha [date]) y los definidos de manera explicita por el diseñador del documento.

Los primeros son además utilizados por los protocolos como HTTP para los encabezados de respuesta a una solicitud y los segundos se definen utilizando las etiquetas <META> de HTML, elementos META de XML, o los formatos RDF o MCF.

Las etiquetas <META> son aquellas mediante los cuales puede proporcionarse información sobre el documento que las incluye, aunque la sintaxis empleada por estas etiquetas es fácilmente entendible por todos, en realidad, se diseñaron para ser utilizadas por programas robots.

Normalmente la información que se incluye en la página es contenido, los metadatos se usan para describir o definir tal contenido pero este no aparece en la ventana de un navegador, en lugar de eso es un elemento silencioso de la sección <HEAD> que añade comentarios o datos descriptivos adicionales para quien esté interesado.

Existen dos tipos de etiquetas <META>: HTTP-EQUIV y las que poseen un nombre (NAME).
HTTP-EQUIV generan campos que son equivalentes a los utilizados por el encabezado HTTP (HEADER). Las que poseen un nombre, proporcionan una manera abierta de definir campos con atributos que describa la información contenida en el documento. Este último tipo de etiqueta (NAME) es específicamente la que nos interesa por la capacidad que provee para trabajar con el robot en la construcción de una base de datos optima para el sistema de búsqueda.

No obstante la etiqueta <META name =”Atributo”> provoca falta de estandarización, debido a que es muy general, por ejemplo:

```
<META name=”Edad”  content=”5 años”>
<META name=”Autor”  content=”Shakespeare”>
<META name=”Compañía”  content=”UDB”>
<META name=”Revision”  content=”7 Marzo 2007”>
<META name=”Revision”  content=”7 Noviembre 2007”>
```

Nótese que en este ejemplo la etiqueta de información pudiera ser útil para el autor de la página pero sin significado para la mayoría de usuarios. Además no tendría significado para un robot en busca de archivos HTML.
En este momento no existen estándares aceptados universalmente, pero si propuestas, que definen que etiquetas <META> pueden usarse, que atributos son adecuados, cuales son las condiciones adecuadas de uso y como debe usarse la información definida en éstas.

Cualquier persona puede inventar sus propias etiquetas <META> y escribir programas que hagan uso adecuado de la información contenida en ellas. Precisamente por eso es fácil encontrar una variedad de estas etiquetas que diversas compañías tienen para su propio uso pero que no significan nada para el resto.
Además la mayoría de los usuarios nunca las verán porque, la existencia de éstas no altera en absoluto la forma de presentar la página Web en un navegador.

También existen herramientas que facilitan la inclusión de estas etiquetas al documento. Para usarlas sólo es necesario capturar el valor adecuado en cada campo y de forma automática se incrustan en el documento HTML, ejemplos:

Meta tag builder (compañía WebPromote)

Meta tag builder (SiteOwner.com)

Meta tag generator (Web_ignite)

Meta Builder (Vancouver Webpages)

Meta tag 101 (Desing Maker)

Dublin core Meta tag script (Vancouver Webpages)

Las etiquetas <META> son una forma de suplir la carencia de texto que el motor de búsqueda pudiera utilizar para obtener la representación de un documento Web (páginas en su mayoría con gráficos o marcos) y aunque no hay reglas concretas existen 2 atributos importantes: *Keyword* y *Description.*

El atributo *Keyword* puede usarse para definir palabras o combinación de ellas que expresan un concepto contenido en la página y que además sirvan para su correcta clasificación o indexación en la base de datos del sistema de búsqueda.

Aunque estas etiquetas son independientes, funcionan excelente usándolas juntas y sus propósitos son además complementarios. A continuación se presenta un ejemplo del uso de estas dos etiquetas:

```
<HEAD>
<TITLE>Robots y agentes inteligentes</TITLE>
<META name="Description"  content="Sistemas de Busqueda en Internet ">
<META name="Keyword"  content="Universidad Don Bosco">
</HEAD>
```

La etiqueta *Description* es útil para proporcionar un resumen del contenido de la página. Los motores de búsqueda usan este resumen cuando despliegan la lista de resultados. Se sugiere que no se usen más de 25 palabras en la descripción. Un desarrollador o un propietario de un sitio Web, podría pensar que usando mas palabras se notara mas en una lista de resultados, pero esto puede resultar contraproducente, porque la mayoría de los motores de búsqueda truncan esta descripción cuando rebasa cierta cantidad de palabras.

Comentario complementario.

Las Etiquetas <META> no son una solución definitiva. Para que estas funcionen y proporcione todas las ventajas que prometen, deberá haber una estrecha cooperación entre los diseñadores/desarrolladores Web y los motores de búsqueda. Cuando no hay metainformación que describa el contenido de una pagina y proporcione palabras clave para su fácil clasificación, es necesario entonces, extraer una representación del documento a través de un análisis de relevancia.

2.3.8.3 ANÁLISIS DE RELEVANCIA.

En una situación cotidiana en la que un usuario de una gran biblioteca (la Web por ejemplo), o cualquier colección de documentos, independientemente de lo bien o mal ordenado que se encuentre, el usuario formula una pregunta o consulta, es decir especifica una *"palabra clave destino"* que representa el tópico sobre el cual se desea obtener información.

El procedimiento de recuperación de información es simple: *buscar en todos los documentos disponibles la palabra clave destino y responder con aquellas referencias, es decir, datos de los documentos, no los documentos en sí, que de una forma u otra hacen alusión a ese tópico.*

Entonces la tarea del usuario seria buscar cada uno de los documentos referenciados en la respuesta, leerlos y conservar aquellos que son relevantes para sus propósitos y descartar aquellos que no lo son. Esto, por supuesto es tortuoso, por lo que surge la necesidad de obtener o recuperar solo documentos relevantes para la investigación o propósitos del usuario.

El análisis de relevancia es un tema complejo debido principalmente a que se intenta hacer un análisis gramatical y semántico del texto a partir de deducciones basadas en métodos matemáticos. Este tipo de análisis tiene un enfoque estadístico más que lingüístico.

2.3.8.3.1 Métodos de Análisis de Textos.

No todas las palabras son igualmente significativas para representar la semántica de un documento. En lenguaje escrito, algunas palabras tienen más significado que otras. Generalmente, los sustantivos (o grupos de sustantivos) son los más representativos del contenido de un artículo. En general, se considera que merece la pena preprocesar el texto de los documentos de una colección para determinar los términos a ser utilizados como *términos índice*[9].

Sin embargo, la representación de los documentos mediante términos índice da lugar a una representación imprecisa de la semántica de los documentos en la colección.

Por ejemplo, un termino como *the* no tiene significado y puede dar lugar a la recuperación de documentos que no están relacionados con la consulta del usuario. La utilización de todas las palabras para indexar un conjunto de documentos genera mucho *ruido* para la tarea de recuperación. Una forma de reducir el ruido es disminuir el conjunto de palabras utilizadas para la indexación. El preprocesamiento de los documentos puede verse como un proceso para controlar el tamaño del vocabulario y mejorar así el resultado de la recuperación.

2.3.8.3.2 Preprocesamiento de Documentos.

El preprocesamiento consiste en transformaciones para reducir el texto:

1. Análisis léxico (tokenization): Tratamiento de dígitos, signos de puntuación y mayúsculas.
2. Eliminación de stopwords: Filtrado de palabras con poco significado para propósitos de recuperación.
3. Stemming (lematizacion): Eliminación de afijos y recuperación de documentos con variaciones sintácticas de los términos de la consulta.
4. Construcción de *thesaurus*: Expansión de la consulta original con términos relacionados.

[9] Término preseleccionado que puede ser utilizado para referenciar el contenido de un documento. Normalmente los términos índices son nombres o grupos de nombres; en general son principalmente sustantivos.

2.3.8.3.3 Análisis Léxico del Texto.

Es el proceso de separar las palabras en el texto. Aunque a primera vista podría pensarse que esto únicamente supone el reconocimiento de los separadores de palabra, sin embargo existen varios casos (problemas) a tener en cuenta:

- Combinación de letras y números: UB40, Windows95, 350AC, 510A.C.

- Números: Para los números se suele hacer otro tipo de indexación. Además, debe tenerse en cuenta que no todos los números significan lo mismo: Motorola 68000 (nombre propio), 68000 dólares (cantidad), 2003 (¿año?).

¿Cómo reconocer los números que son relevantes?
En general, los números no se consideran términos índice a menos que se indique lo contrario (mediante expresiones regulares).

- Guiones y signos: Los guiones se suelen eliminar para evitar inconsistencias de uso. Sin embargo, hay palabras que poseen guiones que forman parte integral de las mismas: Zig-Zag, 3.14, B-49, U.S.A. = USA. Para estos casos, se suele recurrir a la utilización de reglas para especificar estas excepciones.

- Palabras Compuestas: El Salvador, New York.
- ¿Ignorar mayúsculas y minúsculas?: Los analizadores léxicos suelen convertir el texto a minúsculas o mayúsculas. Sin embargo, una vez mas, deben considerarse los escenarios particulares, por ejemplo, MIT/mit, General Motors (Motors != motors), SAIL vs sail.

- Acentos: Résumé o Resume, Papá o Papa.

2.3.8.3.4 Eliminación de Stopwords.

Las palabras que son mas frecuentes en los textos de una colección no son buenos discriminantes y se denominan *stopwords*. Artículos, preposiciones y conjunciones, así como algunos verbos, adverbios y adjetivos son candidatos naturales para formar parte de la lista de *stopwords*. Son característicos de cada lenguaje por lo que se requiere detectar el idioma de cada documento tratado. La eliminación de *stopwords* permite reducir el tamaño de la estructura de indexación.

Sin embargo, la eliminación de *stopwords* puede empeorar el resultado de la consulta. Así, supongamos que un usuario está buscando documentos que contengan la frase *"to be or not to be".* La eliminación de *stopwords* puede dejar únicamente el termino *"be"* de toda la frase, haciendo casi imposible reconocer correctamente los documentos con la frase anterior. Esta es una razón por la que muchos motores Web no lematizan.

2.3.8.3.5 Stemming.

Frecuentemente, la palabra especificada por el usuario en la consulta no aparece exactamente en un documento pero si alguna variante gramatical de la misma como plurales, gerundios, sufijos de tiempo verbal, etc. Este problema puede resolverse con la sustitución de las palabras por su raíz (*stem*).

Un *stem* es la porción de una palabra que resulta de la eliminación de sus afijos (prefijos y sufijos). Un ejemplo podría ser la palabra *"plan"* que es el *stem* de *"planear", "planificación", "planificar".* Los *stems* son interesantes ya que permiten reducir variantes de la misma raíz gramatical a un concepto común. Consecuentemente, el *stemming* permite reducir el tamaño de la estructura de indexación ya que el número de términos índice se reduce. Además, permite ampliar la definición de la consulta con las variantes morfológicas de los términos usados, mejorando así el resultado de la recuperación.

Se pueden distinguir varios tipos de estrategias de *stemming:* mediante un diccionario, n-grams y eliminación de afijos. La aproximación mediante

diccionario consiste en la búsqueda del *stem* en una tabla. Es un proceso simple pero la construcción del diccionario es costosa, por lo que esta aproximación no suele ser practica. El *stemming* mediante *n-grams* se basa en la identificación de diagramas y triagramas y se trata más de un procedimiento de clustering que stemming como tal.

En eliminación de afijos, la parte mas importante es la eliminación de sufijos porque la mayoría de las variantes de una palabra se generan con su introducción. El algoritmo más popular para la eliminación de sufijos es el *algoritmo de Porter* debido a su simplicidad.

El algoritmo de Porter[10] usa una lista para la detección de sufijos. La técnica se basa en aplicar una serie de reglas a los sufijos de las palabras del texto. Por ejemplo, la regla $s->\phi$ se utiliza para convertir las formas plurales en singulares sustituyendo *"s"* por *"nulo".* Siempre se busca el sufijo mas largo de la palabra que empareje con los antecedentes en un conjunto de reglas.

2.3.8.3.6 Thesaurus.

Un thesaurus consta de:

1. Lista de palabras/frases (conceptos) importantes en un dominio. Son los componentes de indexación y generalmente son sustantivos o verbos en gerundio.
2. Un conjunto de palabras relacionadas para cada palabra anterior. Las relaciones entre conceptos pueden ser de sinonimia (car = automobile) o de generalización (Dólar -> unidad monetaria -> unidad de medida, Caballo -> equino -> mamífero -> ser vivo -> entidad).

Los principales objetivos de un thesaurus son proporcionar un vocabulario *"controlado"* para indexación y búsquedas, y ayudar al usuario en la formulación de consultas. La utilización de un vocabulario controlado permite la normalización de los conceptos indexados,

[10] http://tartarus.org/~martin/PorterStemmer/index-old.html

reducción del ruido, identificación de términos índice con significado semántica, y recuperación basada en conceptos, no en palabras.

Estos aspectos son especialmente útiles en dominios específicos, como la medicina o jurisprudencia, para los que existe una cantidad importante de conocimiento recopilado. Sin embargo, para dominios generales, como la Web, no esta tan clara su utilidad.

Procesamiento de Lenguaje Natural.

Existen dos leyes empíricas, ampliamente aceptadas en el campo de la recuperación de información, que estudian la variabilidad de las palabras dentro los *corpus* de lenguaje natural. Se denominan *Ley de Zipf* y *Ley de Heaps*.

Ley de Zipf.

La ley de Zipf también conocida como **"Ley del mínimo esfuerzo"** afirma que un pequeño número de palabras son utilizadas con mucha frecuencia, mientras que frecuentemente ocurre que un gran número de palabras son poco empleadas. Esta afirmación, expresada matemáticamente quedaría de la siguiente forma:

$$P_n \sim 1/n^a$$

Donde $\mathbf{P_n}$ representa la frecuencia de una palabra ordenada $\mathbf{n}^{th}$ y $^{\mathbf{a}}$ es casi 1. Esto significa que el segundo elemento se repetirá aproximadamente con una frecuencia de 1/2 de la del primero, y el tercer elemento con una frecuencia de 1/3 y así sucesivamente.

También se puede expresar de la siguiente manera:

$$F = \frac{k}{R}$$

La frecuencia, F, de aparición de una palabra en un texto es inversamente proporcional a su rango, R.

$$F \times R = k$$

Frecuencia por el rango igual a constante (k)

- Se cuentan las palabras y se ordenan de mayor a menor frecuencia de aparición, F
- El número de orden de cada palabra es su rango, R

La frecuencia, F, de aparición de una palabra en un texto es inversamente proporcional a su rango, R.

Esta ecuación se representa mediante la siguiente gráfica:

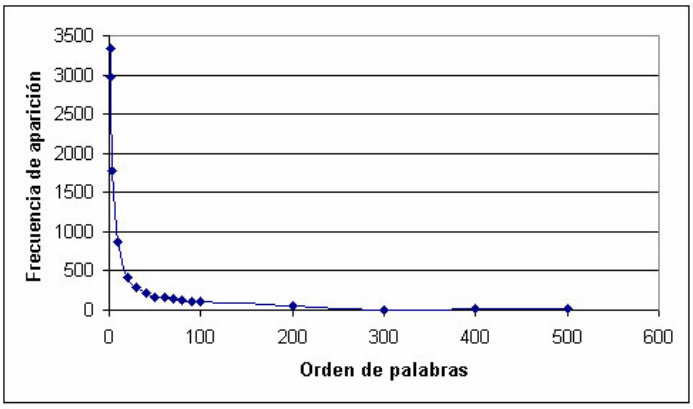


Figura 2.3.7 Representación gráfica de la ecuación de ley de Zipf.

Ley de Heaps.

De manera similar a la ley de Zipf, existe otra ley empírica que describe el comportamiento de los términos dentro de un texto escrito denominada ley de Heaps. En esta ley, se plantea una relación entre el tamaño del texto (cantidad de palabras) y el crecimiento del vocabulario (cantidad de

palabra únicas). En particular, postula que el tamaño del vocabulario (y su crecimiento) es una función del tamaño del texto.

V = N * (K^beta)

donde:
N: Es el tamaño del documento (cantidad de palabras)
K: Constante que depende del texto, típicamente entre 10 y 100.
beta: También es una constante que depende del texto, donde beta <> 0.

- 10 <= K <= 20
- 0.5 <= beta <= 0.6

Por lo tanto, si K = 20 y beta = 0.5, resulta:

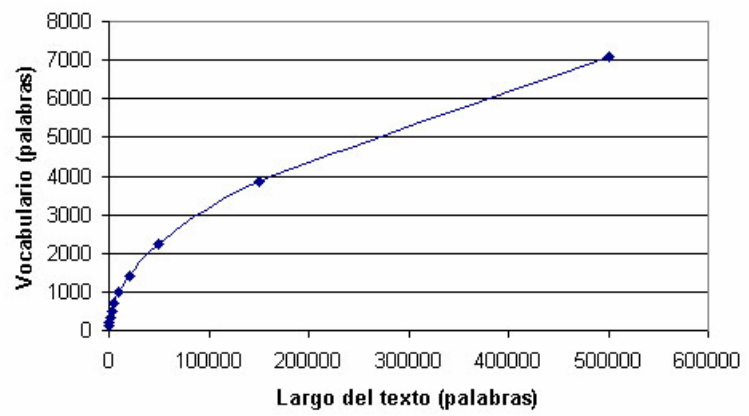


Figura 2.3.8 Representación grafica de la ecuación Ley de Heaps.

Preguntas de repaso.

1. ¿Defina de manera breve cada clasificación del Sistema de Búsqueda?

   _____
   _____
   _____
   _____
   _____
   _____.

2. ¿Mencione los dos tipos de Metadatos que existen?

   _____
   _____
   _____.

3. Mencione los 4 elementos del Preprocesamiento de documentos:

   _____
   _____
   _____
   _____.

4. Defina que es para cada uno el atributo Keyword y el atributo Description:

   _____
   _____
   _____
   _____.

2.3.8.4 ESTANDARIZACIÓN.

Existen diversas organizaciones y grupos de trabajo reconocidos en la comunidad Web con la constante tarea de regularizar los procesos, normas y métodos encontrados en la red (Web). No obstante, en el caso de los mecanismos para la autodescripción de la información contenida en la Web, el problema son las varias entidades hasta la fecha.

Tales organizaciones han elaborado y promovido propuestas formales que ofrecen soluciones prácticas al problema de la estandarización.

Así las principales organizaciones entre las que se encuentras Dublin Core, Internet Engineering Task Force (IETF), World Wide Web Consortium (W3C); han lanzado propuestas como Dublin Core Metadata Initiative (DCMI), Meta Content Framework (MCF) y Resource Description Framework (RDF) que son producto del trabajo de especialistas directamente involucrados con el Internet en general.

Las vías utilizadas para realizar estos trabajos son diversas y van desde la generación de borradores Internet *(Internet draft)* los cuales son documentos generados en pequeñas conferencias sobre un tópico en particular, hasta las solicitudes de comentarios *(RFC request for comment)* que son los documentos que definen los estándares en Internet, haciendo uso además de las listas de correos *(Mailing lists)* que forman grupos virtuales de trabajo y colaboración en Internet.
A continuación se describen cada una de las propuestas para la estandarización de la autodescripcion.

2.3.8.4.1 DCMI, Dublin Core Metadata Initiative.

Es un foro abierto en 1995 comprometido con el desarrollo de estándares de meta datos en línea que soporta un amplio rango de propósitos y modelos.
Las actividades de esta iniciativa incluyen grupos de trabajo dirigidos al consenso, talleres globales, conferencias, alianzas y esfuerzos educacionales para promover la adopción internacional de los meta datos y desarrollo de vocabulario especializado para describir los recursos que doten de mas elementos inteligentes de autodescripción a los sistemas de descubrimiento de información.

La finalidad de esta propuesta es definir una serie de descriptores de documentos y métodos para su uso que faciliten la recuperación y la clasificación (indexación) automática de la información. Estos elementos vienen a fortalecer y completar el propósito de las etiquetas <META> *Keyword* y *Description*.

Los objetivos que motivan al esfuerzo de Dublin Core son:

- Simplicidad de creación y mantenimiento.
- Semántica comúnmente comprendida.
- Acatamiento de patrones existentes y nuevos.
- Ámbito internacional y campo de aplicación.
- Extensibilidad.
- Interoperatividad entre colecciones y sistemas de clasificación.

2.3.8.4.2 MCF, Meta Content Framework.

Meta Content Framework es una forma de representar el contenido de un sitio Web de una manera mas sofisticada de lo que podría hacerse usando las etiquetas <META>.

La compañía *Netscape* envió una propuesta a W3C en junio de 1997 para apoyar en uso de MCF, que tenia la intención de fortalecerla como propuesta y extender su uso. En ese tiempo otras compañías importantes desarrolladoras de productos para Internet (entre ellas Microsoft) estaban dispuestas a respetar esta propuesta si W3C la adoptaban como estándar.

MCF con su característica que facilitaría el trabajo de los motores de búsqueda al momento de la clasificación. Por ejemplo podría proveer de una pagina de resumen de referencias, que incluiría los URL's y descripciones de todas las paginas en el sitio Web, ahorrándoles a los motores de búsqueda el descubrimiento y almacenamiento de cada una de las paginas. Como puede apreciarse con este ejemplo el ahorro en el consumo de recursos seria considerablemente alto. Sin embargo volvió a surgir la falta de adopción de propuestas de estandarización y debido a los inciertos en cuanto a tiempo y forma en que las compañías de motores de búsqueda adoptarían ésta, MCF no progresó, aun cuando no es una propuesta del todo olvidada.

No obstante todas las ventajas de MCF han sido relegadas desde la aparición de RDF, que vino a suplirla y despertó mucho interés de parte de la sociedad Internet.

2.3.8.4.3 RDF, Resource Description Framework .

Resource Description Framework es una iniciativa de W3C generada para ayudar a estandarizar y definir cómo deben ser utilizados los metadatos.

Una de las ventajas que ofrece RDF sobre MCF es la simplicidad en la práctica. Ésta es una de las razones de peso para el desplazamiento de MCF por RDF.

RDF promete un método estructurado para crear y definir términos de metadatos en un esquema, el cual puede ser usado por terceros.

Debido a que un término puede tener diferentes significados en circunstancias y contextos diferentes, los esquemas y nombres de espacios (*schema y namespace*) son de enorme utilidad porque definen cómo debe ser utilizado un término.

Entre las propuestas potenciales que W3C hace con respecto de RDF se encuentran:

Diccionarios y esquemas de clasificación de librerías, mapas de sitios Web, descripción de los contenidos de las páginas Web, sistemas de valoración y firmas digitales.

2.3.9 ESTRUCTURA DE LA BASE DE DATOS.

En este apartado se definirá la manera mas completa posible, sin caer en el detalle excesivo, de cómo es la estructura de la base de datos para un sistema de búsqueda.

Cuando se diseña una base de datos para un sistema de búsqueda, deben tenerse en cuenta varios aspectos, considerando que la cantidad de recursos consumidos por un sistema de éste tipo es muy grande. Entre los aspectos mas importantes se encuentran:

- La estructura de almacenamiento interno. Aún cuando los dispositivos son mucho más rápidos actualmente, un acceso a disco aun es mucho más lento que el acceso de memoria. Por lo tanto la estructura de almacenamiento interno es tan importante como la de almacenamiento externo, debido desde luego a que el sistema de búsqueda debe tener un alto nivel rendimiento manteniendo en memoria principal la información con alto índice de acceso.

- Por otra parte es importante procurar que la estructura de los archivos e índices sea tan sencilla y pequeña como sea posible, para apoyar el factor de rendimiento pero sin perder de vista que una estructura bien diseñada deberá tender a reducir la cantidad de accesos.

La estructura de la base de datos para un sistema de búsqueda se constituye de diversos elementos desde las estructuras internas (variables, arreglos, listas, árboles, etc.)

2.3.9.1 ORGANIZACIÓN.

Los elementos que se mencionarán en esta sección, no son utilizados estrictamente por todos los motores de búsqueda, sino que cada buscador o directorio, de acuerdo a sus metas y sus recursos, crea tantos elementos como sean requeridos.

Existen dos opciones para la creación y uso de la base de datos:

- Base de datos administrada por DBMS.
- Base de datos de bajo nivel.

2.3.9.1.1 Base de Datos Administrada por DBMS.

Consiste en crear una base de datos usando como plataforma de implementación un administrador de base de datos (DBMS) que facilite los accesos para la actualización, clasificación y consulta.

Desventajas.

El costo de esta alternativa puede ser poco atractivo, sin embargo existen productos de bajo costo e igualmente potentes dados los requerimientos.

Ventajas.

La independencia de software es una enorme ventaja debido a principalmente a que la mayoría de los accesos son procesos simples de grabación y consulta, y el administrador del sistema de búsqueda podría cambiar sin demasiados problemas para la base de datos. Otra ventaja es la independencia de dispositivos debido a que los detalles físicos son responsabilidad del DBMS y es posible incluso que se migre parcial o totalmente la base de datos sin que el sistema de búsqueda se vea afectado.

2.3.9.1.2 Base de Datos de Bajo Nivel.

Es una base de datos en la cual el programador es quien se encarga de todos los detalles de almacenamiento y recuperación de la información a nivel físico.

Debido que el diseñador conoce de manera precisa los componentes físicos del sistema, es capaz de dar el máximo aprovechamiento a los recursos y obtener el mayor beneficio en cuanto a velocidad de acceso y de respuesta.

Por otra parte como no es necesaria la adquisición de un producto DBMS, no existe costo en este rubro, pero existe la posibilidad de que el sistema pueda ser tan complicado que el costo de diseño y desarrollo se eleve por encima de un DBMS. Por lo tanto el costo puede se una ventaja o desventaja según las características propias del sistema de búsqueda que adopte esta opción.

Otra gran desventaja es la alta dependencia de dispositivos por los que una migración a un sistema de computo (hardware) distinto para el que fue construido, resulte ser restrictiva o incluso prohibitiva.

2.3.9.2 COMPONENTES.

Los componentes del sistema de una base de datos se dividen en dos grupos que no tienen ningún tipo de orden ya que el orden de creación o de uso es determinado por cada sistema, tales grupos son: internos y externos.

2.3.9.2.1 COMPONENTES INTERNOS.

2.3.9.2.1.1 Archivos Virtuales.

Contienen una fracción de la base de datos en disco, y su objetivo es contener en memoria aquella información que es utilizada de manera constante, evitando en la medida de lo posible, los accesos directos a disco.

2.3.9.2.1.2 Léxico.

Este diccionario tiene la finalidad de contener en memoria principal, una lista de palabras o la parte inicial de ellas (de las que se derivan otras) en diversos idiomas. Tiene entre los objetivos más importantes ser una primera instancia de validación en las búsquedas y servir en un análisis

de relevancia. Este tipo de diccionario en los sistemas de búsqueda actuales contiene entre 15 y 20 millones de palabras[11].

2.3.9.2.2 COMPONENTES EXTERNOS.

2.3.9.2.2.1 Hit List.

Es una lista de ocurrencias de palabras específicas en documentos específicos. Entre los datos que se almacenan están la palabra, la posición de la palabra en el documento, el tipo de fuente y la capitalización de las letras.

Este es uno de los archivos primordiales para el buen funcionamiento del sistema de búsqueda en tiempo de consulta, ya que, es el que ayuda a encontrar una palabra o frase mediante uniones e intersecciones de palabras en un documento y ayuda además a ponderar los documentos.

2.3.9.2.2.2 Almacén.

Contiene todo el HTML completo de cada página Web.

Los documentos se almacenan en secuencia indicando desde luego por medio de un campo previo la longitud de cada uno. Los datos que se registran son en términos generales una clave del documento (no necesariamente URL), la longitud y el contenido.

2.3.9.2.2.3 Índice de Documentos.

Guarda información acerca de cada documento. Es un ISAM (Index Secuential Access Mode, método de acceso secuencial indexado) sencillo de longitud fija ordenado por la clave del documento. Entre los datos almacenados están el estado del documento, un apuntador al almacén, un apuntador a otro archivo que almacena el URL y el titulo, un resumen del documento u otras estadísticas.

2.3.9.2.2.4 Archivo Mínimo.

Este archivo tiene la finalidad de contener la mayor cantidad de información sobre el documento sin tener que guardarlo todo. Esta

[11] http://www.telefonica.net/web2/basedatosbuscador/google.htm

información incluye el URL, el titulo, la descripción y las palabras clave; todo precedido de una clave del documento. La diferencia con *Hit List* es que, éste contiene la posición y atributos de las palabras dentro de los documentos, mientras que el archivo mínimo, contiene información mínima pero elemental sobre los documentos.

2.3.9.3 FUNCIONAMIENTO INTERNO DE LOS SISTEMAS DE BÚSQUEDA.

El proceso es sencillo: cuando el usuario teclea una palabra clave destino, el buscador ejecuta otro programa (motor) que recibe como parámetro la cadena y se encarga de determinar si son varias palabras, una frase o combinación de éstas. Si se trata de palabras utiliza el diccionario para determinar si se trata de palabras validas y el idioma en que están.

Desde luego este proceso es más sofisticado en algunos buscadores que en otros. Por otra parte para hacer más eficiente y rápida la búsqueda, los buscadores que los poseen hacen uso de los archivos virtuales para evitar accesos a disco. Finalmente se hace uso de *Hit List, Archivo Mínimo, índice de documentos* y en última instancia el *almacén* para localizar y ponderar.

2.3.10 APLICACIÓN CLIENTE/SERVIDOR.

Cuando se termina de recopilar y clasificar la información del sistema de búsqueda, el siguiente paso es lograr que dicha información esté disponible para el usuario final. El último componente es la interfaz con el usuario; la interfaz varía de un tipo de sistema de búsqueda a otro y es fácilmente identificable.

En este apartado se verán los diferentes tipos de interfaces existentes que varían para cada sistema de búsqueda.

2.3.10.1 MOTORES DE BÚSQUEDA.

Los motores de búsqueda muestran un cuadro de texto que permite la entrada de la palabra, palabras o frase sobre la cual se desea encontrar información en la Web.

Algunos motores ofrecen realizar la búsqueda en páginas elaboradas en algún lenguaje específico, principalmente ingles y español, por ser de los idiomas más hablados. El resultado de la búsqueda es una lista de referencias que puede ser extensa, pero las 10 primeras son las que se supone son el mejor resultado.

En la siguiente imagen se muestra la interfaz típica de un motor de búsqueda, se eligió como ejemplo *Google*; esta interfaz varia un poco entre buscadores y cada uno agrega su propio diseño y publicidad en la página principal.

Figura 2.3.9 Buscador Google.

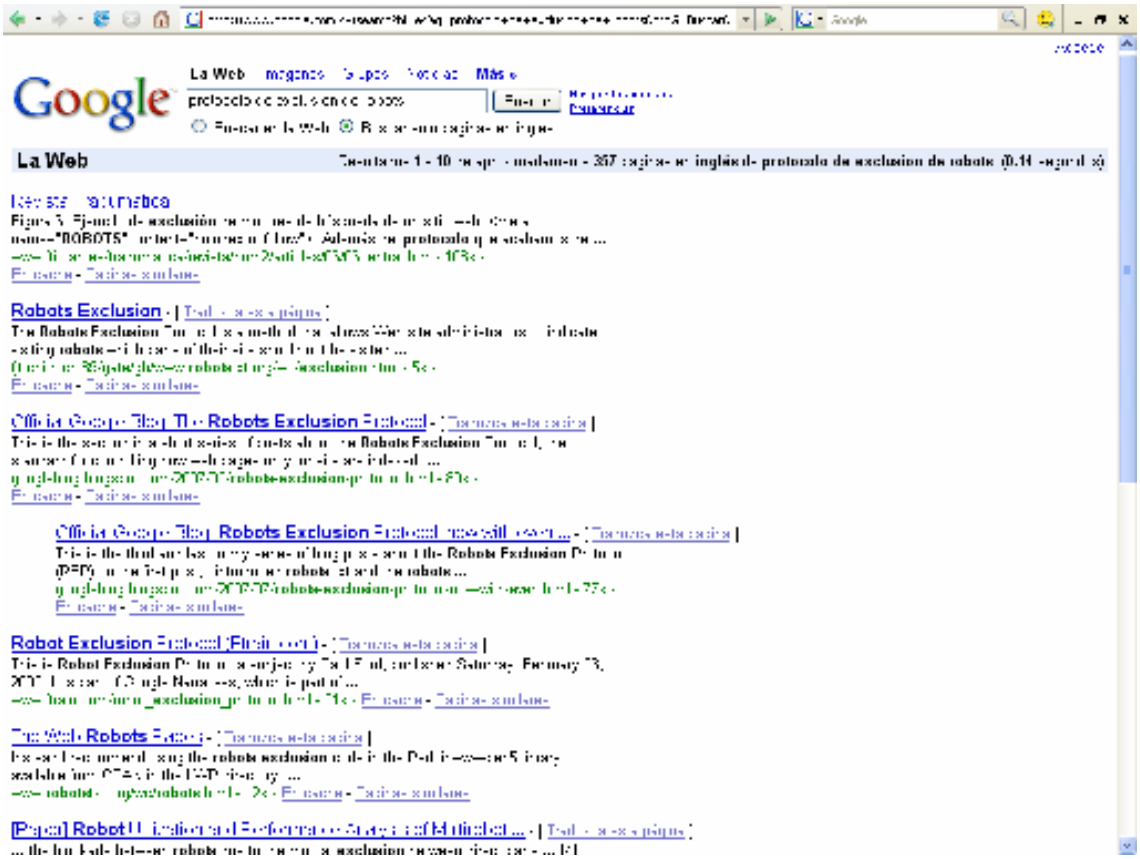


Figura 2.3.10 Resultados de una búsqueda en Google.

Su funcionamiento es sencillo: El usuario proporciona la palabra o frase en el cuadro de texto, y luego, presiona el botón de búsqueda *"Buscar con Google*", lo cual, es la acción que provoca el inicio del proceso de exploración que arroja como resultado la lista de referencias que coinciden con lo especificado anteriormente, entonces el usuario basándose en la descripción que normalmente acompaña a cada referencia de la lista, seleccionará algunos o todos los resultados para tomar la información que se necesite.

A continuación se muestran las pantallas de varios motores de búsqueda:

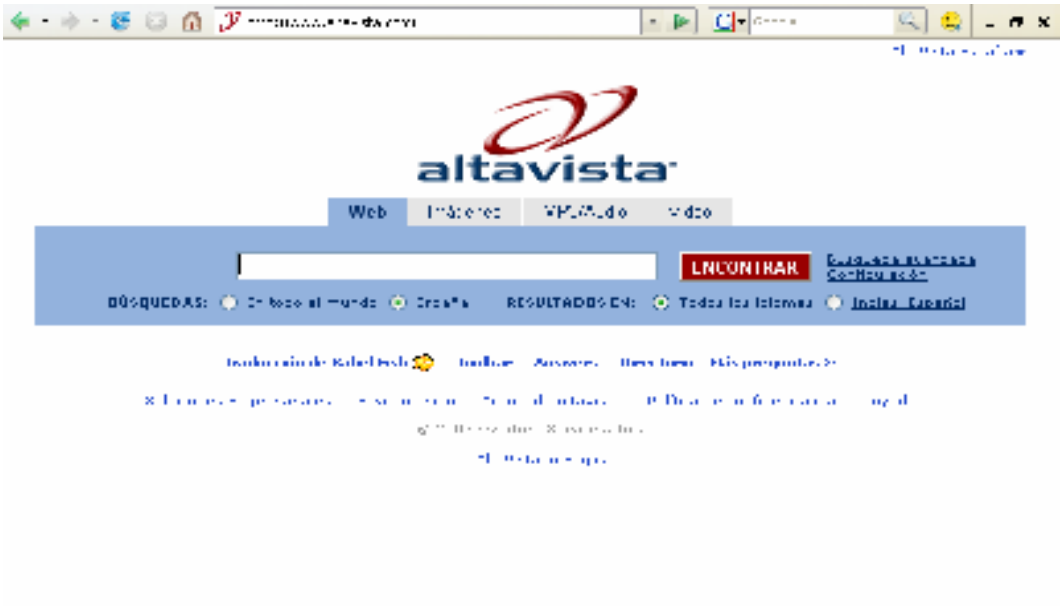


Figura 2.3.11 Buscador Altavista.

Figura 2.3.12 Buscador Yahoo.

Figura 2.3.13 Buscador WebCrawler.

Como puede observarse en las imágenes, aunque cada motor de búsqueda presenta diferente diseño en su página principal y añade publicidad como mejor le conviene a la compañía propietaria, todos proporcionan al usuario un cuadro de texto y un botón para accionar la exploración. Adicionalmente varios de ellos poseen un directorio que los convierte en buscadores híbridos que combinan las características de un motor y directorio. Además la mayoría de buscadores incluye una opción de búsqueda avanzada, que permite al usuario proporcionar con más detalle los parámetros de la búsqueda.

2.3.10.2 DIRECTORIOS.

En el caso de los directorios, la pantalla principal de estos sistemas muestra una lista de categorías, que pueden oscilar entre 10 y 20, dependiendo de cada sistema de directorios.

Figura 2.3.14 Directorio Yahoo.

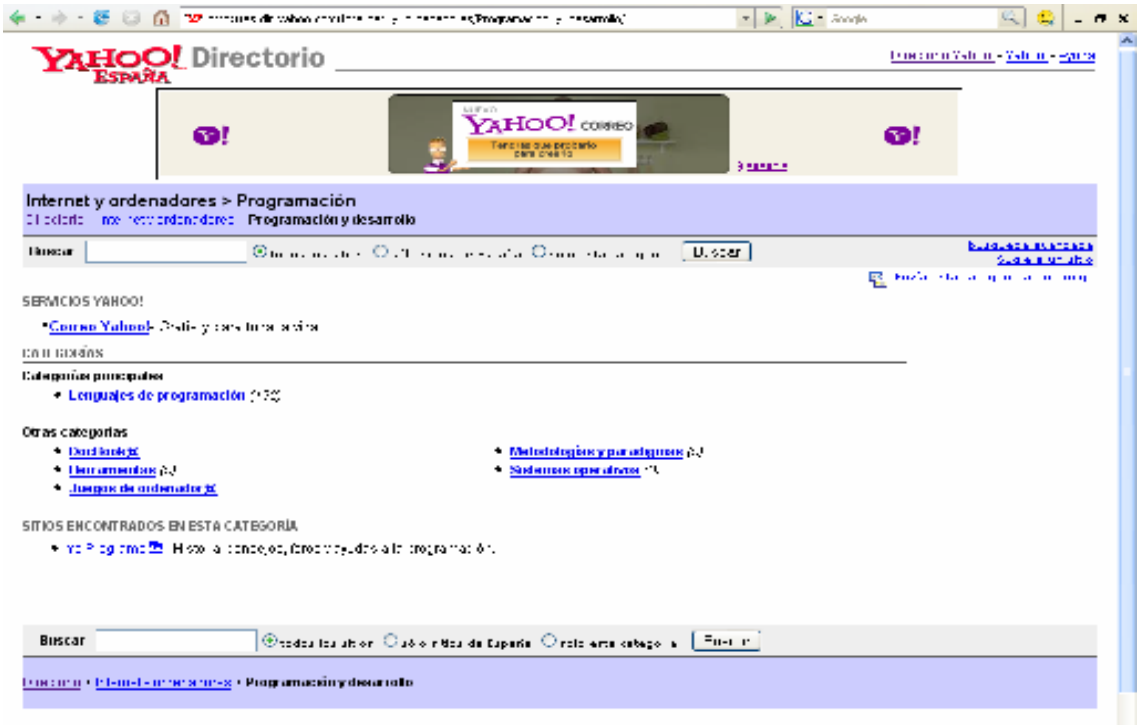


Figura 2.3.15 Categorías y subcategorías de un directorio.

El proceso de búsqueda inicia cuando el usuario elige una categoría de la pagina principal, después el sistema envía al usuario a navegar por la rama elegida cuya primera pagina es ahora una nueva lista de subcategorías, el proceso de elección de estas termina cuando el usuario llega a una lista de referencias a paginas o sitios que corresponden con la búsqueda especifica del usuario.

2.3.10.3 INTERFACES DEPENDIENTES DEL NAVEGADOR.

No todas la interfaces de sistemas de búsqueda dependen de un navegador, o sea, no todas la interfaces se dan mediante páginas Web. El siguiente ejemplo muestra como uno de los muchos metabuscadores existentes (WebFerret) se ejecuta como un programa independiente y de la misma forma arroja los resultados.

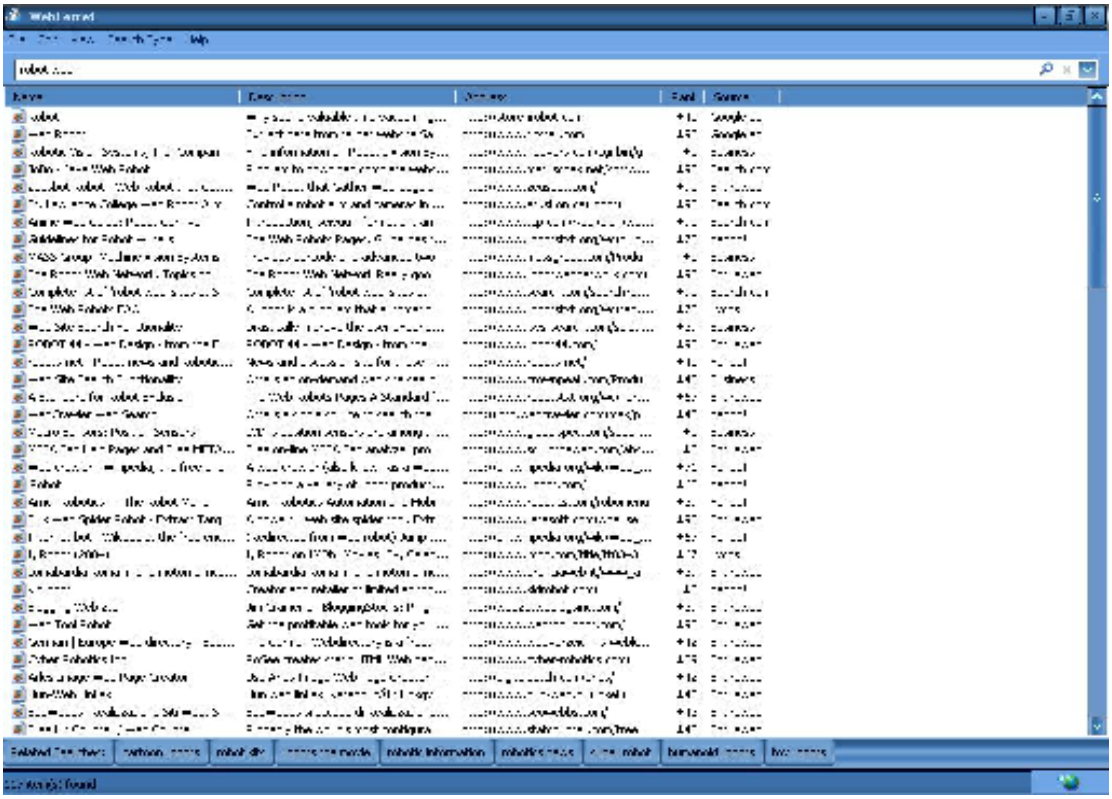


Figura 2.3.16 Ejemplo de un buscador que se ejecuta como un programa independiente.

Como se puede observar, el resultado de una consulta es también una lista de referencias. Aunque depende del software, los datos que se presentan en la lista son muy similares a los mostrados anteriormente.

2.3.10.4 CUANDO USAR UNO U OTRO TIPO DE SISTEMA DE BÚSQUEDA.

Un directorio muestra los enlaces o referencias de páginas, sitios o simplemente archivos por categorías. Frecuentemente esos índices son formados cuidadosamente por conocedores en el área de la clasificación, así que los sitios en estos sistemas de búsqueda son menores, pero de mayor

calidad. Mientras que los motores de búsqueda buscan en billones de páginas, muchas de las cuales no contienen la información solicitada.

Debido a lo anterior, el sistema que mejor se adaptara a nuestras necesidades depende en gran medida a las necesidades específicas de cada usuario. Es decir, si se desea encontrar la mayor cantidad de información sobre un tema en específico y se dispone de tiempo para evaluar los resultados, entonces la mejor opción son los motores de búsqueda; pero si se desea encontrar contenido, no necesariamente en grandes cantidades, entonces se puede optar por los directorios.

2.3.10.5 INSTRUCCIONES PARA EL USUARIO.

Lo más recomendable para un usuario que realiza una búsqueda de información, es dedicar unos minutos a conocer las opciones que el buscador ofrece para aprovechar al máximo sus servicios.

2.3.10.5.1 Opciones de Búsqueda.

Algunas de las opciones mas utilizadas son las siguientes:

- **Comillas:** casi todos lo motores de búsqueda permiten el uso de comillas cuando se pretende encontrar una frase exacta. Si lo que el usuario desea encontrar es la letra de una canción, por ejemplo, puede proporcionar un fragmento entre comillas "*ella es*", seguramente se obtendrán resultados con letras de poesía y canción con el mismo nombre, mientras que al no utilizar las comillas, se obtendrán documentos diversos con la o las palabras de la frase.

- **Operadores lógicos:** el uso de OR, AND y NOT puede permitir una búsqueda más refinada y especifica de tal manera que no solo la cantidad de resultados serán menores, sino que serán mejores en muchos de los casos.

- **Símbolos + y - :** si se desea delimitar la búsqueda a resultados más exactos, muchos sistemas permiten el uso de los operadores aritméticos + y - antes de cada palabra para incluir o excluir tales palabras en los

documentos encontrados. Por ejemplo +rock +español - héroes del silencio; obtendrá páginas con contenido referente a la música rock en español pero omitirá aquellos en los que hablen o aparezcan los héroes del silencio.

- **Usar más palabras:** cuando se agregan mas palabras a la búsqueda se acortan más los resultados. Es muy diferente buscar *esqueleto mamut*, que buscar *esqueleto mamut Siberia 1997*.

2.3.10.5.2 Búsqueda Avanzada.

La gran mayoría de los sistemas de búsqueda ofrecen opciones de búsqueda que permiten restringir aun más la cantidad y la calidad de los resultados obtenidos. Puede especificarse que los documentos se hayan creado o modificado en algún rango de fechas, que contengan las palabras clave destino en alguna sección especifica de la pagina Web, que se encuentre en algún dominio rango de dominios, que estén en determinado idioma, que los resultados sean archivos de un formato especifico, etc.

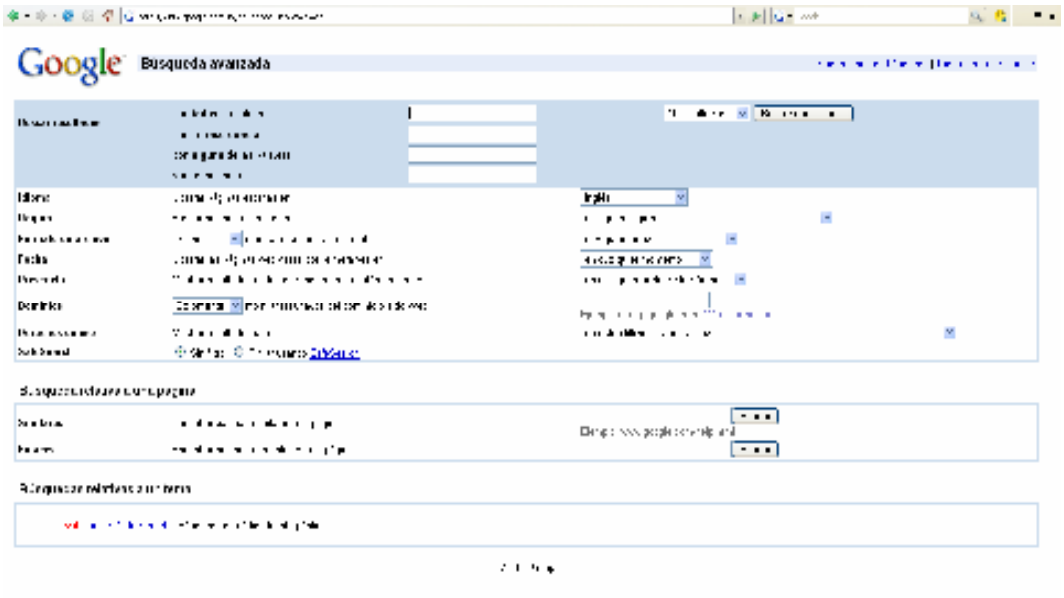


Figura 2.3.17 Búsqueda avanzada en buscador Google.

Preguntas de repaso.

1. Mencione las ventajas y desventajas de las bases de datos administrada por DBMS y bases de datos de bajo nivel.

   _____
   _____
   _____
   _____.

2. Indique los componentes internos de un sistema de base de datos.

   _____
   _____
   _____.

3. Para que se utilizan los símbolos + y – como opciones de búsqueda.

   _____
   _____
   _____.

Auto evaluación.

Instrucciones.

En la hoja provista, subraye la respuesta correcta. Coteje las respuestas al final del auto evaluación.

1. Es uno de los problemas que presenta la arquitectura centralizada y que son evitados por la arquitectura distribuida Harvest.
   a. La pagina Web no carga.
   b. El tráfico Web disminuye.
   c. Los servidores Web se saturan con las peticiones recibidas desde diferentes crawlers.

2. Tipo de aplicación de utilización que puede recibir los robots Web.
   a. Paginas Web.
   b. Agentes de verificación de enlaces.
   c. Navegadores Web.

3. Cuales son los aspectos que debe tener un lenguaje de programación para crear un robot Web.
   a. Funciones para conectividad y comunicación en redes.
   b. Habilidades para el análisis gramatical.
   c. A y B correctas.

4. Indique cuales son los dos tipos básicos en los que pueden clasificarse los sistemas de búsqueda.
   a. Directorios y motores de búsqueda.
   b. Crawler y Spider.
   c. Agentes y Robots.

5. Posee las características de los directorios y de los motores de búsqueda.
   a. Buscador híbrido.
   b. Metabuscador.
   c. Robot combinado.

6. Organización que ha elaborado y promovido propuestas formales ofreciendo soluciones al problema de estandarización de documentos Web.
   a. Microsoft.
   b. World Wide Web Consortium (W3C).
   c. Google.

7. Propuesta cuya finalidad era de definir una serie de descriptores de documentos y métodos para su uso que faciliten la recuperación y la clasificación automática de la información.
   a. MCF, Meta Content Framework.
   b. RDF, Resource Description Framework.
   c. DCMI, Dublín Core Metadata Initiative.

8. Propuesta que incluía descripciones de paginas en un sitio Web, ahorrando a los motores de búsqueda el descubrimiento y almacenamiento de cada una de las páginas.
   a. MCF, Meta Content Framework.
   b. RDF, Resource Description Framework.
   c. DCMI, Dublín Core Metadata Initiative.

9. Dos opciones de búsqueda utilizadas en los buscadores.
   a. Comillas y Operadores Lógicos.
   b. Stemming.
   c. Etiquetas <META>.

10. Opción de búsqueda que permite restringir aun más la cantidad y la calidad de los resultados obtenidos.
   a. Búsqueda avanzada.
   b. Delimitar búsqueda.
   c. Búsqueda por instrucción.

De 10 a 7 preguntas correctas: Puede continuar con el capítulo 4
Menos de 7 preguntas: Repase el capitulo 3 y compruebe el ejercicio.

Contestaciones de las preguntas de las páginas anteriores.

1. C, 2. B, 3. C, 4. A, 5. A, 6. B, 7. C, 8. B, 9. A, 10. A

Investigación Complementaria.

- Investigue sobre otros tipos de arquitecturas de los sistemas de búsqueda no mencionadas en este capítulo.
- Consulte mas información acerca de como funcionan los modelos matemáticos del procesamiento de texto en lenguaje natural.

2.4: ALGORITMOS Y CONCEPTOS UTILIZADOS EN LA ELABORACIÓN DE ROBOTS SPIDER.

Tiempo recomendado de estudio: 3 horas.

Objetivos:

Al concluir este capitulo, el lector estará capacitado para:

- Conocer los diferentes algoritmos que son utilizados para recorrer la Web.
- Explicar los mecanismos que los servidores Web utilizan para restringir el acceso de robots para que no sean sobrecargados sus servidores.
- Describir los aspectos importantes que un robot debe tener en la búsqueda de enlaces.

Introducción.

Un algoritmo es una serie de pasos sucesivos con un orden de operaciones con entradas y salidas que permite hallar la solución a un problema u obtener una solución. Es por tal razón que un robot spider necesita de un algoritmo en que se le especifique por medio de pasos lógicos como debe realizar una búsqueda a través de toda la red. Este capítulo contiene una descripción del funcionamiento de algoritmos como el de Crawling y de Ranking que son utilizados por algunos robots spider.

A la vez contiene información de los diferentes métodos utilizados por algunos servidores Web para que estos robots no visiten sus páginas Web, muchas veces por motivos de seguridad y confidencialidad. Así como también una descripción de los aspectos más importantes que debe tener un robot al momento de realizar las búsquedas de enlaces para ser presentados en un buscador Web.

2.4.1 ALGORITMO DE CRAWLING.

Existen diferentes técnicas a la hora de recorrer la Web. La más simple consiste en comenzar con un conjunto de URL's muy populares o enviadas explícitamente por administradores de sitios Web, y se siguen los links desde allí recursivamente, evitando repeticiones. El recorrido puede ser *breadth-first* (búsquela en anchura) *depth-first* (búsqueda en profundidad).

Para efectuar un recorrido de Internet, los motores de búsqueda permiten a los usuarios acceder a los sitios Web que serán añadidos al conjunto de URL's.

- **Breadth-first** (**Búsqueda en anchura**)**.**
  Breadth-first search o búsqueda en anchura es un algoritmo para recorrer o buscar elementos en un grafo, usado frecuentemente sobre árboles. Intuitivamente, se comienza en la raíz, eligiendo algún nodo como elemento raíz en el caso de un grafo y se exploran todos los vecinos de este nodo. A continuación para cada uno de los vecinos se exploran sus respectivos vecinos adyacentes, y así hasta que se recorra todo el árbol. Éste es un algoritmo de búsqueda sin información, que expande y examina todos los nodos de un árbol sistemáticamente para buscar una solución.

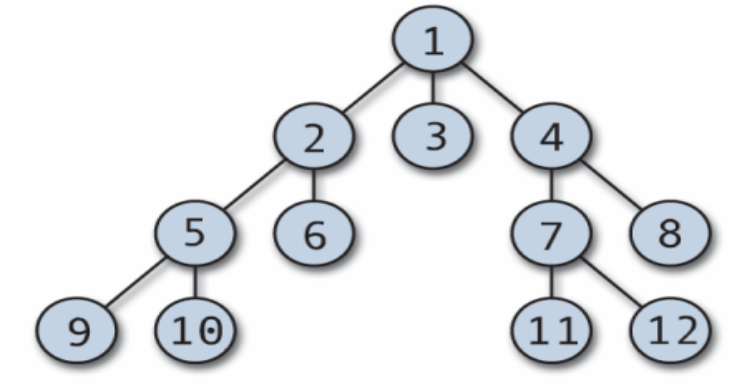


Figura 2.4.1 Orden en la exploración de nodos en Breadth-first.

- **Depth-first (Búsquela en profundidad).**

  Depth First Search o búsquela en profundidad es un algoritmo que permite recorrer todos los nodos de un grafo o árbol de manera ordenada, pero no uniforme. Su funcionamiento consiste en ir expandiendo todos y cada uno de los nodos que va localizando, de forma recursiva, en un camino concreto.

  Cuando ya no quedan más nodos que visitar en dicho camino, regresa, de modo que repite el mismo proceso con cada uno de los hermanos del nodo ya procesado.

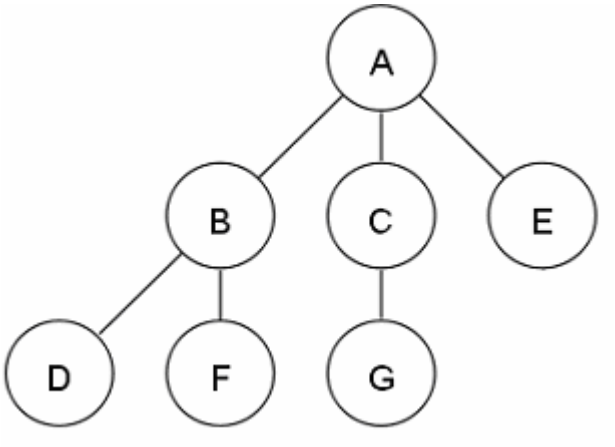


Figura 2.4.2 Exploración de nodos en Depth-first.

2.4.2 IMPLEMENTACIÓN DE ESCRUTINIO DE SITIOS WEB.

En esta sección se describen las consideraciones prácticas de la implementación del escrutinio de sitios. Es decir el hecho de analizar las páginas de un sitio para describir las estructuras de éste encontrando los enlaces a otros sitios.

2.4.2.1 ESCRUTINIO SIMPLE.

El procedimiento para el escrutinio de sitios se realiza de la siguiente manera: un robot encuentras todos los enlaces en un sitio Web, lee las paginas referenciadas por los enlaces y repite el proceso con cada una de ellas.

Descubrimiento de enlaces (Links).

El descubrimiento de enlaces por parte del robot es sencillo, lo único que se tienen que hacer es localizar la etiqueta enlace <*A HREF="enlace"*> en el documento HTML.

2.4.2.2 ESCRUTINIO SOFISTICADO.

En una instancia de búsqueda, es necesario hacer énfasis en la búsqueda de enlaces, sin embargo, para un robot mas completo y eficiente es importante además poner especial atención en aspectos tales como: evitar la redundancia, efectuar discriminación, limitar el ámbito y limitar la profundidad.

2.4.2.2.1 Evitar la Redundancia.

Un robot puede verse atrapado en un ciclo infinito si no se tiene cuidado al seguir los enlaces a otros dominios, o al mismo tiempo puede seguir una ruta cíclica. El escrutinio de sitios genera una lista de URL's visitados, sin embargo, puede evitarse la repetición, revisando que los URL's obtenidos no existan en dicha lista antes de agregarlos y considerarlos para continuar en el escrutinio.

Un URL se compone de varias partes como ya se detallo anteriormente. Pero aun dos URL's parecidos o totalmente diferentes pueden referirse al mismo recurso o pagina.

Un ejemplo seria el siguiente: se tiene una página Web que posee un enlace a *http://www.microsoft.com* y en algún otro de la misma página indica un enlace a *http://www.microsoft.com:80,* ambas refieren a la misma dirección Web aun cuando son cadenas distintas.

Por lo tanto, cuando se programa un robot Web, éste debe tener la suficiente inteligencia como para detectar tales situaciones para no caer en ciclo redundante.

2.4.2.2.2 Efectuar la Discriminación.

El escrutinio es el primer paso para encontrar enlaces dentro de una página Web. Cuando ya se obtienen dichos enlaces, es necesario que se determine hacia donde se esta apuntando y que es, ya que, frecuentemente se tendrán que ignorar enlaces a otro tipo de datos como imágenes y/o sonidos, debido a que el robot Web solo se interesará por seguir los enlaces hacia otras paginas o sitios Web.

2.4.2.2.3 Discriminación por Tipos de Archivos.

Debido a que no es regla que los enlaces deben precisar su extensión, es decir, el tipo de archivo al que están apuntando, el robot Web debe ser capaz de decidir cuando seguir el enlace y cuando ignorarlo. Aun cuando fuera obligatorio, seria una regla fácil de romper, dada la flexibilidad de sintaxis de HTML y los navegadores por su alta tolerancia a fallas de sintaxis.

Los mecanismos para la recuperación de información son variados, sin embargo la lógica para la especificación de tipos es muy sencilla: los documentos HTML mayormente son definidos con extensiones *.html, .htm, .php, .asp* y si el enlace no incluye extensión, generalmente se asume que está a un archivo diferente a una página Web.

Si el robot utiliza el criterio anterior ¿Que decisión tomaría el robot cuando se encuentre con una archivo con extensión *.dll o .exe*?

Tales extensiones se refieren a un archivo que tiene que descargarse *(download)* y un archivo ejecutable respectivamente, además también existe la posibilidad que se encuentre un archivo con extensión desconocida, por lo tanto, ¿se deben seguir esos enlaces o se deben ignorar?

Una solución, para ayudar este problema seria la de evaluar y leer los primeros y últimos bytes del archivo para encontrar etiquetas *<HTML> y </HTML>.* Sin embargo, surge otra interrogante ¿Que pasaría con documentos de texto *(.doc, .txt, etc.),* que poseen valor informativo pero no en formato HTML?

En respuesta a lo último, se construye una lista con todos los tipos conocidos de archivos con sus extensiones asociadas. Tal lista incluiría extensiones las cuales se confía que regresaría un documento HTML cuando se accedan.

2.4.2.2.4 Discriminación Server Polling.

Una técnica conocida por encuesta al servidor (Server Polling), es una forma detallada de abordar la discriminación. Consiste en solicitar el encabezado del archivo apuntado a través del comando HEADER del HTTP para cada enlace, con la finalidad de investigar cuales son los datos del documento que el servidor tiene y que podrían ser de utilidad al momento de decidir si se discrimina o no un determinado enlace. El encabezado devuelto por el servidor, tiene información que normalmente incluye una descripción del tipo de archivo.

Cuando se esta realizando una búsqueda de páginas HTML, el campo de encabezado *Content-Type: text/html* determina que el enlace apunta a un documento HTML.

El uso de esta técnica es más confiable que hacer suposiciones basados en el tipo de archivo según su extensión. Sin embargo nos encontramos con que el robot tiene que emitir un comando HEADER por cada URL encontrado en cada página. Estos accesos adicionales a Internet pueden reducir el rendimiento del proceso del escrutinio, lo que significa, que lo que se gana en detalle se sacrifica en rendimiento.

Desde luego que, se pueden combinar ambas técnicas. Se podrían discriminar los archivos que no son HTML tales como .gif, .jpg, .mp3, etc. Y el resto, tratarlos según la técnica de discriminación por encuesta al servidor.

2.4.2.2.5 Limitar el Ámbito.

Debido a las características que posee un robot Web tales como el alto consumo de recursos (memoria, comunicación, atención del servidor, entre otros), el alto nivel de control que se debe ejercer sobre el mismo y la posibilidad de caer en ciclos infinitos, es lógico y entendible porqué debe limitarse al ámbito del escrutinio.

Un determinado sitio o pagina Web, puede hacer referencia (apuntar) a cualquier otra en la Web y realizar el escrutinio en un sitio Web, puede convertirse en una indagación a una multitud de sitios. Una forma común de limitar al ámbito es seguir solo aquellos enlaces que guíen al mismo servidor Web o al mismo dominio.

Limitar el ámbito al servidor Web significa que el robot sólo atenderá dos tipos de enlaces: URL absolutos que incluyan el nombre del servidor actual y URL relativos que por definición guían al mismo servidor.
Limitar el ámbito al dominio es casi igual, en lugar de cuidar que la referencia absoluta sea al mismo servidor Web, deberá cuidarse que sea al mismo dominio. Bajo ese esquema el robot no deberá seguir enlaces a la misma página para evitar ciclos infinitos.

2.4.2.2.6 Limitar la Profundidad del Escrutinio.

Aun evitando la redundancia, realizando la discriminación y limitando el ámbito, el escrutinio puede presentar problemas, principalmente para los servidores quienes lo atienden. El escrutinio es recursivo por naturaleza. Un robot Web analiza una pagina Web en la exploración de enlaces hacia otras paginas Web y repite el proceso ahora con cada uno de esos documentos HTML.

Lo que se tiene que hacer durante el escrutinio es, disminuir un nivel cada vez que sigue un enlace y aumentarlo cuando regresa al mismo nivel del que partió.

2.4.2.2.7 Otras Consideraciones.

2.4.2.2.7.1 Marcos (Frames).

Algunas páginas Web definen áreas de su contenido como Marcos y hacen referencia a otras páginas que proveen el contenido de tales marcos.

Para definir tales marcos se una las etiquetas HTML <FRAMESET> y </FRAMESET>. Ejemplo:

```
<FRAMESET ROWS="*,25" FRAMEBORDER="0" FRAMESPACING="0" BORDER="NO">
        <FRAMSE COLS="100,*">
                <FRAME SCR="index.htm" NAME="principal" MARGINWITH=0>
                <FRAME SCR="pagina1.htm" NAME="pagina" MARGINWITH=10>
        </FRAMESET>
        <FRAME SCR="enlaces.htm" NAME="enlaces" SCROLLING=NO
MARGINWIDTH=0>
</FRAMESET>
```

El código HTML anterior hace referencia a las paginas *index.htm, pagina1.htm y enlaces .htm.*

Un robot Web completo tendría que distinguir y manejar sin problemas los enlaces a través de marcos, al igual, que con enlaces directos usando la etiqueta *<A HREF=url>.*

2.4.2.2.7.2 Contenidos Dinámicos.

El escrutinio es una técnica que funciona bien con páginas estáticas en las cuales la página principal se enlaza con otras en el sitio. Pero las páginas que se genera dinámicamente en base a parámetros representan un problema para la función de escrutinio del robot ya que no reconocerán como enlaces en una página Web.

Como por ejemplo las paginas de grupo de noticias. Dicha información se presenta en una página solo cuando el usuario especifica ciertos enlaces que determinan el contenido de ésta.

En este caso *no* solo no habrá enlace a la pagina, sino, que la pagina *no existirá* hasta que el cliente la solicite, que es el momento en que el servidor la crea.

Preguntas de repaso.

1. Mencione los dos recorridos y su significado que utiliza el Algoritmo de Crawling.

   _____
   _____
   _____.

2. ¿Enumere las extensiones con las cuales son definidos los documentos HTML?

   _____
   _____.

3. De una definición de discriminación Server Polling.

   _____
   _____
   _____.

4. En limitar el ámbito al servidor, que tipo de enlaces se siguen para guiar al robot Web desde y hacia el mismo servidor,

   _____
   _____.

2.4.3 ALGORITMO DE RANKING.

No existe mucha información pública detallada sobre los algoritmos de clasificación utilizados por los motores actuales, debido a que la mayoría de ellos están patentados.

Lo que se documenta, es el número de hiperenlaces que apuntan a una página que proporcionan una medida de su popularidad y calidad. Además, paginas conectadas entre sí referenciadas desde la misma pagina podrían tener contenido similar.

A continuación se presentan ejemplos de rankings basados en análisis de links.

- **WebQuery.**

  Utiliza métodos locales a partir de la respuesta normal, permite visualizar las páginas Web. Toma un conjunto de páginas Web (por ejemplo, la respuesta a una consulta) y ordena en base a cuan conectadas están. Además, extiende el resultado con las páginas que están muy conectadas al conjunto recuperado.

- **HITS (*Hypertext Induced Topic Search*).**

  Desarrollado por Kleinberg en 1998. Depende de la consulta. Este esquema toma el conjunto de páginas de la respuesta más las que están a un enlace de distancia (directo o inverso). Intenta determinar computacionalmente las autoridades y los *hubs* para un tema determinado.

  Llámese *autoridades* a las páginas que reciben muchos links desde S (que deberán tener contenido relevante) y *hubs* a las que apuntan mucho a S (deberán apuntar a contenido similar). Siendo S el conjunto base.

  Los *hubs* y *autoridades* tienden a formar un grafo bipartito[12] en donde los *hubs* apuntan a múltiples *autoridades* y las *autoridades* son apuntadas por múltiples *hubs.*

  El algoritmo funciona calculando las autoridades y *hubs* para un tema concreto, a partir de una consulta normal. Luego determina el conjunto de páginas relevantes para la consulta, que es el conjunto base S.

  Luego analiza la estructura de hiperenlaces del subgrafo definido por S, para buscar *autoridades* y *hubs* en ese conjunto.

[12] Grafo bipartito, es aquel cuyos vértices pueden formarse dos conjuntos distintos de tal manera que no haya adyacencia entre vértices pertenecientes al mismo conjunto.

Funcionamiento del Algoritmo:

- Para una consulta Q, se denomina R al conjunto de documentos recuperados por un motor de búsqueda estándar.
- Inicializar S a R.
- Añadir a S todas las páginas a las que apunta cualquier página de R.

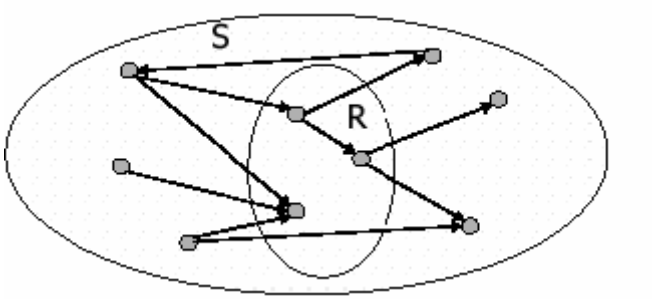


Figura 2.4.3. **Funcionamiento Algoritmo HITS.**

El problema de este sistema es que el contenido se puede difuminar, pero la precisión y recuperación mejoran significativamente.

- **PageRank.**

Es parte del algoritmo de ranking utilizado por Google.

Es un sistema de votación que mide la importancia de una página en función de todos los enlaces de la Web.

Asigna un valor numérico a cada página que representa su importancia en Internet. Así, si una página coloca un enlace a otra realiza un voto para esta última. Cuantos más votos tenga una página, más importante es. La importancia de la pagina que vota determina el *"peso del voto".*

El algoritmo original del PageRank fue descrito en varios trabajos Sergey Brin y Lawrence Page.

Se define el PageRank por:

$$PR(A) = \frac{(1-d)}{N} + d\left(\frac{PR(T_1)}{C(T_1)} + \ldots + \frac{PR(T_n)}{C(T_n)}\right)$$

Figura 2.4.4 Fórmula de PageRank.

Donde:

N es el número total de páginas Web desde las que salen vínculos.

n es el numero total de paginas Web desde las que salen vínculos a la página A.

PR(A) es el PageRank de la pagina A.

PR(Ti) es el PageRank de las paginas *Ti* que tienen un vínculo hacia la pagina A.

C(Ti) es el numero de vínculos salientes de la pagina *Ti.*

d es un factor de amortiguación que puede ser tomado entre 0 y 1.

- **Ponderación de Páginas.**

Encontrar documentos con el contenido deseado no resulta tan difícil, lo difícil es encontrar documentos indicados sobre el tema que se esta buscando.

En la Web se encuentran millones de documentos y difícilmente una persona podrá realizar un trabajo productivo si el buscador le entrega miles de páginas para que las evalúe por separado. Así que resulta indispensable ponderar dichas páginas. Existen varios métodos para hacerlos y la mayoría se basa en la presencia (mayor o menor) de determinado sitio en la Web. Dicho de otra manera, los sitios son importantes en la medida en que el resto de los sitios de la Web lo consideren así.

- **Ponderación por conteo comparativo.**

  Una forma de ponderar consiste en contar el número de veces que se localizan las palabras de la consulta en los documentos encontrados, así mismo se considera un factor extra*: si una o alguna de las palabras de la consulta se encontraron en un documento en particular y no en los demás, entonces ésa es una buena evidencia de que el documento es importante y se le asigna puntuación adicional.*

- **Ponderación por referencias**

  Una alternativa de ponderación de páginas Web es que utiliza los enlaces como evidencia de buenos contenidos. Una forma simple de plantear esto es la siguiente hipótesis: una pagina con buen contenido, es seguramente referenciada desde muchos índices.

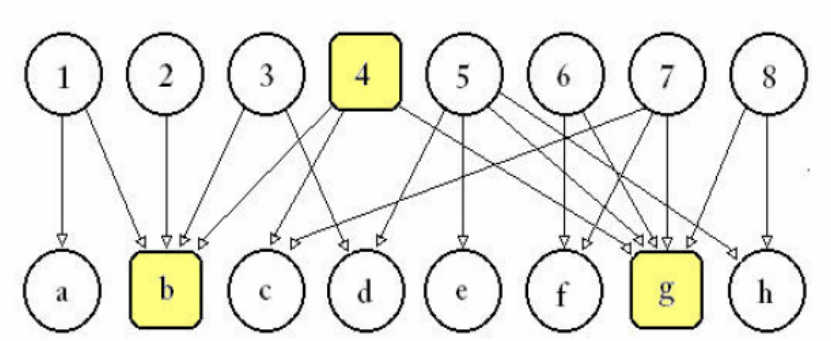


Figura 2.4.5 Ponderación de páginas Web por referencias.

En la figura anterior se tienen numerados algunos índices potenciales y con las letras los potenciales paginas de contenido. Observe que *"b"* y *"g"* atraen a la mayoría de los enlaces desde otras páginas, por lo cual se marcan como mejores páginas de contenido. De la misma forma la pagina *"4"* se detecta como mejor pagina índice, precisamente porque apunta a las dos mejores paginas de contenido. No es la página con mayor número de enlaces, de hecho *"5"* tiene más enlaces pero no son los más apropiados. Al mismo tiempo la puntuación para "*c"* como página de contenido aumenta porque aparece una fuente de información, este procedimiento se ejecuta recursivamente sumando puntos a las páginas como índices y como contenido.

2.4.4 EXCLUSIÓN DE ROBOTS.

Debido a que los robots spiders funcionan de manera automática, éstos pueden causar un problema de sobrecarga hacia los servidores Web debido a las constantes peticiones que realiza un robot, con el fin de indexar un sitio Web específico.

También se puede dar el caso sobre la creación de un robot spider de forma experimental y posterior ejecución en Internet, el cual conlleva, una serie de riesgos e inconvenientes por el mal diseño en la programación del mismo, cuando se le proporciona una URL desde donde se comienza a realizar el escrutinio de sitios Web.

Lo anterior llevo a la implementación de métodos para propietarios o administradores de sitios Web que no quieren ser explorados (o que desean poner algún tipo de limitación) hacia los robots spiders.

Para tal efecto existen dos mecanismos que los servidores Web implementan para limitar el acceso de dichos robots: *El protocolo de exclusión de robots y las etiquetas <META>.*

2.4.4.1 PROTOCOLO DE EXCLUSIÓN DE ROBOTS.

El *Robots Exclusion Protocol* es un método que permite a los administradores de sitios Web indicar a los robots que visiten sus páginas qué partes de sus sitios no deberían ser visitados por éste. Es una manera de proteger a los servidores de accesos indeseados.

En Junio de 1994, Martjin Koster, Jonathon Fletcher y Lee McLoghlin, con un grupo de autores y entusiastas de los robots, crearon un documento llamado "*A Standard for Robot Exclusión"*[13]. Este documento provee de una vía en la cual un administrador de servidores Web podía controlar a qué robots se le permitían el acceso al servidor, y a qué sitios de dicho servidor podían acceder los robots Web.

El *Standard for Robot Exclusion* simplemente establece que un robot lo primero que va a mirar en el *root* del servidor es un fichero llamado *robots.txt.*

[13] http://fjt.cnitn.cn:89/gate/gb/www.robotstxt.org/orig.html

Si encuentra el fichero, el robot lo lee y actúa de acuerdo a las limitaciones impuestas en dicho fichero.
El nombre del fichero *robots.txt* fue elegido porque la mayoría de sistemas operativos podían usarlo, y además no requiere de configuraciones complicadas por parte del administrador, ya que la extensión **.txt* lo permiten la mayoría de servidores. Además el nombre es fácil de recordar, y fácil de reconocer.

El fichero *robots.txt.*
El "Protocolo de Exclusión de Robots" se basa en la especificación del contenido de un fichero *ROBOTS.TXT* en el que se presentan las instrucciones de comportamiento para los robots en relación con las páginas Web. Dicho fichero debe residir en el directorio raíz del servidor Web (es decir, en el directorio donde se encuentra la página por defecto). Por ejemplo, en el Web de la revista: http://en.wikipedia.org. Es el archivo *ROBOTS.TXT* sería accesible a través del siguiente URL:
http://en.wikipedia.org/robots.txt

Este fichero está compuesto por una serie de registros separados por líneas en blanco.
La presencia de un archivo *"robots.txt"* vació equivale la no existencia de éste. Y esto a su vez equivale a que no hay restricción de acceso para robots, o sea, todos los robots son bienvenidos.

Cada registro tiene el siguiente formato:
DIRECTIVA:" espacio opcional " VALOR "espacio opcional"
Comentario

El nombre de la directiva puede ir en mayúsculas o minúsculas indistintamente. Las directivas posibles son:

- *User-agent* (agente de usuario).
- *Disallow* (no permitir).

Cada registro comenzará con una o más líneas con directivas *User-agent*, y a continuación, una o más líneas con directivas *Disallow.* El significado concreto de cada una de estas dos directivas es el siguiente:

- *User-agent* (agente de usuario): Define, para el registro en el que se encuentre, el nombre del robot para el que se dan normas de acceso. Se pueden definir varios agentes en el mismo registro si las normas son las mismas para todos. Si el valor es *"*"*, significa que el registro describe reglas de acceso para el resto de los robots que no han sido incluidos en otros registros. Por tanto, sólo un registro del fichero *ROBOTS.TXT* podrá indicar como *"*"* el *User-agent,* si se quiere mantener todos los robots fuera del servidor.

- *Disallow* (no permitir): Define la *URL* que no debe ser inspeccionada por el robot o robots indicados en *User-agent*. La *URL* puede ser parcial o completa. Por ejemplo:

  Disallow: /doc

  No permite el acceso ni *a /doc.html, /doc/doc1.htm, /doc/doc2.htm,* etc.

  Disallow: /doc/

  No permite el acceso a todos los archivos del directorio */doc* pero sí permite el acceso a */doc.htm.*

  Si el valor de la directiva está vacío significa que el robot indicado en el registro puede inspeccionar todas las *URL's.*

Protocolo de exclusión de robots: Robots Bien Portados Vs. Robots Mal Portados.

Un Robot "bien portado" es aquel que respeta el estándar de exclusión de robots. Un robot que ignora este estándar, simplemente no leyendo al archivo *robots.txt* y hacer peticiones constantes al servidor, sin importar los problemas que sus acciones podrían causar, se convertirá en un Robot "mal portado" y por lo tanto, los administradoras de sitios Web tendrán que adoptar medidas para el control de accesos no deseados.

Ventajas de utilizar el archivo robots.txt:

- Evita problemas con un robot en particular.
- Optimiza recursos y no da paso a muchos robots spiders.

Desventajas de utilizar el archivo robots.txt:

- Se podría negar el acceso a motores de búsqueda importantes y que son utilizados por la mayoría de usuarios.
- No se puede asegurar que todos los robots spider respeten las reglas que se encuentren en el archivo *robots.txt.*

Ejemplos de ficheros ROBOTS.TXT:

1. Si no queremos ningún robot inspeccione nuestro sitio Web, el fichero deberá contener lo siguiente:

   User-agent: *
   Disallow: /

2. Si queremos impedir que cualquier robot visite cualquier URL que comience por */doc/confidencial/ y /doc/personal/,* deberá contener un registro con las siguientes directivas:

   User-agent: *
   #documentos confidenciales de la empresa
   Disallow: /doc/confidencial/

   #documentos personales
   Disallow: /doc/personal/

3. Si se quiere conseguir lo mismo que en el ejemplo anterior excepto para un robot especifico (por ejemplo *robot_x*), el fichero será:

   # Todos los Robots tienen restringido el acceso.
   User-agent: *
   Disallow: /doc/confidencial/
   Disallow: /doc/personal/

   # El Robot: "robot_x" tiene permitido el acceso.
   User-agent: robot_x
   Disallow:

En las siguientes imágenes se muestra como esta hecha la configuración del archivo *robots.txt* para diferentes páginas Web.

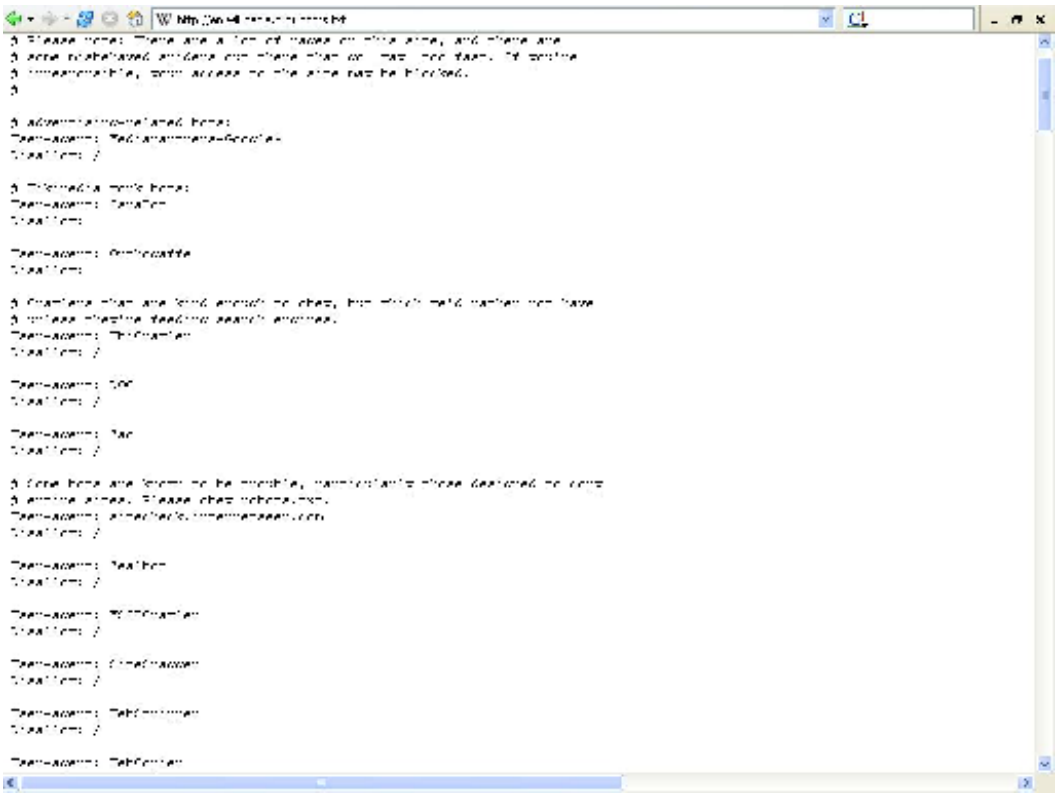


Figura 2.4.6 Ejemplo de archivo *robots.txt* para la página Web *http://en.wikipedia.org/robots.txt*

Figura 2.4.7 Ejemplo de archivo *robots.txt* para la pagina Web *http://www.cnn.com/robots.txt*

Figura 2.4.8 Ejemplo de archivo *robots.txt* para la pagina Web *http://www.udb.edu.sv/robots.txt*

2.4.4.2 ETIQUETAS <META> PARA ROBOTS.

Las etiquetas <META> para robots es una técnica que permite a los autores de sitios Web, indicar a los robots visitantes si tienen permiso para que dicho sitio sea indexado o usado para descubrir nuevos enlaces. No es necesaria ninguna acción por parte del administrador del servidor.

Sintaxis de etiqueta <META> para robots:
<META NAME="Robots", CONTENT="instrucciones">

Donde *"instrucciones"* especifican dos tipos de acciones que deberías seguir los robots que analicen gramaticalmente la pagina Web en donde se incruste esta etiqueta META. Estas acciones son [NO]INDEX y [NO]FOLLOW. En donde INDEX indica al robot que puede incluir el documento en su base de datos. Si INDEX es precedido por NO, la instrucción es entonces que la pagina no deberá ser incluida en la base de datos del robot.

FOLLOW indica que la pagina puede ser analizada para encontrar enlaces, si se precede de NO, indica que no debe ser analizada en busca de enlaces.

Ejemplo:

<META NAME="robots" CONTENT="noindex, nofollow">

Esto especifica que el robot que solicite la pagina Web, no debería incluirlo en la base de datos (indexar), ni analizarla para descubrir nuevos enlaces.

Otras combinaciones posibles son:

<META NAME="robots" CONTENT="noindex, follow">

<META NAME="robots" CONTENT="index, nofollow">

<META NAME="robots" CONTENT="index, follow">

Las etiquetas <META> para robots y contenido (CONTENT) no son sensibles a mayúsculas ni minúsculas.

Preguntas de repaso.

1. Clasifique los siguientes algoritmos de Ranking mencionados de acuerdo a sus características:

   Depende de la consulta y forman un grafo bipartito. _____.

   Toma un conjunto de páginas Web y las ordena en base a cuan conectadas están.

   _____.

   Realiza un sistema de votación para medir la importancia de una página en función de todos los enlaces de la Web. _____.

2. Explique que es la ponderación por conteo comparativo.

   _____

   _____

   _____

   _____.

3. Explique en que consiste el Protocolo de Exclusión de Robots
_____
_____
_____.

4. Escriba el concepto (*User-agent, Disallow, valor asterisco (*), INDEX, [NO] INDEX, FOLLOW, [NO] FOLLOW*) en el espacio en blanco que mejor se adapte a la definición dada:

   Indica al robot que puede incluir el documento en su base de datos. _____.

   Define la *URL* que no debe ser inspeccionada por el robot.

   _____.

   Indica que no debe ser analizada para búsqueda de enlaces. _____.

Auto evaluación.

Instrucciones.

En la hoja provista, subraye la respuesta que corresponda. Coteje las respuestas al final de la auto evaluación.

1. Procedimiento el cual un robot encuentra enlaces en un sitio Web y repite el proceso en cada enlace.
    a. Depth-fist.
    b. Escrutinio simple.
    c. Breadth-first.

2. Técnica que consiste en solicitar el encabezado del archivo apuntado a través del comando HEADER del HTTP
    a. Escrutinio simple.
    b. Escrutinio sofisticado.
    c. Discriminación Server Polling.

3. Es seguir solo aquellos enlaces que guíen al mismo servidor Web o al mismo dominio.
    a. Discriminación Server Polling.
    b. Limitar el ámbito del escrutinio.
    c. Escrutinio de sitios Web.

4. Algoritmo de ranking que mide la importancia de una página utilizado por Google.
    a. WebQuery.
    b. HITS.
    c. PageRank.

5. Mencione los dos mecanismos que existen para la exclusión de robots.
    a. Protocolo de Exclusión de Robots y etiquetas <META>.
    b. HTML y Etiquetas <META>.
    c. Fichero ROBOTS.TXT y HELP.TXT.

De 5 a 3 preguntas correctas: Puede continuar con el capitulo 5.
Menos de 3 preguntas: Repase el capítulo 4 y compruebe el ejercicio.

Respuestas:
1. B, 2. C, 3. B, 4. C, 5. A

Investigación Complementaria.

- Explique con sus propias palabras que es el protocolo de exclusión de robots.
- Describa que son las etiquetas <META> en la exclusión de robots.
- Consulte un fichero ROBOTS.TXT en una página Web, identifique y analice las directivas que ahí se encuentran.

2.5: COMBINE SYSTEM.

Tiempo recomendado de estudio: 4 horas

Objetivos:

Al concluir este capitulo el lector estará capacitado para:

- Instalar el robot Combine System en sistemas Linux Debian y Linux Ubuntu de una manera rápida haciendo uso del sistema de paquetes *apt.*

- Configurar dominios, tópicos específicos, componentes del sistema y variables para realizar una búsqueda de prueba en Combine System desde la consola.

- Realizar una búsqueda de tópicos específicos, definiendo los archivos de configuración y de URL's semillas.

Introducción:

El Combine System es un sistema de búsqueda de recursos en Internet él cual es abierto, de libre distribución y altamente configurable. También posee una robusta y eficiente herramienta para la creación de bases de datos de un tamaño moderado (cerca del millón de registros), acerca de temas específicos.

La velocidad de escrutinio ronda las 200 URL's por minuto y una completa estructura de registros toma unos 25 Kilobytes de espacio en disco duro.

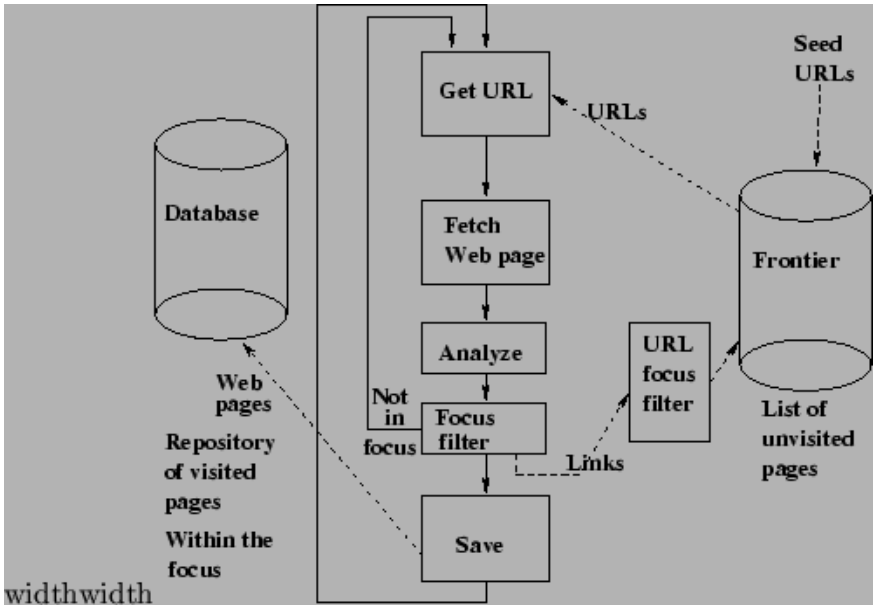


Figura 2.5.1 Búsqueda enfocada en Combine System.

Las características principales del robot Combine System son:

- Parte de un sistema de búsqueda de ingeniera en una caja (SearchEngine-in-a-Box System).
- Grandes posibilidades de configuración.
- Filtro de tópicos integrado (clasificador de tópicos automático) para el modo de escrutinio especifico. (focused crawling mode).
- Posibilidad de utilizar cualquier filtro de tópico (Si se ha dado el Perl Plug-In module) en el modo de escrutinio especifico.
- Limitaciones de escrutinio basadas en expresiones regulares en URL's.
- Detección de lenguaje.
- Limpiador HTML.

- Extracción de metadatos.
- Detección de duplicados.
- Análisis de HTML para estructurar cada uno de los registros por cada página examinada.
- Soporte de muchos formatos de documentos (Text, HTML, PostScript, MSWord, MSPowerPoint, MSExcel, RTF, TeX, images).
- Uso de una base de datos SQL para guardar y administrar los datos.

Naturalmente Combine System, obedece el protocolo de exclusión de robots y se comporta bien con los servidores Web. Además posee escrutinios enfocados (generando bases de datos de tópicos específicos), soporta reglas de configuración para el escrutinio basado en expresiones regulares en URL's. El Crawler esta diseñado para correr continuamente para mantener la base de datos lo mas actualizada posible, esté puede ser detenido o reiniciado en cualquier momento sin perder información o estado.
La operación del Combine System mostrada en la figura 2.5.1, de una búsqueda enfocada esta basada en la combinación de un Web Crawler general y un clasificador automático de temas.

El tema enfocado lo da un filtro de enfoque usando una definición de tema implementado como un diccionario, donde cada termino esta conectado a una clase temática.
Los datos del escrutinio son almacenados en una base de datos local relacional.

2.5.1 INSTALACIÓN DEL COMBINE SYSTEM.

El Crawler enfocado ha sido reestructurado como un paquete Debian, esto con el objetivo de hacer más fácil su distribución e instalación. El paquete contiene información de las dependencias para asegurarse que todo el software que se necesite para correr el Crawler sea instalado al mismo tiempo. En unión con esto se tiene también empaquetado un número necesario de módulos de Perl como paquetes Debian.

2.5.1.1 INSTALACIÓN EN DISTRIBUCIÓN LINUX DEBIAN 4.0.

Los siguientes pasos fueron realizados en la distribución Debian 4.0, pero, al ser Ubuntu un derivado de la distribución antes mencionada, es posible utilizar los mismos pasos para las distribuciones Ubuntu, estos pasos son los más sencillos y se propone que la instalación del Combine System se realice de preferencia en cualquiera de estas dos distribuciones o en sus derivados.

Pasos:

1- Añada las siguientes líneas en el archivo /etc/apt/sources.list:
deb http:// combine.it.lth.se/ /debian

2- Digite los siguientes comandos para actualizar el repositorio de paquetes:
apt-get update

3- Digite los siguientes comandos para instalar el Combine System:
apt-get install combine

A continuación se muestra el procedimiento completo y sus resultados:

- Iniciamos sesión como el usuario *root* y añadimos las líneas de los repositorios de software del Combine System en el archivo */etc/apt/sources.list.* (En este caso se hizo uso del editor de texto Pico de Unix).

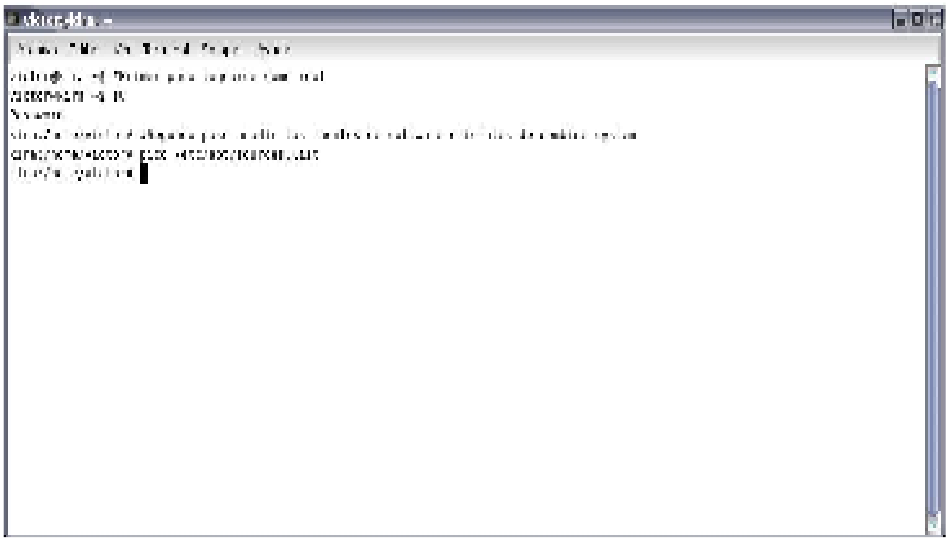


Figura 2.5.2 Inicio de sesión como usuario root.

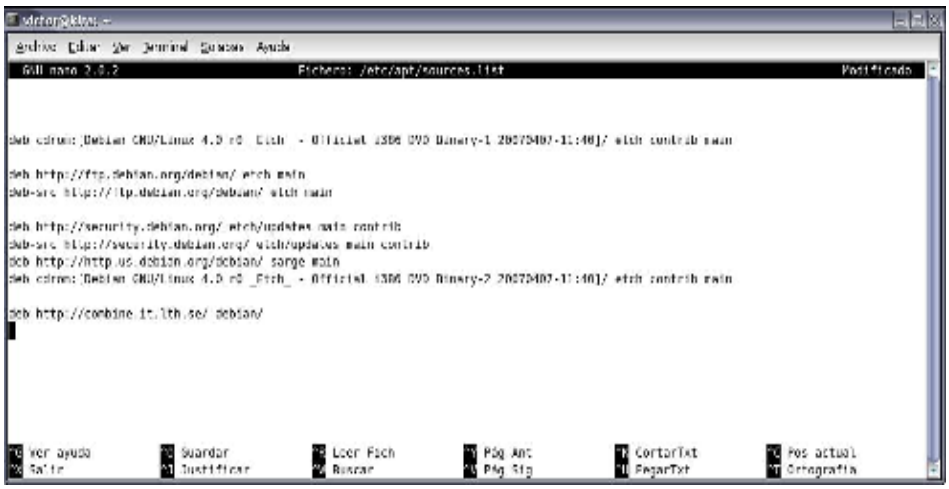


Figura 2.5.3 Archivo sources.list.

- A continuación actualizamos los repositorios de software del apt

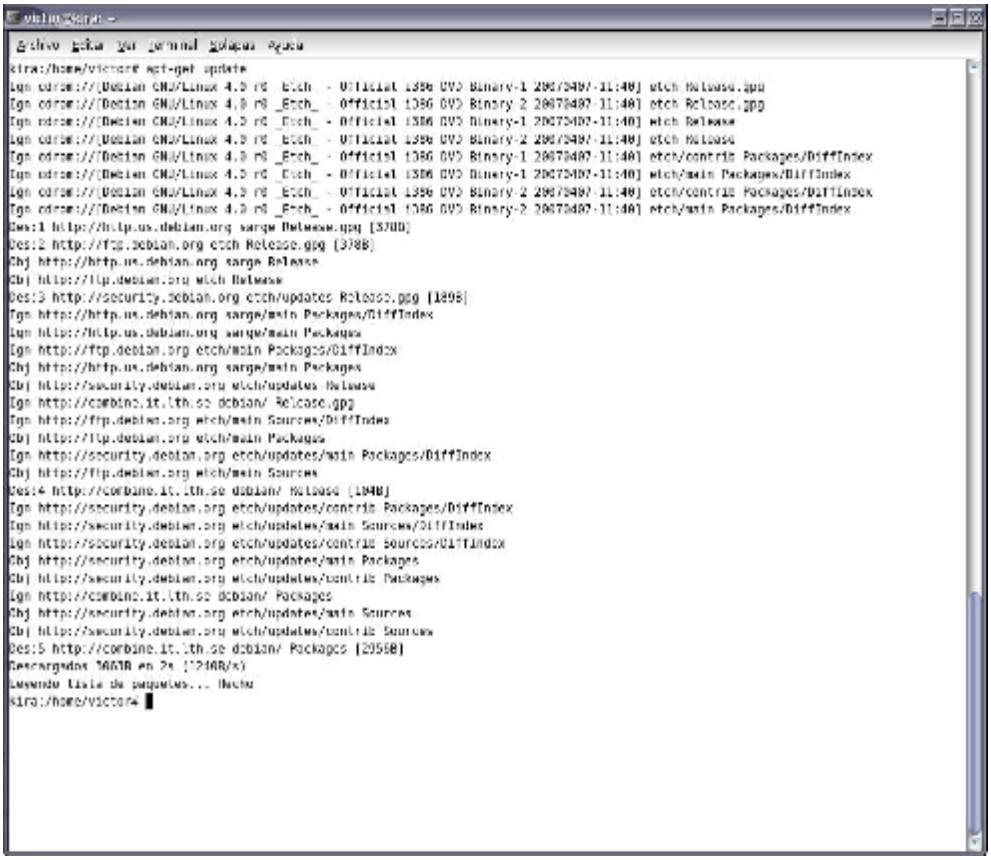


Figura 2.5.4 Actualización de los repositorios de software Debian.

- Instale el Combine System haciendo uso del comando a*pt-get install combine* esto automáticamente instalará además MySQL 5.0 y muchos módulos de Perl necesarios.

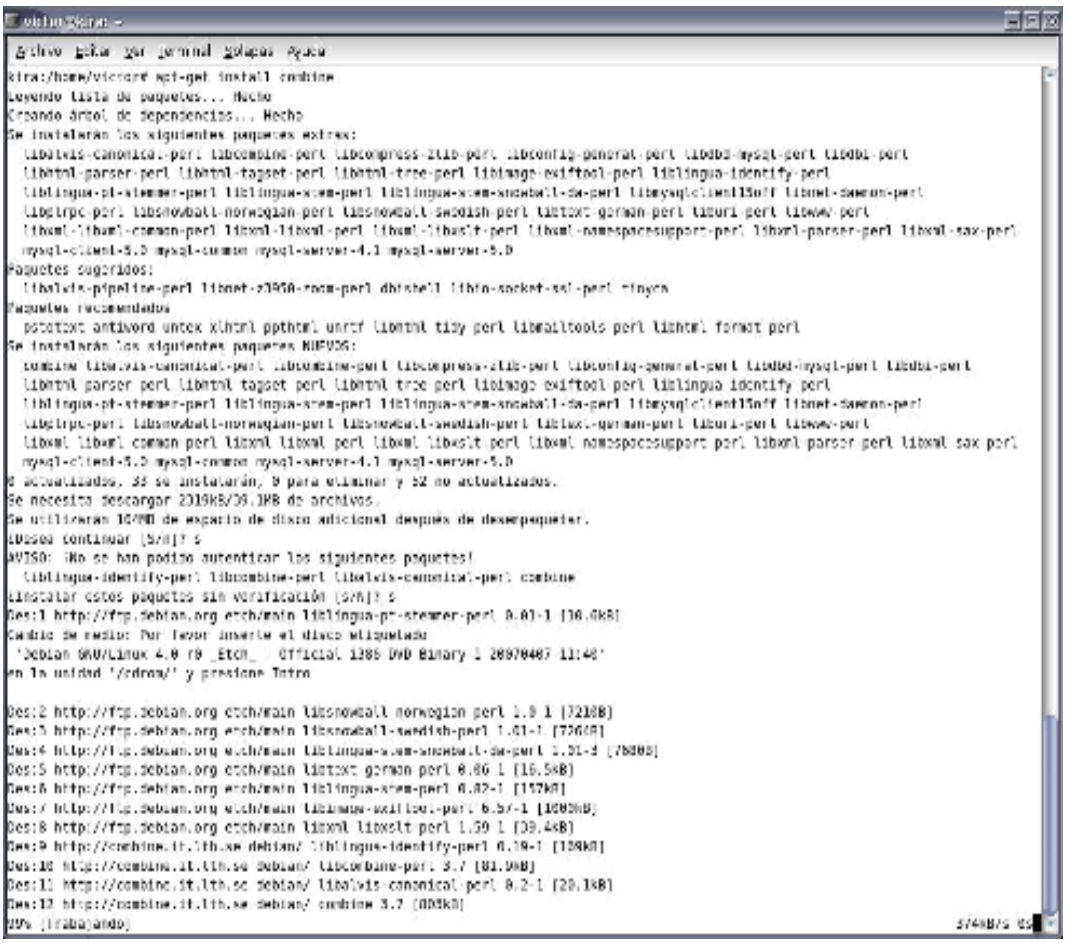


Figura 2.5.5 Comando de instalación del Combine System.

Puede que algunos de los paquetes del repositorio no se encuentren autenticados, entonces para instalar estos paquetes el programa pedirá los discos de la distribución Debian, por ende nos será de utilidad tenerlos a la mano.

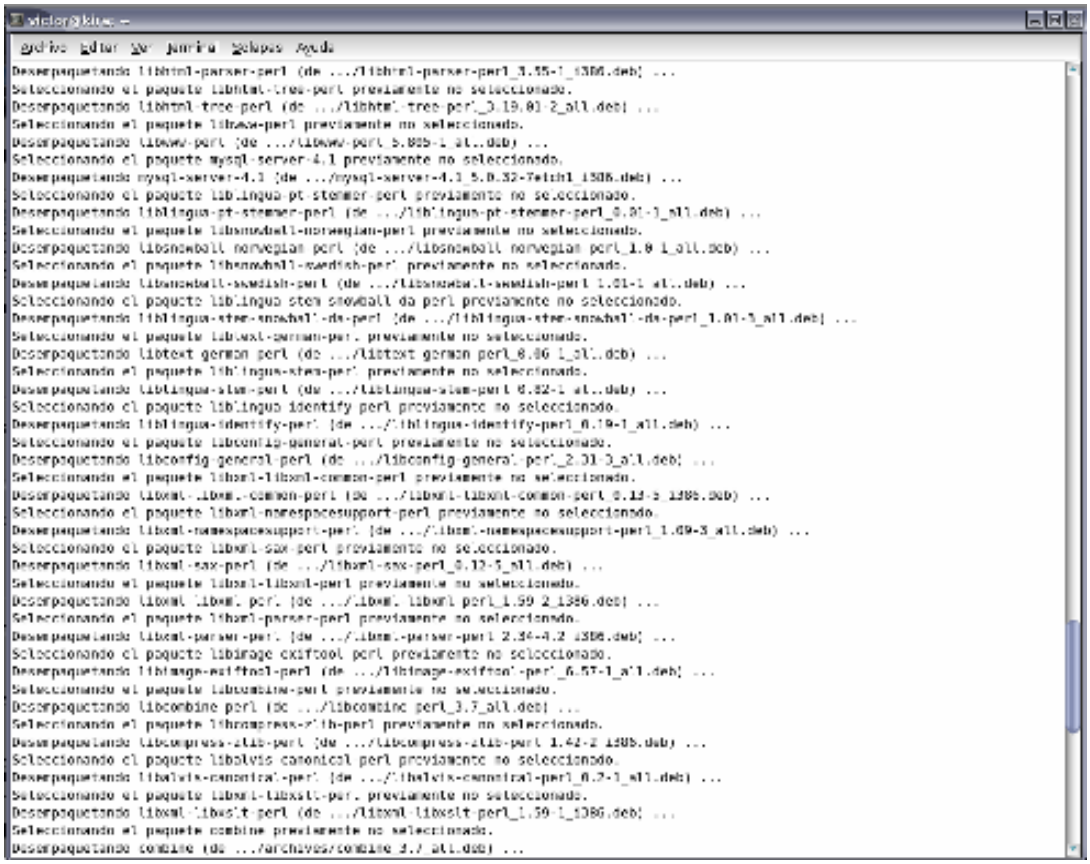


Figura 2.5.6 Instalación del Combine System.

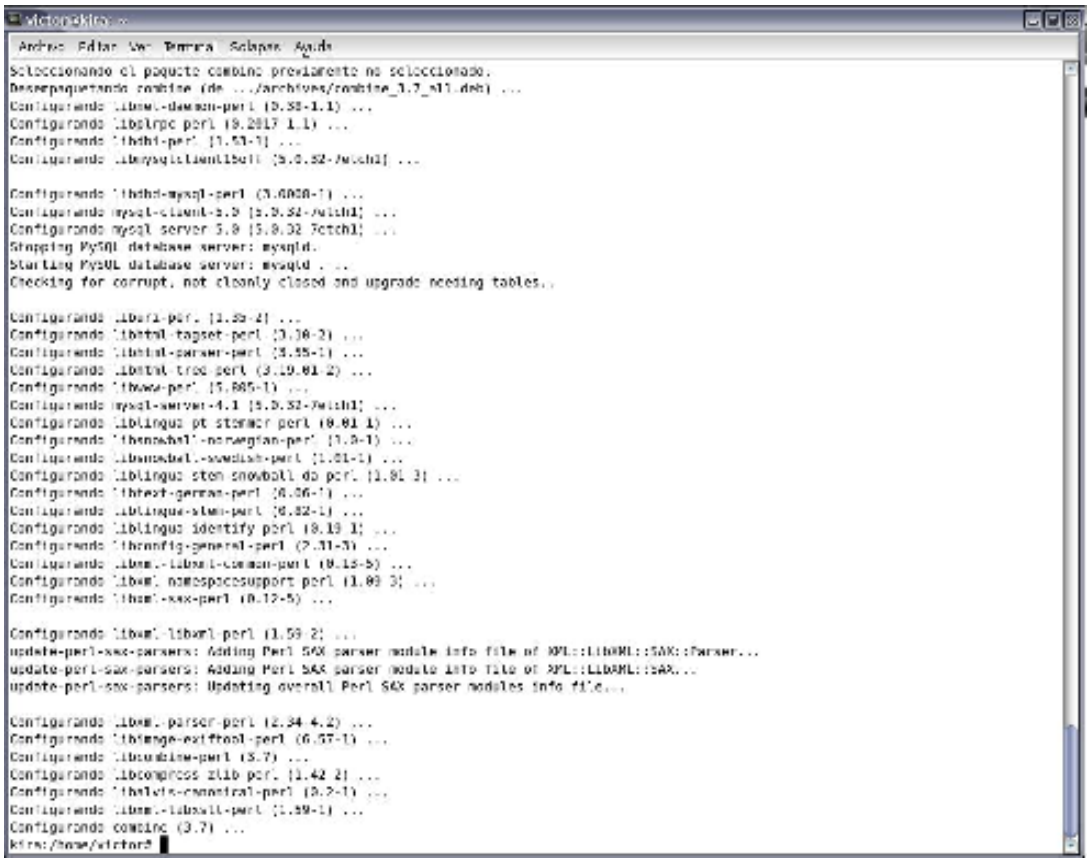


Figura 2.5.7 Instalación del Combine System (Continuación).

2.5.1.2 REQUISITOS DE INSTALACIÓN EN SISTEMAS OPERATIVOS NO SOPORTADOS.

Para poder llevar el sistema a otras plataformas, se tendrá que verificar la disponibilidad para la plataforma a seleccionar, de dos paquetes específicos:

- Perl 5 o superior.
- MySQL versión 4.1 o superior.

Si estos dos sistemas son soportados se tiene grandes probabilidades de poner a funcionar el Combine System en esa plataforma.
Además los módulos externos de Perl deberían de ser verificados para que el Combine System trabaje en la nueva plataforma.

A continuación se listan los módulos externos:

2.5.1.2.1 Módulos externos.

Estos son los módulos de Perl de los cuales depende el Combine System. Los módulos marcados con *"*"* no son indispensables.

Alvis::Canonical
Alvis::Pipeline *
Compress::Zlib
Config::General
DBI
Data::Dumper *
Digest::MD5
Encode
Getopt::Long
HTML::Entities
HTML::Tidy *
HTML::TokeParser
HTTP::Date

HTTP::Status
Image::ExifTool
LWP::UserAgent
Lingua::Identify
Lingua::Stem
MIME::Base64
Net::hostent
URI
URI::Escape
URI::URL
XML::LibXML
XML::LibXSLT
ZOOM *

Cada uno de estos módulos puede ser descargado desde **http://www.cpan.org/.**

Para instalar los módulos de Perl con mayor facilidad y de una manera automatizada se puede utilizar el sistema CPAN.
Las líneas utilizadas para manipular el CPAN son las siguientes:

Perl –MCPAN –e shell

Opcionalmente los siguientes programas externos pueden ser utilizados en el sistema:

- antiword (parsing MSWord files).
- detex (parsing TeX files) .
- pdftohtml (parsing PDF files).
- pstotext (parsing PS and PDF files, needs ghostview) .
- xlhtml (parsing MSExcel files).
- ppthtml (parsing MSPowerPoint files).
- unrtf (parsing RTF files).
- tth (parsing TeX files).
- untex (parsing TeX files).

2.5.2 PRUEBA DEL COMBINE SYSTEM.

Una manera sencilla de hace un ensayo con el Combine System, es escudriñar solo una página Web y exportarla en un documento XML. Esto probará una buena parte del código y nos garantizará que las funciones básicas de la búsqueda enfocada funcionen.

A continuación se realizarán algunas pruebas para verificar que nuestro sistema este funcionando correctamente.

Ejecute los comandos *sudo combineINIT --jobname aatest* como procedimiento de usuario *root* y genere un trabajo de búsqueda llamado *aatest.*

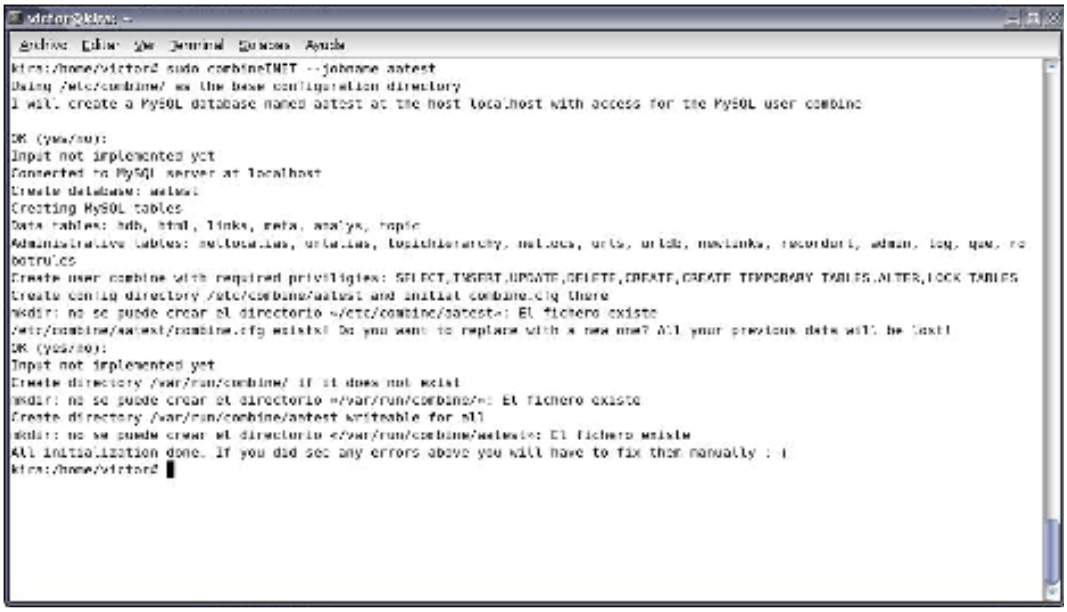


Figura 2.5.8 Inicio del trabajo de escrutinio aatest.

Acaba de iniciar el trabajo *aatest,* el Combine System ha creado la base de datos de trabajo para *aatest* y creado los directorios necesarios, en caso de que algunos de estos directorios ya existan Combine System ignora la creación de los mismos.

Ahora le proporcionaremos al programa una URL semilla la cual es necesaria para iniciar la búsqueda.

Digite los comandos *echo 'http://combine.it.lth.se/'* | *combineCtrl load --jobname aatest* para proporcionar la dirección del sitio de Combine System como semilla.

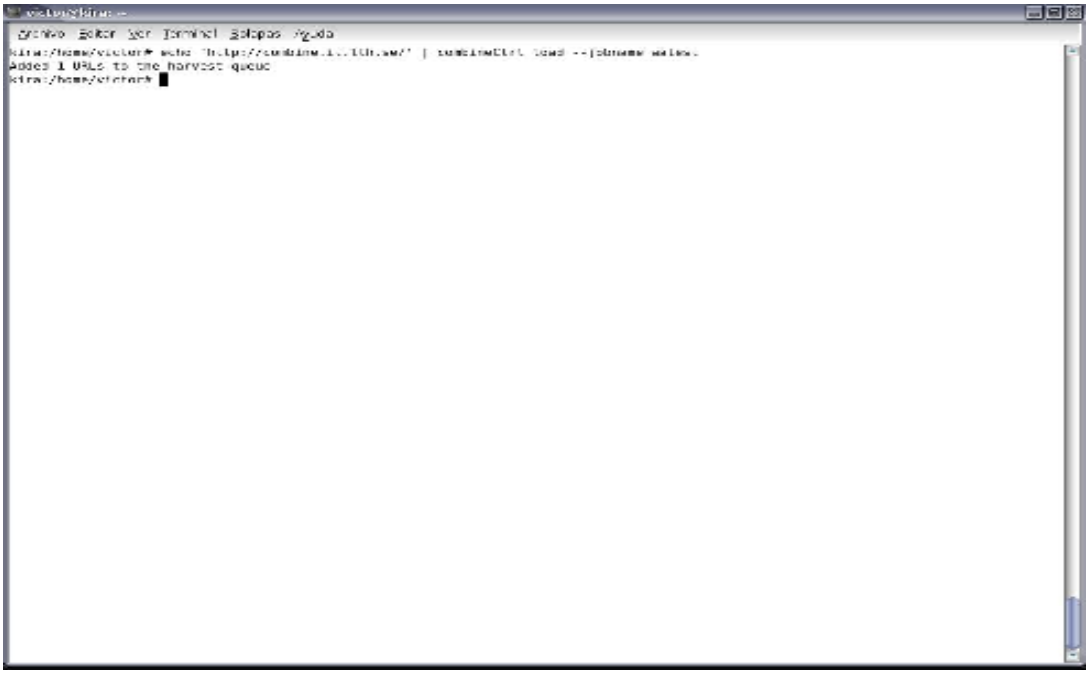


Figura 2.5.9 Adición del URL semilla para el trabajo de escrutinio aatest.

A continuación inicie un proceso de harvesting por medio del comando *combineCtrl --jobname aatest Start*.

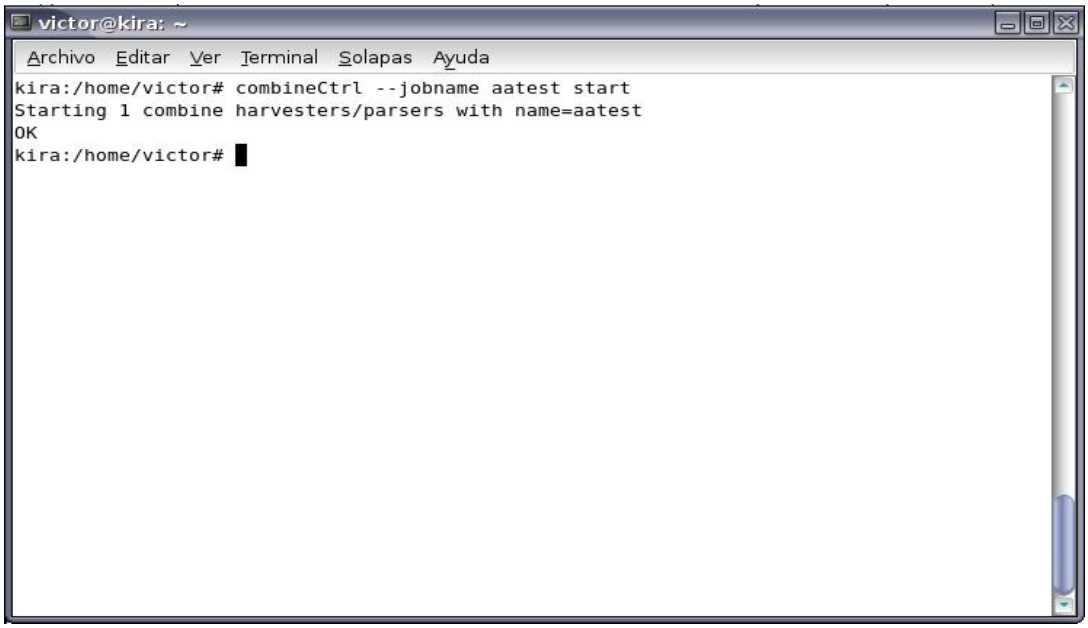


Figura 2.5.10 Inicio del proceso de harvesting para el trabajo de escrutinio aatest.

Después de un tiempo termine el proceso haciendo uso del comando *combineCtrl Kill --jobname aatest*

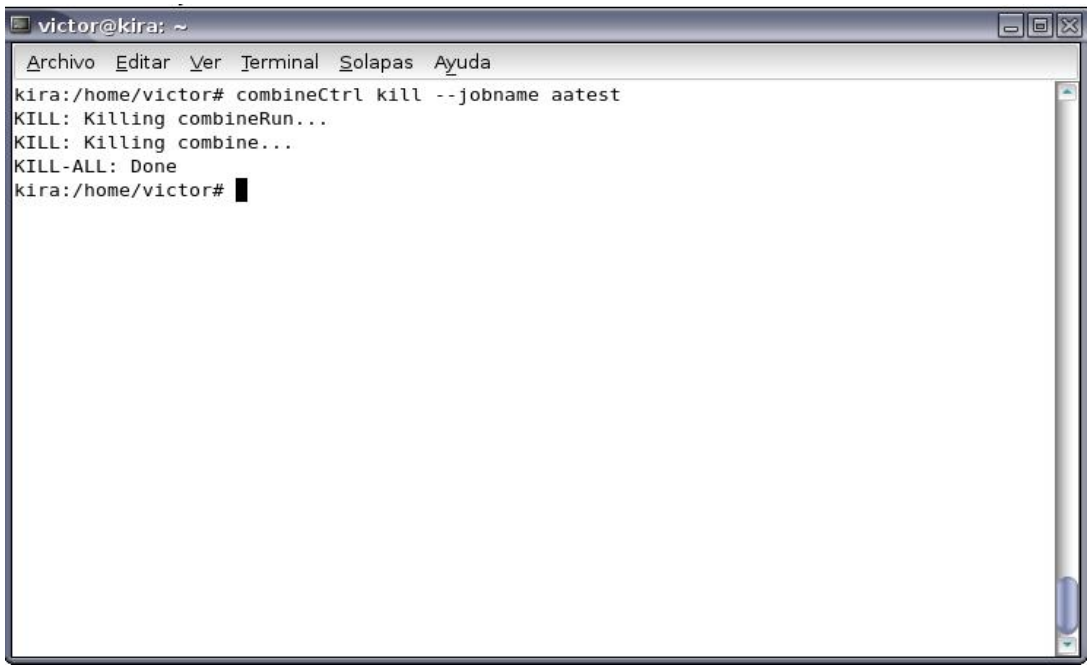


Figura 2.5.11 Finalización del proceso de escrutinio del trabajo aatest.

Para finalizar, exporte el resultado del escrutinio a un archivo XML Alvis, haciendo uso del comando *combineExport --jobname aatest --profile dc > pruebaxml.*

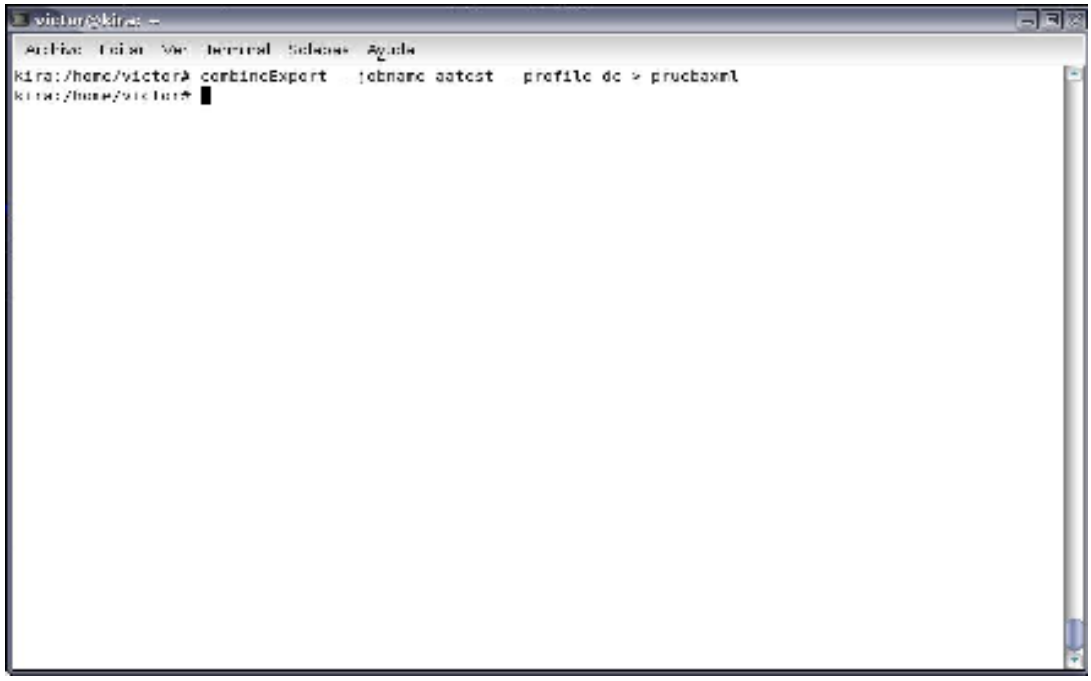


Figura 2.5.12 Comandos para exportar los resultados del trabajo de escrutinio aatest al archivo pruebaxml.

Este procedimiento nos guardara un archivo XML Alvis con las direcciones encontradas por el Combine System, éste archivo puede ser abierto por cualquier editor de texto.

Ahora, se debe verificar que el archivo sea como este (a excepción de las fechas y el orden):

```
<?xml version="1.0" encoding="UTF-8"?>
<documentCollection version="1.1" xmlns:dc="http://purl.org/dc/elements/1.1/">
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
<dc:format>text/html</dc:format>
<dc:format>text/html; charset=iso-8859-1</dc:format>
<dc:subject>Carnivorous plants</dc:subject>
<dc:subject>Drosera</dc:subject>
<dc:subject>Nepenthes</dc:subject>
<dc:title transl="yes">Installation test for Combine</dc:title>
<dc:description></dc:description>
<dc:date>2006-05-19 9:57:03</dc:date>
<dc:identifier>http://combine.it.lth.se/CombineTests/InstallationTest.html</dc:identifier>
<dc:language>en</dc:language>
</metadata>
```

2.5.3 UTILIZANDO ESCENARIOS DE BÚSQUEDA DENTRO DEL COMBINE SYSTEM.

2.5.3.1 BÚSQUEDAS GENERALES SIN RESTRICCIONES.

Para realizar una búsqueda sin ninguna restricción se puede utilizar el procedimiento que ha sido detallado en el apartado 2.5.2 de este documento. No es recomendado utilizar este tipo de búsqueda ya que el Combine System lanzará una base de datos demasiado grande, además que la búsqueda no tendrá ningún fin específico.

2.5.3.2 ESCRUTINIO ENFOCADO – RESTRICCIÓN DE DOMINIOS.

En este tipo de búsqueda se crea una base de datos enfocada con todas las páginas de un sitio Web. En este ejemplo realizáremos una búsqueda en el sitio oficial del Combine System y en el sitio de ALVIS.

La base de datos debe ser continuamente actualizada, todas las páginas deben ser probadas frecuentemente buscando cambios o links inservibles, que deben de ser borrados de la base de datos, y las nuevas páginas se deben añadir a ésta.

A continuación se muestran los pasos realizados para la búsqueda en los sitios del Combine System y de Alvis:

- Inicialice un proceso llamado *focustet* haciendo uso de los comandos *sudo combineINIT --jobname aatest focustest*

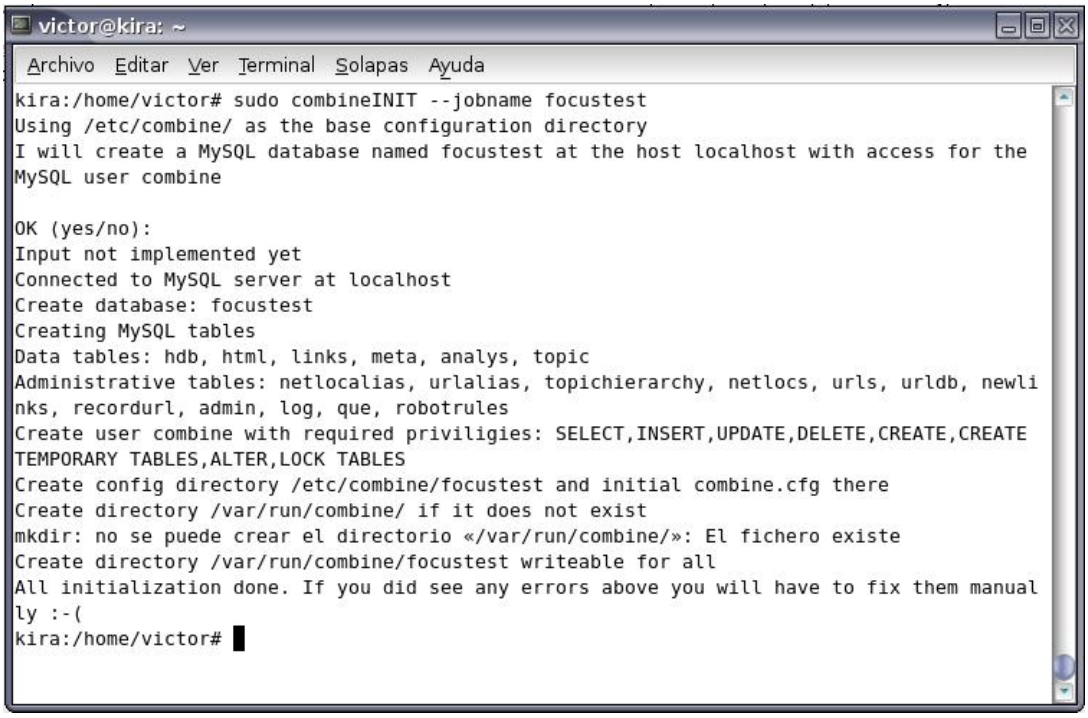


Figura 2.5.13 Inicialización del trabajo de escrutinio focustest.

- Haga uso de un editor de texto para leer el archivo de configuración que se encuentra en */etc/combine/focustest/combine.cfg*

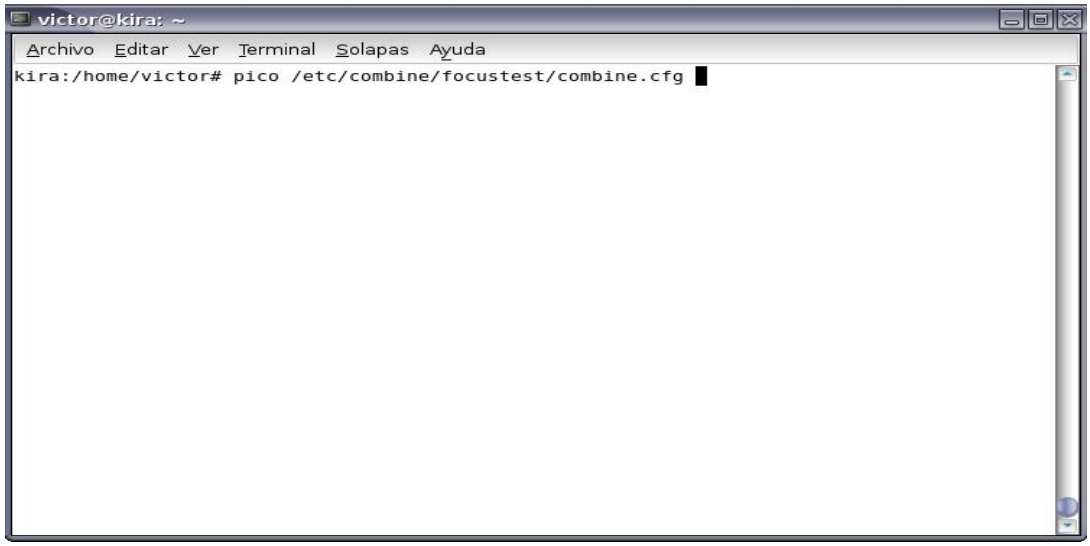


Figura 2.5.14 Abriendo el archivo combine.cfg en el editor de texto pico.

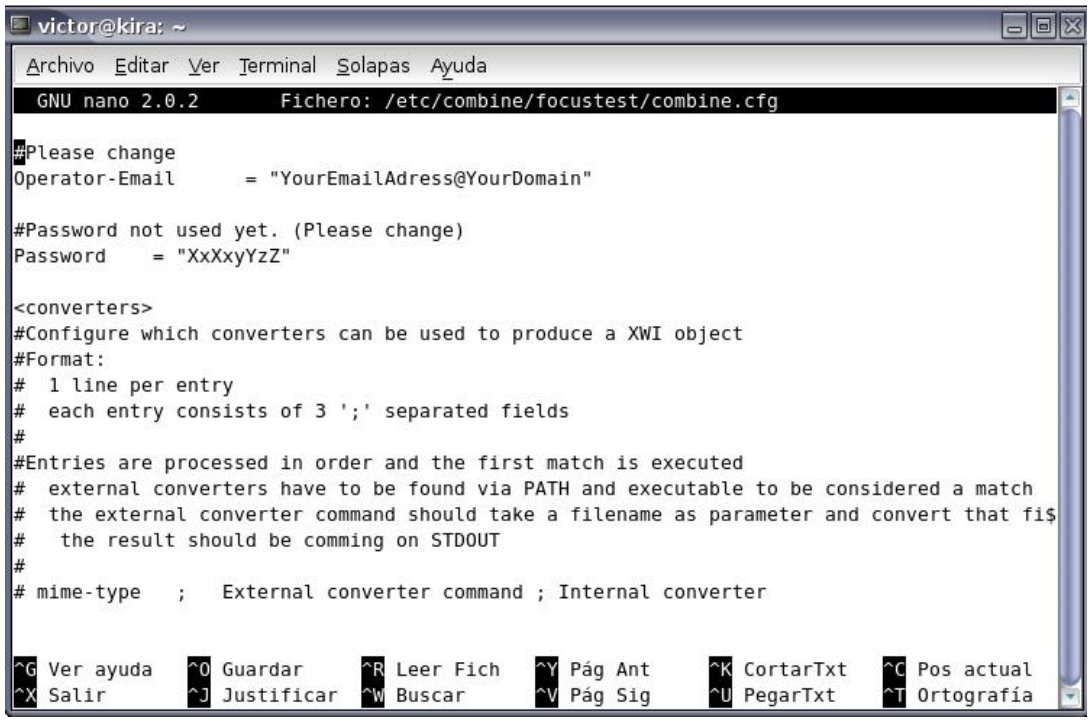


Figura 2.5.15 Archivo de configuración combine.cfg

- Edite la parte <*Allow*> para agregar los *HOST* de la siguiente manera:

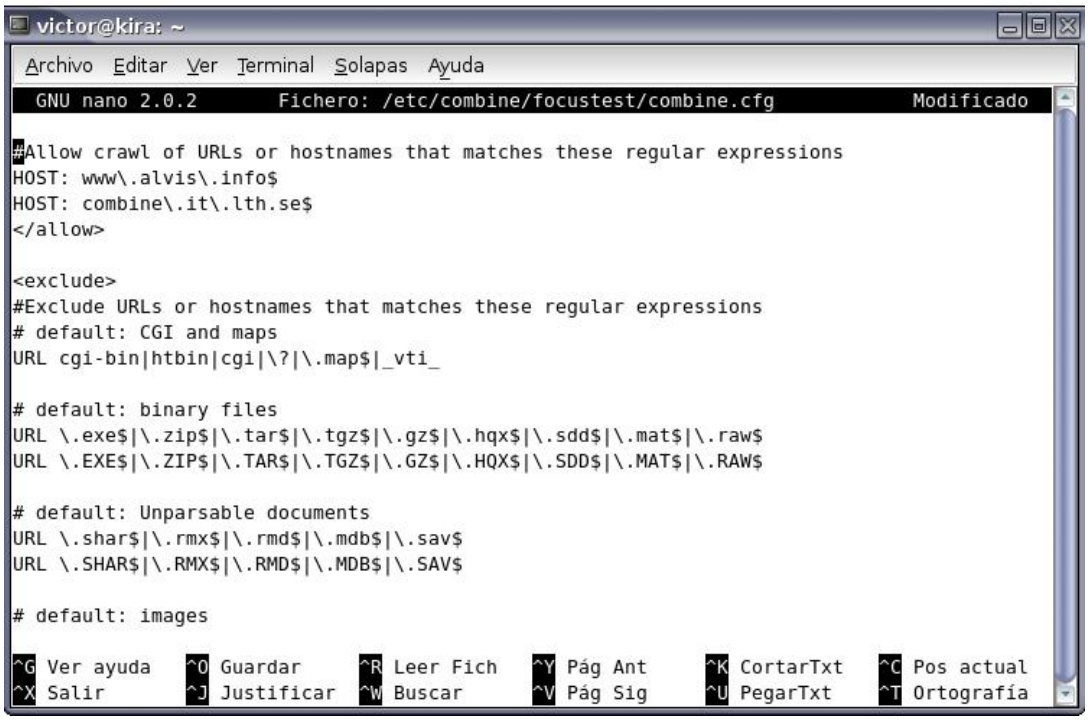


Figura 2.5.16 Archivo de configuración combine.cfg con dos HOSTS de inicio agregados.

El símbolo para omitir el *'.'* en el archivo de configuración es *'\'* esto es necesario debido a que en éste archivo se manejan expresiones de Perl.

De manera similar se utiliza el símbolo de dólar *'$'* para indicar que el final del texto del Host debe finalizar ahí, esto quiere decir que por ejemplo un servidor Web en *www.alvis.info.com* (si existiera) no seria escudriñado debido a que la búsqueda termina en el dominio *.info.*

- Ahora haga uso de los comandos *echo 'http://combine.it.lth.se/'* | *combineCtrl load --jobname focustest'* y *echo 'http://www.alvis.info/* | *combineCtrl load --jobname focustest'* para que el trabajo de búsqueda focustest lea dos URL's semillas.

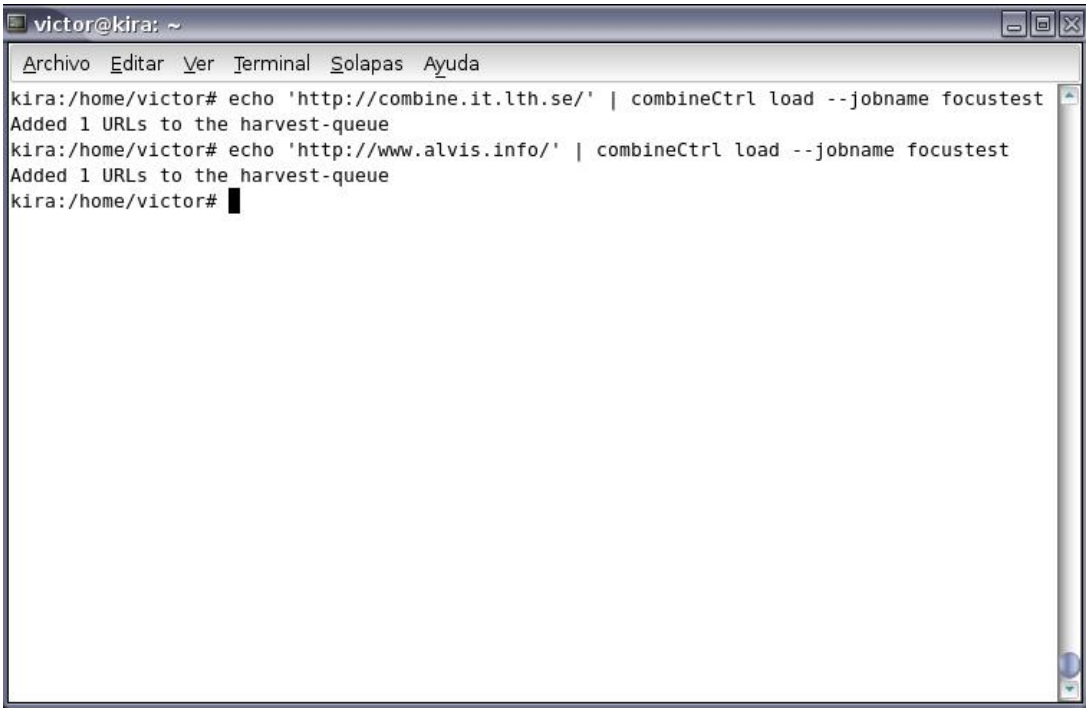


Figura 2.5.17 Comandos para agregar dos URL's semillas.

- Ahora inicie un proceso de harvesting, haciendo uso del comando *combineCtrl start --jobname focustest*

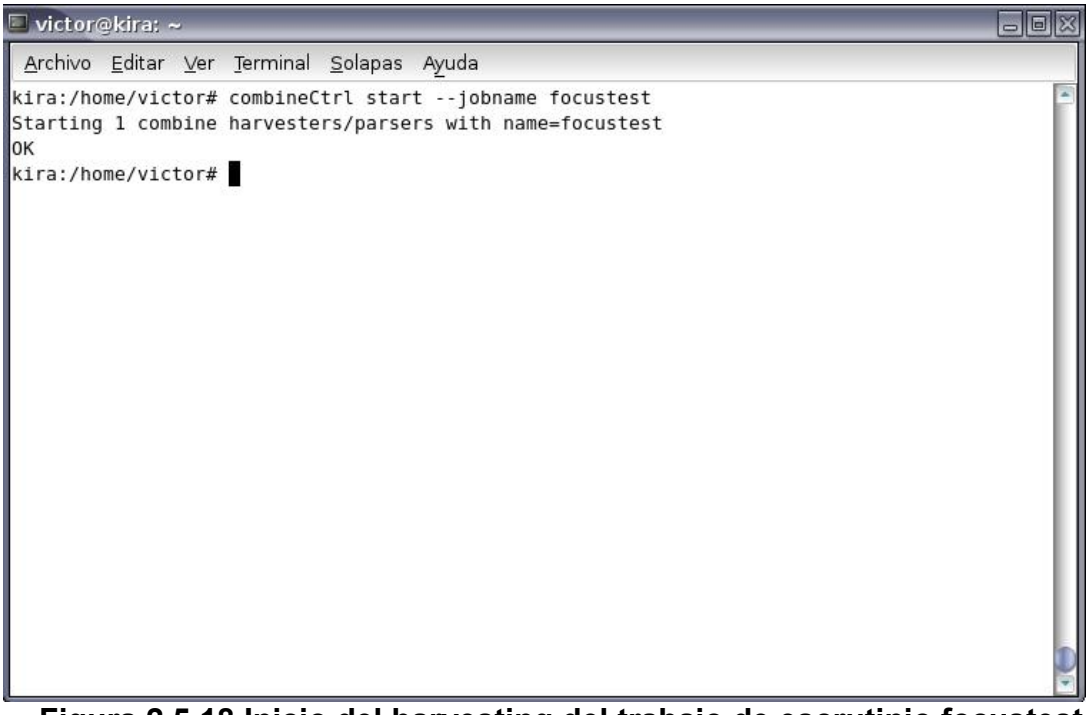


Figura 2.5.18 Inicio del harvesting del trabajo de escrutinio focustest.

- Después de esperar unos dos minutos se exportaran los datos y se guardaran en el archivo *xmlfocus.*

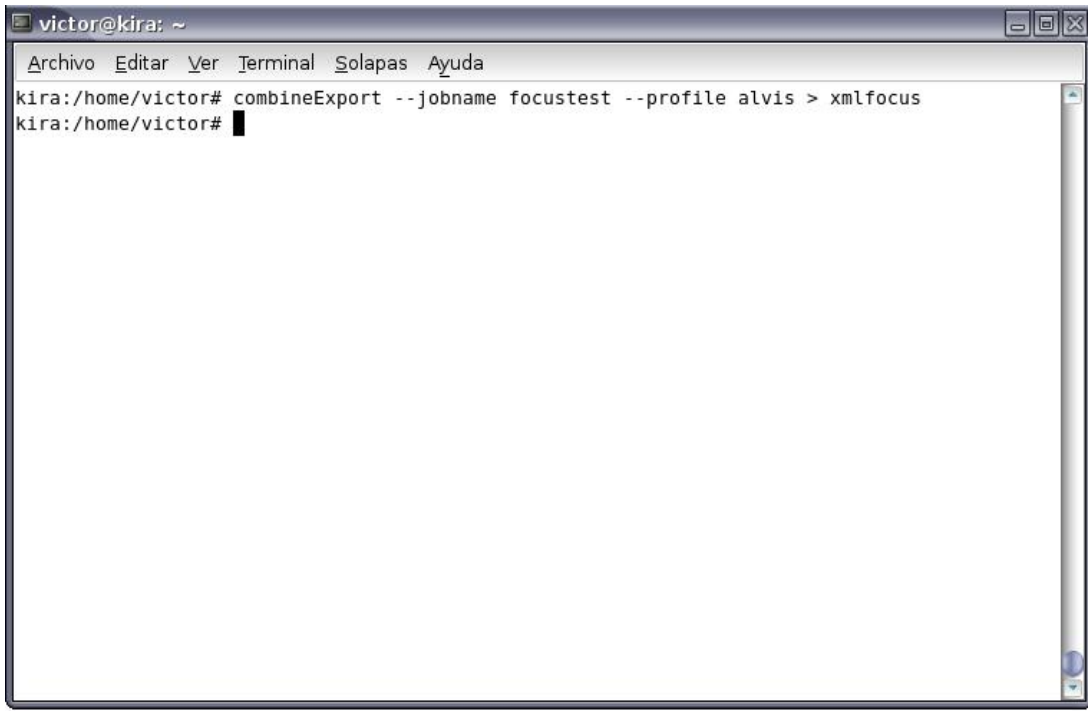


Figura 2.5.19 Comandos para exportar los resultados del trabajo de escrutinio focustest al archivo xmlfocus.

2.5.3.3 ESCRUTINIO HACIENDO USO DE DEFINICIONES DE TÓPICOS.

Localizado en */etc/combine/<jobname>/topicdefinitio,* las definiciones de tópicos utilizan tripletas (término, peso de relevancia, clases de tópico) como sus entradas básicas. Los pesos son asignados como enteros e indican la relevancia del término con su respectiva clase de tópico. Los valores más altos indican más relevancia del término, quiere decir que tendrán más importancia al momento de realizar la búsqueda, un número grande pero negativo puede ser usado para excluir documentos que contengan ese término.

Los términos pueden ser:

- Una simple palabra.
- Una frase (todas las palabras en el orden exacto).
- Una expresión booleana con términos de conexión como las expresiones AND (todas las palabras deben estar presentes, pero en cualquier orden). El operador booleano AND esta codificado como '@and'.

El operador de expresión OR tiene que ser insertado como dos tripletas separadas. Por ejemplo la expresión: 'Universidad AND (Bosco OR UCA)' tendría que separarse en las tripletas: 'Universidad @and Bosco' y 'Universidad @and UCA'.

Los términos pueden incluir expresiones regulares de Perl como:

- '?' se utiliza para que los caracteres posteriores sean opcionales por ejemplo el termino 'moneda?' tomara 'moneda' y 'monedas'.
- '[^ \s]*' se utiliza para que se tomen en cuenta todos los caracteres posteriores excluyendo los espacios por ejemplo 'perl 5[^ \s]*' tomara en cuenta 'perl 5', 'perl 5.1', 'perl 5.2' y así sucesivamente.

Es importante entender que cada termino dentro de las tripletas en la definición de tópico, es considerado por si mismo sin ningún contexto, entonces para realizar una búsqueda exitosa se deberán de organizar los términos en sub-clases.

Términos muy generales como “historia”, “examen”, “cosa” no deberían de ser utilizados. Si realmente se necesita utilizar dichos términos, éstos deberían convertirse en tópicos específicos.

A continuación se muestran los pasos realizados para la búsqueda de tópicos específicos, en éste caso, se busca en páginas de restaurantes de México y Argentina, comida común como tacos, pollo, vinos y carne.

- Realice un archivo para la configuración de tópicos de nombre *restTopic.txt* en cualquier editor de texto, en este caso se utilizo el editor de texto pico.

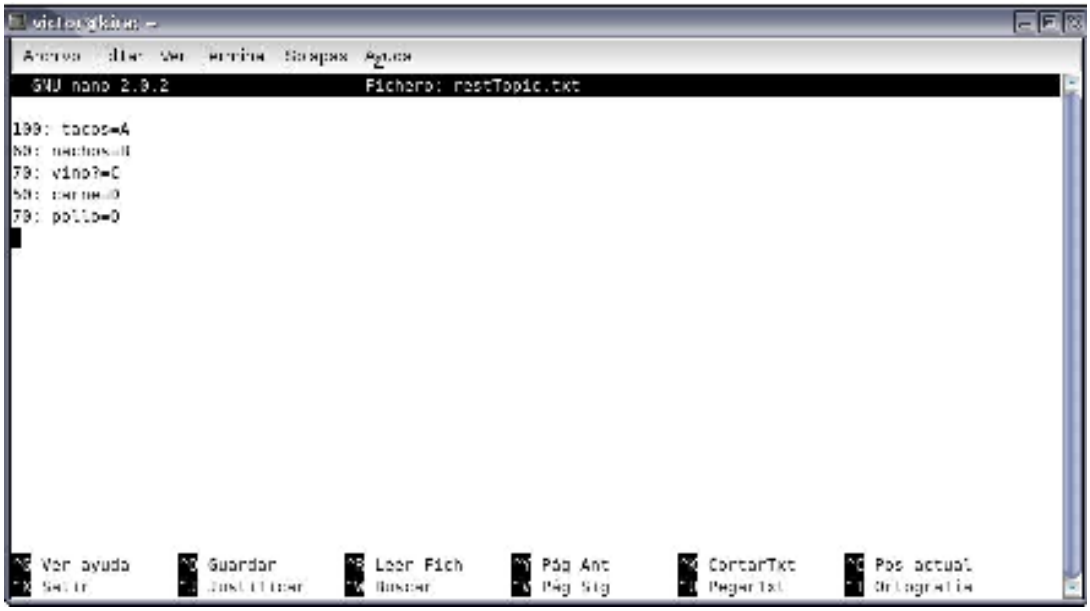


Figura 2.5.20 Archivo de configuración de tópicos RestTopic.txt del trabajo restaurantes.

- Realice un archivo con las URL's semillas, de nombre *restSeed.txt* en cualquier editor de texto.

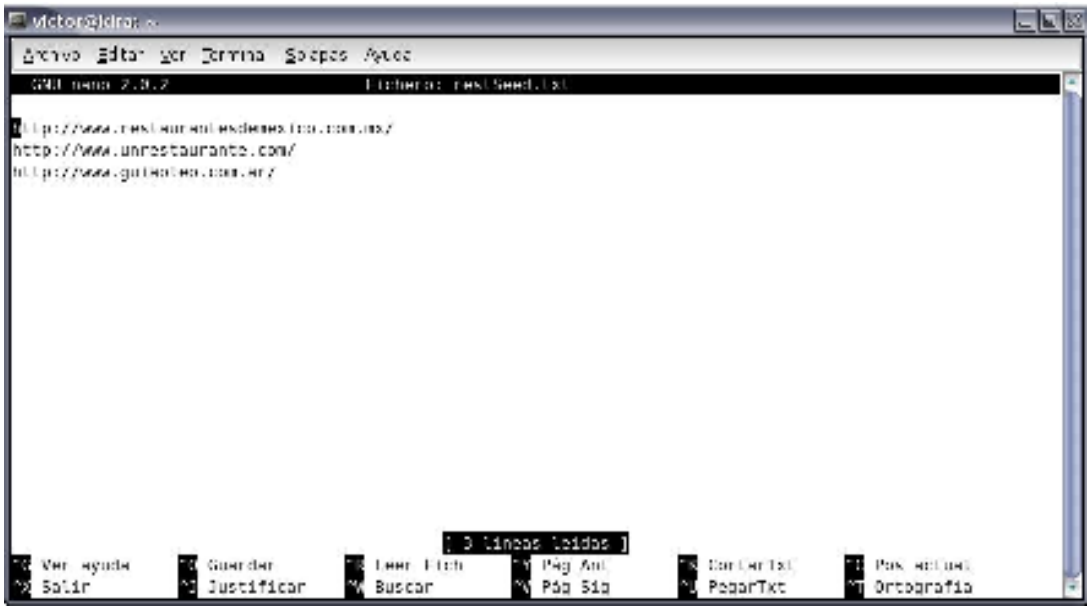


Figura 2.5.21 Archivo de configuración de semillas RestSeed del trabajo restaurantes.

- Inicialice un trabajo de escrutinio llamado restaurantes haciendo uso de los comandos *combineINIT --jobname restaurantes --topic restTopic.txt.*

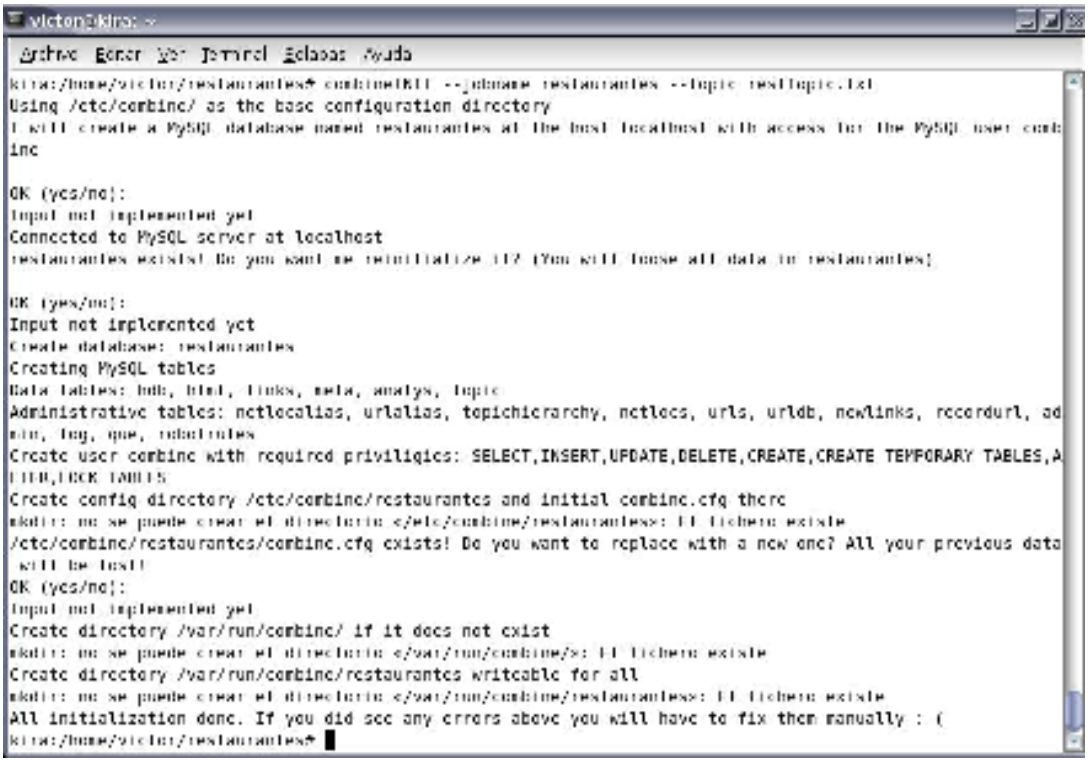


Figura 2.5.22 Iniciación del trabajo de escrutinio restaurantes.

- Cargue las URL's semillas definidas en archivo *restSeed.txt*, al trabajo restaurantes haciendo uso de los comandos *CombineCtrl load --jobname restaurantes < restSeed.txt*

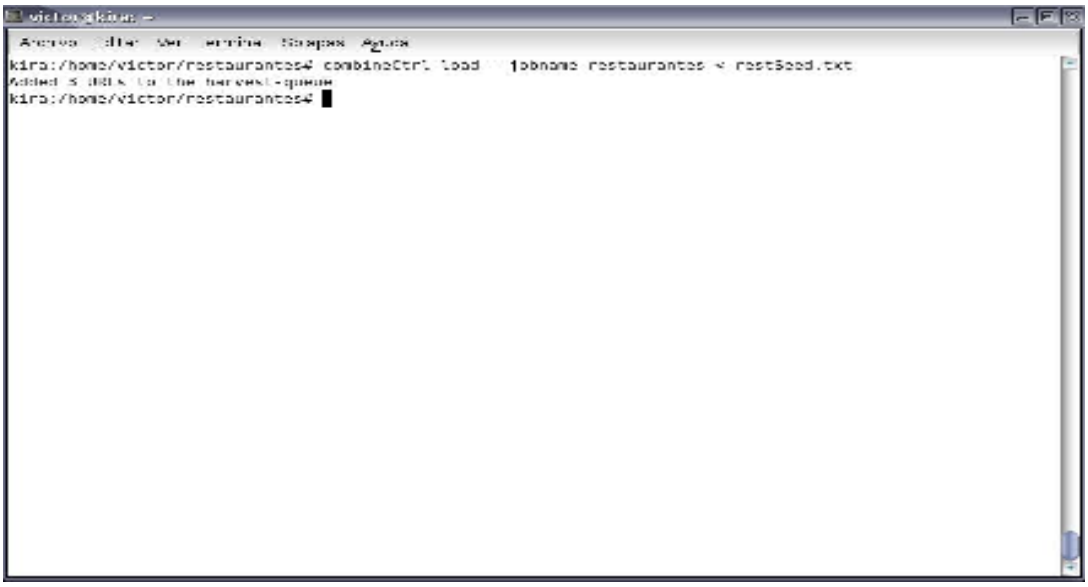


Figura 2.5.23 Comandos para cargar URL's semillas al trabajo de escrutinio restaurantes.

- Inicie tres procesos de harvesting para el trabajo restaurantes haciendo uso los comandos *combineCtrl start --jobname restaurantes --harvesters 3*

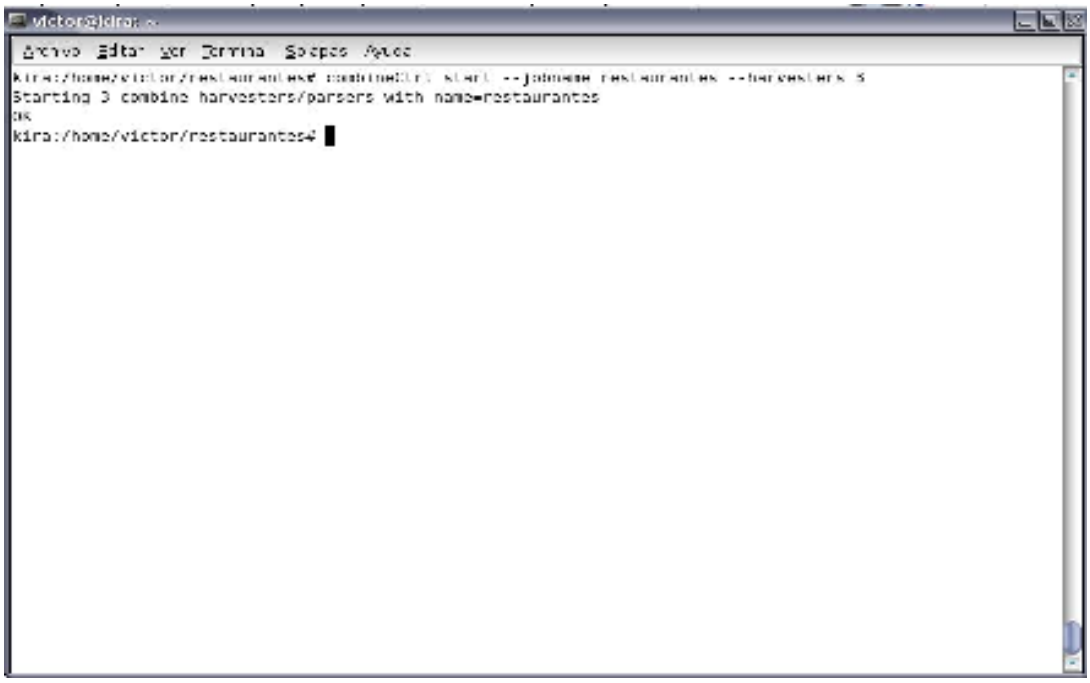


Figura 2.5.24 Inicio de 3 procesos de harvesting para el trabajo de escrutinio restaurantes.

- Después de esperar unos minutos podrá comprobar los resultados en la base de datos restaurantes en MySQL, en éste ejemplo se utilizó la interfaz grafica Emma de la siguiente manera:

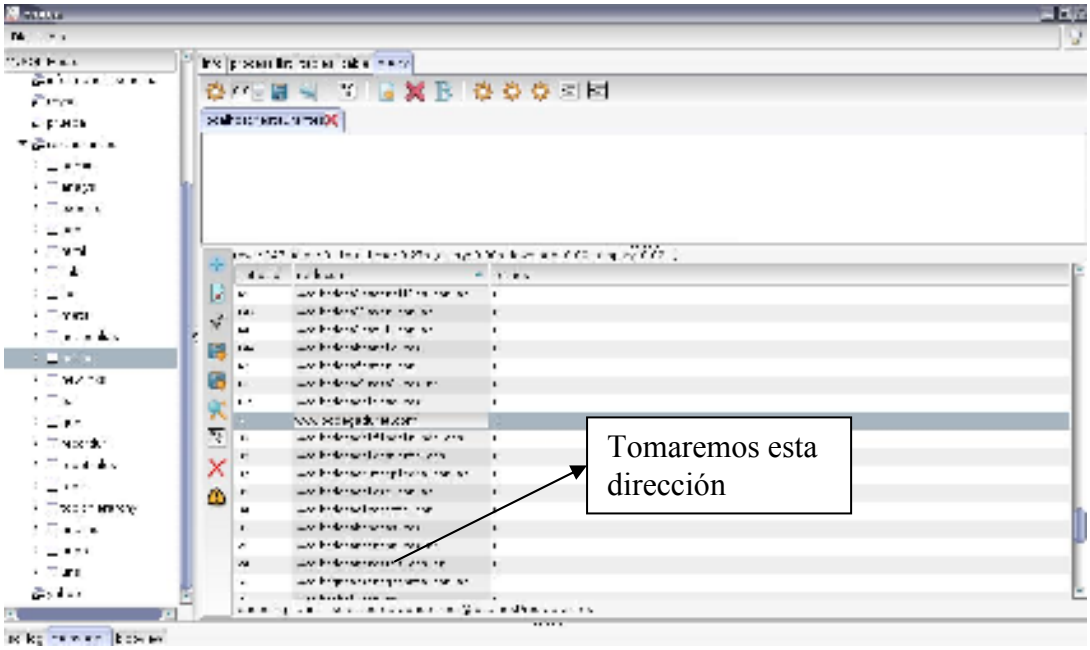


Figura 2.5.25 Base de datos resultado del trabajo de escrutinio restaurantes vista en Emma.

Figura 2.5.26 Página encontrada por medio del trabajo de escrutinio restaurantes.

2.5.3.4 ESCRUTINIO HACIENDO USO DE DEFINICIONES DE TÓPICOS Y UN DOMINIO ESPECIFICO .SV.

A continuación se mostrará el proceso paso a paso, donde se combinan los dos tipos de búsqueda que se pueden realizar con el Combine System.

- Realice un archivo para la configuración de tópicos de nombre *usvtopic.txt* en cualquier editor de texto, en éste caso se utilizó el editor de texto pico.

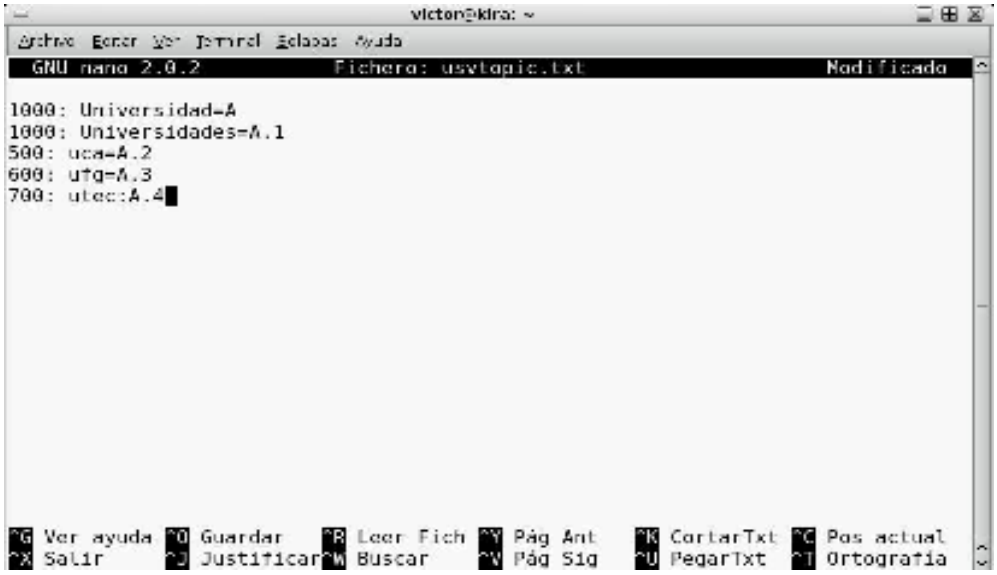


Figura 2.5.27 Archivo de configuración de tópicos usvtopic.txt del trabajo buscausv.

- Realice un archivo con las URL's semillas, de nombre *buscaseed.txt* en cualquier editor de texto.

Figura 2.5.28 Archivo de configuración de semillas buscaseed.txt.

- Inicialice un trabajo de escrutinio llamado buscausv haciendo uso de los comandos *combineINIT --jobname buscausv --topic usvtopic.txt.*

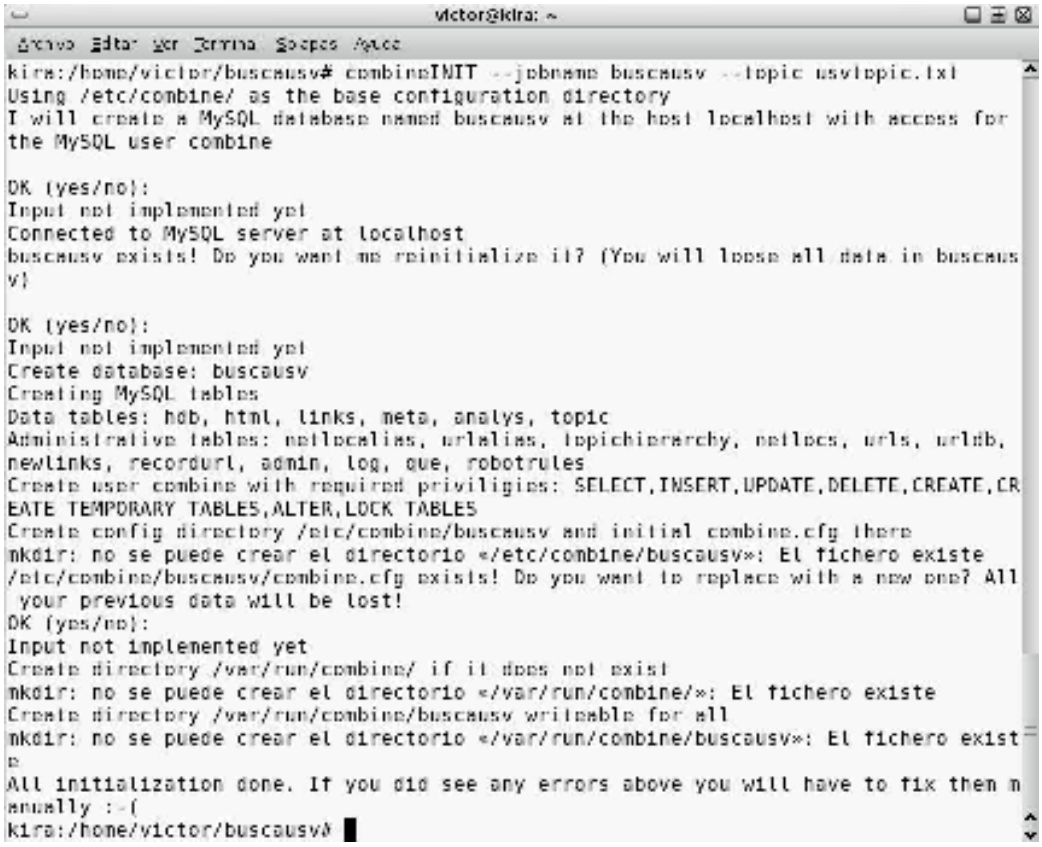


Figura 2.5.29 Iniciación del trabajo de escrutinio buscausv.

- Cargue las URL's semillas definidas en archivo *buscaseed.txt*, al trabajo buscausv haciendo uso de los comandos *CombineCtrl load --jobname buscausv < buscaseed.txt.*

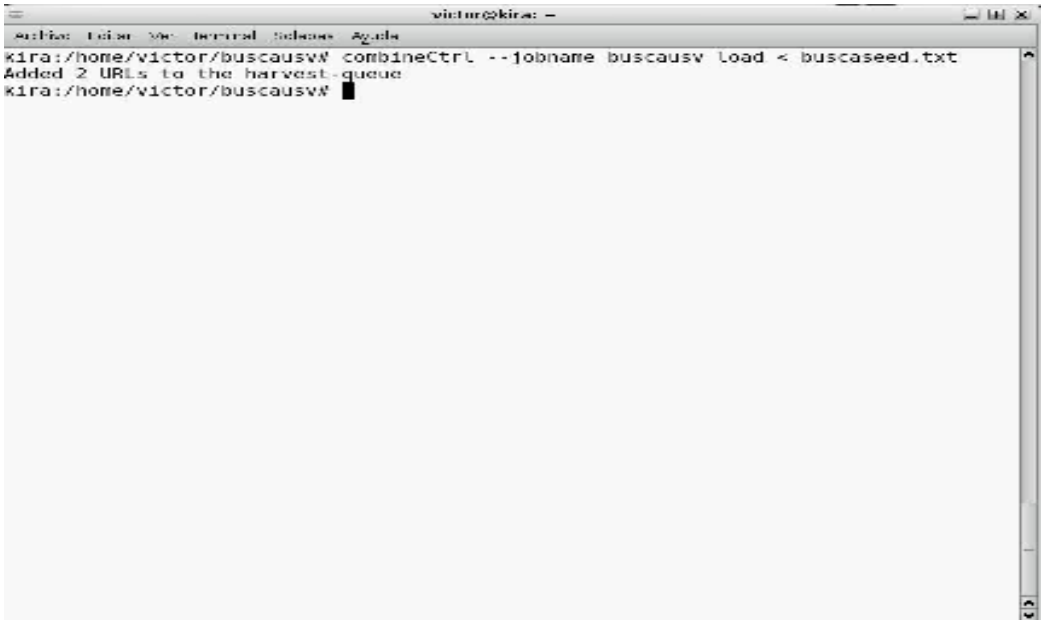


Figura 2.5.30 cargando URL's semillas buscausv.

- Haga uso de un editor de texto para leer el archivo de configuración que se encuentra en */etc/combine/buscausv/combine.cfg*

Figura 2.5.31 Abriendo el archivo *combine.cfg* en el editor de texto pico.

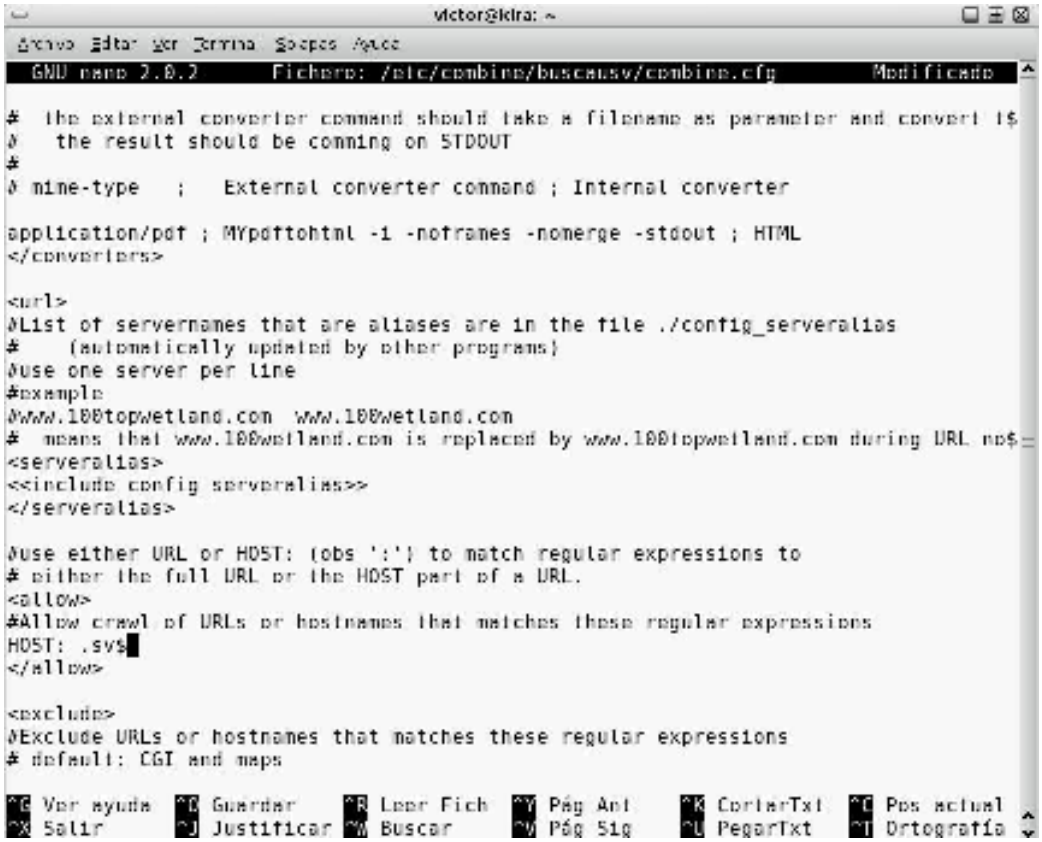


Figura 2.5.32 Archivo de configuración *combine.cfg*

- Inicie tres procesos de harvesting para el trabajo buscausv haciendo uso de los comandos *combineCtrl start --jobname buscausv --harvesters 3*

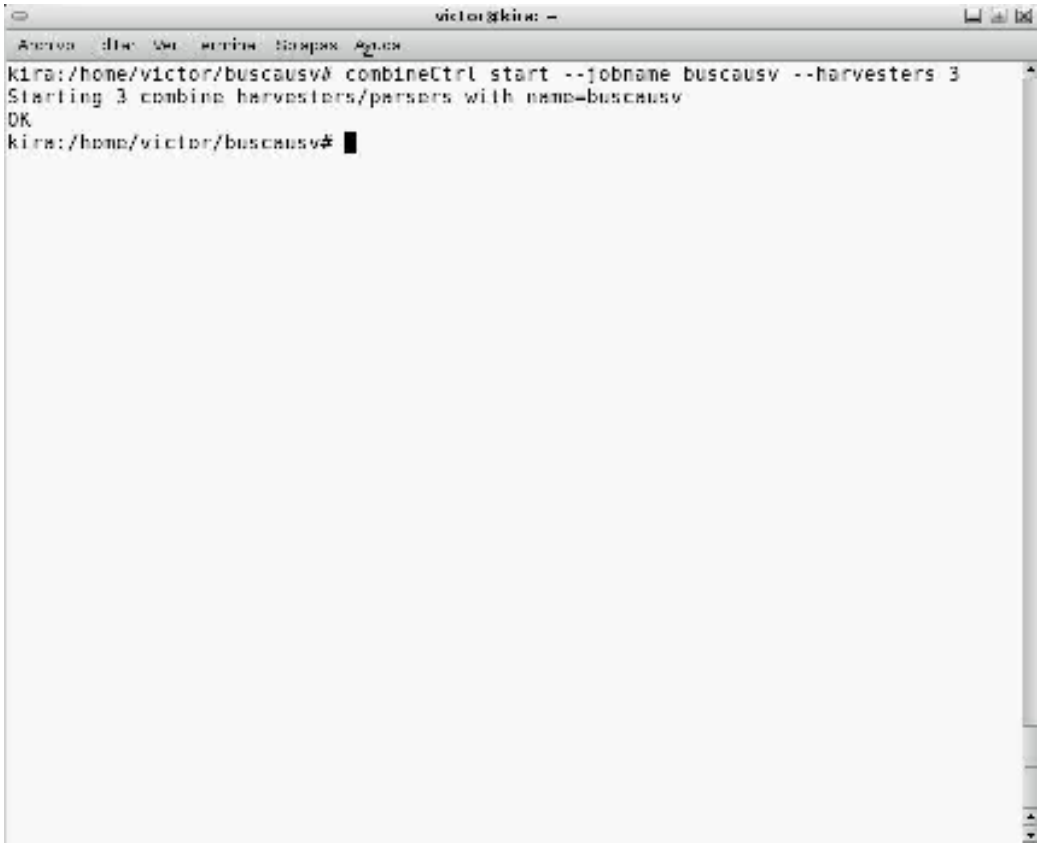


Figura 2.5.33 Inicio de 3 procesos de harvesting para el trabajo de escrutinio buscausv.

Auto evaluación.

Instrucciones

Conteste, o realice los ejercicios que se le solicitan.

1. Escriba los comandos para iniciar un trabajo de escrutinio general llamado prueba.

2. Ahora escriba los comandos para agregar la URL semilla http://www.yahoo.com al trabajo de escrutinio prueba.

3. Escriba los comandos para agregar tres *harvesters* al trabajo de escrutinio prueba

4. Escriba los comandos necesarios para terminar los *harvesters* del trabajo prueba.

5. Escriba los comandos necesarios para exportar los resultados del trabajo de escrutinio prueba a un archivo llamado *xmlprueba.*

Respuestas de las preguntas de la página anterior.

1. *combineINIT --jobname prueba*
2. *echo "http://www.yahoo.com" | combineCtrl load --jobname prueba*
3. *combineCtrl start --jobname prueba --harvesters 3*
4. combineCtrl kill --jobname prueba
5. *combineExport --jobname prueba --profile dc > xmlprueba.*

De 5 a 3 preguntas correctas: Puede continuar con el capitulo 6
Menos de 3 preguntas: Repase el capitulo 5 y compruebe el ejercicio.

Investigación Complementaria

- Investigue como funciona el sistema XML Alvis.
- Investigue como funciona el algoritmo de Combine System, las formulas, el algoritmo que utiliza y su recorrido.
- Realice una búsqueda de artículos de computadoras, monitor o monitores, teclado, mouse, microprocesadores.

2.6: IMPLEMENTACIÓN DEL ROBOT COMBINE SYSTEM EN UN BUSCADOR WEB.

Tiempo recomendado de estudio: 6 horas

Objetivos:

Al concluir este capítulo el lector estará capacitado para:

- Entender y poner en práctica el concepto de Search engine in a box System sus principales componentes y su funcionamiento.

- Hacer uso del lenguaje *php* para modificar las plantillas proporcionadas por el sistema de ingeniería en una caja.

- Hacer uso de los archivos de configuración del Combine System para personalizar su propio buscador, restringiendo la búsqueda a dominios .*sv.*

Introducción:

El proyecto ALVIS fue desarrollado en Estados Unidos, el cual es un sistema de búsqueda basado en el robot Combine System y el sistema de indexación de texto Zebra. Este sistema nos permite crear un sistema de búsqueda en pocos pasos.

Zebra es un "motor" de indexación y recuperación de información de texto estructurado de alto rendimiento, reconoce registros en una variedad de formatos como el correo electrónico, XML y MARC; proporciona acceso a ellos a través de una combinación de expresiones de búsqueda booleana y consultas de ranking de relevancia.

También soporta bases de datos grandes y seguras (10 millones de registros) además de permitir actualizaciones constantes, debido a que utiliza el protocolo estándar de recuperación de información Z39.50, con el cual, se pueden realizar búsquedas en bases de datos, usando una variedad de programas y paquetes de herramientas (toolkits).

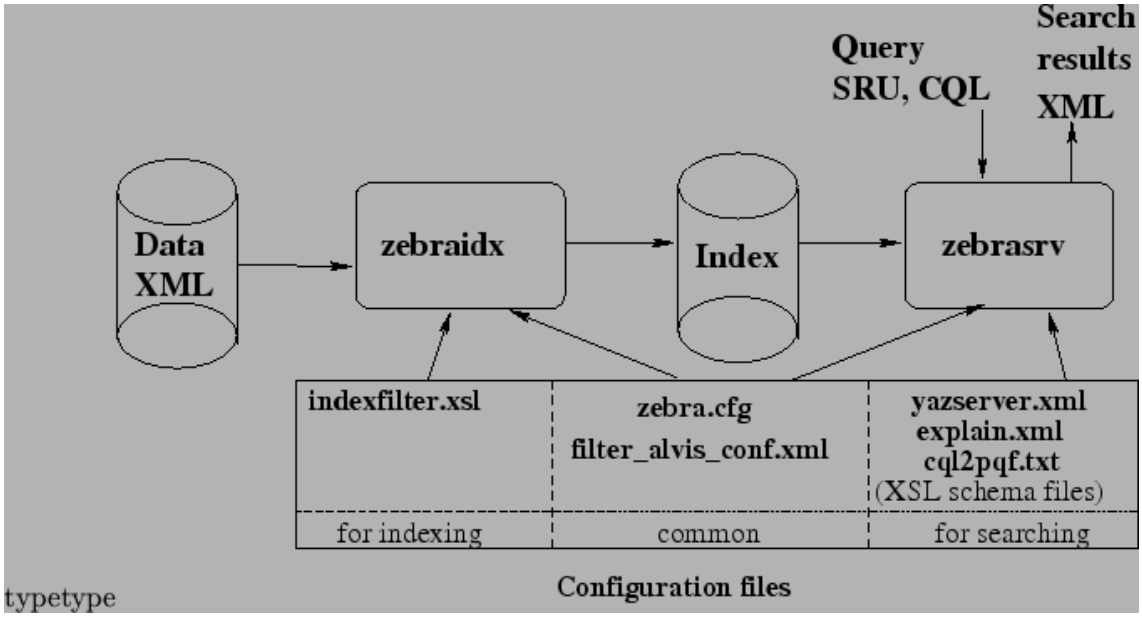


Figura 2.6.1 Arquitectura de Zebra

Zebra coloca pocas restricciones sobre el tipo de información que se puede indexar y manejar. La información, es "parseada"[14] por un filtro de entrada especifico a ese tipo y convertido en una estructura interna que Zebra sabe como manejar. Éste proceso toma lugar cada vez que el registro es accesado, ya sea este para indexación o recuperación de la información.

El parámetro *RecordType* en el archivo *zebra.cfg* o la opción *–t* para crear un archivo de configuración del indexador, le dice a Zebra como procesar los registros de entrada.

Están disponibles dos tipos básicos de procesamiento: texto puro o sin tratar e información estructurada.
El texto puro o sin tratar, es como su nombre lo indica, texto seleccionado para proporcionar un argumento de texto a Zebra.
La información estructurada son todos los registros manejados internamente usando mecanismos básicos.

YAZ.
Es una librería hecha en C/C++ para la aplicación de recuperación de información usando los protocolos Z39.50/SRU.

Algunas de las propiedades que presenta YAZ son:
Conjunto de utilidades como por ejemplo MARC parser, CCL parser, CQL parser y rutinas para el manejo de memoria.

Search in a box system (Sistema de ingeniería en una caja).
El sistema de ingeniería en una caja es la combinación principalmente de dos sistemas: el robot Combine System y el servidor indexador Zebra, que a la vez necesitara de un cliente Yaz para poder funcionar.

La configuración del Combine System es un tanto sencilla y ya fue explicada en el capitulo 2.5 de este documento, pero la configuración del servidor Zebra es mucho mas compleja, para hacer las cosas un poco mas sencillas, Alvis realizó

[14] Derivado de Parseo. Ver Glosario.

un archivo que permite la configuración de Zebra de una manera automática para que se integre con el Combine System.

El siguiente esquema muestra la configuración de Zebra y los archivos involucrados en la configuración automatizada:

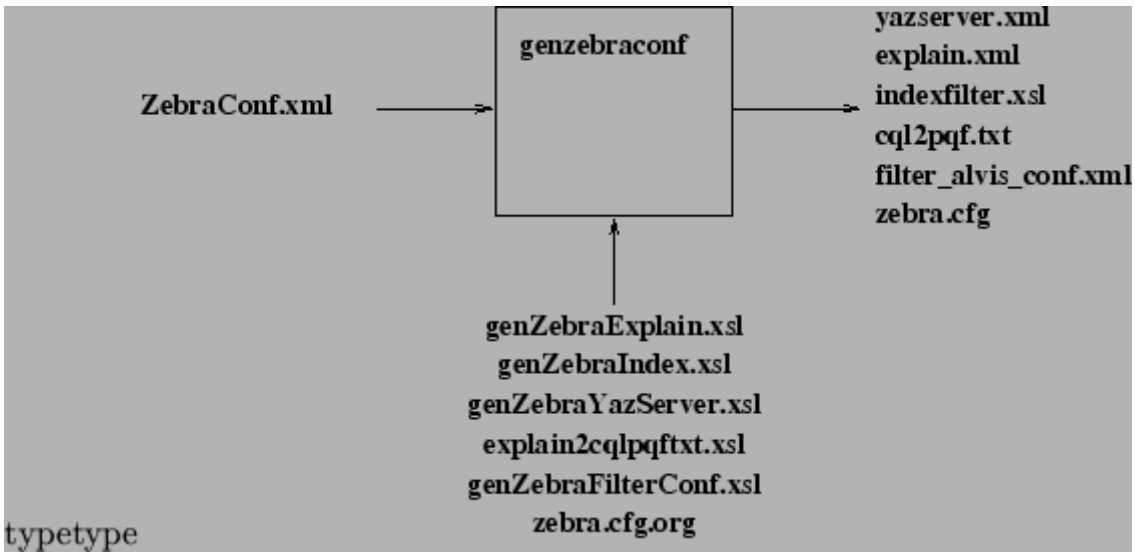


Figura 2.6.2 Configuración de Zebra.

El archivo de configuración Zebra.conf posee tres secciones principales:

- **Información del servidor:** ésta sección contiene información acerca de cómo se ejecutara el servidor, como el nombre del *host* y el número de puerto que utilizará.

- **Información de base de datos:** éste apartado contiene información textual acerca de la base de datos.

- **Definición de indexado:** define el formato de los registros como éstos deben de ser indexados.

2.6.1 INSTALACIÓN DEL PAQUETE SEARCH IN A BOX SYSTEM.

El archivo *SEbox.tgz* incluido en el CD adjunto es la base para generar nuestro buscador, la configuración de Zebra y Yaz la realiza el sistema de ingeniera en una caja, ya que se generan archivos dependiendo del nombre de nuestro servidor y del puerto que utilicemos para nuestro sistema de búsqueda.

Pasos:

1- Edite el archivo */etc/apt/sources.list* para agregar los URL's necesarios para la instalación del servidor Zebra, Yaz y XSLT (en este caso utilizamos el editor de texto pico).

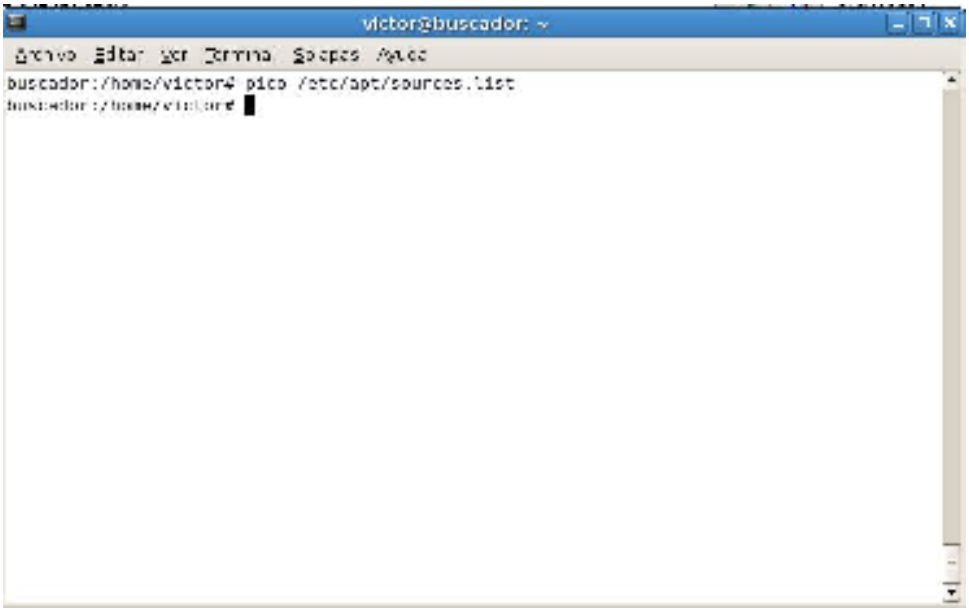


Figura 2.6.3 Uso de pico para modificar el archivo *sources.list*

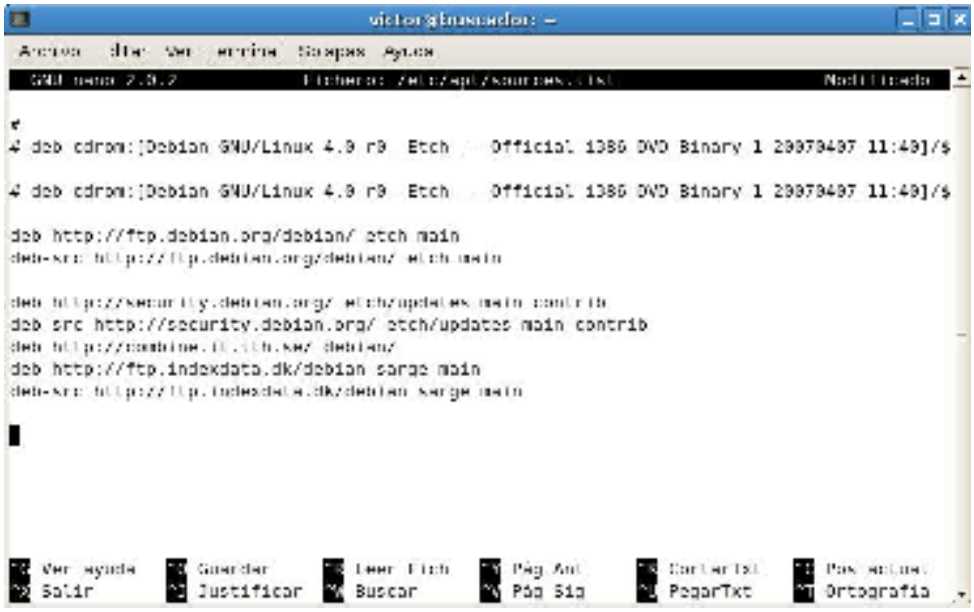


Figura 2.6.4 Archivo de configuración *sources.list*

2- Utilice el comando *apt-get* para instalar los componentes utilizados en el sistema de ingeniería en una caja.

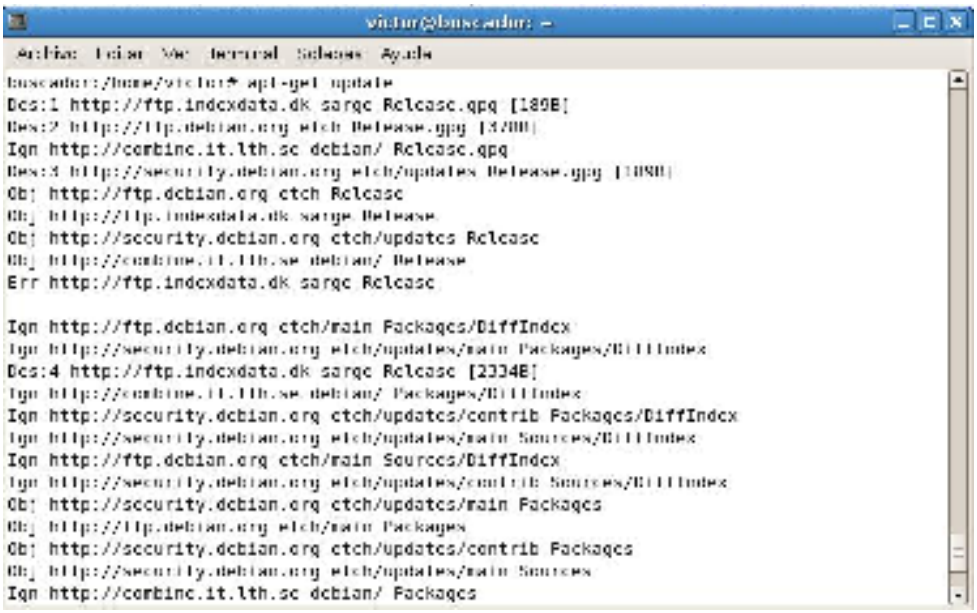


Figura 2.6.5 Actualización de los repositorios de Debian.

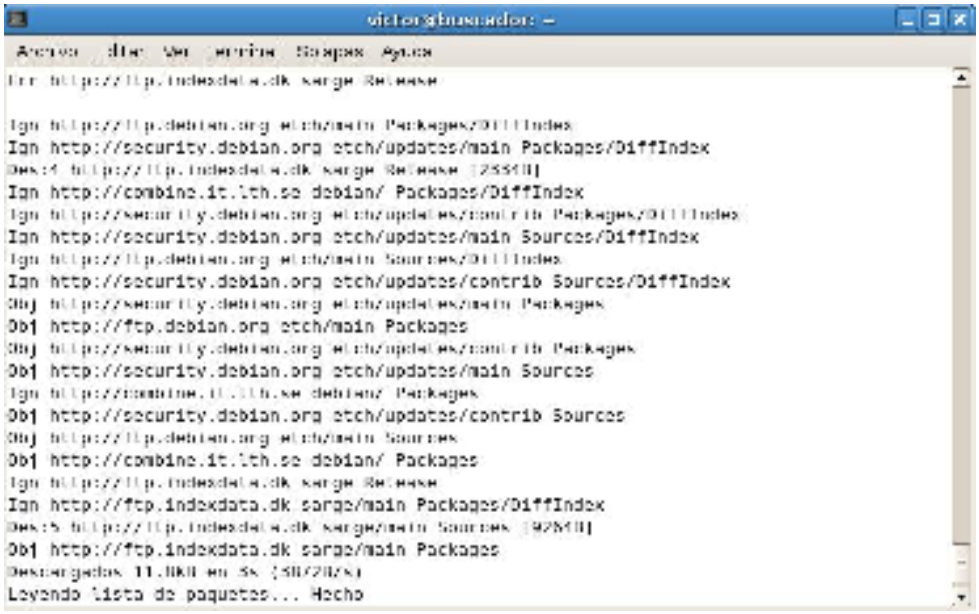


Figura 2.6.6 Actualización de los repositorios de Debian.

Figura 2.6.7 Instalación de Zebra, Yaz y XSLT.

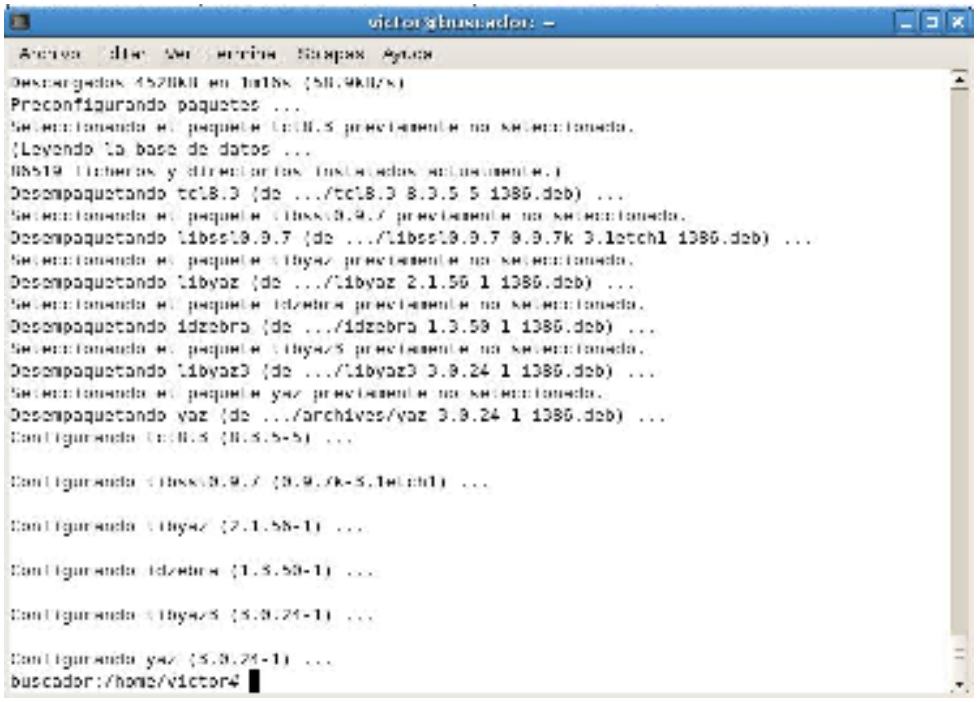


Figura 2.6.8 Instalación de Zebra, Yaz y XSLT.

3- Utilice el comando *tar xzf* para descomprimir el archivo *SEbox.tgz* en la carpeta actual, luego entre a la carpeta *SearchEngineBox* que acaba de descomprimir haciendo uso del comando *cd.*

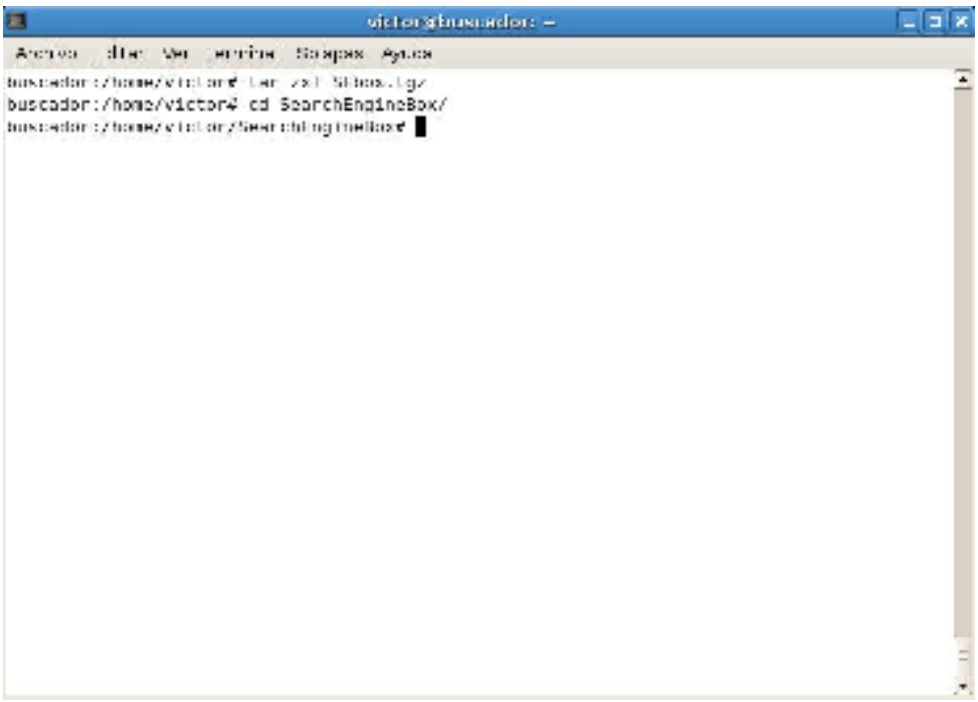


Figura 2.6.9 Descomprimiendo el archivo SEbox.tgz

4- Inicialice el proceso de escrutinio buscador con el comando *combineINIT --jobname buscador*

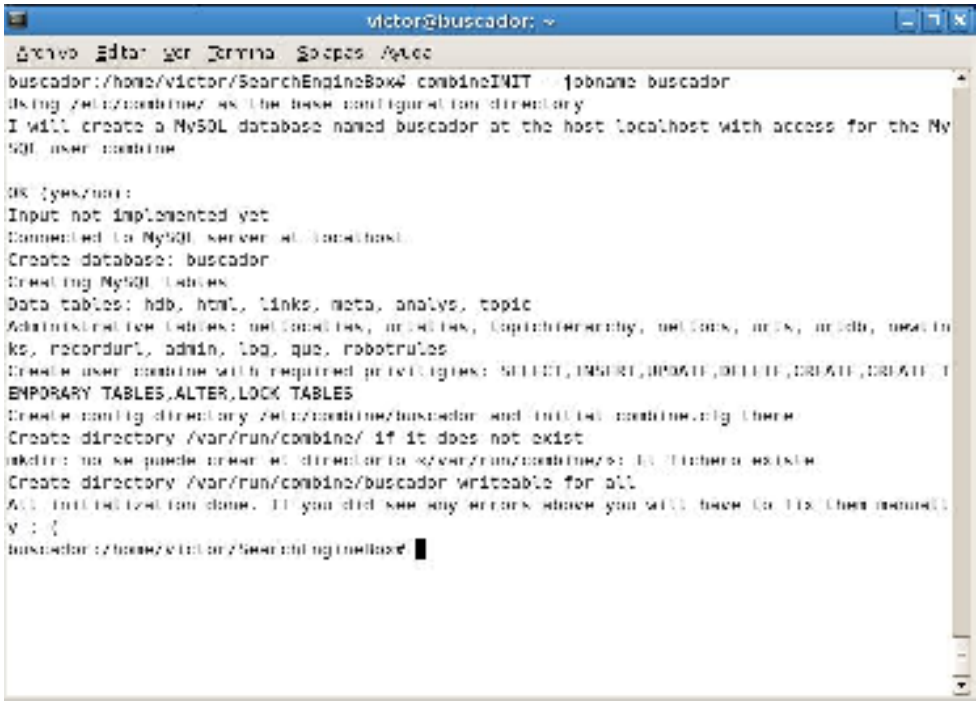


Figura 2.6.10 Inicio del trabajo de escrutinio buscador.

5- Edite el archivo */etc/combine/buscador/combine.cfg* en la parte de *HOST* para delimitar el Combine System a dominios *.sv.*

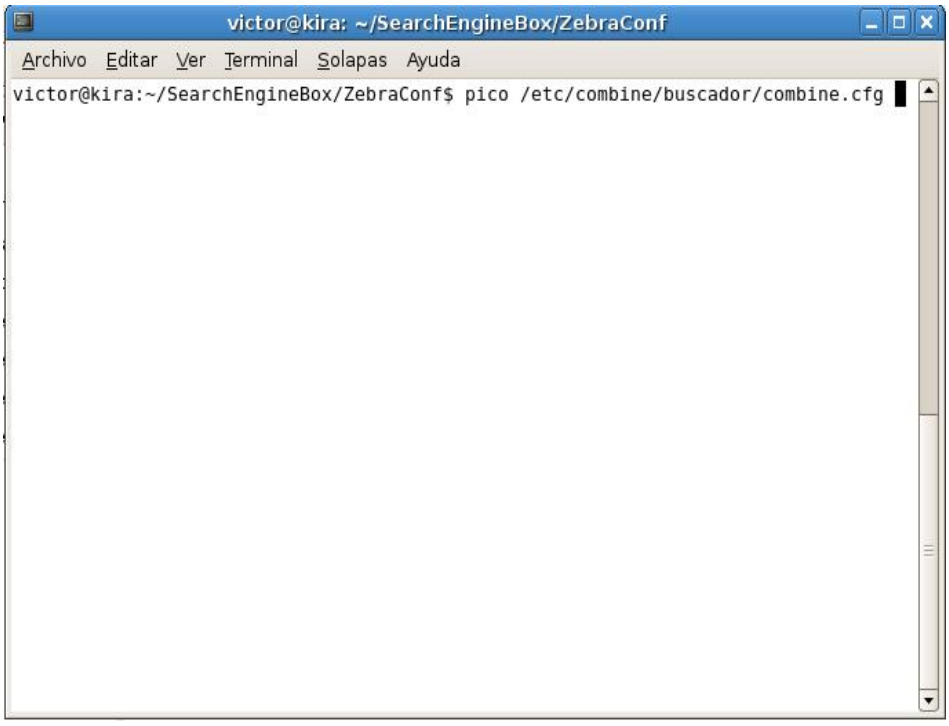


Figura 2.6.11 Editando el archivo combine.cfg

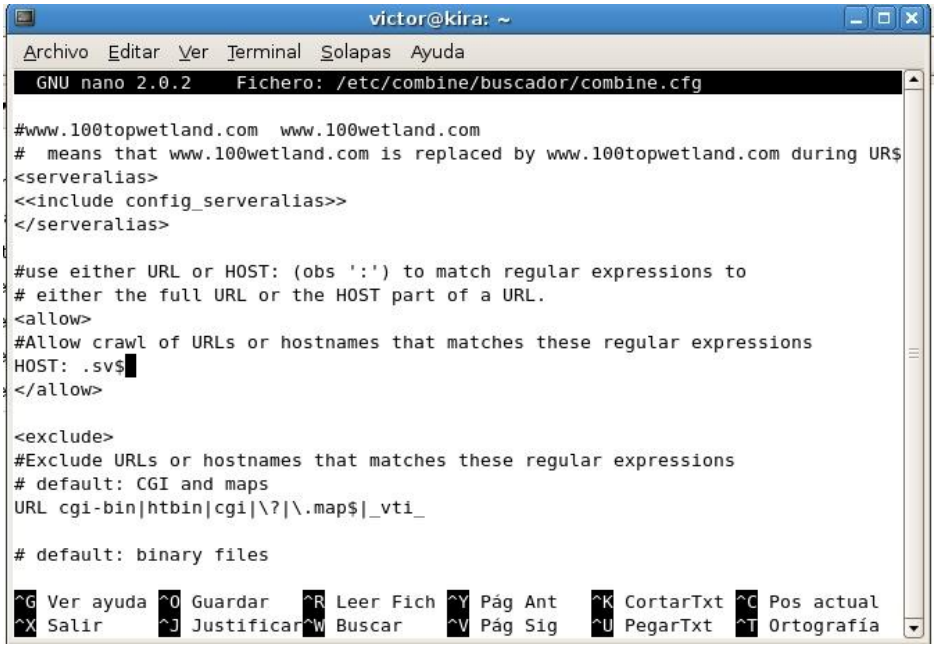


Figura 2.6.12 Editando el archivo combine.cfg

6- Agregue los URL's del archivo *buscaseed.txt* que vienen incluidos en el disco ajunto, haciendo uso del comando *combineCtrl --jobname buscador load < buscaseed.txt*

Figura 2.6.13 Inclusión de las URL's semillas del archivo buscaseed.txt

7- Ingrese a la carpeta *ZebraConf* con el comando *cd*, y edite el archivo *ZebraConf.xml* añadiendo entre *<host>…</host>* su *hostname* (nombre de la PC en el sistema) y entre *<port>…</port>* el numero de puerto que utilizara el servidor Zebra y añada la línea *ZebraHost=hostname:puerto* al final del archivo /etc/combine/buscador/combine.cfg.

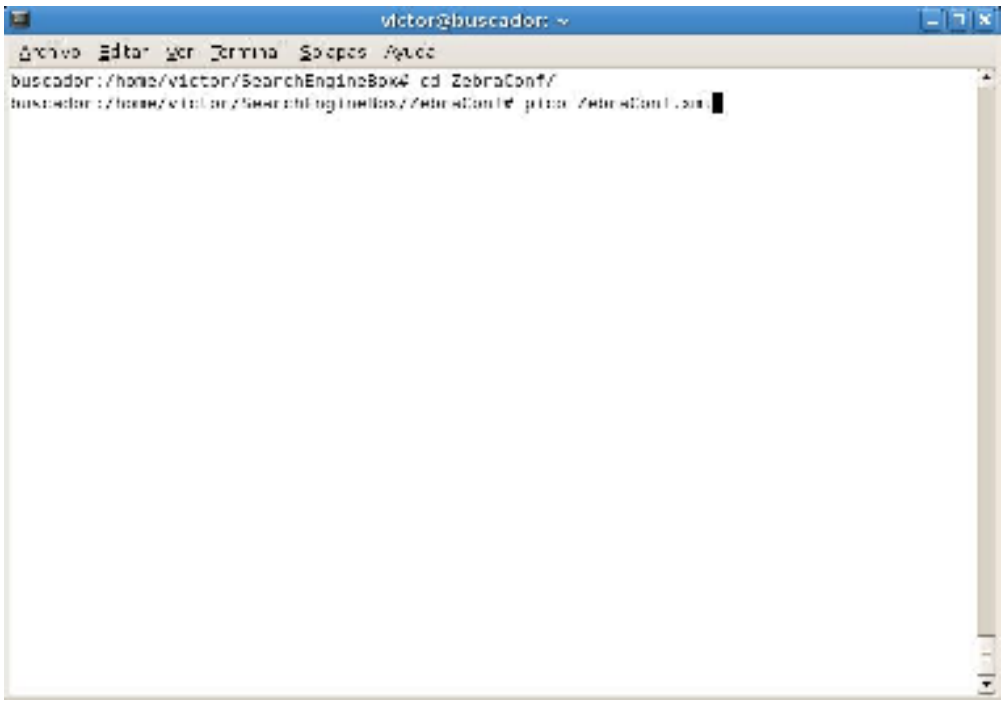


Figura 2.6.14 Editando el archivo ZebraConf.xml

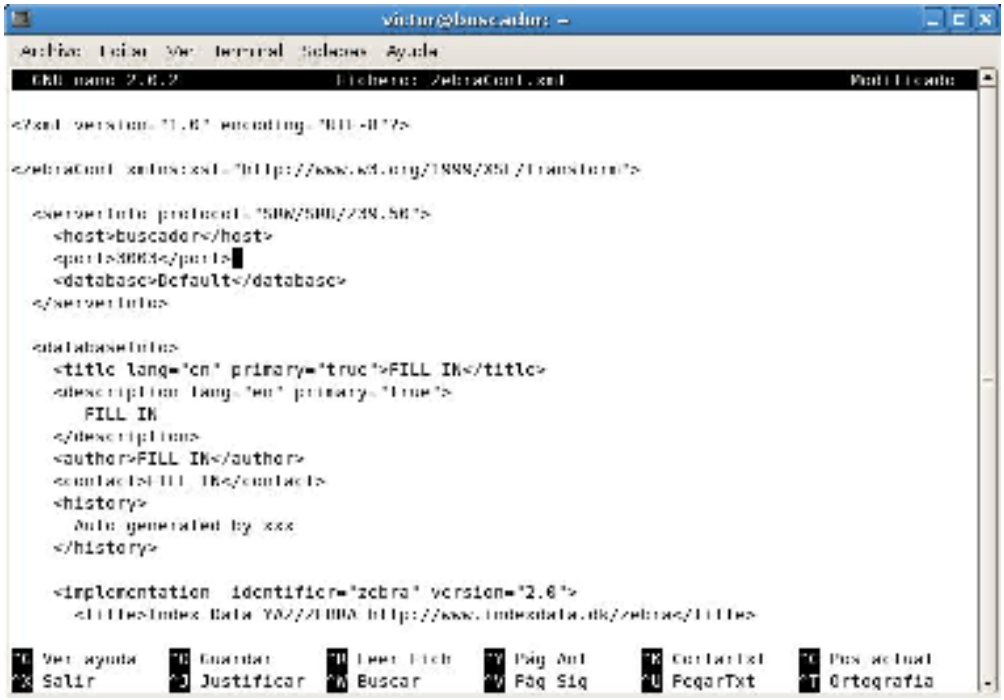


Figura 2.6.15 Editando el archivo ZebraConf.xml

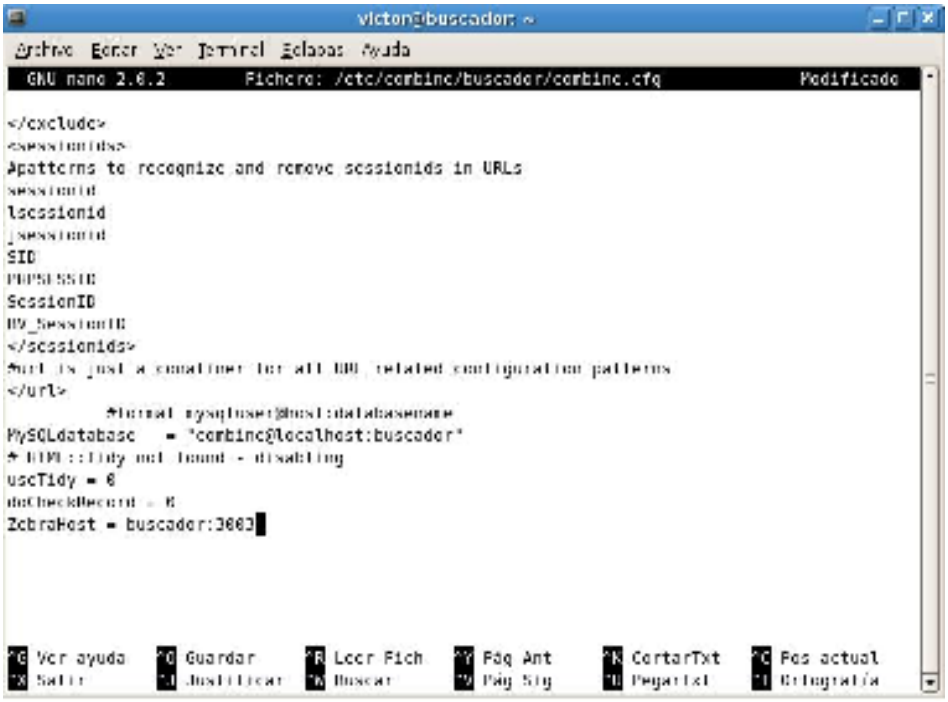


Figura 2.6.16 Editando el archivo combine.cfg

8- Genere el archivo de configuración Zebra haciendo uso de los comandos *make setup* y *make* dentro de la carpeta *ZebraConf.*

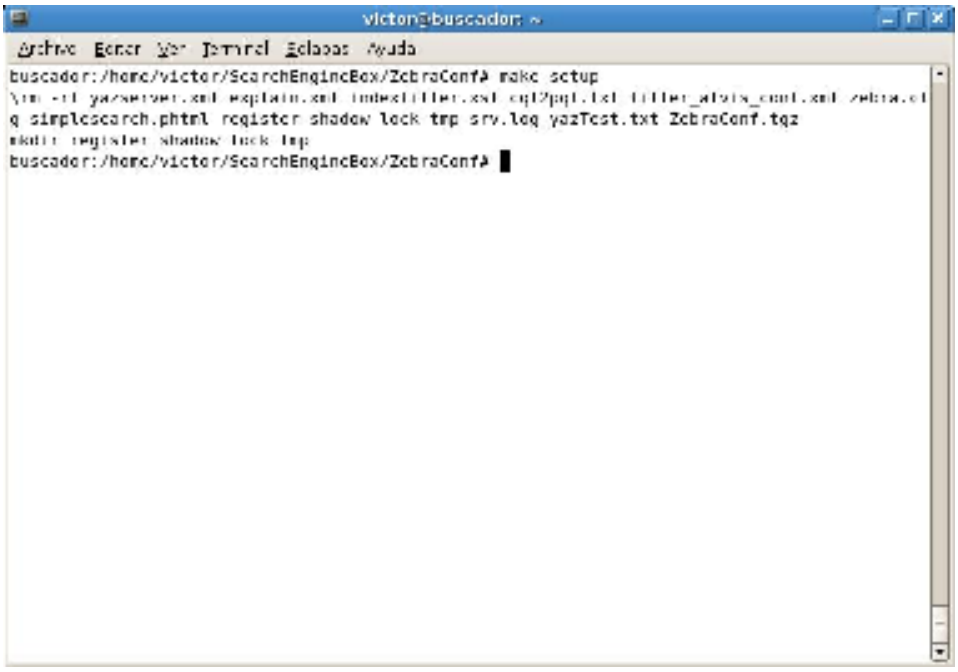


Figura 2.6.17 Uso del comando make setup.

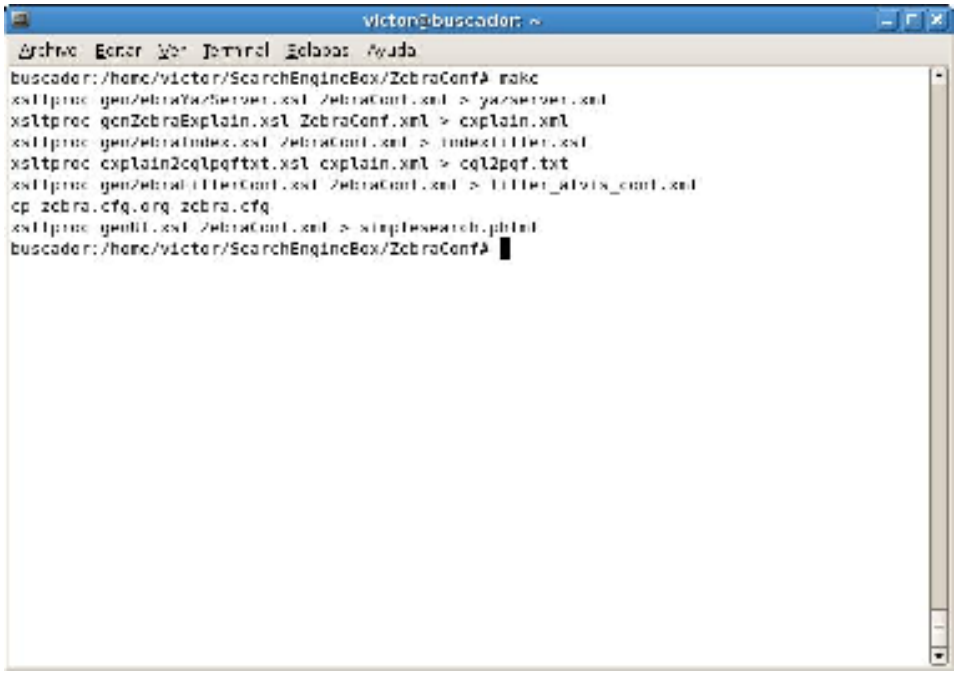


Figura 2.6.18 Uso del comando make.

9- Edite el archivo *zebra.cfg*, cambie *tmpdir* por *settmpdir*

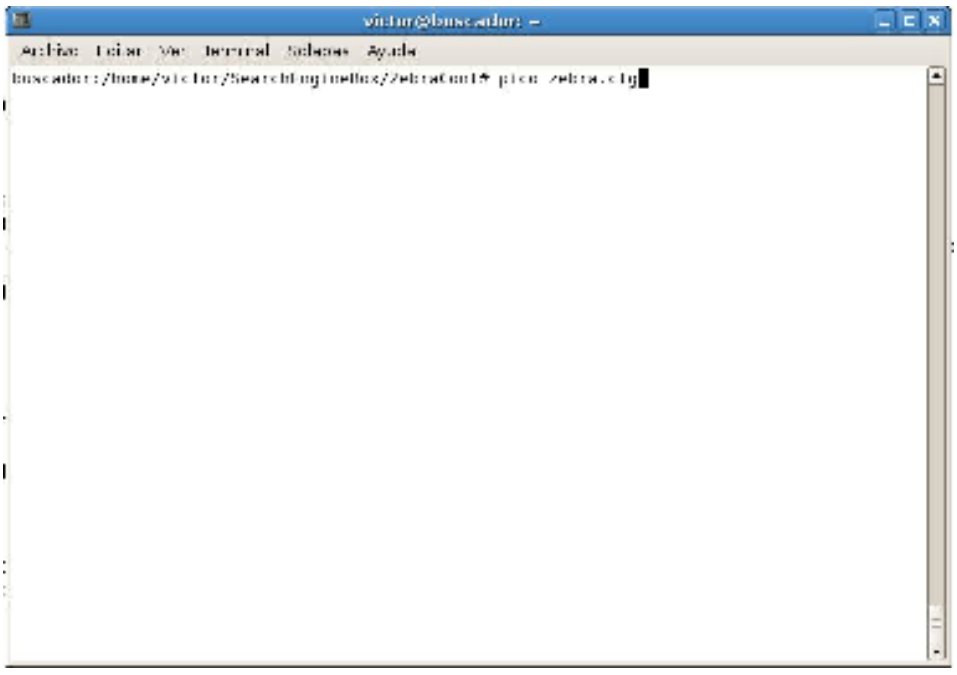


Figura 2.6.19 Editando el archivo zebra.cfg

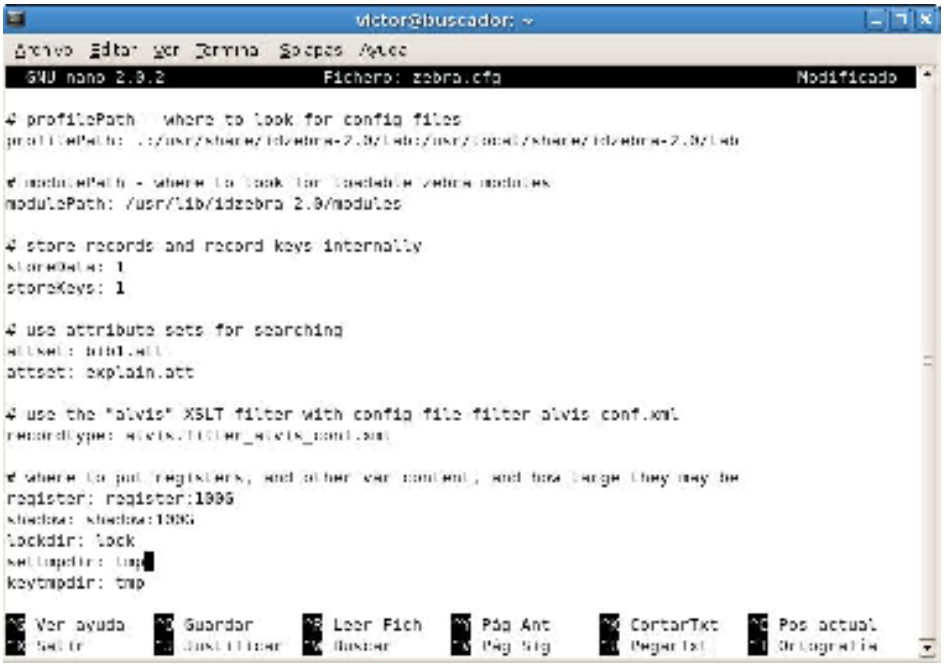


Figura 2.6.20 Editando el archivo zebra.cfg

10-Inicie el servidor Zebra, Yaz y el servidor de base de datos haciendo uso del *comando zebrasrv –f yazserver.xml –l Server.log &*

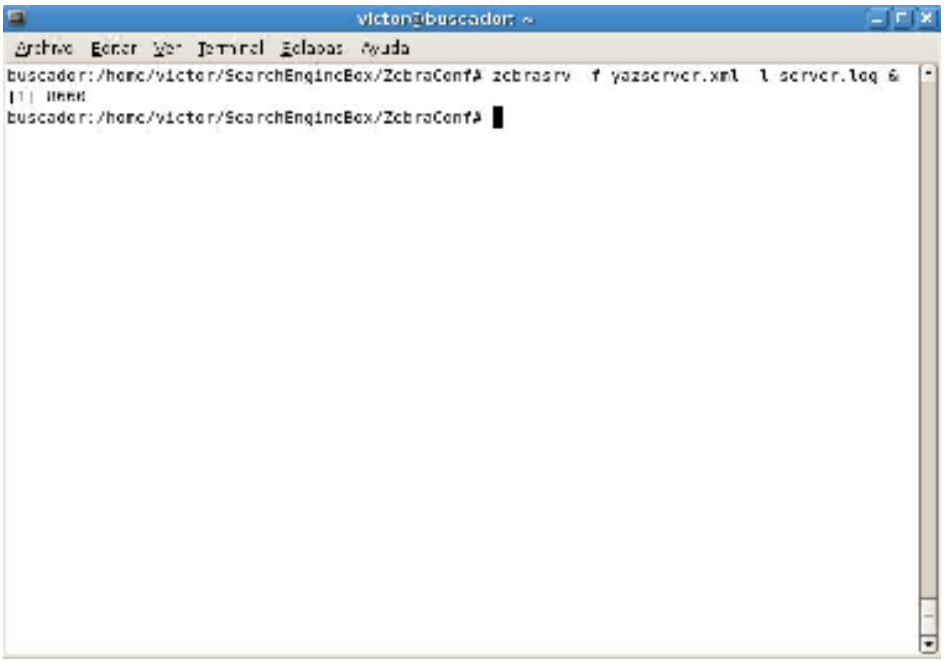


Figura 2.6.21 Iniciando el servidor Zebra.

11-Inicie el trabajo de escrutinio buscador haciendo uso del comando *combineCtrl --jobname buscador start*.

Figura 2.6.22 Iniciando trabajo de escrutinio buscador.

12-Identifique los archivos plantilla *simplesearch.phtml* y *extrRecordData.xsl*, éstos archivos los utilizará en su servidor Web como interfaz grafica de su buscador Web.

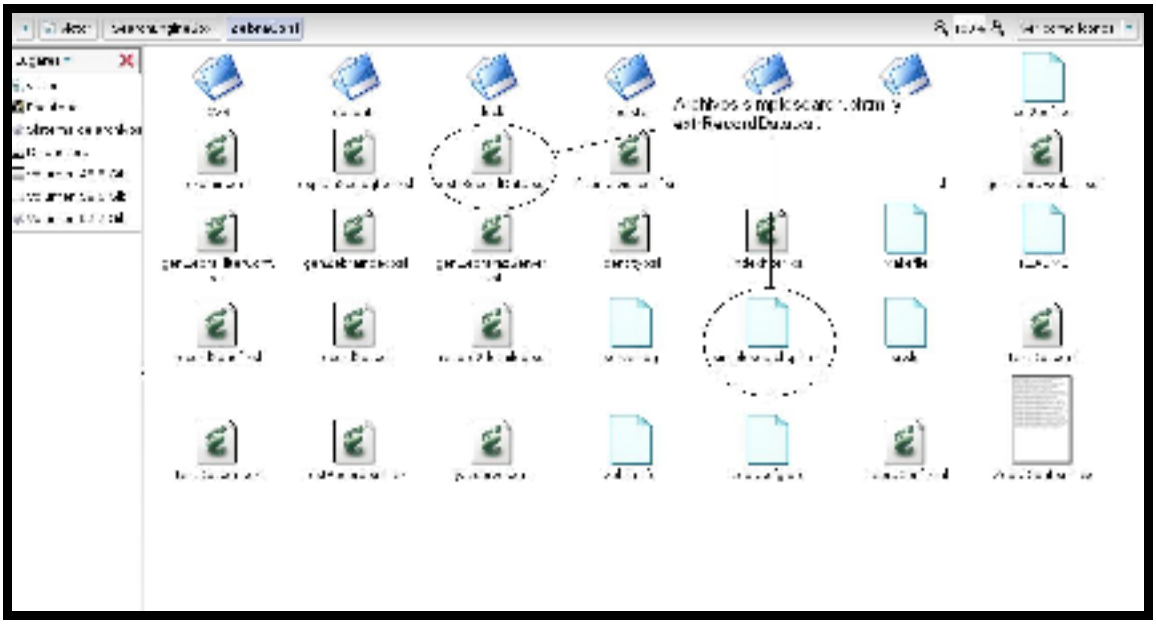


Figura 2.6.23 Localizando archivos plantilla.

13-Personalice el archivo *simplesearch.phtml* haciendo uso de algún editor de archivos PHP/HTML añadiendo etiquetas HTML y rellenando algunos campos o suprimiendo las opciones que no necesite, en éste ejemplo se utilizo *Quanta*, a continuación se muestra el cambio de este archivo para el buscador Web de paginas .sv. al cual se le denomino *index.phtml* (Este archivo modificado se encuentra en el CD adjunto).

Figura 2.6.24 Archivo index.phtml

Figura 2.6.25 Archivo index.phtml

14-Copie el archivo *index.phtml* y *extrRecordData.xsl* a la carpeta del servidor Web apache que, en Debian se encuentra en */var/www/*, haciendo uso del comando *cp*

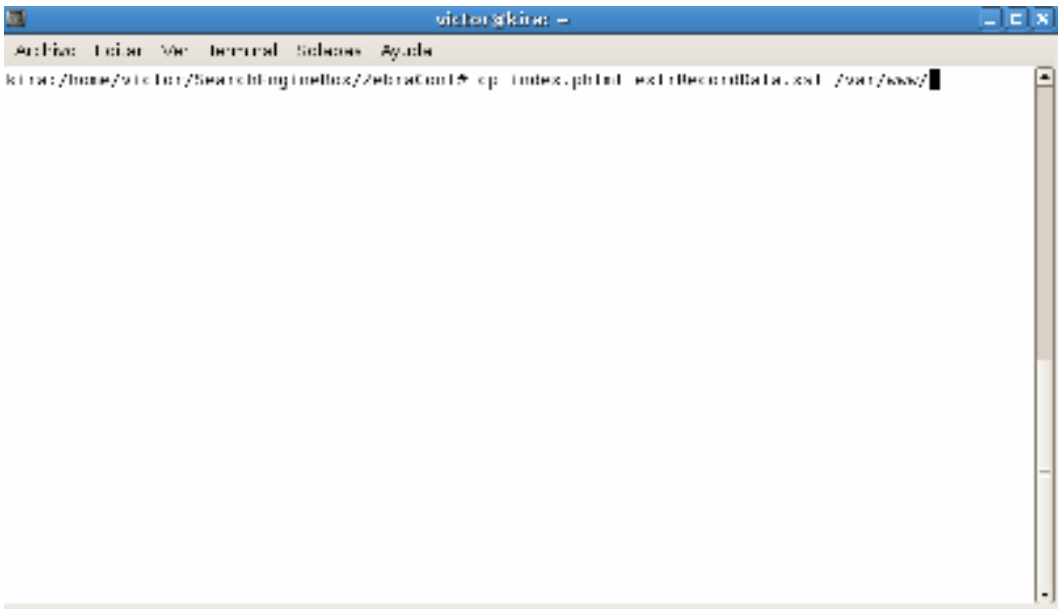


Figura 2.6.26 Copiando index.phtml y extrRecordData.xsl a la carpeta del servidor Web.

15-Compruebe que el buscador funcione correctamente.

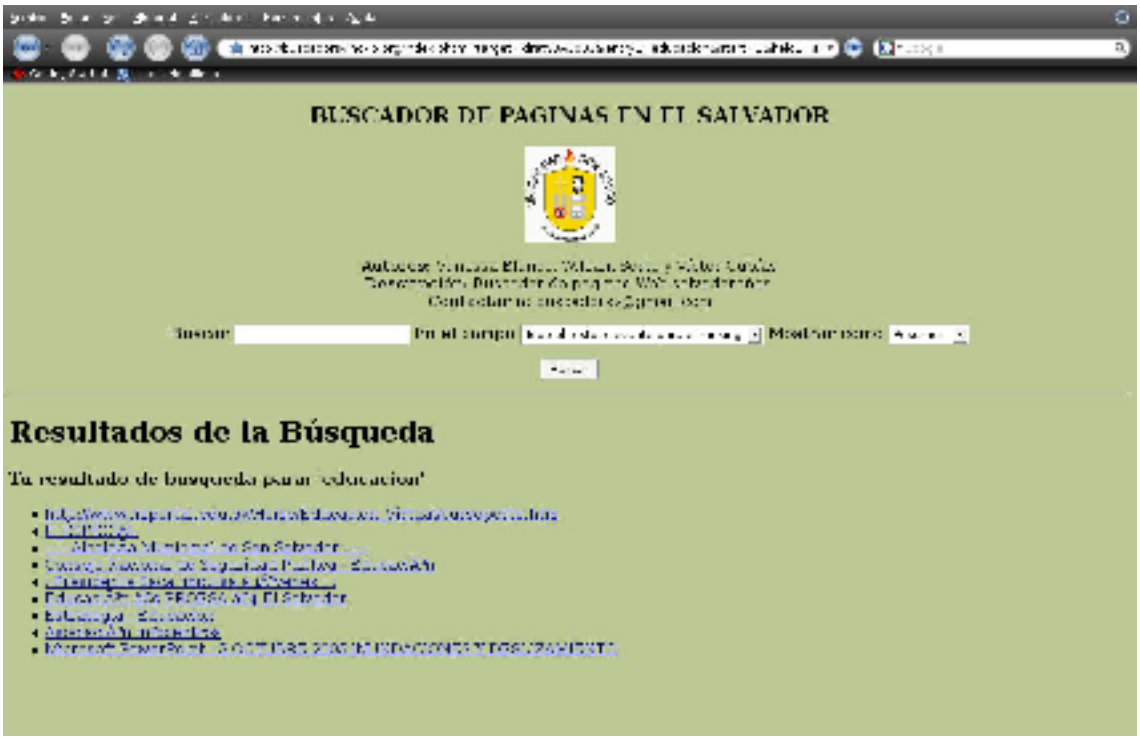


Figura 2.6.27 Buscador paginas .sv

2.6.2 LISTADO DE ARCHIVOS BÁSICOS PARA REALIZAR EL BUSCADOR.

Archivo	Ubicación	Descripción
sources.list	/etc/apt/	Archivo de configuración de Linux Debian encargado de controlar las fuentes de los paquetes de software
SEbox.tgz	/media/cdrom (cd adjunto)	Archivo comprimido de ingeniería en una caja con las plantillas y archivos de configuración de Zebra y Yaz.
combine.cfg	/etc/combine/buscador	Archivo de configuración del trabajo de escrutinio buscador, en el se limita el dominio a .sv y se agregan líneas para que Combine trabaje con Zebra.
buscaseed.txt	/media/cdrom (cd adjunto)	Archivo que contiene los URLs semillas utilizados por el Combine System como punto de partida para su búsqueda.
ZebraConf.xml	/…/SearchEngineBox/ZebraConf	Archivo utilizado por el sistema de ingeniería en una caja, para generar el archivo de configuración zebra.cfg del servidor Zebra.
zebra.cfg	/…/SearchEngineBox/ZebraConf	Archivo de configuración creado por el sistema de ingeniería en una caja utilizado por el servidor Zebra.
Yazserver.xml	/…/SearchEngineBox/ZebraConf	Archivo de configuración utilizado por el sistema de ingeniería en una caja para el cliente Yaz
Server.log	/…/SearchEngineBox/ZebraConf	Archivo creado para seguir las acciones del servidor Zebra.
simplesearch.phtml	/…/SearchEngineBox/ZebraConf	Archivo plantilla incluido en el sistema de ingeniería en una caja para realizar la interfaz grafica del buscador

		Web.
extrRecordData.xsl	/…/SearchEngineBox/ZebraConf	Archivo generado por el sistema de ingeniería en una caja, que permite extraer los datos encontrados en el Combine System.
index.phtml	/media/cdrom (cd adjunto)	Plantilla modificada para el buscador de páginas Web salvadoreñas bajo el dominio .sv.

Auto evaluación.

Instrucciones

Conteste, o realice los ejercicios que se le solicitan.

1. ¿En que carpeta se encuentra el archivo de configuración ZebraConf.xml?

2. ¿Cuáles son los dos archivos plantilla que utiliza la ingeniera en una caja para manejar la interfaz grafica del buscador?

3. ¿Qué es el servidor Zebra?

4. ¿Para que se utiliza Yaz?

5. Cree un trabajo de escrutinio llamado buscadorespañol y configurelo para el dominio .es

Respuestas de las preguntas de la página anterior.

1. …/SearchEngineBox/ZebraConf
2. simplesearch.phtml y extrRecordData.xsl
3. Zebra es un motor o servidor de indexación y recuperación de información de texto
4. para la recuperación de información usando los protocolos Z39.50/SRU, es el cliente utilizado por el servidor Zebra.
5. combineINIT –jobname buscadorespañol
   pico /etc/combine/buscadorespañol/combine.cfg
   HOST: .es$

De 5 a 3 preguntas correctas: Finalizo el curso satisfactoriamente
Menos de 3 preguntas: Repase el capitulo 6 y compruebe el ejercicio.

Investigación Complementaria

- Realice un buscador de páginas Web españolas con el dominio .es, realice todos los pasos necesarios y modifique los archivos plantilla para personalizar su buscador.

GLOSARIO.

A

Algoritmo: Descomposición en pasos u operaciones elementales de cualquier operación de cálculo o proceso analógico para su resolución óptima.

Agente: Es un sistema de *hardware y/o software* que interactúa con su entorno.

Apache: El servidor HTTP Apache es un software (libre) de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

B

Base de datos: Conjunto de datos organizados de modo tal que resulte fácil acceder a ellos, gestionarlos y actualizarlos.

Buscador: Conjunto de programas en Internet cuya finalidad es proporcionar información o referencias de información sobre uno o varios tópicos a los usuarios de la Web.

C

Combine System: Robot Spider escrito en Perl, su fuente se encuentra bajo la licencia GPL.

Cable coaxial: Se trata de un cable de cobre rodeado de aislamiento, un conductor secundario que actúa como *"tierra"* y una cubierta de plástico externa. Gracias a estas dos capas de blindaje el coaxial es relativamente inmune a la interferencia eléctrica.

CGI (Common Gateway Interface): La interfaz de entrada común, es una importante tecnología de la World Wide Web que permite a un cliente solicitar datos de un programa ejecutado en un servidor Web.

D

Datagrama: Entidad de datos autocontenida e independiente que transporta información suficiente para ser encaminada desde su computadora de origen a su destino sin tener que depender de que se haya producido anteriormente tráfico alguno entre ambos y la red de transporte.

DB2: Es una marca comercial, propiedad de IBM, bajo la cual se comercializa el sistema de gestión de base de datos.

Dominio: Es un conjunto de computadores conectados en una red que confían a uno de los equipos de dicha red la administración de los usuarios y los privilegios que cada uno de los usuarios tiene en dicha red.

E

Emma: interfaz grafica en Linux para el control de base de datos en Mysql.

F

Freeware: Aplicaciones de uso gratuito que pueden encontrarse en Internet.

FTP: Siglas de “File Transfer Protocol” (Protocolo de Transferencia de Archivos), es un método efectivo para transferir archivos de información.

FQDN (Fully Qualified Domain Name): Es un nombre entendible por personas que incluye el nombre de la computadora y el nombre de dominio asociado a la misma.

Fibra óptica: tecnología para transmitir información como pulsos luminosos a través de un conducto de fibra de vidrio. La fibra óptica transporta mucha más información que el Cable de cobre convencional. La mayoría de las líneas de larga distancia de las compañías telefónicas utilizan la fibra óptica.

Firebird: Es un sistema de administración de base de datos relacional (o RDBMS) de código abierto, basado en la versión 6 de Interbase, cuyo código fue liberado por Borland en 2000. Su código fue reescrito de C a C++.

H

HTTP: Protocolo que permite la transacción WWW, o sea, permite “navegar” a través de los hiperenlaces de los documentos tipo página Web.

Hub o concentrador: es un dispositivo que permite centralizar el cableado de una red.

I

IETF(Internet Engineering Task Force): Grupo de Trabajo en Ingeniería de Internet, es una organización internacional abierta de normalización, que tiene como objetivos el contribuir a la ingeniería de Internet, actuando en diversas áreas, tales como transporte, encaminamiento, seguridad. Fue creada en EE.UU. en 1986.

Información estructurada: son todos los registros usados por Zebra y manejados internamente usando mecanismos básicos.

Internet: Es una red informática capaz de conectar entre si miles de computadoras de todo el mundo. También es una amplia fuente de información que cambia y se expande constantemente.

IP: Protocolo no orientado a conexión usado tanto por el origen como por el destino para la comunicación de datos a través de una red de paquetes conmutados.

ISO 3166: Estándar que codifica los nombres de países y áreas dependientes y sus principales subdivisiones.

K

Knowbots: Localizan referencias hipertextuales dirigidas hacia un documento o servidor concreto.

M

Motor de búsqueda: Un buscador automático. Funciona Proporcionando una palabra al motor de búsqueda luego se espera la respuesta y en pocos segundos se obtienen los enlaces a las páginas Web relativas a la palabra proporcionada.

MySQL: Es un sistema de gestión de base de datos relacional, multihilo y multiusuario. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Metadatos: Son datos que describen otros datos. En general, un grupo de metadatos se refiere a un grupo de datos, llamado recurso.

Modelo Instruccional: Es un método didáctico el cual utiliza la elaboración de ensayos, diseño de mapas conceptuales, esquemas, la discusión y trabajo cooperativo mediante los espacios virtuales, la ilustración y análisis de casos concretos en el contexto inmediato de los estudiantes, la revisión crítica de textos, ejercicios y tareas con un grado progresivo de dificultad e integración de los conceptos, tanto como su transferencia a situaciones de la vida cotidiana, para concluir en la formulación de un diseño instruccional que aproveche los recursos y lenguajes generados en la integración de medios y nuevas tecnologías de información.

N

Navegador: También se le llama Browser, un Navegador es un paquete de Software especialmente diseñado que permite explorar las páginas en el Web e ir de una página a otra.

Network: (red) Una red de computadoras es un sistema de comunicación de datos que conecta entre si sistemas informáticos situados en diferentes lugares. Puede estar compuesta por diferentes combinaciones de diversos tipos de redes.

O

Oracle: Sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation. Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su soporte de transacciones, estabilidad, escalabilidad y multiplataforma.

ODBC (Open Database Connectivity): Estándar de acceso a Bases de Datos desarrollado por Microsoft Corporation, el objetivo de ODBC es hacer posible el acceder a cualquier dato de cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos (DBMS por sus siglas en inglés) almacene los datos, ODBC logra esto al insertar una capa intermedia llamada manejador de Bases de Datos, entre la aplicación y el DBMS.

P

PageRank: Es una familia de algoritmos utilizados para asignar de forma numérica la relevancia de los documentos (o páginas Web) indexados por un motor de búsqueda.

Parseo: Método mediante el cual se transforma una entrada de texto en una estructura de datos (usualmente un árbol) que es apropiada para ser procesada.

Par trenzado: Cable similar a los pares telefónicos estándar, que consiste en dos cables aislados "trenzados" entre sí y encapsulados en plástico. Los pares aislados vienen en dos formas: cubiertos y descubiertos.

PostgreSQL: Servidor de base de datos objeto relacional libre, liberado bajo la licencia BSD. Como muchos otros proyectos *open source*, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo, dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

Protocolo: Es un conjunto estricto de reglas o procedimientos que se requieren para iniciar y mantener las comunicaciones.

Q

Quanta: Es una herramienta de desarrollo Web para el escritorio KDE, permite editar archivos HTML y PHP entre otros.

R

Robot: Dispositivo electrónico o mecánico, que desempeña tareas automáticamente, ya sea de acuerdo a supervisión humana directa, a través de un programa predefinido o siguiendo un conjunto de reglas generales.

Robot Web: Un robot es un término aplicado a programas de comunicación que utilizando HTTP como protocolo de comunicación, exploran grandes porciones del Web y de manera recursiva extraen información de ellas.

RFC: Request For Comments (abreviado como RFC), se traduce como *"petición de comentarios"*; es un documento cuyo contenido es una propuesta oficial para un nuevo protocolo de la red Internet (originalmente de ARPANET), que se explica con todo detalle para que en caso de ser aceptado pueda ser implementado sin ambigüedades.

S

Servidor Web: Es una computadora que contiene una colección de páginas Web interrelacionadas, además de otros recursos complementarios como imágenes, documentos, etc.

Sitio Web: También se le llama Home Page, y es un archivo que ha sido escrito utilizando generalmente el lenguaje de programación HTML, y que esta alojado en un servidor (computador) que es reconocido dentro de Internet.

Spider: Programa diseñado para recorrer la Web siguiendo los enlaces entre páginas.

Shareware: Software protegido por leyes de copyright, que se encuentra disponible gratuitamente durante cierto tiempo para su evaluación por el usuario. Tras pasar dicho tiempo, el programa expira y no podrá volver a ser utilizado, a no ser que el usuario registre el programa por un precio.

Softbot* (*software robot): Es un agente que usa herramientas de software y servicios basados en el comportamiento de las personas.

SQLite: Es un sistema de gestión de bases de datos relacional compatible con ACID, y que está contenida en una relativamente pequeña librería en C. SQLite es un proyecto de dominio público creado por D. Richard Hipp.

SRU: Versión avanzada y mas actualizada del protocolo Z39.50

T

TCP: Protocolo que garantiza los datos serán entregados a su destino sin errores y en el orden en que se transmitieron.

Texto puro o sin tratar: texto seleccionado para proporcionar un argumento de texto a Zebra.

U

URL: Localizador uniforme de recursos, sirve para invocar un documento Web desde los programas navegadores para Internet.

W

Web: Son páginas que utilizan un lenguaje especial llamado HTML, que permite presentar en pantalla texto y gráficos. Estas páginas contienen referencias o enlaces que permiten acceder a otras páginas.

Web Ants*:* Son robots que cooperan en un mismo objetivo.

Web crawlers: Programa que inspecciona las páginas del World Wide Web de forma metódica y automatizada.

Worms: Es una réplica de un programa, a diferencia de un robot que es un programa original.

Wanderes: *S*on robots que realizan estadísticas sobre la Web.

Y

Yaz: librería hecha en C/C++ para la aplicación de recuperación de información usando los protocolos Z39.50/SRU.

Z

Z39.50: Protocolo que especifica el formato y los procedimientos que gobiernan el intercambio de mensajes entre un cliente y un servidor, para la recuperación estructurada de la información distribuida.

Zebra: Motor de indexación y recuperación de información de texto estructurado de alto rendimiento.

CONCLUSIONES.

1. Se realizo una monografía que consta de una base teórica y una guía practica de los componentes de un robot spider o explorador de la red.

2. Se implementó un robot spider ya existente, con las modificaciones necesarias para que funcionase dentro de un buscador Web.

3. Se realizó una guía que sirve de herramienta de apoyo a la comunidad académica para poder desarrollar un buscador propio.

4. Se logro culminar la monografía con el desarrollo de un buscador para páginas salvadoreñas, delimitado al dominio .sv.

BIBLIOGRAFÍA.

a) Libros y Documentos.

- "Diseño, desarrollo e implementación de un buscador de sitios Web nacionales", Rolando Francisco Álvarez Campos, Jorge Alexander Cruz Bautista, Guadalupe Mejía Salguero, Tesis Universidad Don Bosco (UDB), 2000.

- "Diseño y desarrollo de un prototipo de robot Web para instituciones educativas", Rafael Alejandro Alvarenga Medrano, Juan Gerardo Molina Braun, Rodrigo Humberto Romero Orellana, Tesis Universidad Don Bosco (UDB), 2006.

- "Desarrollo e implementación de un buscador de sitios Web para El Salvador", Elio David Vides, Tesis Universidad Centroamericana "José Simeón Cañas" (UCA), 1998.

- "Cómo Programar en Java", Deitel, Harvey M., MEXICO, MEXICO: PRENTICE HALL, 2001, 1a Edición.

- **"**Fundamentos de Programación en Perl" Wyke, Allen, BOGOTÁ, COLOMBIA, McGRAW HILL, 2001, 1a Edición.

- **"**Como elaborar y asesorar una investigación de tesis", Carlos Muñoz Razo, PRENTICE HALL Hispanoamérica, 1998, 1a Edición.

- "Redes de Telecomunicaciones: Protocolos, Modelado y Análisis", Schwartz, Mischa, ARGENTINA: ADDISON WESLEY, 1ª Edicion, 1995.

- "Creación de Sitios Web con XML" Floyd, Michael, MADRID, ESPAÑA : PRENTICE HALL, 2000, 1ª Edicion.

- La Biblia de PHP 5, Coggeshall, Jojn, MADRID, ESPAÑA, 2005, 1ª Edición.

- Fundamentos de Programación en Perl, Wyke, Allen, BOGOTÁ, COLOMBIA, McGRAW HILL, 2001, 1a. Edición.

- Guía de referencia y aprendizaje Linux, Welsh, Matt; Kalle Dalheimer, Matthias; Kaufman, Lar, Editorial Anaya, 2000, Madrid.

- Unix y Linux: guía práctica, Sánchez Prieto, Sebastián, Editorial Alfaomega, 1999, México.

b) Información consultada en Internet.

- http://combine.it.lth.se/ Sitio oficial del robot Combine System, apartado de documentación del sistema Combine y la ingeniería en caja, el sitio contiene descarga de los archivos de instalación del combine y el manual completo acerca de su uso. Sitio realizado por el departamento de información y tecnología de la Universidad de Lund.

- http://www.w3c.es/Consorcio/ W3C Consorcio del World Wide Web.

- http://proton.ucting.udg.mx/materias/robotica/ Universidad de Guadalajara.

- http://es.geocities.com/robot_de_busqueda/Contenidos.html#rbSheila García Pérez, Página Web creada para la asignatura: Sistemas Avanzados de Recuperación de Información, Universidad Carlos III de Madrid.

- http://www.java.com/es/about/ Sitio oficial de JAVA.

- http://perlenespanol.baboonsoftware.com/tutoriales/aprendiendo_perl/perl_basico_parte_1.html Sitio de Perl en español realizado por el programador Uriel Lizama.

- http://gluc.unicauca.edu.co/wiki/index.php/Curso_Python

  Grupo GNU/linux de la Universidad del Cauca (Colombia).

- http://www.ujaen.es/sci/redes/web/norm-r120.pdf Normativa de uso del servicio Web en la Universidad de Jaén.

- http://www.robotstxt.org/wc/norobots-rfc.html Documentación en idioma ingles que define un método para los administradores de sitios en Internet, que dan instrucciones a los robots visitantes de páginas Web especificando el "protocolo de exclusión de robots".

- http://www.towercom.es/nsp00016.html Enlace Web que contiene información técnica acerca de los robots Web, utilidades y protocolo de exclusión de robots.

- http://www.abcdatos.com/buscadores/robot.html Página Web que contiene información general referente a los robots Web, agentes y motores de búsqueda.

- http://manuales.ojobuscador.com/ Página Web que contiene La historia de los buscadores cronológicamente desde sus inicios, evoluciones y sus creadores. Así como también una breve descripción de cada uno de los buscadores más importantes existentes en la Web.

- http://www.unam.edu.mx/ Archivo pdf obtenido por medio de la pagina de la UNAM, elaborado por José Flores Peñalosa. Contiene información sobre la historia, arquitectura y diferentes algoritmos de búsqueda de los buscadores Web.

- http://modelosrecuperacion.50webs.com/difusa.htm Sitio Web del programador Julio A. Ayala acerca de Recuperación y organización de la información.

- http://es.wikipedia.org/ Enciclopedia, entendida como soporte que permite la recopilación, el almacenamiento y la transmisión de la información de forma estructurada.

- http://www.unav.es/cti/manuales/TutorialJavaScript/indices/ Sitio del centro de tecnología informática de la Universidad de Navarro, tutorial de Javascript.

- http://www.php.net/ Sitio oficial de PHP, en el cual se encuentran foros, manuales, descargas y documentación acerca de este lenguaje de programación.

- http://www.itq.edu.mx/vidatec/espacio/aisc/windowsnt/ServidorDNS.html Sitio oficial del Instituto Tecnológico de Querétaro de México, apartado sobre el sistema DNS, Introducción, historia y explicación del sistema de dominio de nombres en general.

- http://infolab.stanford.edu/~backrub/google.html Sitio oficial de la Universidad de Stanford Infolab, apartado funcionamiento de Google y PageRank.

- http://www.definicion.org/ Diccionario de definiciones de conceptos técnicos.

- http://www.gueb.org/ Sitio creado por el programador Calderón de la Barca, apartado del funcionamiento de Google acerca de PageRank, ranking en Google y peso.

- http://www.programacionenc.net/ Sitio de programación en C/C++ creado por el Webmaster Adrián Fernando Vaca, apartado de sockets, funcionamiento y usos.

- http://www.arrakis.es/~tobal/ia.htm#proceso Sitio del programador Javi Tobal, apartado de indexación automática de documentos, proceso general de indexación y como evaluar el resultado de una búsqueda.

- http://www.uam.es/ Sitio oficial de la Universidad Autónoma de Madrid, apartado de knowbots, específicamente ejemplos.

- http://servidorti.uib.es/adelaida/tice/modul6/memfin.pdf Sitio Web de la Universidad De Les Iles Balears Departamento de Ciencias matemáticas e informática, documento PDF Mecanismos de recuperación de información en la WWW.

ANEXOS.

Apéndice A. Etiquetas HTML.

Etiquetas y estructura.

Las etiquetas usadas para definir la estructura de los documentos HTML son:

- <HTML> y </HTML>. Se usan para indicar el inicio y el fin de un documento HTML. Todos los demás elementos del documento deben aparecer entre estas etiquetas.
- <HEAD> y </HEAD>. Definen la sección del encabezado de los documentos HTML. Éste suele contener información sobre el documento que no se muestra directamente al usuario. Como por ejemplo el título de la ventana de su navegador.

  Dentro de la cabecera <HEAD> podemos encontrar:

  - <TITLE> y </TITLE>. Define el título de la página. Por lo general, el título aparece en la barra de título encima de la ventana.
  - <link>: para vincular el sitio a hojas de estilo o íconos Por ejemplo:<link rel="stylesheet" href="/style.css" type="text/css">.
- <BODY> y </BODY>. Define el contenido principal o cuerpo del documento, ésta es la parte del documento HTML que se muestra en el navegador, dentro de esta etiqueta pueden definirse propiedades comunes a toda la página, como color de fondo y márgenes.

  Dentro del cuerpo <BODY> podemos encontrar numerosas etiquetas.

  A continuación se indican algunas a modo de ejemplo:

  - <H1>, <H2>, <H3>, <h4>, <H5>, <H6>. Encabezados o títulos del documento con diferente relevancia.
  - <a>. Hipervínculo o enlace, dentro o fuera del sitio Web. Debe definirse el parámetro de pasada por medio del atributo href. Por ejemplo: Google se representa como Google.
  - . Área de la página.
  - <img>. Imagen, requiere del atributo src, que indica la ruta en la que se encuentra la imagen. Por ejemplo: .

- <font color="#rrggbb">. Color del texto, representado por un código hexadecimal. Cada par puede variar entre 00(el tono más oscuro) a ff(más claro).
- <marquee>"texto"</marquee>. Esta etiquete presenta el texto en movimiento horizontal.

Etiquetas de estilo.

Son etiquetas que afectan la apariencia del texto en un documento HTML.

- y . El texto entre estas etiquetas aparece en negrita.
- y . El texto entre estas etiquetas aparece en cursiva.
- <u> y </u>. El texto entre estas etiquetas aparece en subrayado.

Etiquetas de formato.

Estas etiquetas afectan el formato de línea del texto del documento HTML.

-
. Esta etiqueta inserta un salto de línea, provocando que el navegador muestre el siguiente texto en una nueva línea.
- <P>. Inicia un nuevo párrafo en el texto.

Etiquetas para tablas.

Las tablas son usadas pasa controlar la presentación y alineación del texto plano e imágenes, los recientes estándares de HTML dinámico y la hojas de estilo en cascada (CSS, Cascading Style Sheets) están enfocadas a la presentación de detalles.

Las etiquetas usadas para definir tablas son:

- <TABLE> </TABLE>. Definen una tabla. El resto de las etiquetas de tablas se definen dentro de éstas.
- <TR> y </TR>. Estas etiquetas definen un nuevo renglón en la tabla.
- <TD> y </TD>. Se usan para definir una celda dentro de un renglón.
- <TH> y </TH>. Se usan igual que <TD y <>/TD>, solo que definen un encabezado de columna en lugar de una celda de datos.

Apéndice B. Guía de CPAN.

CPAN – *"The Comprehensive Perl Archive Network",* básicamente es un directorio central donde se pueden encontrar módulos y programas creados por terceras personas en Perl.

Su dirección es *http://search.cpan.org/*, donde se ve un mecanismo de búsqueda junto con él directorio organizado en categorías específicas.

Aquí podemos buscar módulos creados por otras personas, descargarlos y usarlos en nuestros propios desarrollos.

¿Qué es un módulo?

Un módulo es una parte independiente de algo, en este caso los módulos de Perl son partes de códigos independientes que podemos ensamblar con nuestros propios programas.

Estos módulos son pedazos de código diseñados para ser rehusados por otros programadores que se vean en la misma necesidad.

¿Cómo busca los módulos Perl?

Cuando queremos usar un módulo la sintaxis que usamos es:

```
use Modulo;
```

Con esta directiva le estamos diciendo a Perl que queremos usar un módulo que se llama *"Modulo",* entonces lo que va a hacer Perl es leer varios directorios, estos directorios se encuentran en un arreglo (array) que se llama *@INC* y son los directorios conocidos como librerías.

Es decir Perl, en este arreglo, tiene varias direcciones a directorios donde va a buscar los módulos, entonces cuando busca un módulo va directorio por directorio hasta encontrarlo, de lo contrario regresa un error.

Hay veces que nuestra sintaxis al llamar un módulo es:

```
use Modulo::ModuloChico;
```

Al hacer esto le estamos diciendo que queremos que busque el *"ModuloChico"* dentro de un directorio llamado *"Modulo"* que debe de encontrarse en algún directorio de @*INC.*

En la llamada de modulo anterior, los dos puntos son similares a *"/"* en Windows y en UNIX, así que una llamada a un módulo se realizaría de esta manera:

```
use Modulo::ModuloChico::Modulito;
```

Lo que interpretaría Perl sería, busca el módulo *"Modulito"* en el directorio *"ModuloChico"* que esta en el directorio "Modulo" que debe de encontrarse en algún directorio de @*INC.*

PPM.exe

PPM.exe es una aplicación que viene con la distribución de Perl. PPM significa *"Perl Package Manager",* que significa *"administrador de paquete de Perl".*

Lo que hace este programa es que con unos cuantos comandos instala, actualiza y desinstala el módulo que queramos.

Ejecutando el PPM.exe

Para poder ejecutar esta aplicación primero debemos de entrar a la ventana de comandos de Windows ("COMMAND"), para hacer esto, se debe ir al Menú Inicio → Ejecutar, teclear *"command".*

Luego ir al directorio *"bin"* que se encuentra donde está instalado Perl, normalmente es *"C:\perl\bin"*, así que se teclea:

```
C:\perl\bin
```

Ya que se esta en el directorio ahora se teclea:

```
ppm
```

Como es la primera vez que se ejecuta el programa va comenzar un proceso paso a paso tras una secuencia de configuración, lo recomendable es que siempre se ponga que sí a los *"defaults"* que da el programa.

Las siguientes veces que se ejecute el programa debe de aparecer el siguiente mensaje:

PPM interactive shell (2.1.5) - type 'help' for available commands.
PPM>

Instalar Módulos.

Ya que se esta dentro del programa para instalar un módulo todo lo que se debe hacer es escribir:

install MODULO

Por ejemplo para instalar el módulo *MIME::Lite*, se digita:

install MIME::Lite

Al escribirlo el programa preguntara si en realidad se quiere instalar el módulo, todo lo que deba hacer es poner *"y"* - *(yes)* y listo, el programa les instalará todo.

NOTA: Se debe estar conectado a Internet para poder realizar este pasó.

Apéndice C. Respuestas a las preguntas de repaso.

Preguntas de repaso del apartado 1.1 al 1.2.3.

1. ¿Qué es un robot aplicado al Web?

   Es un programa que rastrea recursos a través del Internet siguiendo los vínculos que contienen las páginas Web.

2. ¿Qué es un agente?

   Es un sistema de *hardware y/o software* que interactúa con su entorno.

3. Clasifique los siguientes programas según el agente al que pertenece autónomo, inteligente o de usuario. Ejemplo*: Mozilla Fire Fox usuario*

Navegador Web Opera	**Usuario.**
Googlebot	**Autónomo.**
Formulario de relleno gmail	**Inteligente.**
MsnBot	**Autónomo.**
Navegador Internet Explorer	**Usuario.**

Preguntas de repaso del apartado 1.3 al 1.3.6.

1. ¿En que se basan las clasificaciones de los robots spider?

   En la forma en que se comportan o en la manera como éstos se mueven a través de Internet.

2. Mencione y explique brevemente tres clasificaciones de los robots spider.

   Arañas *(Spiders*): es un programa usado para rastrear la red. Lee la estructura de hipertexto y accede a todos los enlaces referidos en el sitio Web.

   Hormigas (*Web Ants*): Trabajan de forma distribuida, explorando simultáneamente diferentes porciones de la Web.

Orugas (*Web crawlers*): Los Web crawlers se utilizan para crear una copia de todas las páginas Web visitadas para su procesado posterior por un motor de búsqueda que indexa las páginas proporcionando un sistema de búsquedas rápido.

3. Mencione al menos un robot spider que se comporte como:
   Gusano: ***Bagle, NetSky o Passer****.*
   Oruga: ***WebCrawler y MetaCrawler.***
   Araña: ***Googlebot, Combine System y BDDBot.***

Preguntas de repaso del apartado 1.4 al 1.4.5.

1. Marque con X en el enunciado que mejor defina el comportamiento de los robots de mantenimiento:
   __ Explora y reúne grandes porciones del Web.

   __ Extrae información y la utiliza para fines estadísticos

   _X_ Esta orientado a la administración de la estructura hipertexto en servidores Web.

2. Defina los siguientes tipos de robots:
   Robot de copia a espejo:
   Son los robots utilizados para mantener estructuras de hipertexto redundante con el objetivo de proveer una respuesta rápida y segura a los fallos en los servicios Web.

   Robot estadístico:
   Este tipo de robot extrae información y la utiliza para realizar cálculos de tipo estadísticos.

Robot combinado:

Entra en esta clasificación todo robot capaz de realizar tareas propias de otros tipos de robot, pero que no cumplen con algunas características propias de alguno de ellos.

Preguntas de Repaso del apartado 2.1 al 2.1.1.7.

1. Complete las oraciones de acuerdo a la capa del modelo OSI que corresponde.
   a) Proporciona transferencia de tramas de datos desde un nodo a otro. **Capa de Enlace.**
   b) Controla el funcionamiento de la subred que decide la ruta física que deben seguir los datos. **Capa de Red.**
   c) Sirve como ventana para que los usuarios y procesos de aplicación tengan acceso a servicios de red. **Capa de Aplicación.**
   d) Se encarga de las conexiones físicas de la computadora hacia la red. **Capa Física.**
   e) Lleva los datos libres de errores de la maquina origen a la destino. **Capa de Transporte.**
   f) Da formato a los datos presentados a la capa de aplicación. **Capa de Presentación.**
   g) Mantiene el enlace entre las dos computadoras que estén transmitiendo archivos. **Capa de Sesión.**

2. ¿Capa que transmite la información en Bloques?
   Capa de Enlace.

3. ¿Capa que divide el mensaje recibido en trozos o datagramas?
   Capa de Transporte.

Preguntas de repaso del apartado 2.1.2 al 2.3.11.

1. ¿Que es protocolo?

   Es un conjunto estricto de reglas o procedimientos que se requieren para iniciar y mantener las comunicaciones.

2. ¿Que es dirección IP?

   Una dirección IP es un número que identifica de manera lógica y jerárquica a una computadora dentro de una red que utilice el protocolo IP (Internet Protocol).

3. Dentro del modelo Cliente/Servidor defina:

   Proceso cliente:

   El cliente es un proceso que envía un mensaje a un proceso servidor, requiriéndole a éste la realización de una tarea (servicio).

   Proceso servidor:

   Es el proceso servidor que satisface las requisiciones del cliente realizando la tarea solicitada.

4. ¿Nombre del lenguaje de programación que se describe como optimizado para leer archivos de texto, extraer información y crear reportes de los mismos?

   Perl.

5. Defina el concepto de script.

   Es un lenguaje de programación que fue diseñado para ser ejecutado por medio de un intérprete.

6. ¿Mencione uno de los dos tipos de aplicaciones, la cuales, PHP es frecuentemente usado en su creación?

   Creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web.

Preguntas de repaso del apartado 2.4.1 al 2.4.5.5.

1. ¿Mencione 3 tipos de dominios de primer nivel?
   edu, org com, net, mil, gov, uucp.

2. ¿Que es URL?
   Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

3. ¿Protocolo orientado a objetos usado para distintas tareas como servidores de aplicaciones y sistemas de control de distribución de información?
   HTTP.

4. ¿Que es FTP?
   Es un protocolo de transferencia de archivos entre sistemas conectados a una red TCP basado en la arquitectura cliente-servidor.

5. ¿Cual es el nombre de la capa de modelo OSI que ofrece el servicio FTP?
   Capa de aplicación.

Preguntas de repaso del apartado 3.1 al 3.5.2.

1. ¿Que es un Sistema de Búsqueda?
   Es un conjunto de programas en Internet, cuya finalidad es la de proporcionar información y/o referencias de información sobre uno o varios tópicos a usuarios de la Web.

2. Mencione las partes por las que se compone un Sistema de Búsqueda:
   Robot (o crawler), Base de datos, Aplicación Cliente/Servidor.

3. Complemente cada idea agregando si corresponde a un significado de: Generador de consulta, Buscador, Generador de presentación.

   Encargado de acceder a la estructura interna de datos para satisfacer la petición de datos.

   Buscador.

   Construye la vista a ser incluida en el documento HTML que recibirá el usuario a través del servidor Web.

   Generador de Presentación.

   Recibe la consulta realizada en forma de una cadena de texto ingresada por el usuario.

   Generador de consultas.

4. Mencione los elementos de la Arquitectura Centralizada.

   Crawler, Indexador, Maquina de búsqueda, Interfaz.

5. Mencione los dos elementos que la arquitectura *Harvest* introduce para resolver los problemas de la arquitectura crawler:

   Gatherer que recopila páginas de determinados servidores Web. Brokers que proporcionan el mecanismo de indexación y la interfaz de consulta de los datos recuperados.

Preguntas de repaso del apartado 3.6 al 3.8.3.6.

1. ¿Defina de manera breve cada clasificación del Sistema de Búsqueda?

   Directorio

   Los directorios mantienen una lista de documentos de la Web clasificada por categorías las cuales forman una estructura en forma de árbol previamente diseñada.

   Motor de búsqueda.

   El sistema de búsqueda que utiliza robots y permite a los usuarios buscar información específica de la World Wide Web.

Buscador híbrido.

Posee las características de los directorios y de los motores de búsqueda.

Metabuscador

Son un tipo de buscador que aprovecha el trabajo de recopilación que realizan otros sistemas de búsqueda.

2. ¿Mencione los dos tipos de Metadatos que existen?

   Implícitos en las propiedades de los recursos y los definidos de manera explicita por el diseñador del documento.

3. Mencione los 4 elementos del Preprocesamiento de documentos:

   Análisis léxico del texto, Eliminación de stopwords, Stemming, Construcción de thesaurus.

4. Defina que es para cada uno el atributo Keyword y el atributo Description:

   Keyword son las palabras o frases que representan el contenido de un documento, página.

   La etiqueta Description proporciona un resumen del contenido de la página.

<u>Preguntas de repaso del apartado 3.8.4 al 3.10.5.2.</u>

1. Mencione las ventajas y desventajas de las bases de datos administrada por DBMS y bases de datos de bajo nivel.

   Una base de datos administrada por DBMS suele ser más costosa pero es independiente tanto del software como de los dispositivos de hardware.

   Una base de datos de bajo nivel resulta de ser de bajo costo, pero tiene una alta dependencia de dispositivos pues una migración de una base de datos a un hardware distinto para el que fue construido puede ser restrictiva.

2. Indique los componentes internos de un sistema de base de datos.

   Archivos virtuales y léxico.

3. Para que se utilizan los símbolos + y – como opciones de búsqueda.

   Se utilizan si se desea delimitar la búsqueda a resultados más exactos.

Preguntas de repaso del apartado 4.1 al 4.2.2.7.2.

1. Mencione los dos recorridos y su significado que utiliza el Algoritmo de Crawling.

   Breadth-first (Búsqueda en anchura), Depth-first (Búsqueda en profundidad).

2. ¿Enumere las extensiones con las cuales son mayormente definidos los documentos HTML?

   .html, .htm, .php, .asp

3. De una definición de discriminación Server Polling.

   Consiste en solicitar el encabezado del archivo apuntado a través del comando HEADER del HTTP para cada enlace, con la finalidad de investigar cuales son los datos del documento que el servidor tiene y que podrían ser de utilidad al momento de decidir si se discrimina o no un determinado enlace.

4. En limitar el ámbito al servidor, que tipo de enlaces se siguen para guiar al robot Web desde y hacia el mismo servidor.

   URL absoluto URL relativo.

Preguntas de repaso del apartado 4.3 al 4.4.2.

1. Clasifique los siguientes algoritmos de Ranking mencionados de acuerdo a sus características:

   Depende de la consulta y forman un grafo bipartito.

   HITS.

   Toma un conjunto de páginas Web y las ordena en base a cuan conectadas están.

   Web Query.

   Realiza un sistema de votación para medir la importancia de una página en función de todos los enlaces de la Web.

   PageRank.

2. Explique que es la ponderación por conteo comparativo.

   Es contar el número de veces que se localizan las palabras de la consulta en los documentos encontrados.

3. Explique en que consiste el Protocolo de Exclusión de Robots.

   Es un método que permite a los administradores de sitios Web indicar a los robots que visiten sus páginas qué partes de sus sitios no deberían ser visitados.

4. Complemente cada idea agregando si corresponde a un significado de: *User-agent, Disallow, valor asterisco, INDEX, [NO] INDEX, FOLLOW, [NO] FOLLOW.*

   Indica al robot que puede incluir el documento en su base de datos. **INDEX**.

   Define la *URL* que no debe ser inspeccionada por el robot

   Disallow.

   Indica que no debe ser analizada para búsqueda de enlaces.

   NO FOLLOW.