**Universidad de San Carlos de Guatemala**
**Facultad de Ingeniería**
**Escuela de Ingeniería en Ciencias y Sistemas**

# CONSTRUCCIÓN DE UN MOTOR DE BÚSQUEDA PARA LA OPTIMIZACIÓN DE CONSULTAS EN INTERNET

**Nestor Giovanni García Enríquez**
**Asesorado por Ing. Raymond Theodore Usher Vargas**

**Guatemala, septiembre de 2004**

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA


FACULTAD DE INGENIERÍA

# CONSTRUCCIÓN DE UN MOTOR DE BÚSQUEDA PARA LA OPTIMIZACIÓN DE CONSULTAS EN INTERNET

TRABAJO DE GRADUACIÓN

PRESENTADO A JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

**NESTOR GIOVANNI GARCÍA ENRÍQUEZ**
ASESORADO POR ING. RAYMOND THEODORE USHER VARGAS
AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, SEPTIEMBRE DE 2004

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA


## NÓMINA DE JUNTA DIRECTIVA

| | |
|---|---|
| DECANO: | Ing. Sydney Alexander Samuels Milson |
| VOCAL I: | Ing. Murphy Olympo Paiz Recinos |
| VOCAL II: | Lic. Amahán Sánchez Álvarez |
| VOCAL III: | Ing. Julio David Galicia Celada |
| VOCAL IV: | Br. Kenneth Issur Estrada Ruiz |
| VOCAL V: | Br. Elisa Yazminda Vides Leiva |
| SECRETARIO: | Ing. Pedro Antonio Aguilar Polanco |

## TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

| | |
|---|---|
| DECANO: | Ing. Sydney Alexander Samuels Milson |
| EXAMINADOR: | Inga. Elizabeth Domínguez Alvarado |
| EXAMINADOR: | Inga. Virginia Tala Ayerdi de Alemán |
| EXAMINADOR: | Ing. Marlon Antonio Pérez Turk |
| SECRETARIO: | Ing. Pedro Antonio Aguilar Polanco |

**HONORABLE TRIBUNAL EXAMINADOR**

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**CONSTRUCCIÓN DE UN MOTOR DE BÚSQUEDA PARA LA OPTIMIZACIÓN DE CONSULTAS EN INTERNET**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 25 de febrero de 2002.

Nestor Giovanni García Enríquez

## ÍNDICE GENERAL

**43**

## 2. CARACTERISTÍCAS DE LAS BASES DE DATOS EN INTERNET

# ÍNDICE DE ILUSTRACIONES

## FIGURAS

## GLOSARIO

**Alfanumérico** Aplicado a la clasificación fundada a la vez en el alfabeto y la numeración.

**Algoritmo** Pasos ordenados a seguir para la realización de una actividad. Ciencia del cálculo aritmético y algebraico.

**Análisis semántico** Análisis basado en el estudio del significado de los signos lingüísticos y de sus combinaciones desde un punto de vista sincrónico o diacrónico.

**ARPA** *(Advanced Research Projects Agency)* Es la Agencia de Proyectos de Investigación Avanzados creada por la Unión Soviética (1957) con la misión de desarrollar tecnologías que pudieran ser útiles a la milicia. La ARPA decidió que la red que necesitaba la DOD durante la Guerra Fría debía ser una red de paquete conmutador, que consistía en una subred y computadoras *hosts*.

***Backbones*** Es una estructura de red que proporciona un respaldo para las conexiones perdidas.

**Base de datos** Archivo en el que se almacena información de forma ordenada y se encuentra integrada por tablas, columnas y registros.

**Caché** Memoria extendida de la computadora a la cual se puede acceder en primer instancia.

**CGI** *(Common Gateway Interfase)* Lenguaje que permite la construcción de páginas en Internet.

**Conmutador** Dispositivo por medio del cual se transfiere información entre dos redes diferentes.

**DBMS** *(Data Base Manager System)*. Administrador de una base de datos o estructura en donde se encuentra el conjunto de todos los objetos posibles de una red.

**Enlace** Objeto que permite establecer un vínculo con otro objeto. Comúnmente llamado link.

***Excite*** Es uno de los motores de búsqueda modelo para la creación de otros motores de búsqueda.

**FTP** *(File Transfer Protocol)* Servicio de la *World Wide Web* que permite transferir archivos.

**Host** Terminal de red.

**HTML** *(Hyper Text Markup Language)* Lenguaje de marcación de hipertexto que permite a los usuarios producir páginas

| | |
|---|---|
| | de Web que incluyen texto, gráficos y apuntadores a otras páginas de Web. |
| ***Hypertext Links*** | Conexiones de texto resaltado. |
| **IDC** | *(Internet Data Base Connector)* Archivos que permiten establecer una conexión Internet-Base de Datos. Es un componente integral de un servidor de Internet. |
| **Lógica de *Bool*** | Se encarga de las proposiciones lógicas AND, OR, NOT. |
| **Meta *Tags*** | Son palabras que se utilizan para describir páginas de la *Web*. |
| **Motor de búsqueda** | Son “máquinas” que ayudan al usuario a definir una búsqueda en lenguaje casi natural y luego pulsar clic en el ratón para obtener grandes flujos de información sobre el tema solicitado. |
| **Navegador** | Programa que permite visualizar páginas HTML, CGI, ASP. Comúnmente llamado Visor (*Browser*). |
| **Página *Web*** | Documentos que integran en conjunto la *Web*. |
| **Proceso robot** | Método utilizado para rastrear información nueva en la red con el fin de actualizar el índice de la base de datos del motor de búsqueda |

| | |
|---|---|
| **Red** | Colección interconectada de computadoras autónomas, con el fin de compartir los recursos para que todos los programas, el equipo y especialmente los datos estén disponibles para cualquier usuario de la misma. |
| **Retroalimentación** | Proceso por medio del cual las entradas de información generan información de salida, la cual genera nuevamente información de entrada. |
| **Servidor** | Ordenador que proporciona servicios a sus clientes. |
| **Telnet** | Protocolo de comunicaciones |
| **URL** | *(Uniform Resource Locator)* Localizador uniforme de recursos |
| **Virtual** | Que tiene existencia aparente pero no real. |
| **WWW** | *(World Wide Web)* Red mundial de información. |

# RESUMEN

Al hablar de sistemas de información se puede pensar en diversos elementos que interactúan entre sí para proporcionar a los usuarios finales resultados que permitirán la formación en la educación, tomar decisiones en ámbitos empresariales, etc.

El presente proyecto presenta en su primera parte las condiciones sobre las cuales la información era accedida por los usuarios de la red internacional. Con lo anterior se pudieron examinar métodos que ayudarán a la obtención de información de modo más fácil y rápido.

En las partes finales se presentan las definiciones básicas para la construcción de un motor de búsqueda que permita optimizar la calidad de la información obtenida así como también el tiempo utilizado para recabar la información solicitada.

El proyecto consta de diversos diagramas que ejemplifican el flujo de las peticiones y las respuestas, así como también de los elementos necesarios en la construcción de la herramienta.

## OBJETIVOS

- *General*

Que el trabajo de investigación sirva para mostrar los pasos, procedimientos, algoritmos y herramientas que se utilizan para la creación de un motor de búsqueda que permita optimizar las consultas en las bases de datos en Internet, así como también proponer técnicas que ayuden a redactar consultas que permitan obtener la información esperada.

- *Específicos*

1. Investigar el funcionamiento de los motores de búsqueda para poder así, resumir los procedimientos y reglas que permitan crear un motor de búsqueda eficiente que optimice las peticiones de las consultas por parte de los usuarios.

2. Describir el funcionamiento de un motor de búsqueda, su estructura y consideraciones básicas para poder así definir las partes de desarrollo del nuevo motor de búsqueda que persigue optimizar el procesamiento de consultas.

3. Investigar nuevas herramientas *Web* que ayuden a la construcción de un motor de búsqueda, que a diferencia de los ya existentes muestre ciertas ventajas en relación al proceso de análisis, desarrollo y utilización.

# *INTRODUCCIÓN*

La cantidad de información a la cual tenemos acceso por medio de las redes computacionales permite definir la necesidad de que el acceso a ella sea de manera fácil y rápida, debido a que para realizar nuestras actividades tanto de trabajo como académicas necesitamos información clara y fidedigna. Se considera que la información que se pretende reunir para conformar el trabajo de investigación de tesis permitirá mostrar de forma clara el proceso de creación de un motor de búsqueda que permita optimizar las consultas que los usuarios realizan en Internet, cuando se habla de optimizar se refiere al tiempo de respuesta de una consulta así como calidad de la información obtenida.

El desarrollo de un proyecto como lo es un motor de búsqueda involucra de forma estrecha las bases de datos en Internet, éstas contienen características importantes que brindan un gran desempeño para este tipo de aplicaciones.

Este trabajo de investigación muestra información necesaria para la construcción de un motor de búsqueda, en donde también se presentan las distintas etapas que permiten la construcción de la aplicación tomando en cuenta las bases de datos en Internet, herramientas de desarrollo *Web* como XML, HTML y tecnologías como lo son los servidores de aplicaciones.

# 1. EL DESARROLLO DE INTERNET

## 1.1 Internet

A mediados de la década de 1960, en la cúspide de la Guerra Fría, el DoD necesitaba una red de comando y control que pudiera sobrevivir a una guerra nuclear. Las redes telefónicas tradicionales de circuito conmutado se consideraban muy vulnerables, debido a que con la pérdida de una línea o un conmutador, ciertamente terminaría toda conversación que se estuviera ejecutando, pudiendo incluso partir la red. Para resolver este problema el DoD acudió a su rama de investigación, ARPA o *Advanced Research Projects Agency* (más tarde DARPA y ahora ARPA de nuevo), es decir la Agencia (de Defensa, periódicamente) de Proyectos de Investigación Avanzados.

ARPA se creó en respuesta al lanzamiento del *Sputnik* de la Unión Soviética en 1957 y tuvo la misión de desarrollar tecnologías que pudieran ser útiles a la milicia. ARPA nunca tuvo científicos ni laboratorios, de hecho, no tenía más que una oficina y un presupuesto pequeño (para los estándares del pentágono). Sin embargo, cumplió con su misión al ofrecer financiamiento y contratos a universidades y compañías cuyas ideas le parecían prometedoras.

Después de discusiones con varios expertos, la ARPA decidió que la red que necesitaba el DoD debía ser una red de paquete conmutado, que consistía en una subred y computadoras terminales.

La cantidad de redes, máquinas y usuarios conectados a la ARPANET creció con rapidez a partir del 1 de enero de 1983, cuando se interconectaron la NSFNET (*National Science Foundation*) y la ARPANET, el crecimiento se hizo exponencial; se unieron varias redes regionales y se hicieron conexiones con redes en Canadá, Europa y el Pacífico. En algún momento durante la década de los años ochenta, la gente empezó a ver la aglomeración de redes como una interred, y más tarde como la Internet, no habiendo reconocido los derechos de invención por tan importante creación.

El crecimiento continuó en forma exponencial y para 1990 la Internet había crecido a 3,000 redes y 200,000 computadoras. En 1992 se adhirió la terminal número un millón. Para 1995, había múltiples redes principales, cientos de redes de nivel medio (regionales), decenas de miles de redes de área local, millones de *hosts* y decenas de millones de usuarios. El tamaño se duplica aproximadamente cada año.

Realmente, estar en Internet significa que una máquina puede operar con ciertos protocolos que son capaces de recibir y enviar, de y para todas las demás maquinas que se encuentra conectadas.

Con el crecimiento exponencial, la antigua manera informal de operar la Internet ya no funciona. En enero de 1992 se integró la Sociedad Internet para promover el uso de Internet y quizá en algún momento hacerse cargo de su gestión.

Tradicionalmente, Internet ha tenido cuatro aplicaciones principales, que son las siguientes:

### 1.1.1 Correo electrónico

La capacidad de redactar, enviar y recibir correo electrónico ha estado disponible desde los primeros días de la ARPANET y es enormemente popular. Mucha gente recibe docenas de mensajes al día y lo considera su forma primaria de interactuar con el mundo externo, dejando muy atrás al teléfono y al correo lento. En estos días, los programas de correo electrónico están disponibles virtualmente en todo tipo de computadoras.

### 1.1.2 Noticias

Los grupos son foros especializados en los que usuarios con un interés común pueden intercambiar mensajes. Existen miles de grupos de noticias, con temas técnicos y no técnicos, lo que incluye computadoras, ciencia, recreación y política. Cada grupo de noticias tiene su propia etiqueta, estilo y costumbres.

### 1.1.3 Sesión remota

Mediante el uso de Telnet, los usuarios en cualquier lugar de la Internet pueden ingresar en cualquier otra máquina en la que tenga una cuenta autorizada.

### 1.1.4 Transferencia de archivos

Con el programa FTP, es posible copiar archivos de una máquina en Internet a otra. De esta manera está disponible una vasta cantidad de artículos, bases de datos, así como otro tipo de información.

Hasta casi fines de la década de los años noventa, la Internet se poblaba en gran medida de investigadores académicos del gobierno y de la industria. Una aplicación nueva, la WWW (*World Wide Web*, red mundial) cambió todo eso y atrajo millones de nuevos usuarios no académicos a la red. Esta aplicación, inventada por el físico del CERN, Tim Berners-Lee, no cambió ninguno de los recursos subyacentes pero los hizo más fáciles de usar. Junto con el visor de *Mosaic*, escrito en el Centro Nacional para Aplicaciones de Supercomputadoras, la WWW hizo posible que una localidad estableciera varias páginas de información conteniendo textos, dibujos, sonido y hasta vídeo con enlaces intercalados a otras páginas. Al accionar el ratón en el enlace, el usuario se ve transportado de inmediato a sitio al que apunta ese enlace. Por ejemplo, muchas compañías tienen una página local con entradas que apuntan a otras páginas que ofrecen información de productos, listas de precios, ventas, apoyo técnico, comunicación con los empleados, información para accionistas, etc.

En un tiempo muy corto, han aparecido muchos otros tipos de páginas, incluso mapas, tablas del mercado de valores, catálogos de tarjetas de bibliotecas, programas de radio grabados y hasta una página que apunta al texto completo de muchos libros cuyos derechos de autor han expirado.

En el primer año en que salió a la luz *Mosaic*, la cantidad de servidores WWW creció de 100 a 7,000. Sin duda, el crecimiento en los años por venir continuará siendo enorme y es probable que sea la fuerza que impulse la tecnología y el uso de Internet hacia el próximo milenio.

En el siglo XX, la tecnología clave ha sido la obtención, procesamiento y distribución de la información. Entre otros avances, hemos visto la instalación de redes telefónicas mundiales, la invención del radio y televisión, el nacimiento y crecimiento sin precedentes de la industria de las computadoras y el lanzamiento de satélites de comunicación.

Al iniciar la década de los años noventa, las redes de computadoras comenzaron a prestar servicios a particulares en su hogar. Estos servicios y la motivación para usarlos son muy diferentes y dependen de la situación de los usuarios.

El acceso a la información remota vendrá en muchas formas. Un área en la cual ya está sucediendo es el acceso a las instituciones financieras. Mucha gente paga sus facturas, administra sus cuentas bancarias y maneja sus inversiones en forma electrónica. Las compras desde el hogar se están haciendo populares, con la facilidad de inspeccionar los catálogos en línea de miles de compañías.

Algunos de estos catálogos pronto ofrecerán un vídeo instantáneo de cualquier producto que se pueda ver con sólo hacer clic en el nombre del producto. Los periódicos se publicarán en línea y serán personalizados.

Podremos decirle al periódico que queremos saber todo lo referente acerca de los políticos corruptos, los grandes incendios, los escándalos de celebridades y las epidemias, pero nada de fútbol, en la noche mientras usted duerme, el periódico se bajará al disco de su computadora o se imprimirá en su impresora. El siguiente paso más allá de los periódicos, es la biblioteca digital en línea. Otra aplicación en esta categoría es el acceso a sistemas de información como la actual red mundial (*World Wide Web*), la cual contiene información sobre arte, negocios, cocina, gobierno, salud, historia aficiones, recreación, ciencia, deportes, viajes y muchos otros temas, demasiado numerosos para mencionarlos aquí. Todas las aplicaciones antes mencionadas implican la interacción entre una persona y una base de datos remota. La segunda categoría extensa de redes que se usa implica la interacción persona a persona, básicamente la respuesta del siglo XXI al teléfono del siglo XIX. Millones de personas utilizan ya el correo electrónico y pronto contendrá en forma rutinaria audio y video además de texto. El correo electrónico de tiempo real permitirá a los usuarios remotos comunicarse sin retraso, posiblemente, viéndose y escuchándose. Esta tecnología hace posible realizar reuniones virtuales, llamadas videoconferencias, entre gente muy alejada. Se dice que el transporte y la comunicación están en competencia, y cualquiera que gane hará al otro obsoleto. La introducción ampliamente difundida de redes significará nuevos problemas sociales, éticos y políticos. Sólo mencionaremos en forma breve algunos de ellos, debido a que un estudio minucioso requiere una investigación completa. Uno de los grandes problemas surge cuando los grupos de noticias tratan de temas que a la gente en verdad le importan, como la política, la religión o el sexo. Las opiniones expresadas en tales grupos pueden ser profundamente ofensivas para algunas personas, además, los mensajes no necesariamente están limitados al texto, fotografías a color de alta definición e incluso pequeños videos pueden transmitirse ahora con facilidad por la Internet.

## 1.2 La necesidad de encontrar y clasificar información

En la WWW nos provee una serie de servicios a los cuales podemos acceder a través de una red, toda estos servicios han ido evolucionando y por consiguiente han hecho más fácil la tarea de obtención de información. Con el crecimiento del invento de la red mundial, muchos usuarios fueron creando sus propios segmentos o apartados con el fin de recabar una serie de datos que fueran de utilidad para muchos otros, esos apartados y/o segmentos los conocemos ahora como: páginas, sitos *Web*, entre otros, el crecimiento exponencial de esos sitios causó una lluvia de información, la cual es muy difícil de controlar, debido a que proviene de todos los estratos sociales, en donde los temas de conversación y/o estudio son variados y en muchos casos desconocidos. La capacidad que nos da la red para comunicarnos con otras personas a través de una máquina así como la capacidad de acceder a recursos que pueden estar geográficamente muy lejanos, creó la necesidad de clasificar toda la información que andaba en la red sin un lugar o segmento propio. Esta clasificación tiene varias razones y se puede enumerar sin tomar en cuenta la importancia de cada área, debido a que el flujo de la misma es tan grande que no se sabe a ciencia cierta cual tiene mayor importancia, dentro de esta clasificación se puede mencionar: la información de tipo académico-científico, política, económica, religiosa, deportiva, entretenimiento, así como muchas otras clasificaciones que se encontraban dispersas por la red, sin embargo, se debería de tratar de reunir la información por temas afines para facilitar su búsqueda e interpretación. La cantidad de temas que se pueden encontrar en Internet es extremadamente extensa y si alguien necesitaba encontrar información sobre: “Las plagas que afectaron la región sur de Europa durante la segunda guerra mundial“ resultaba tan complicado, como realizar la búsqueda de ese mismo tema en una biblioteca con unos 500 libros con información sobre la segunda guerra mundial.

La necesidad de obtener información obligó a los programadores a buscar la forma de clasificar la información, de tal forma que fuera fácil de acceder, en poco tiempo y exacta. La gran cantidad de información que existe actualmente en la red sería imposible de consultar si no existieran los motores de búsqueda, los cuales son “máquinas” que ayudan al usuario a definir una sentencia de selección en lenguaje casi natural y luego pulsar *clic* en el ratón para poder obtener grandes flujos de información sobre el tema solicitado.

La concentración de información en la red creció de manera un poco desordenada, si lo queremos llamar así; todas las personas que contribuían a la creación de la biblioteca más grande del mundo se esmeraban en introducir información, de hecho lo continúan haciendo esperando que ésta pueda ser consultada con algún programa que facilite su visualización, así como su utilización, este tipo de programas se les conoce como “Buscadores”, “Motores de búsqueda” o “*Search Engines*”, los cuales recopilan la información basados en la solicitud del usuario; estos conceptos los detallaremos a continuación.

## 1.3 ¿Qué son los motores de búsqueda?

### 1.3.1 Motor de búsqueda

Una de las partes principales de un portal o página *Web* es el metabuscador o motor de búsqueda, que no son más que sistemas que utilizan varias bases de datos donde efectúan consultas de forma simultánea.

También, se puede decir que un motor de búsqueda o metabuscador es un sistema automático de recuperación que utiliza para sus consultas bases de datos de distintos buscadores.

Otra definición de motor de búsqueda, es que es una base de datos escudriñable formada por archivos de Internet. Un programa de computación recolecta estos archivos sin utilizar criterios específicos de selección, y con estos archivos forma un índice de búsqueda que otro programa consulta cuando el usuario lo solicita.

### 1.3.2 *World Wide Web*

Comparable a una tela de araña, con conexiones directas de un punto a otro, es una colección de páginas del *Web* almacenadas en computadoras a lo largo y ancho del mundo, se puede mostrar texto, gráficos, sonido y vídeo. Contiene "*hypertext links*" (conexiones de texto resaltado), las cuales son vínculos electrónicos a otras páginas del *Web* (locales o muy lejanas).

### 1.3.3 *Web Site*

Se puede decir que es la computadora donde está almacenada la página *Web*. La computadora debe estar conectada a Internet para que la página esté disponible globalmente.

### 1.3.4 *Web Server*

Una definición, muy fácil de entender es: software que permite realizar la conexión entre un "*Web Site*" y un usuario de computadora.

### 1.3.5 Página de *Web*

Es un documento electrónico, escrito en HTML (*hypertext mark-up language* = lenguaje de texto resaltado y con señales), puede incluir sonido, vídeo, gráficos, animación, así como *links* (enlaces) a otros documentos en la misma página o en cualquier otro lugar en la red (WWW).

### 1.3.6 *Home Page*

Es la primera página que usted ve cuando se conecta a una página *Web*. Frecuentemente sirve de introducción a la información contenida en un "*Web Site*".

### 1.3.7 La interfase

Comúnmente la interfase de un motor de búsqueda es un programa que está desarrollado en un lenguaje que permita utilizar todas las herramientas de Internet, la extensión de una interfaz de un motor de búsqueda es HTML.

Sin embargo, sólo se pueden considerar motor de búsqueda a los sistemas automáticos de recuperación de información, que almacenan información sobre páginas *Web* en una base de datos, la cual se puede interrogar desde un simple formulario.

## 1.4 Características de los motores de búsqueda

Como ya se mencionó, todos los motores de búsqueda tienen como objetivo devolver la mayor cantidad de información en respuesta a una petición realizada por el usuario. Los motores de búsqueda tienen características muy particulares, las cuales le permiten funcionar de determinada forma o modo, dependiendo de la calidad de información que se le introduzca; un motor de búsqueda puede ser capaz de devolver resultados óptimos en porcentajes altos dependiendo de su estructura de funcionamiento. Los motores de búsqueda para realizar sus procedimientos se apoya en gran parte en la lógica *booleana* o lógica de *bool* tácita (sin necesidad de utilizar operadores complejos), esto ayuda a los algoritmos de búsqueda a verificar los campos de la base de datos y devolver resultados precisos muy rápidamente.

### 1.4.1 Reconocimiento de lenguaje natural

El lenguaje natural, es aquel que utilizamos todos para poder comunicarnos con los demás, aunque existen marcadas diferencias entre algunas palabras que tienden a ser confusas, el motor de búsqueda puede interpretar palabras y hasta frases escritas en lenguaje natural, para poder interpretar el lenguaje natural el motor de búsqueda se vale de una gramática y de un pequeño pre-compilador que le permite analizar palabra por palabra y llevar la secuencia de la frase, este tipo de gramática debe de ser de tipo LL(5), la cual, es capaz de identificar expresiones de lenguaje natural.

### 1.4.2 Palabras clave

Las palabras clave dentro de un motor de búsqueda permiten que la redacción de la consulta sea más precisa y que el motor de búsqueda utilice sus algoritmos de búsqueda de manera eficiente, todas las palabras clave están definidas por el propio motor de búsqueda. Las palabras clave, son aquellas palabras que usamos para describir los conceptos o ideas que buscamos. No son sólo las palabras habituales, sino, también cualquier secuencia de caracteres que sirva para localizar nuestro objeto, aunque no sean pronunciables.

Estas palabras están normalmente separadas por espacios en blanco y no se debe incluir signos diferentes a los alfanuméricos, a no ser que representen alguna función especial.

## 1.5 Cómo trabaja un motor de búsqueda

Existen diversas modalidades con las que los motores de búsqueda realizan su trabajo; el objetivo principal del motor de búsqueda es devolver la información más precisa que se pudo haber requerido mediante una secuencia de caracteres que forman la propia consulta.

Después de introducirle una petición de búsqueda, el motor de búsqueda la coteja con la base de datos y devuelve una lista ordenada de las coincidencias. La lista está ordenada según la relevancia de la consulta colocándose primero las más coincidentes.

Al conectar con algún motor de búsqueda nos encontraremos con una página que contiene un formulario para definir nuestra búsqueda y las opciones

de la misma; tras rellenarla, enviarla y esperar unos segundos, el buscador nos devolverá una lista de lugares donde figura el resultado de nuestra búsqueda.

Así, pues, tendremos dos áreas según el propósito:

- Formular la búsqueda y enviarla.
- Lista de resultados, ordenados según su semejanza con las palabras claves introducidas.

Si no conseguimos obtener solo los resultados deseados debemos volver al inicio, pero modificando la estrategia de búsqueda según la observación de los resultados.

### 1.5.1 Desarrollo de la relevancia

Algunos motores de búsquedas ahora funcionan en conexión con otros para que los primeros resultados provengan de su propio índice, y los que siguen provengan de los resultados de otro motor en donde se ejecutó la consulta con anterioridad.

Algunos otros motores aplican reconocimiento automático de las frases, tratando los términos de búsqueda como frases, en vez de combinarlos con los operadores *booleanos*. Por ejemplo, algunos buscadores como Altavista y *Excite* proveen frases y palabras adicionales posibles con los resultados, para que se pueda formar un criterio preciso.

### 1.5.2 Evitar el SPAM

Para que los algoritmos tomen a las páginas como muy relevantes, diseñadores agregan términos a sus páginas, en la forma de "meta *tags*". Meta *tags* son palabras que se usan para describir páginas de la *Web*. Algunos motores de búsqueda buscan en estas palabras. Idealmente, los diseñadores usan palabras que precisamente describen sus páginas. Pero la verdad es que los diseñadores llenan los meta *tags* con palabras comunes que aparecen en muchas búsquedas, para que sus páginas suban al principio de los resultados.

En respuesta a esta técnica, los programadores de los motores de búsqueda han cambiado los algoritmos para no permitir el abuso. De hecho, el motor *Excite* no revisa los meta *tags* durante la búsqueda. Pero los meta *tags* pueden ser muy útiles aún cuando son usados de manera inadecuada.

Otro abuso nuevo es cuando los diseñadores agregan palabras clave a sus páginas usando el mismo color del background en letras pequeñas. Son invisibles al ojo humano, pero los motores de búsqueda las leen como parte del texto, es por eso que a veces los resultados parecieran no tener nada que ver con la búsqueda.

### 1.5.3 Lenguaje de consulta

El motor de búsqueda de texto permite que se formen consultas a partir de palabras clave como AND, OR y NOT. Es recomendable escribir las palabras con acentos y en minúsculas. Por ejemplo:

- Información or recuperación

Busca documentos que contenga 'información' o 'recuperación'

- Información and recuperación

Busca documentos que contengan 'información' y 'recuperación'

- Información not recuperación

Busca documentos que contengan 'información' pero no 'recuperación'

- ( información not recuperación ) and WAIS

Busca documentos que contengan 'WAIS' e 'información' pero no 'recuperación'.

- *Web**

Busca documentos que contengan palabras que empiecen por '*Web*'

### 1.5.4 Tipos de búsqueda

Es necesario que un buscador posea varios criterios para ejecutar su búsqueda, debido a que como el lenguaje natural es extenso y rico en palabras, en determinadas situaciones, un criterio de búsqueda puede fallar, de tal forma que en ese momento los otros criterios definidos puedan interpretar la petición.

Se puede mencionar los criterios basados en:

#### 1.5.4.1 Búsqueda por conceptos o contexto

Criterio en el cual los motores de búsqueda utilizan el concepto de que las palabras que se introdujeron forman la consulta, pero con la variante de que pueden utilizar sinónimos para formular la petición, este criterio a menudo tiene muchos errores al momento de presentar los resultados finales.

#### 1.5.4.2 Palabras claves

Este criterio está basado en las palabras claves que están definidas por el motor de búsqueda y que a partir de la redacción de la consulta escrita por el usuario, comienza a realizar sus comparaciones para determinar qué camino puede tomar para llegar a presentar la información más congruente, este método o criterio es muy utilizado.

#### 1.5.4.3 Retroalimentación

La retroalimentación se maneja basada en a un historial o bitácora de algún ejemplo o páginas encontradas previamente. El criterio de retroalimentación tiene mucha aceptación dado que los resultados basados en este criterio son completos y con un tiempo de respuesta muy corto.

#### 1.5.4.4 Ocurrencias

Cuando el número de ocurrencias es muy elevado, quiere decir que el resultado de la consulta con este criterio tiende a ser aceptado.

La mayoría de motores buscan ocurrencias que contengan, como mínimo una de las claves:

- "Y" lógico
- "NO" no lógico
- Dos claves próximas
- Algunas claves como una sola cadena de caracteres "Frases"

Considerar las palabras claves como:

- Palabras relacionadas y el comodín *
- Subcadenas de caracteres terminal o inicial
- "Truncar" Palabras enteras
- Limitar el número de ocurrencias
- Seleccionar el grado de detalle en el listado de resultados

- Compacta: es decir palabras tomadas como abreviaturas
- Detallada: aquellas en las cuales se define con un amplio criterio el concepto de la búsqueda

### 1.5.5 Empezar una búsqueda

Para empezar una búsqueda, simplemente se teclea la palabra o palabras que se desea buscar en el campo destinado a tal efecto, y se hace *clic* sobre el botón Buscar o el botón que indique que la búsqueda inicia. En algunos programas de navegación, puede que no aparezca un campo donde teclear, en cuyo caso hay que atenerse a las instrucciones propias del programa.

Si se introduce más de un término, el servidor buscará todos los documentos que contengan al menos una de las dos palabras. Por ejemplo, si se teclea:

- Conferencia or debate

El resultado de la búsqueda será un listado de todos los documentos que contienen o bien el término conferencia, o el término debate, o ambos.

Para hacer búsquedas más ajustadas, es muy recomendable el uso de estrategias de búsqueda avanzada

En el ejemplo anterior, si lo que interesa es buscar documentos que contengan ambos términos (conferencia y debate), será necesario utilizar el operador "*and*".

No hay que preocuparse demasiado por escribir los términos o los operadores en mayúsculas o minúsculas, ya que la base de datos está configurada de manera que esto no sea relevante.

### 1.5.6 Modo de resultados

#### 1.5.6.1 Normal

Esta forma de presentar los resultados de una búsqueda la realiza sin tomar en cuenta el detalle del resultado, únicamente toma en cuenta las características que posea el título o encabezado del resultado.

#### 1.5.6.2 Verificación o comprobación

Para poder presentar los resultados, este modo se basa exclusivamente en la verificación o comparación de la cantidad de repeticiones que contenga el detalle del resultado con las palabras claves.

### 1.5.7 Deficiencias de los motores de búsqueda

Los motores de búsqueda en su mayoría no manejan información imprecisa. Dará mayor importancia a un documento que tiene mayor cantidad de ocurrencias y manejará mal la información imprecisa pero relevante para el usuario. Es mejor usar varias herramientas de búsqueda para manejar grandes volúmenes de datos y hacer nosotros el análisis semántico y el proceso de filtrado definitivo.

## 1.6 Componentes básicos de un motor de búsqueda

### 1.6.1 Componentes

Los dos componentes de un motor de búsqueda son:

#### 1.6.1.1 Robot

Un programa especial "atraviesa" la red saltando de vínculo en vínculo. Este programa, puede recibir diferentes nombres: *spider* (araña), robot, *worm* (gusano), *wanderer* (vagabundo) o *crawler* (reptil). El programa lee el contenido de los archivos para incorporarlo a la base de datos, y busca nuevos vínculos que visitar para obtener más archivos. El mismo programa continuará visitando periódicamente los mismos archivos para detectar si se los han actualizado o eliminado.

#### 1.6.1.2 Índice

Base de datos que contiene una copia completa o parcial de los documentos reunidos por el robot, la información derivada de ellos es obtenida a través de programas especiales que facilita la labor de los mecanismos de búsqueda, dándole significado a los resultados.

### 1.6.2 Mecanismo de búsqueda

Se refiere a un software que permite al usuario indagar el índice a través de una página *Web* y que devuelve resultados significativos a la búsqueda, habitualmente ordenados según la relevancia asignada.

#### 1.6.2.1 Buscador sin robot

Las direcciones añadidas se ubican en secciones dentro de una estructura de árbol, debiéndose indicar las categorías bajo las que se desea que queden ubicadas en el proceso de alta.

Los contenidos, en muchos casos, son analizados y procesados por personas que visitarán la dirección añadida, determinando si éste cumple con los requisitos necesarios para ser dado de alta y si los datos introducidos son correctos. Es imprescindible dar de alta manual el sitio *Web* para figurar dentro de la base de datos de los directorios.

Para lograr una buena posición normalmente no es necesario el uso de palabras clave, usualmente, es suficiente que el nombre del sitio comience con la letra más cercana a la A en el alfabeto o con alguno de los primeros caracteres de la codificación.

- Hay menos resultados totales debido al menor contenido de sus bases de datos. Estos son mejores, más fiables y presentan menos enlaces erróneos o poco efectivos.

- Los resultados aparecen por orden alfabético en la mayoría de los casos y ordenados por categorías temáticas.

- Las altas demoran entre 2 a 4 días. Salvo excepciones en que por las características del directorio, llegan a las 2 a 8 semanas en producirse.

- Son ideales para hallar páginas sencillas con temas comunes, pues las búsquedas se facilitan.

#### 1.6.2.2 Buscador con robot

Los contenidos son indizados por medio de un robot, araña o gusano. No es imprescindible dar el alta a un sitio *Web* para figurar en él. Es aconsejable pero no imprescindible, pues la mayoría de los robots buscan por la *Web* por ellos mismos, indizando todo lo que encuentran a su paso. Aún así, el contenido de sus bases de datos no suele superar el 10% del total de la red.

Para lograr una buena posición es necesario el correcto uso de palabras clave y etiquetas dentro del código HTML, los resultados aparecen por orden de popularidad, dependiendo de las características del robot y puede tomar las palabras clave del título, descripción o contenido.

Las altas pueden llegar a demorar varios meses en algunos casos y son ideales para hallar temas intrincados o prohibidos en los directorios de búsqueda o para encontrar temas poco comunes.

### 1.6.3 Características y convenciones típicas

La mayoría de los motores de búsqueda comparten las mismas prestaciones básicas y poseen una interfase de usuario similar. Sin embargo, varían de manera significativa al momento de incluir características avanzadas.

En este párrafo nos centraremos en las características comunes que presenta la interfase de usuario de los principales motores de búsqueda. Éstas poseen:

#### 1.6.3.1 El formulario de entrada de la búsqueda

Es un espacio que el usuario activa mediante un *clic* del ratón y donde se escriben los términos o frases que definen el criterio de búsqueda.

#### 1.6.3.2 El botón de búsqueda

En la pantalla de su navegador, un botón rotulado *SEARCH* (busque), *SUBMIT* (preséntelo), *GO* (vaya) o *GO TO GET IT* (vaya y consígalo) debe pulsarse para que éste envíe los criterios al motor de búsqueda.

#### 1.6.3.3 Listado de sitios que se ajustan al criterio

Los motores de búsqueda presentan como primera respuesta de 10 a 100 documentos que satisfacen más estrechamente el criterio de búsqueda, ordenados según su relevancia con el mismo. Por defecto, presentan el título del documento y unas pocas líneas que ilustran sobre su contenido.

#### 1.6.3.4 Tipo de búsqueda

Se ofrecen diversos mecanismos para el refinamiento de la búsqueda, consistentes en formularios y casillas de verificación, algunas veces como parte de la búsqueda básica, pero por lo general, como parte del servicio de búsqueda avanzada.

#### 1.6.3.5 Sitios revisados

La mayoría de los motores de búsqueda presentan como alternativa a la lista de sitios, uno o varios directorios de sitios *Web* a los que el personal del servicio de búsqueda ha pasado revista y clasificado (de manera similar a los directorios temáticos).

#### 1.6.3.6 Ayuda en línea

Como suele suceder con los programas de computación, es necesaria una ayuda útil y fácil de usar, aunque no siempre usada con la suficiente asiduidad.

### 1.6.4 Servicios analizados

A continuación podrá acceder a una explicación detallada sobre diferentes motores de búsqueda. Posteriormente encontrará definidas las características, fortalezas y debilidades de cada uno de los servicios.

- *InfoSeek*
- Altavista
- *HotBot*
- *Excite*
- *Northern Light*
- *Google*

#### 1.6.4.1 Índices o directorios

Los índices o directorios como su propio nombre indica, se encargan de indexar las direcciones *Web* en categorías, similar a las páginas amarillas. La principal característica de éstos, es que las direcciones que encontremos en cada categoría han sido asignadas a ésta por una persona, no por un programa. Para localizar algo en un índice podemos navegar las diferentes secciones hasta que encontremos lo que buscamos, o podemos realizar una búsqueda. Cuando buscamos por una palabra o frase, un índice utiliza principalmente cuatro criterios para decidir qué resultados va a mostrarnos:

- **Título**

  Ésta es la palabra clave en el título del sitio *Web*.

- **URL**

  Ésta es la palabra clave en el URL del sitio *Web*.

- **Descripción**

  Es la palabra clave, que forma parte de la descripción del sitio *Web* (estas descripciones suelen ser redactadas por el propio editor del índice).

- ***Clics***

Algunos directorios y cada vez más máquinas de búsqueda consideran relevante el número de veces que sus clientes dan clic en su enlace a una *Web*. Básicamente, este mecanismo funciona de la siguiente manera: si el enlace del índice a nuestra página *Web* recibe más peticiones (*clics*), que la que se encuentra inmediatamente por encima de nuestro enlace, automáticamente, se realiza un intercambio de posiciones, quedando por arriba el enlace que más *clics* haya obtenido.

Los principales retos con los que nos encontraremos trabajando con directorios serán:

- Que nuestra *Web* sea aceptada.
- Que nos sitúen en la categoría/s correctas.

Muchos *Webmasters* dan por hecho que su página será inmediatamente aceptada por los índices y cuatro meses después de su primer intento están preguntándose qué pudo pasar. Es importante seguir las reglas de estos buscadores a “rajatabla”, debido a que la *Web* será revisada por un ser humano. Si intentamos aparecer en una sección en la que no pertenecemos o damos una descripción confusa podemos ser rechazados, o lo que puede ser peor aún, que el editor nos sitúe en un lugar equivocado y escriba su propia descripción del sitio.

## 1.7 Ambiente bajo el cual trabaja un motor de búsqueda

Muy especialmente, podemos definir el ambiente de un motor de búsqueda como la ciudad o el mundo en el cual se encuentra, es decir, que tiene límites, alcances y está regido por normas, las cuales las define la persona que crea estructura lógica del motor de búsqueda; dentro de los alcances, podemos mencionar el tipo de información a la cual se le dará prioridad y por mencionar un límite podemos mencionar el idioma en el cual se realizará la búsqueda, por ejemplo:

La programación de los motores de búsqueda se hace por medio de lenguajes especiales que tienen las características de conectarse a una red de información como Internet, así como la capacidad de manejar distintas bases de datos, entre otros. Algunos de los lenguajes más comunes para la programación de páginas y/o motores de búsqueda son: CGI, ASP, JAVA, entre otros.

### 1.7.1 El sistema operativo

Sin duda alguna el factor más importante para un buen desempeño en el rendimiento, es el sistema operativo, éste determina la posible potencia de la base de datos. Los sistemas operativos han desarrollado bases sólidas para este tipo de aplicaciones. Existen en el mercado sistemas UNIX, Windows, OS/2 con grandes capacidades y con un alto grado de estabilidad. En los últimos años, se han desarrollado sistemas operativos experimentales con herramientas de comunicación, un ejemplo de ellos es Linux que se ha convertido en el sistema operativo para PC con capacidades para Internet, debido a que configura todos los servicios en una computadora sin exigir gran requerimiento de hardware.

Las máquinas UNIX que corren SunOS, Solaris, HP-UX o cualquier otra variedad de UNIX son las plataformas preferidas para bases de datos de gran tamaño. No hay que olvidar la aceptación que Windows NT ha tenido en el mercado de los sistemas operativos de red y la madurez que ha alcanzando en los últimos *Resource Kit*.

### 1.7.2 Las metaetiquetas

Las metaetiquetas (*metatags*) son parte del código HTML donde se añaden valores especificando información sobre el documento.

En el caso que nos ocupa, la descripción y las palabras (o frases) clave (*keywords*) de su página y que serán utilizadas por los robots para indexar sus páginas.

Las metaetiquetas se ubican en la cabecera del código HTML entre las etiquetas <HEAD> y </HEAD>.

```
<HEAD>
<META name="description" content="descripción de la página">
<META name="keywords" content="palabras clave separadas por comas">
</HEAD>
```

## 1.8 Relación con las bases de datos

En este punto vamos a mencionar las características principales de un motor de búsqueda con una base de datos, para ello, se hará énfasis en las conexiones de las interfaces, las cuales involucran variables, tipos de datos, funciones, DBMS's entre otros.

### 1.8.1 Interfase dbCGI

Es una interfase muy utilizada entre bases de datos y la World Wide *Web*, desarrollada por CorVu Pty Ltd (Australia). Esta interfaz puede interactuar con diversos sistemas abiertos de bases de datos, como Informix, Ingres, Oracle, Progress y Sybase, a través del *Web*.

La herramienta dbCGI posibilita crear dinámicamente documentos personalizados en formato, tales como formularios, tablas y reportes complejos. Además, dbCGI está diseñado para soportar codificación interlineal de sentencias para interactuar con las bases de datos. Cuando un cliente desde el navegador hace una requisición a documentos dbCGI, el programa ejecuta las requisiciones a las bases de datos e interpola los resultados dentro del respectivo archivo para devolverlos al *Web* navegador.

### 1.8.2 Características y funciones

DbCGI puede utilizarse con cualquier bases de datos que soporte acceso dinámico con el lenguaje C.

Las funciones principales de esta interfase son:

- Flexibilidad en el uso del lenguaje para diseñar formularios y páginas *Web* que recolecten las requisiciones y desplieguen los resultados de las consultas de bases de datos.

- Independencia de la base de datos con los módulos ejecutables, específicos para cada uno, que contienen alrededor de 250 líneas de código.

- Posibilita el uso de variables del servidor y de los datos recolectados en un formulario, dentro de las sentencias.

- Funciones de seguridad que permiten la verificación de los datos remitidos por el usuario a través del servidor *Web*.

- Se pueden lograr estilos profesionales y detallados del reporte de los resultados obtenidos de la interacción con la base de datos.

- Soporta datos BLOB, tales como imágenes, sonido y vídeo.

- Sintaxis similar al formato estándar.

Estas funciones de dbCGI permiten crear fácilmente páginas *Web* donde el motor de búsqueda pueda acceder y/o actualizar bases de datos.

### 1.8.3 Funcionamiento de un servidor *Web*

En la siguiente figura se muestra la arquitectura de un servidor *Web* con la interfase dbCGI. Cuando un cliente *browser* hace una requisición de algo, que es interpretado como un archivo dbCGI, el servidor por medio del motor de búsqueda pasa la requisición al programa dbCGI. Éste lee el archivo especificado, luego procesa las requisiciones a la base de datos contenidas en dicho archivo. Finalmente compone los resultados en formato para que el servidor los remita al cliente *browser*.

## 1.9 Interpretación de información

### 1.9.1 Interpretación de los resultados

Al efectuar una búsqueda se obtiene un listado de los documentos que satisfacen la petición, o, en su caso, un mensaje informando de que no existe ningún documento que coincida con la petición realizada.

Los documentos aparecen listados en forma decreciente según su índice de relevancia, que consiste en un cálculo que realiza el buscador y que depende de los siguientes factores:

- Número de veces que aparece el término buscado en cada documento.
- Tamaño del documento.
- Número total de documentos que cumplen con la petición realizada.

El servidor devolverá un máximo de  entradas en la lista de documentos encontrados, y el resto serán ignorados. Si el resultado de una búsqueda contiene un porcentaje elevado en cada una de las bases de datos en las cuales realice la búsqueda, muy probablemente, existen más documentos que satisfacen el criterio utilizado, por lo que sería recomendable intentar de nuevo la misma búsqueda haciendo uso de alguna de las estrategias de búsqueda avanzada.

### 1.9.2 Estrategias de búsqueda avanzada

Las siguientes estrategias ayudan a mejorar la exactitud de la búsqueda.

#### 1.9.2.1 El carácter '*' como truncador

En principio, la búsqueda se efectúa por palabra completa, pero se puede emplear el carácter '*' (asterisco) como truncador. De esta manera, si se introduce el término:

- electr*

Se buscarán simultáneamente electricidad, electrónica, electrificar, etc., esta técnica solamente funciona cuando el truncador aparece como el último carácter del término de búsqueda, y no sirve como comodín genérico en otras posiciones.

El truncador incrementa el número de resultados de una búsqueda, por lo que debería ser usado con cierta precaución. Si se quieren buscar dos palabras muy similares, y no se tiene la total certeza de que con el truncador no se extenderá excesivamente el término de búsqueda, es preferible usar el operador.  Por ejemplo, se puede teclear:

- electrónica OR electricidad

Para asegurarse de que en la búsqueda no se van a incluir otros términos como electrónico o electrificar.

#### 1.9.2.2 El operador *AND*

El operador *AND* se utiliza para buscar aquellos documentos que contienen dos o más términos. Por ejemplo, si se teclea:

- iniciativa AND comunicación

Se buscarán los documentos que contienen ambas palabras. Para utilizar este operador, simplemente, hay que teclearlo en el campo de búsqueda, como si fuera un término más. Puede escribirse indistintamente en mayúsculas (*AND*) o en minúsculas (*and*).

El uso de *AND* reduce el número de resultados, debido a que el servidor únicamente incluirá en la lista aquellos documentos que contengan todos los términos teclados en el campo de búsqueda, y no sólo uno de ellos. Sin embargo, las palabras buscadas no necesariamente aparecerán seguidas en el documento. Si se quiere buscar términos seguidos, será necesario efectuar una búsqueda literal.

#### 1.9.2.3 El operador *OR*

El operador *OR* se usa para unir en una sola búsqueda todos aquellos documentos que contengan al menos una de las palabras clave que interesa buscar. Así, si se introduce en el campo de búsqueda:

- iniciativa OR comunicación

El resultado será un listado de todos los documentos que contengan uno cualquiera de los dos términos introducidos. El uso de *OR* incrementa el número de resultados. Al igual que los demás operadores, puede escribirse indistintamente en mayúsculas (*OR*) o en minúsculas (*or*).

Este operador es el que actúa por defecto cuando se introducen dos o más palabras separadas por un espacio, sin ninguna otra indicación. Por tanto, para hacer la misma búsqueda del ejemplo anterior, sería suficiente con introducir:

- iniciativa comunicación

El resultado en ambos casos será exactamente el mismo.

#### 1.9.2.4 El operador *NOT*

El operador *NOT* se utiliza para excluir de la búsqueda todos los documentos que contengan un determinado término. Si por ejemplo se introduce:

- iniciativa *NOT* comunicación

Se estará pidiendo al servidor que busque aquellos documentos que contengan el primer término pero no el segundo.

Dicho de otra manera, el servidor buscará los documentos que contengan la palabra iniciativa, y de ese conjunto excluirá aquellos que contengan la palabra comunicación.

El uso de *NOT* reduce el número de resultados de la búsqueda. Puede usarse indistintamente en mayúsculas (*NOT*) o en minúsculas (*not*).

#### 1.9.2.5 Uso de comillas para buscar expresiones

Para buscar dos (o más) términos seguidos, formando una expresión, se han de introducir en el campo de búsqueda entrecomillados. Por ejemplo, si se desea encontrar documentos que contengan la expresión «República Checa», debe escribirse:

- "República Checa"

Si no se incluyen las comillas, el servidor efectuará la búsqueda de documentos que contengan al menos uno de los dos términos, no necesariamente seguidos. En cambio, si van entrecomillados, el servidor los trata como si fueran un solo término. Pueden utilizarse indistintamente comillas dobles (") o comillas sencillas (').

Cuando se teclea en el campo de búsqueda una expresión entrecomillada, la frase completa se convierte en el término buscado, de manera que si se incluye algún operador dentro de las comillas, éste no funcionará como tal, sino que formará parte de la expresión buscada. Por ejemplo, si introducimos en el campo de búsqueda la expresión:

- "*lost and found*"

La palabra *and* no será interpretada como operador, sino como parte de la expresión que se desea buscar.

#### 1.9.2.6 Uso de paréntesis para búsquedas complejas

El uso de paréntesis abre la puerta a la realización de búsquedas complejas, que contengan varios operadores y términos. Así, la expresión:

- (reforma OR modificación) AND cambio NOT institucional

Buscará todos aquellos documentos que contengan al menos uno de los términos reforma o modificación, y que además también contengan el término cambio, excluyendo los que contengan la palabra institucional.

Si no se está seguro de si una determinada palabra aparece correctamente escrita en la base de datos donde se va a buscar, es muy recomendable utilizar el operador. Por ejemplo, para buscar el término televisión, es muy conveniente hacerlo con y sin acento, de la siguiente manera:

- television OR televisión

### 1.9.2.7. Búsqueda por título

El buscador tiene también en cuenta el nombre de los documentos al efectuar la búsqueda, lo que permite incluirlo en los términos en el momento de buscar. Esto puede ser útil, por ejemplo, si en la base de datos del boletín electrónico Info-Europa se desea limitar la búsqueda a un año concreto, de la siguiente manera:

- agricultura AND 97

Puesto que los ejemplares de Info-Europa, se encuentran en el servidor clasificados por fechas, una búsqueda como la anterior limitará el número de resultados, puesto que sólo devolverá documentos que contengan, además de la palabra agricultura, la cifra 97, lo que muy probablemente se refiere a ejemplares del boletín publicado en ese año.

De la misma manera, en la base de datos del Diario Oficial de las Comunidades Europeas (DOCE), si se conoce el número del diario en que se quiere buscar, fácilmente se le puede incluir en la búsqueda. Por ejemplo, la siguiente expresión buscará la palabra aduana en el número L121 del año 96 del DOCE.

- aduana AND L121 AND 96

### 1.9.3 Algunos detalles técnicos

El servicio de búsqueda debe todo su mérito a un excelente CGI, llamado WWWWAIS. Se trata de un pequeño programa, cuya función básica es hacer de puente entre una página *Web* que contenga uno o más campos de búsqueda y los índices creados por un programa indexador de documentos.

Es decir, WWWWAIS no busca en los documentos por sí mismo, sino que pasa la consulta a un servidor WAIS en el mismo o diferente servidor, el cual rastrea sus índices y devuelve la respuesta al CGI. Éste entonces le da el formato adecuado para una página *Web*, y rápidamente se la envía al usuario que se encuentra efectuando la búsqueda. Algunos ejemplos de indexadores *WAIS* pueden ser *freeWAIS*, entre otros.

#### 1.9.3.1 Formatos de búsqueda

No posee, sólo puede hacerse una búsqueda simple por palabras claves al no ser muy grande su base de datos, los resultados se muestran rápidamente.

#### 1.9.3.2 Modos de ordenamiento de resultados

No se sigue un orden determinado, aparecen a medida que los sitios van dando sus altas. Los más nuevos son mostrados en primer lugar.

Aparecen luego ordenados por área temática, toma la palabra clave introducida en la búsqueda para hallar las descripciones que la contengan.

#### 1.9.3.3 Procedimiento de alta de un motor de búsqueda

Sólo es necesario llenar un formulario con los datos de su sitio *Web*, descripción, su nombre y su e-mail.

#### 1.9.3.4 Medición de parámetros

- **Tamaño**

Se refiere a la cantidad de *URL* distintos, páginas, sitios, portales, etc. Se pueden mandar sitios para ser incluidos, pero los editores toman la decisión de incluirlo o no.

- **Cobertura**

Se refiere a las categorías, idiomas que puede encontrar durante la consulta.

- **Lógica *boolean***

Indica la lógica *booleana* que utiliza para poder optimizar sus consultas, utilizando las siguientes palabras: "and", "or", "not", "+", "-" , “*”, “/”.

- **Los Resultados**

La búsquedas en algunos motores de búsqueda suelen abarcar tres sectores principales: las categorías de directorios, los sitios *Web* enumerados por idioma, y las páginas *Web* contenidas en los índices. En el caso de los dos primeros, algunos motores buscan las correspondencias en su propia base de datos y después ordena los resultados comenzando por los más relevantes y terminando por los que no lo son. Algunos de los factores que determinan el nivel de relevancia son: el número de palabras clave que se hayan encontrado (mientras más palabras clave haya en la correspondencia, más alto será el nivel); las correspondencias exactas con las palabras (que tienen mayor nivel que las correspondencias aproximadas); y en qué lugar del documento se encontraron las palabras buscadas (se le concede mayor nivel a una correspondencia en el título del sitio *Web* que a las que ocurren en los comentarios o en la URL)."

### 1.9.4 Búsqueda avanzada

La pantalla para la búsqueda avanzada es muy amplia respecto de las opciones disponibles para limitar la búsqueda. Se puede escoger entre combinaciones *boolean*, formas de las palabras, de qué parte de la página vienen las palabras, cuantos resultados por pantalla, fecha, tipo de medio, idioma.

### 1.9.5 Los resultados

Primero, RADAr encuentra todas las correspondencias con las palabras clave, y después clasifica los resultados de acuerdo con su grado de relevancia dentro de cada sección específica. RADAr clasifica los resultados de la siguiente forma.

#### 1.9.5.1 Correspondencias con múltiples palabras clave

Aquellos documentos que corresponden a más palabras clave tendrán un rango superior a los que corresponden a menos de éstas.

#### 1.9.5.2 Ponderación de la sección del documento

Los documentos que corresponden a palabras incluidas en el título se sitúan por encima de los que incluyan la correspondencia en el resto del texto o URL.

#### 1.9.5.3 Generalidad de la categoría

Las categorías que estén situadas en una posición más elevada dentro de la jerarquía ramificada de RADAr (es decir, las categorías más generales) ocupan un rango superior a las categorías de nivel inferior (es decir, las que tienen un alcance más restringido).

# 2. CARACTERISTICAS DE LAS BASES DE DATOS EN INTERNET

Sin duda alguna, la infraestructura de Internet ha dado cabida a nuevos tipos de aplicaciones y servicio a los usuarios desarrollados por empresas, organizaciones y gobiernos. Parte fundamental de estos servicios son la implementación de bases de datos accedidas a través de la World Wide *Web*. El alcance y el fácil acceso a ellas, así como la reducción de costos y la popularidad que ha cobrado la *Web*, son los principales atractivos que ofrece una aplicación de esta naturaleza.

Este capítulo pretende dar un panorama amplio sobre el funcionamiento e implementación de bases de datos en Internet, accediendo a sus datos a través de la *Web*.

## 2.1 ¿Por qué utilizar bases de datos en la *Web*?

La *Web* es un medio para localizar/enviar/recibir información de diversos tipos, aún con las bases de datos.

En el ámbito competitivo, es esencial ver las ventajas que esta vía electrónica proporciona para presentar la información, reduciendo costos y el almacenamiento de la información, y aumentando la rapidez de difusión de la misma.

Internet provee de un formato de presentación dinámico para ofrecer campañas y mejorar negocios, además de que permite acceder a cada sitio alrededor del mundo, con lo cual se incrementa el número de personas a las cuales llega la información.

Alrededor de 14 millones de personas alrededor del mundo hacen uso de Internet, lo cual demuestra el enorme potencial que esta red ha alcanzado, con lo cual se puede decir que en un futuro no muy lejano, será el principal medio de comunicación utilizado para distintos fines.

Pero, no sólo es una vía para hacer negocios, sino también una gran fuente de información, siendo éste uno de los principales propósitos con que fue creada. Una gran porción de dicha información requiere de un manejo especial, y puede ser provista por bases de datos.

En el pasado, las bases de datos sólo podían utilizarse al interior de las instituciones o en redes locales, pero actualmente, la *Web* permite acceder a bases de datos desde cualquier parte del mundo. Éstas ofrecen, a través de la red, un manejo dinámico y una gran flexibilidad de los datos, como ventajas que no podrían obtenerse a través de otro medio informativo.

Con estos propósitos, los usuarios de Internet o Intranet pueden obtener un medio que puede adecuarse a sus necesidades de información, con un costo, inversión de tiempo, y recursos mínimos. Asimismo, las bases de datos serán utilizadas para permitir el acceso y manejo de la variada información que se encuentra a lo largo de la red.

## 2.2 Seguridad

La evaluación de este punto es uno de los más importantes en la interconexión de la *Web* con bases de datos.

Al nivel de una red local, se puede permitir o impedir, a diferentes usuarios el acceso a cierta información, pero en la red mundial de Internet se necesita de controles más efectivos en este sentido, ante posible espionaje, copia de datos, manipulación de éstos, etc.

La identificación del usuario es una de las formas de guardar la seguridad. Las identidades y permisos de usuarios están definidas en los Archivos de Control de Acceso.

Pero la seguridad e integridad total de los datos puede conservarse, permitiendo el acceso a distintos campos de una base de datos, solamente a usuarios autorizados para ello.

En este sentido, los datos pueden ser presentados a través de la *Web* de una forma segura, y con mayor impacto a todos los usuarios de la red mundial. Para la integración de bases de datos con la *Web*, es necesario contar con una interfase que realice las conexiones, extraiga la información de la base de datos, de un formato adecuado, de tal manera, que pueda ser visualizada desde un visor de la *Web*, y permita lograr sesiones interactivas entre ambos, dejando que el usuario haga elecciones de la información que requiere.

## 2.3 Los programas como interfases para acceder a bases de datos

Un *gateway* es una conexión con el sistema operativo externo, la acción de llamar un programa desde un navegador *Web* es muy sencilla para el usuario, lo cual es uno de los principales atractivos del *Common Gateway Interface*, el tener acceso a una base de datos a través de un programa tiene una metodología propia, comúnmente el usuario hace *clic* sobre un botón predeterminado o sobre un vínculo, en este momento el navegador envía una solicitud de ejecutar el programa al servidor *Web*, el servidor *Web* revisa la configuración y los archivos de acceso para asegurarse que se cuenta con el permiso de ejecución del programa y se asegura de que éste exista, cualquier resultado producido por el programa se devuelve al navegador *Web* que despliega el resultado.

Los programas son un *gateway* de doble sentido. Los datos pueden transferirse al programa para su procesamiento, al igual los programas pueden devolver información al servidor *Web*. Con esto, la información introducida por el usuario puede afectar el comportamiento del programa y los resultados devueltos por el programa son resultado directo de lo que introduce el usuario. En algunos casos, los programas se ejecutan al cargar la página y los resultados se despliegan como parte de la misma.

### 2.3.1 La interfase

La interfaz proporciona un método para interactuar con una base de datos. Los proveedores de bases de datos pueden proporcionar una interfase de varios niveles y complejidad dependiendo de las necesidades. Los *gateway* reciben datos transferidos desde un navegador *Web* mediante un servidor de y los convierten a un formato que la base de datos pueda entender. La información convertida se transfiere a la interfaz de la base de datos y ésta la ejecuta. Los resultados se devuelven al programa de *gateway*, el cual los convierte a un formato de manera que el navegador los pueda desplegar.

Un *gateway* no puede trabajar solo, requiere de un tipo de canal para hacer contacto con la base de datos, este canal lo proporciona la interfase de la base de datos, el cual es un software especial que suministra el proveedor. El *gateway* se comunica con la interfase, la cual se pone en contacto con la base de datos.

Dependiendo de las necesidades de la aplicación, el diseño de base de datos puede o no tener capacidades de redes, esto lo determina el software del proveedor, el método más sencillo se alcanza con una base de datos sin capacidad de red, debido a que el servidor de los programas y el servidor de la base de datos están ubicados en la misma máquina, de esta forma los programas tienen acceso a cualquier programa de interfase que deseen utilizar, las consultas y los resultados devueltos no van y vienen a los programas a través de la red, de esta manera se optimiza el tiempo de respuesta.

Sin embargo, se supone que los navegadores están accediendo a la información de manera remota, si la base de datos es de mediano o gran tamaño y hay muchos usuarios, no es buena idea tener el servidor de base de datos y el servidor de los programas en la misma computadora. Si la base de datos incluye capacidades interconstruidas de redes tiene la opción de correr el servidor de en una máquina remota y acceder a la base de datos utilizando el software de bases para red, muchas veces es preferible ubicar en el servidor de los programas y tener una máquina dedicada al servidor de la base de datos para evitar problemas potenciales y proteger los datos tanto como sea posible, este método funciona bien si se tiene más de un servidor de base de datos, un solo servidor que puede acceder a varias máquinas con bases de datos mediante programas.

Este método aligera la carga de una base de datos y se refleja un notable incremento en el rendimiento; sin embargo, esto sólo vale la pena en el caso de bases de datos verdaderamente grandes.

## 2.4 Diferencias en relación con otras bases de datos

A diferencia de la implementación de bases de datos comunes, las bases de datos en Internet tienen características especiales, podemos mencionar algunas como la conexión, las interfases, las cuales interactúan con el usuario entre otras. Las bases de datos en Internet utilizan los modelos conocidos para la creación de su diseño. Para darle un enfoque particular a las diferencias con otras bases de datos se tomarán puntos importantes de las interfases y la recuperación de información de las bases de datos.

Cuando alguien usa un navegador *Web* para acceder a una base de datos hay varios componentes que intervienen para transferir la consulta del usuario a la base de datos y devolver los resultados al navegador, la acción se desarrolla de la siguiente manera:

a. El usuario llama a un programa *Gateway* que utiliza, haciendo *clic* en un hipervínculo u oprimiendo un botón del formulario. El navegador reúne toda la información escrita por el usuario para enviarla al programa.

b. El navegador contacta al servidor de la máquina donde reside el programa, pidiéndole que localice a este último y le transfiere la información. El servidor corrobora si la máquina solicitante tiene autorización de acceso al programa. Si el usuario tiene acceso, el servidor de localiza el programa *Gateway* y transfiere a éste la información del navegador *Web*.

c. Se ejecuta el programa *Gateway*.

d. El proceso *Gateway* convierte la información recibida a un formato que la base de datos sea capaz de entender.

e. El *Gateway* usa el módulo de la base de datos para transferir la consulta a la interfaz de la base.

f. La interfase de la base de datos analiza la sintaxis de la consulta para asegurar que sea precisa.

g. Si la interfase encuentra un error de sintaxis en la consulta, se envía un mensaje de error al programa *Gateway*.

h. El mensaje de error se envía al servidor de, el cual lo transfiere al navegador *Web* para que éste lo despliegue al usuario.

i. Si no hay error, la interfase envía la consulta a la bases de datos.

j. La base de datos atiende la consulta y devuelve los resultados al programa *Gateway* a través de la interfase.

k. El programa *Gateway* formatea los resultados y los envía al servidor, por medio de él, para su envío al navegador *Web*.

l. El navegador *Web* despliega los resultados.

Hacer inserciones, actualizaciones y eliminaciones requiere de un procedimiento más complejo que el utilizado en base de datos locales, para hacer simples consultas. El código de inserción debe verificar si existen los datos en la base de datos, con el fin de evitar duplicidad en la información, el código de actualización debe asegurarse que la información exista, antes de modificarla, en caso contrario debe originarse un error. Las acciones de eliminación y edición debe de asignarse a un grupo reducido y confiable para evitar la eliminación de datos necesarios.

Para procesar algo más que una simple consulta en un ambiente de esta naturaleza, es necesario algún tipo de control de acceso como lo puede ser comparar el identificador de conexión del usuario contra una lista de usuarios autorizados para determinada acción, asignar a cada tipo usuario un identificador de conexión y asignar a cada tipo nivel un nivel de acceso basándose en las funciones que tienen permitidas o utilizar la autenticación de clave de acceso de la base de datos destino para validar los niveles de acceso a los usuarios. En definitiva, el método que utilice depende de cómo planee utilizar el programa de *Gateway*.

### 2.4.1 Proceso de consultas integrado

El proceso de consultas para una base de datos en Internet varía, ya que cuando un usuario realiza una petición el motor de búsqueda pasa las peticiones al servidor principal, si éste no encuentra ninguna respuesta, el proceso integrado de consultas, envía la petición hacia otro servidor del sistema distribuido de base de datos.

### 2.4.2 Procesamiento de la transacción distribuida

Otra diferencia es el proceso de la transacción de la base de datos, ahora este proceso se hará con un sistema de base de datos distribuido que se encuentra en distintos servidores de Internet.

## 2.5 Características de modelos utilizados

### 2.5.1 Modelo entidad - relación

Uno de los modelos más conocidos para los sistemas de base de datos es el modelo entidad - relación el cual consta de:

- Tablas
- Campos
- Relaciones
  - De uno a uno
  - De uno a muchos
  - De muchos a muchos
- Cláusulas SQL
  - Select
  - Insert
  - Update
  - Delete
- Transacciones
  - Commit
  - Rollback

### 2.5.2 Modelo orientado a objetos

El modelo orientado a objetos en los últimos tiempos ha tenido un crecimiento bastante considerable, debido a las nuevas herramientas 4GL que existen en el mercado, en este modelo podemos mencionar las siguientes características:

- Objetos
- Eventos
- Funciones
- Procedimientos
- Polimorfismo
- Encapsulamiento
- Relaciones

## 2.6 Relaciones con las bases de datos

Interfase de puerta de enlace o *Gateway* común (CGI) es un conjunto de especificaciones para transferir información entre el explorador de un cliente *Web*, un servidor *Web* y una aplicación CGI. El explorador de un cliente *Web* puede iniciar una aplicación CGI completando un formulario HTML o haciendo *clic* en un vínculo de una página HTML del servidor *Web*. Como ocurre con ISAPI, la aplicación CGI puede aceptar información escrita por el usuario y tratarla de cualquier modo que se pueda programar, y después devolver los resultados en una página HTML o enviar la información a una base de datos. Como las aplicaciones CGI sencillas a menudo están escritas con lenguajes de archivos de comandos como Perl, a las aplicaciones CGI también se les conoce como "archivos de comandos".

Microsoft Internet Information Server puede usar la mayoría de las aplicaciones de 32 bits que se ejecuten en Windows NT y cumplan las especificaciones CGI. La siguiente ilustración muestra cómo intercambian información un explorador, un servidor y una aplicación CGI utilizando CGI. El resto de esta sección trata sobre este proceso que consta de cinco partes.

### 2.6.1 El cliente envía una petición

El explorador de un cliente puede realizar una petición CGI a un servidor mediante uno de estos dos métodos:

- **GET**

El cliente añade los datos a la dirección URL que pasa al servidor.

- **POST**

El cliente envía los datos al servidor mediante el campo de datos de mensajes, por lo que se pueden superar las limitaciones de espacio inherentes al método GET. El cliente inicia un proceso CGI haciendo *clic* en cualquiera de los elementos siguientes de una página HTML:

- Un vínculo de hipertexto que ejecute el archivo de comandos directamente.
- El botón "Enviar" de un formulario HTML.
- Un objeto en línea recuperado con el método GET.
- Un objeto de búsqueda (es decir, uno que utiliza la etiqueta ISINDEX de HTML).

### 2.6.2 Servidor recibe la petición

La dirección URL que el explorador del cliente envía al servidor contiene el nombre del archivo de comandos CGI o la aplicación que va a ejecutarse. El servidor compara la extensión del archivo con la clave de registro ScriptMapping del servidor para decidir qué ejecutable debe iniciar.

El servidor tiene entradas ScriptMap para archivos .cmd y .bat, que inician Cmd.exe; y para archivos IDC, que inician el conector de bases de datos de Internet. Para permitir que el servidor inicie un tipo de aplicación CGI sin asignación de extensión, agregue una entrada para dicho tipo de aplicación a la clave del registro. Por ejemplo: para permitir la ejecución de los archivos de comandos de Perl, agregue una entrada similar a la siguiente:

pl: REG_SZ: C:\RESKIT\PERL\BIN\PERL.EXE %s %s

Donde

- \Reskit\Perl\Bin\ es el directorio que contiene el ejecutable.

- Perl.exe es el comando ejecutado.

- El primer %s es la ruta de acceso traducida del archivo de comandos de PERL (la dirección URL traducida a una ruta de acceso local).

- El segundo %s es la cadena de consultas (información de la dirección URL) y sólo se pasa como un parámetro de la línea de comandos si la cadena de consultas no contiene el signo igual (=).

- El servidor pasa la petición a la aplicación.

- El servidor pasa la información a la aplicación CGI utilizando variables de entorno y, a continuación, inicia la aplicación. Algunas de estas variables están relacionadas con el servidor pero la mayoría vienen del explorador del cliente y tienen relación con el explorador del cliente o con la petición que se está enviando.

### 2.6.3 La aplicación CGI devuelve los datos al servidor

La aplicación realiza su procesamiento. Si son adecuados, la aplicación escribe los datos en un formato que el cliente pueda recibir en el flujo de salida estándar (STDOUT). La aplicación debe seguir un formato específico a la hora de devolver los datos:

a. La primera o primeras líneas contienen las directivas del servidor, así como el tipo de contenido MIME. Otras directivas del servidor son *Location* (que el cliente redirige o devuelve a otro documento) y *Status*.

b. A continuación de las directivas del servidor debe haber una línea en blanco.

c. Los datos que la aplicación devuelve al cliente siguen a la línea en blanco.

El servidor devuelve los datos al cliente. El servidor toma los datos que recibe de STDOUT y agrega encabezados estándar y, a continuación, devuelve el mensaje al cliente.

### 2.6.4 API de Internet server

ISAPI para Windows NT se puede utilizar para escribir aplicaciones que los usuarios de *Web* pueden activar completando un formulario HTML o haciendo *clic* en un vínculo de una página HTML de su sitio *Web*. La aplicación remota puede aceptar información introducida por el usuario y tratarla de cualquier modo que se pueda programar, y después devolver los resultados en una página HTML o enviar la información a una base de datos.

ISAPI se puede usar para crear aplicaciones que se ejecuten como DLL en su servidor *Web*. Si ha utilizado archivos de comandos CGI anteriormente, encontrará que las aplicaciones ISAPI tienen un mejor rendimiento porque se cargan en memoria durante la ejecución del servidor. Requieren menos tiempo de espera porque cada petición no inicia un proceso distinto.

Otra característica de ISAPI es que permite el preprocesamiento de peticiones y el postprocesamiento de respuestas, administrando las peticiones y respuestas con el protocolo de transferencia de hipertexto que sean específicas del sistema.

Puede usar filtros ISAPI en aplicaciones como autenticaciones personalizadas, accesos o registros, así también, se pueden crear sistemas muy complejos usando filtros y aplicaciones ISAPI. Puede combinarse extensiones ISAPI con el conector de bases de datos de Internet para crear sitios altamente interactivos.

### 2.6.5 Características del conector de bases de datos de Internet

El conector de bases de datos de Internet tiene varias características que facilitan la creación de páginas *Web* que contengan datos de una base de datos

### 2.6.6 Archivos del conector de bases de datos de Internet

Los archivos del conector de bases de datos de Internet contienen la información que se utiliza para tener acceso a la base de datos. La siguiente sección describe las características de los archivos del Conector de bases de datos de Internet.

#### 2.6.6.1 Parámetros

Una consulta se puede definir completamente en un archivo del conector de bases de datos de Internet, aunque, este tipo de consulta es útil, pueden crearse páginas *Web*, son más potentes mediante la utilización de parámetros. Los parámetros son los nombres y valores de los controles del formulario de HTML, como por ejemplo "<INPUT…>", así como los nombres especificados directamente en las direcciones URL. Estos nombres y valores los envían los exploradores de *Web* y pueden utilizarse en instrucciones del servidor.

Supongamos que la página *Web* solicita al usuario la cifra de ventas anuales y a continuación, da el nombre "sales" a la variable asociada. Se muestra un formulario con un campo de entrada que se utiliza para obtener el número.

La sintaxis del campo de entrada y el botón de ejemplo en formato HTML es la siguiente:

<FORM METHOD="POST" ACTION="/scripts/ejemplos/ejemplo.idc">
Escriba las ventas anuales hasta la fecha: <INPUT NAME="sales" VALUE="5000" >
<INPUT TYPE="SUBMIT" VALUE="Ejecutar consulta">
</FORM>

En el código anterior en el conector de bases de datos de Internet se utiliza el parámetro que aparece en negrita en lugar del número VALUE ingresado que es igual a 5000:

- Sentencia SQL
  SELECT au_lname, ytd_sales
  from pubs.dbo.titleview
  where ytd_sales > %sales%

Aquí el nombre del parámetro debe ser "sales" para que corresponda a <INPUT NAME= "sales" …> de la página *Web*. Los parámetros deben estar entre signos de porcentaje (%) para distinguirlos de un identificador normal. Cuando el Conector de bases de datos de Internet encuentra el parámetro en el archivo .idc, sustituye el valor enviado por el explorador de *Web* y, posteriormente, envía la instrucción al controlador.

El signo de porcentaje (%) es también un carácter comodín para las consultas. Los comodines se utilizan en las consultas para buscar un elemento de una tabla que contenga determinados caracteres. Para insertar un único signo "%" en un comodín, utilice "%%". Esto evita que el IDC intente utilizar % como marcador de parámetro. Por ejemplo:

- Sentencia

  SELECT au_lname, ytd_sales, title
  from pubs.dbo.titleview
  where title like '%%%título%%%'

Para que un signo de porcentaje se reconozca como un comodín de debe especificarlo dos veces y, a continuación, agregar los caracteres de porcentaje alrededor del parámetro para distinguir la cadena como parámetro. En el ejemplo, la consulta busca la palabra título en todas las entradas de la columna de títulos. Esta consulta devuelve lo siguiente:

- Título
- Título y hecho
- Página del título principal
- Autor y título

Para devolver todas las entradas que contienen la palabra título en las seis primeras letras, debe dar el siguiente formato a las consultas:

- Sentencia

```
SELECT au_lname, ytd_sales, title
 from pubs.dbo.titleview
 where title like '%título%%%'
```

En este ejemplo se devuelven los siguientes resultados:

- Título
- Título y hecho

Para devolver todas las entradas que contengan la palabra título como las seis últimas letras, debe dar el siguiente formato a las consultas:

- Sentencia

```
SELECT au_lname, ytd_sales, title
 from pubs.dbo.titleview
 where title like '%%%título%'
```

En este ejemplo se devuelven los siguientes resultados:

- Título
- Autor y título

Es posible crear potentes conjuntos de páginas *Web* utilizando el resultado de una consulta para proporcionar vínculos con otras consultas. Por ejemplo, para mostrar los títulos de un autor individual, en lugar de devolver el nombre del autor como texto normal, puede darle formato como un vínculo y, a continuación, utilizar el vínculo para realizar otra consulta.

### 2.6.7 Campos obligatorios en un archivo IDC

- **Private** -- Identifica el tipo de acceso de las variables que controlan la transferencia de datos.

- **Datasource** -- El nombre que corresponde al nombre del origen de datos (DSN) del sistema que ha creado previamente utilizando el administrador de o la herramienta proporcionada para el desarrollo.

- **Template** -- El nombre del archivo con extensión HTML que da formato a los datos devueltos por esta consulta. Por convención, estos archivos utilizan la extensión .HTX.

- **SQL Statement** -- La instrucción que se va a ejecutar. La instrucción SQL puede contener valores de parámetros, que deben ir entre signos de porcentaje (%), del cliente. En el archivo del conector de bases de datos de Internet, la instrucción SQL puede ocupar varias líneas. Después del campo SQLStatement, todas las líneas que comiencen con un signo más (+) se considerarán parte del campo SQLStatement. En el mismo archivo pueden aparecer varias instrucciones SQL.

### 2.6.8 Campos opcionales en un archivo IDC

- **Expires** -- El número de segundos que hay que esperar antes de actualizar una página con salida en caché. Si una petición posterior es idéntica, la página de caché se devolverá sin tener acceso a la base de datos. Este campo es útil cuando desee forzar una nueva consulta de la base de datos tras un determinado período de tiempo. De forma predeterminada, el IDC no pone en caché las páginas de salida, sólo las incluye en caché cuando se utiliza el campo *Expires*.

- **MaxFieldSize** -- El espacio de búffer máximo por campo que asigna el IDC. Los caracteres posteriores a éste se truncarán. El parámetro se aplica sólo a aquellos campos de la base de datos que superen los 8192 bytes. El valor predeterminado es 8192.

- **MaxRecords** -- El número máximo de registros que el IDC devolverá de cualquier consulta. El valor MaxRecords no se define de forma predeterminada, lo que significa que una consulta puede devolver hasta 4.000 millones de registros. Defina este valor para limitar los registros devueltos.

- **ODBCConnection** -- Inserte este campo con el valor de agrupación para agregar la conexión a la agrupación de conexiones, que conserva la conexión con la base de datos abierta para futuras peticiones. A continuación, el IDC envía los datos a través de una conexión agrupada para la posterior ejecución de un archivo .IDC que contiene los mismos valores de *Datasource*, *Username* y *Password*. Defina esta opción para mejorar el rendimiento utilizando el conector de bases de datos de Internet.

  Además, hay una opción sin agrupación, que especifica que la conexión del archivo .idc en la que se define esta opción no debe tomarse de la agrupación de conexiones. Para administrar la caché de conexiones con más precisión, defina el valor de este campo como *nopool*. Además, si hay un límite en cuanto al número de conexiones actuales, probablemente no desee que la agrupación de conexiones monopolice todas las conexiones; de lo contrario, nadie podría conectarse a SQL Server.

- **Passwords** -- Contraseña que corresponde al nombre de usuario. Si no existiera ninguna contraseña, este campo puede dejarse en blanco.

- **Required Parameters** -- Los nombres de los parámetros, si existe alguno, que Httpodbc.dll se asegura que pasarán del cliente; de lo contrario, devolverá un error. Los nombres de los parámetros se separan mediante comas.

- **Translationfile** -- Ruta de acceso al archivo que asigna caracteres no ingleses (como à, ô o é) para que los exploradores puedan mostrarlos adecuadamente en formato HTML. Si el archivo de traducción no está en el mismo directorio que el archivo .idc, debe escribir la ruta de acceso completa al archivo de traducción. Sintaxis:

  Translationfile: C:\nombre_directorio\nombre_archivo.

  Si publica una base de datos en un idioma que no sea inglés, utilice el campo Translationfile. Un archivo de traducción es un archivo de texto en el que los caracteres especiales se asignan en el siguiente formato: valor=cadena<CR>, donde valor es un carácter internacional y cadena es el código de traducción de HTML.

- **Username** -- Nombre de usuario válido para el origen de datos proporcionado por el campo *datasource*.

  Nota: Si utiliza Microsoft SQL Server con la opción de seguridad integrada, los campos de nombre de usuario y contraseña del archivo .idc se ignoran. La conexión con SQL Server se realiza utilizando las credenciales del usuario *Web*.

  Si la petición se realiza como un usuario anónimo, el nombre de usuario y la contraseña están determinados por la configuración del usuario anónimo (el valor predeterminado es IUSR_nombreequipo), del administrador de servicios de Internet. Si la petición del cliente contenía credenciales para la conexión, el nombre de usuario y la contraseña proporcionados por el usuario final se utilizan para conectarse a SQL Server.

### 2.6.9 Campos opcionales avanzados de ODBC

Las opciones avanzadas de ODBC permiten depurar y ajustar el controlador utilizado por el conector de bases de datos de Internet. Para obtener más detalles acerca de estas opciones, consulte la documentación de su controlador. El formato del archivo IDC es:

ODBCOptions: Nombre_opción=Valor[,Nombre_opción=Valor…]

Por ejemplo, para que la instrucción SQL deje de ejecutarse durante más de 10 segundos y activar el seguimiento de las llamadas a funciones en el archivo IDC tiene que especificar lo siguiente:

ODBC Options:
SQL_QUERY_TIMEOUT=10,
SQL_OPT_TRACE=1,
SQL_OPT_TRACEFILE=path\Sql.log

Nombre de la opción Valor Propósito SQL_ACCESS_MODE
0 = Lectura/escritura - 1 = Sólo lectura.

Indicador para el controlador del origen de datos de que no requiere la conexión para ser compatible con instrucciones SQL que hacen que se produzcan actualizaciones. Este modo puede utilizarse para optimizar estrategias de bloqueo, la administración de las transacciones u otras áreas apropiadas para el controlador o el origen de datos. El comportamiento del controlador y del origen de datos cuando se les pide que procesen instrucciones SQL que no son de sólo lectura durante una conexión de sólo lectura está definido por la implementación.

De forma predeterminada SQL_ACCESS_MODE se establece como 0, lo que permite la lectura y la escritura. SQL_LOGIN_TIMEOUT Entero, se refiere al número de segundos que debe esperar para que finalice un inicio de sesión antes de desconectarse.

El valor predeterminado depende del controlador y debe ser distinto de cero. Si el valor es 0, el tiempo de espera se inhabilita y un intento de conexión esperará indefinidamente. Si el tiempo de espera especificado sobrepasa el tiempo de espera de inicio de sesión máximo del origen de datos, el controlador sustituirá dicho valor: SQL_OPT_TRACE

0 = Desactivar el seguimiento
1 = Activar el seguimiento

Cuando se activa el seguimiento, todas las llamadas a funciones que realiza a través de una archivo como por ejemplo Httpodbc.dll, y se escriben en el archivo de seguimiento. Con la opción SQL_OPT_TRACEFILE se puede especificar un archivo de seguimiento. Si este archivo ya existiera, el controlador lo agrega al archivo, de lo contrario, crea el archivo. Si se activa el seguimiento sin especificar ningún archivo de seguimiento, escribe en el archivo Sql.log. SQL_OPT_TRACEFILE (Nombre de Archivo). El nombre del archivo de seguimiento que se va a utilizar cuando SQL_OPT_TRACE=1.

Nota: Muchos orígenes de datos no son compatibles con esta opción o solo pueden devolver el tamaño del paquete de red. Si el tamaño especificado sobrepasa o es menor que el tamaño mínimo del paquete, el controlador sustituye dicho valor.

En SQL_TRANSLATE_DLL(Nombre de archivo) se incluye el nombre de una DLL que contiene las funciones SQLDriverToDataSource y SQLDataSourceToDriver, que el controlador carga y utiliza para realizar tareas como la traducción del juego de caracteres.

En SQL_TRANSLATE_OPTION se incluye el valor que controla la funcionalidad de traducción, que es específica de la DLL de traducción que se utiliza:

SQL_TXN_ISOLATION (entero)
1=Lectura no confirmada
2=Lectura confirmada
4=Lectura repetible
8=Serializable
16=Control de versiones

Define el nivel de aislamiento de las transacciones. El conector de bases de datos de Internet no es compatible con las transacciones que se extienden más que la petición del archivo .idc. Sin embargo, en el caso de algunos DBMS, definir la opción SQL_TXN_ISOLATION como 1 (Lectura no confirmada) dará como resultado una mayor concurrencia , por tanto, se obtendrá un mejor rendimiento. Sin embargo, con esta configuración pueden recuperarse aquellos datos que otras transacciones no hayan confirmado con la base de datos.

SQL_MAX_LENGTH se incluye la cantidad máxima de datos de una columna de caracteres o binaria que devuelve el controlador. Esta opción reduce el tráfico de red y sólo debe utilizarse cuando el origen de datos (lo opuesto al controlador) de un controlador con varias capas pueda implementarlo.

SQL_MAX_ROWS indica el número máximo de filas que devuelve una instrucción SELECT. Si el valor es 0 (valor predeterminado), el controlador devuelve todas las filas. Esta opción reduce el tráfico de red cuando el propio origen de datos puede limitar las filas que se devuelven, frente a la variable incorporada MaxRecords del conector de bases de datos de Internet, que limita las filas recuperadas.

El parámetro SQL_NOSCAN puede tomar los siguientes valores:

0= Buscar y convertir cláusulas de escape
1= No buscar y convertir cláusulas de escape

Especifica si no es necesario que el controlador busque cadenas SQL en cláusulas de escape. Si se define como 0 (el valor predeterminado), el controlador busca cláusulas de escape en cadenas SQL. Si se define como 1, el controlador no busca cláusulas de escape en cadenas SQL; en su lugar el controlador envía la instrucción directamente al origen de datos. En caso de que la instrucción SQL no contenga ninguna cláusula de escape, la utilización de una sintaxis especial encerrada entre llaves ( { } ) y la posterior definición de esta opción como 1 proporcionará una pequeña mejora en el rendimiento indicando al controlador que no debe buscar en la cadena SQL.

El parámetro SQL_QUERY_TIMEOUT indica el número de segundos que hay que esperar para que se ejecute una instrucción SQL antes de cancelar la consulta. Si se define como 0 (el valor predeterminado) no hay tiempo de espera. Si el tiempo de espera especificado sobrepasa el tiempo de espera máximo del origen de datos o es menor que el tiempo de espera mínimo, el controlador sustituirá dicho valor. Entero Específico del Controlador, se refiere a los valores de la opción específicos que el controlador puede especificar con el formato número=valor. Por ejemplo: 4322=1, 234=Cadena

## 2.7 Manejo de información con herramientas de desarrollo *Web*

Como podemos ver, sobre la base de temas anteriores, un servidor *Web* es un programa que se ejecuta en una computadora conectada a Internet, con la función de escuchar en un puerto TCP/IP predefinido las solicitudes de cliente y luego responder a navegadores *Web* con el contenido basado en esas solicitudes. Cuando se escribe una dirección en el navegador, es proyectado en una dirección y puerto IP correspondiente a un servidor *Web* específico. Después de haber establecido una conexión, el cliente y el servidor se comunican con el protocolo de transferencia de hipertexto Por lo general, el servidor *Web* envía un bloque de texto en el navegador, analiza y puede solicitar contenido adicional como información gráfica. El modelo trabaja bien para la información estática, sin embargo, si queremos hacer que nuestra página presente información basada en las peticiones tecleadas por el usuario, (*COMMON GATEWAY INTERFACE*) es la respuesta. El servidor ejecuta un programa como un proceso por separado para satisfacer las solicitudes de los usuarios, como puede ser una consulta a una base de datos.

Dado que un programa es externo al navegador *Web*, puede ser escrito casi en cualquier lenguaje, ya sea compilado o interpretado. Los lenguajes populares de C, e incluso el *Shell* de UNIX. Algunos servidores *Web* ofrecen bibliotecas e intérpretes para y Visual Basic para ser utilizados por programas

.

En la actualidad, existen diferentes tipos de herramientas para implantar una base de datos en la *World Wide Web*. La capacidad y alcance del software disponible depende directamente del hardware con que se cuente para la implantación de la base de datos, del sistema operativo en función y del diseño de la misma, éste puede ser centralizado o distribuido. El servidor de base de datos puede alojarse junto con el servidor que atiende al navegador o en un servidor o servidores de bases de datos diferentes, en la misma red o en diversas redes.

Otro factor importante es el número de usuarios esperados para consultar la base de datos, debiéndose determinar un rango de usuarios contemplado, así como un crecimiento a futuro de la base de datos, debido a que del tamaño de las tablas dependerá también la rapidez de consulta. La calidad de los dispositivos de comunicación y la capacidad del servidor tienen gran relevancia para ofrecer un buen servicio a los usuarios.

# 3. MÉTODOS NUEVOS DE SELECCIÓN DE INFORMACIÓN

## 3.1 Descripción de métodos

Cuando alguien usa un navegador *Web* para acceder a una base de datos existen varios componentes que intervienen para transferir la consulta del usuario a la base de datos y devolver los resultados al navegador, la acción se desarrolla utilizando: el navegador *Web*, el motor de búsqueda y la base de datos, de la siguiente manera: (El enfoque que le daremos utilizará CGI).

El usuario llama a un programa que sea puerta de enlace (en inglés: *gateway*) que utiliza CGI, haciendo *clic* en un hipervínculo u oprimiendo un botón del formulario, de una página *Web*.

El navegador reúne toda la información escrita por el usuario para enviarla al programa CGI. El navegador contacta al servidor de HTTP en la máquina donde reside el programa CGI, pidiéndole que localice a este último y le transfiere la información.

El servidor de *HTTP* corrobora si la máquina solicitante tiene autorización de acceso al programa CGI. Si el usuario tiene acceso, el servidor de *HTTP* localiza el programa de puerta de enlace en donde se encuentra el motor de búsqueda y transfiere a éste la información del navegador *Web*.

El proceso del motor de búsqueda convierte la información recibida a un formato que la base de datos sea capaz de entender. El motor de búsqueda usa el módulo de la base de datos para transferir la consulta a la interfaz de la base, así mismo también analiza la sintaxis de la consulta para asegurar que sea precisa.

Si el motor de búsqueda encuentra un error de sintaxis en la consulta envía un mensaje de error al programa de la puerta de enlace. El mensaje de error se envía al servidor de HTTP, el cual lo transfiere al navegador *Web* para que éste lo despliegue al usuario. Si no hay error, el motor de búsqueda envía la consulta a la base de datos.

La base de datos atiende la consulta y devuelve los resultados al motor de búsqueda a través de la interfase. El motor de búsqueda formatea los resultados y los envía al servidor, por medio del CGI, para su envío al navegador *Web*.

Para procesar algo más que una simple consulta de información en un ambiente de esta naturaleza, es necesario algún tipo de control de acceso como lo pueden ser comparar el identificador de conexión del usuario contra una lista de usuarios autorizados para determinada acción, asignando a cada tipo usuario un identificador de conexión y asignando a cada tipo nivel un nivel de acceso basándose en las funciones que tienen permitidas o utilizar la autenticación de clave de acceso de la base de datos destino para validar los niveles de acceso a los usuarios. En definitiva, el método que utilice depende de cómo planee utilizar el motor de búsqueda y como está estructurado el mismo.

Existen tres métodos, los cuales son:

- Uso de cuadros de lista de selección múltiples en formularios HTML
- Consultas por lotes
- Consultas múltiples

## 3.2 Cuadros de lista de selección múltiples en formularios HTML

Este método utiliza para la seguridad de los permisos de los usuarios los conectores de bases de datos de Internet, estos conectores pueden contener información SQL, estos archivos poseen extensión IDC. Los permisos de NTFS pueden definirse y utilizarse con aplicaciones que utilicen el IIS (*Internet Information Server*, el cual explicaremos posteriormente).

La mayoría, sino es que todos los buscadores poseen un formulario HTML que contiene etiquetas <SELECT MULTIPLE…>, el conector de bases de datos de Internet convierte los elementos seleccionados en listas separada por cualquier signo; esas listas pueden utilizarse en los archivo.idc de la misma forma que otros parámetros. No obstante, puesto que el parámetro es realmente una lista, sólo suele utilizarse en instrucciones de SQL Select con una cláusula que permita validar la cantidad de elementos que se encuentran en ella, como ejemplo podemos mencionar IN.

El problema de utilizar el método de selección múltiple es dé que no existe la posibilidad de redactar la consulta a conveniencia del usuario, sino que, por otro lado el mismo navegador va restringiendo las categorías y temas en los que se desea buscar información, haciéndolo un método eficiente pero tardado en la forma de redactar la consulta, por lo que es poco utilizado.

A continuación se presenta una consideración importante del método, en donde se hace referencia a los elementos seleccionados en una variable de selección múltiple.

Si el nombre del parámetro del archivo.idc está entre comillas simples, cada elemento de la lista también estará entre comillas simples. Siempre que la columna de la cláusula IN sea una columna de caracteres o de cualquier otro tipo en el que los literales vayan entre comillas (por ejemplo, fechas y horas) debe escribir el nombre del parámetro entre comillas simples. Si no se encuentran comillas simples alrededor del nombre del parámetro, no se colocará ninguna comilla alrededor de cada elemento de la lista. Si la columna de la cláusula IN es de tipo numérico o de cualquier otro tipo en el que los literales no vayan entre comillas simples, no debe escribir el nombre del parámetro entre comillas simples.

Por ejemplo, si un formulario HTML contenía el cuadro de lista de elección múltiple que aparece a continuación:

```
<SELECT MULTIPLE NAME="región">
<OPTION VALUE="Oeste">
<OPTION VALUE="Este">
<OPTION VALUE="Norte">
<OPTION VALUE="Sur">
</SELECT>
```

Es posible construir un archivo.IDC con una instrucción SQL:

```
SELECT
nombre,
región
FROM cliente
WHERE
región IN ('%región%')
```

Si el usuario ha seleccionado "Norte", "Oeste" y "Este" en el formulario HTML, la instrucción SQL se convertiría en:

```
SELECT
nombre,
región
FROM cliente
WHERE
región IN ('Norte', 'Oeste', 'Este')
```

A continuación aparece otro ejemplo de formulario HTML, pero esta vez se utilizan datos numéricos, por lo que esta vez el parámetro del archivo.idc no aparece entre comillas.

```
<SELECT MULTIPLE  NAME="año">
<OPTION VALUE="1994">
<OPTION VALUE="1995">
<OPTION VALUE="1996">
</SELECT>
```

Es posible construir un archivo.IDC con una instrucción SQL:

```
SELECT
producto,
año_ventas
FROM
ventas
WHERE
año_ventas IN (%año%)
```

Si el usuario ha seleccionado "1994" y "1995" en el formulario HTML, la instrucción SQL se convertiría en:

```
SELECT
product,
sales_year
FROM sales
WHERE
sales_year IN (1994, 1995)
```

### 3.3 Consultas por lotes y consultas múltiples

Para la selección de información en una base de datos se pueden emplear distintos métodos que devolverán información algunas veces precisa y otras no. Nos apoyaremos en el *Structured Query Language* (SQL), el cual es un lenguaje estructurado de consultas de información que nos permitirá seleccionar información de una base de datos, en este caso de una base de datos del *Web*.

El motor de búsqueda es el encargado de tomar la información escrita por el usuario en el navegador, asi como de construir la sentencia SQL que brinde mejores resultados al usuario.

Regularmente la sentencia SELECT es la principal cláusula para construir una consulta de información en una base de datos en Internet, un motor de búsqueda tiene distintas funciones como por ejemplo actualizar en un servidor HTTP información que los usuarios envían a través de formularios. Muchos de los motores de búsqueda actuales utilizan técnicas poderosas para la selección de información y lo más importante es que esta información sea devuelta en pocos segundos, tal es el caso de los metabuscadores.

Haremos un análisis de técnicas que nos pueden ayudar en la optimización de los resultados de las consultas, tanto en tiempo como en calidad de resultados. En un archivo idc, es posible agrupar consultas SQL de dos formas distintas, como consultas por lotes o como consultas múltiples.

### 3.3.1 Consultas por lotes

Las consultas por lotes son métodos que se utilizan en bases de datos que permiten procesar varias consultas en una única instrucción. En una base de datos en Internet este método es bastante apropiado, debido a que pueden existir consultas en las cuales los “títulos” o palabras clave que se incluyen pueden ser diversas y poco congruentes, de tal forma, que el motor de búsqueda debe de poder separarlas en varias consultas y crear un archivo por lotes dentro del cual se puedan incluir todas las subconsultas si así le queremos llamar, la base de datos regresará la información al motor de búsqueda y éste debe crear N listas con los resultados posibles que devuelva cada una de las búsquedas que se ejecutaron.

Para que la consulta sea ejecutada por lotes se debe dar a las instrucciones el formato por lotes con el fin de optimizar el rendimiento.

SELECT campo1-*metatags*, campo2-*metatags*, campo_n-*metatags* from usuario.permiso.basededatos.tabla WHERE <condición>

SELECT titulo1-*metatags*, titulo2-*metatags* from usuario.permiso.basededdatos.tabla. WHERE <condición>

SELECT campo-*metatags* as “nombre” from usuario.permiso.basededatos.tabla WHERE <condición>

Los ejemplo anteriores, muestran como se escriben las sentencias SQL, en un archivo por lotes para que el motor de búsqueda envíe el archivo hacia la base de datos para que se ejecute la consulta. Si la base de datos devuelve resultados por las tres sentencias anteriores el motor de búsqueda creará tres índices por cada base de datos consultada, esta cantidad de información se presenta al usuario a través del navegador y los índices son enlaces hacia los sitios en donde se encuentra la información, se debe de tomar en cuenta que la mayoría de las veces los índices son construidos por información que se almacena en una bitácora de páginas y sitios visitados frecuentemente, pero en algunas situaciones algunos sitios han sido dados de baja por la poca frecuencia de uso, es por ello que se puede decir que algunos índices se encuentran desactualizados y esto depende de cómo el servidor HTTP maneje la información que se encuentre en la base de datos.

Básicamente, el núcleo de este método lo constituyen los famosos *MetaTags* y palabras claves.

El funcionamiento del motor de búsqueda cuando adentro de la consulta se incluyen *metatags* y/o palabras claves varia, se debe de tomar en cuenta que el motor de búsqueda crea un índice el cual le sirve para incluir los *metatags* más recientes o más buscados, a la vez, se le agrega o adjunta a las expresiones que se escriben en el navegador los *metatags* como una especie de comodines que le ayudarán al motor a realizar la búsqueda dentro de los índices que pudo haber creado, se debe mencionar que el motor de búsqueda crea tantos índices como sea necesario, dependiendo de la cantidad de palabras que se hayan escrito, esto, ayudará a separar la búsqueda en distintos criterios, como por ejemplo, puede separar la búsqueda por “títulos”, “frases”, “encabezados”, y “texto independiente”, muchos de los buscadores actuales no pueden interpretar la secuencia de caracteres como una frase indivisible, lo cual los hace poco eficientes, debido a que si se busca información relacionada con el tema de "tecnología en el siglo XXI”, los resultados de la selección de información nos mostrarán registros en donde se encuentra la palabra “tecnología” y nos mostrará por separado resultados en donde se encuentre la frase “siglo XXI”. Entre más robusto sea el índice que crea un motor de búsqueda, los resultados serán devueltos en menor cantidad de tiempo.

### 3.3.2 Consultas múltiples

Las consultas múltiples son un método que se utiliza en las bases de datos que usan los motores de búsqueda, este método consiste en procesar las consultas una por una, y no como en el método de consultas por lotes, en donde se procesan todas juntas.

Este método está enfocado en la calidad de la información devuelta y no tanto en el tiempo de respuesta, todas y cada una de las consultas de este método se ejecutan hasta que la última haya brindado sus resultados, es por ello, que el tiempo de respuesta es más lento que el método anterior, pero por cada consulta que ejecuta crea un índice, en donde no pueden existir más de 10 resultados posibles, dentro de los cuales el 90% de los mismos satisface los requerimientos del usuario. En este método también se hace uso de los *metatags* y/o de las palabras claves. Una consideración importante de este método es que tiene mucho qué ver la forma de transferencia de información entre la base de datos y el servidor de la aplicación que le presta los servicios al motor de búsqueda, esto toma mucha relevancia, debido a que cuando la base de datos es distribuida, el motor de búsqueda crea subíndices por cada partición de base de datos que encuentre y es en estos subíndices en donde los índices principales realizan su búsqueda, por ello, la comunicación entre la base de datos y el servidor es importante, porque al regresar la consulta los posibles resultados los debe de almacenar en el servidor de aplicaciones sin perder ningún byte, esta consideración se debe de tomar en cuenta en bases de datos que no soportan flujos de información muy grande.

## 3.4 Algoritmos de búsqueda

Con la alternativa del .COM y todo lo que en ello encierra Internet, todos los investigadores del mundo se han volcado a realizar sus búsquedas de información por medio de la red, con la plena certeza de que, “en la red se encuentra todo”.

Si se analiza la anterior proposición, se puede concluir con que efectivamente todo lo que necesitamos saber y/o conocer sobre determinado tema lo encontramos en la *Web*, el proceso de búsqueda está estructurado de tal forma que su principal elemento lógico es: el algoritmo que le sirve al motor de búsqueda para definir sus reglas y procesos elementales.

Desde el inicio de la década de los años setenta, se han venido escribiendo una gran cantidad de libros sobre algoritmos de búsqueda, pero no es hasta mediados de la década de los años ochenta, cuando se les clasifica por “generaciones”.

### 3.4.1 Algoritmos de búsqueda de primera generación

Si se comprende que los ordenadores son, en realidad, grandes dispositivos para el cálculo de enormes volúmenes de información, es necesario entender que los primeros algoritmos de búsqueda confiarán en esa capacidad bruta y consistirán en implantaciones correctas de la lógica que debe seguirse en el proceso de búsqueda. Los conceptos de algoritmos de primera generación se basaban en el hecho de maximizar utilidades y minimizar costos, esto por medio de grafos y algoritmos que pretenden resolver el problema de encontrar el camino más corto, están inmersos en éstos, los algoritmos de búsqueda en árboles B, árboles B+ y árboles B*, que años después se convirtieron en los algoritmos de vanguardia en búsquedas en sistemas complejos de información, en donde la capacidad de los procesadores se quedaba corta y la solución a los problemas de búsqueda era implementar algoritmos de búsqueda eficientes.

Algunos algoritmos de primera generación:

- Minimax
- Negamax
- Alfa-Beta
- Falfa-Beta
- Lalfa-Beta
- Palfa-Beta
- Scout

### 3.4.2 Algoritmos de segunda generación

Cuando a inicios de la década de los años ochenta, los algoritmos presentados anteriormente eran bien conocidos, muchos pensaron que ya había poco que decir sobre los algoritmos de búsqueda, y así lo demuestra la pobre bibliografía de esa década. A partir de entonces, se sucedieron multitud de esfuerzos en el diseño de hardware especializado para realizar la búsqueda cada vez más rápida e investigaciones en el aprendizaje automático para suplir las deficiencias de las funciones de evaluación en la estimación local de la calidad de una posición o para guiar la búsqueda.

Con los pocos algoritmos de búsqueda, hasta ese entonces, se consideraba la posibilidad de crear algoritmos genéricos de búsqueda potentes e infalibles, el crecimiento en la velocidad de procesamiento se veía mermado con la calidad de los algoritmos de búsqueda, el volumen de información que las empresas comenzaban a presentar era alarmante y a la vez motivante para desarrollar algoritmos de búsqueda que pudieran suplir a los usuarios de una herramienta lógica que complementara el hardware en crecimiento.

Sin embargo, mientras que una máquina examinaba miles de posiciones para tomar una decisión, su oponente humano sólo consideraba algunas decenas de ellas y ganaba. Por lo tanto, se comprendió la necesidad de mejorar la calidad de la búsqueda y surgieron nuevas ideas. No fue hasta la segunda mitad de la década de los años ochenta, cuando aparecieron los primeros resultados concluyentes. Mc Allester presentó los números conspiratorios, que fueron mejorados por Schaeffer, los cuales a su vez, sirvieron como base para el desarrollo de otro algoritmo de búsqueda; los números p-n propuestos por Victor Allis. Horacek presentó un modelo de razonamiento con incertidumbre para jugar al ajedrez y Althöfer introdujo el algoritmo incremental Negamax. Se reconoció la importancia de que los algoritmos de búsqueda sean selectivos, han aparecido muy pocas aportaciones en este sentido. Smith y Nau proponen, por primera vez, un algoritmo para podar anticipadamente algunas ramas del árbol de búsqueda. Finalmente, la aportación más brillante ha sido la presentación de un nuevo algoritmo por Richard Korf y David Maxwell, el *Best-First-Minimax-Search*. El crecimiento en el estudio de la inteligencia artificial también ayudó al seguimiento de la construcción de algoritmos de búsqueda eficientes.

Los algoritmos de segunda generación que se han estudiado son:

- *Best-First-Minimax-Search*
- *Simple Best-First-Minimax-Search*
- *Extension Best-First-Minimax-Search*

En los últimos años de la década de los ochenta y principios de la década de los años noventa se comenzó a tomar en cuenta un área en donde los algoritmos de búsqueda no habían sido puestos a prueba por ser esta área muy nueva y de “poco uso”.

Con el crecimiento del WWW se comenzaron a ver los frutos de los algoritmos de búsqueda en las bases de datos en Internet, el creciente uso y surgimiento de los buscadores, metabuscadores y/o motores de búsqueda obligaron a la creación de algoritmos de búsqueda en un ambiente nunca estudiado y ni siquiera contemplado, estos algoritmos debían poseer un desenvolvimiento mayor o igual a los algoritmos ya existentes; ¿por qué deberían de considerarse así?, la respuesta es sencilla, la cantidad de información que deberían de manipular tendía a crecer de manera exponecial.

### 3.4.3 Algoritmo basado en dominios

Como la alternativa del ".com" no siempre nos traerá el sitio que buscábamos, existen otras opciones. Una propuesta interesante es *Domainsurfer*, un motor de búsqueda que contiene la base de datos de dominios.com, org y .net inscritos y activos. Lo malo de *Domainsurfer* es que al tener indizados todos los dominios activos, no nos asegura que todos nos llevarán a la página real y muchos de los dominios que pulsemos serán páginas de "Este Dominio está a la venta" o "Reservado por...". En la primera versión de nuestro algoritmo de búsqueda (que es el que actualmente se usa), se contemplan únicamente cierto dominios, pero para hacer que la nueva versión del algoritmo de búsqueda sea eficiente y realmente optimice las consultas debemos ingresar todos los dominios existentes, así como un proceso de actualización de dominios y uno de actualización de índices para que cuando una página caduque, el índice se actualice y siempre en el 99% de las veces muestre información actualizada.

Debemos tomar en cuenta que el índice para ser utilizado en el algoritmo lo tenemos que relacionar con el directorio de sitio *Web*, este índice debe de estar sobre campos como: tema, título, dirección de página, descripción, categoría, categoría, subcategoría, idioma, dominio, país.

Los directorios de sitios *Web* son aplicaciones controladas por humanos que manejan grandes bases de datos con datos como los descritos anteriormente, estas bases de datos son alimentadas, cuando los administradores revisan las direcciones que les son enviadas las clasifican en subdirectorios de forma temática.

Básicamente, el proceso de optimización de consultas debe de actualizar el directorio del sitio automáticamente, esto quiere decir que cuando existan nuevas páginas para ser incorporadas al directorio, la clasificación debería realizarse por tema, título y/o cualquiera de los demás campos, esto le permitirá al motor de búsqueda tener más opciones por donde ejecutar su búsqueda y no estar restringido sólo a algunos campos.

Una de las opciones más importantes y que no debe de faltar en el directorio del sito *Web* es la categorización por idioma, debido a que, no toda la información que existe en un idioma existe en otro. La traducción y la búsqueda por idioma es un proceso netamente del motor de búsqueda, el cual está definido como un motor de búsqueda “aparte” con diccionario de términos.

Uno de los directorios que utiliza el algoritmo basado en dominios es el Dmoz, el cual es un directorio que se encuentra alimentado por miles de colaboradores, tanto en información de páginas como en terminología.

El motor de búsqueda es una máquina que va guardando en su índice copias de millones de páginas de Internet, así como asignándoles criterios que luego servirán para su búsqueda. Estos criterios o pesos le sirven al algoritmo para almacenar en un estado de "Recientes" o "Más utilizados" todas las páginas con pesos mayores. Un motor de búsqueda no cuenta con subcategorías como los directorios, pero esta deficiencia ha sido cubierta con avanzados algoritmos de búsqueda como el basado en dominios que analizan las páginas que tienen en su memoria (índice) y con ello nos proporcionan el resultado más adecuado a nuestra búsqueda.

La siguiente es una lista de los motores de búsqueda que utilizan algoritmo basado en dominios.

- Altavista (El más popular con búsqueda también de gráficos, música, idiomas, etc.)
- *Lycos*
- *Excite*
- *Go*
- *Hotbot*
- *Google* (Una nueva opción que destaca en facilidad de carga, buenos resultados y tiene incluso una versión para dispositivos inalámbricos, esto va orientado a la computación móvil.)

### 3.4.4 Algoritmo de búsqueda

Mostramos un ejemplo de una clase principal encargada de ejecutar los conectores, las cuales contienen un filtro especial para cada buscador en particular.

Los conectores extienden la clase GusPlugin que contiene rutinas necesarias para que cualquier conector funcione, (no será mostrada) como conectarse al buscador y devolver la búsqueda.

Ésta es una clase abstracta, así que puede ser de difícil comprensión, pero es necesario entenderla para conocer su funcionamiento. En el caso de que un conector no funcione, por razones de cambio en la página original, entonces se puede cambiar la extensión, por la de un conector ya hecho y dar la sensación de que la funcionalidad no se perdió. Esta clase contiene un pequeño cargador de clases, así que sólo es necesario copiar los nuevos conectores a la carpeta con los demás sin tener que modificar el resto del código, por supuesto los conectores son como “hilos” que a medida que van terminando se interpretan y devuelven los resultados, mientras los otros siguen buscando. En este caso, se limitó la carga de conectores a 2 [0-1] por la razón de que son demasiados resultados y los usuarios, pueden terminar cancelando la búsqueda.

Un conector es tomado como un enchufe a la base de datos, como una palabra clave para devolver el resultado de una consulta (ver apéndice 1). La clase que se describe en el apéndice 1, permite obtener máximo dos conectores de base de datos para realizar sus pruebas, debido a que en un buscador normal la cantidad de conectores está determinada por la cantidad de servidores de puerta de enlace a los cuales tiene derecho de acceder, esta clase, para optimizar los resultados de sus consultas, está basada en *metatags*, pero de igual forma se puede implementar para devolver pesos de los enlaces encontrados en las bases de datos, esto último, depende del motor de búsqueda.

## 3.5 Análisis de los métodos descritos

Para determinar cuál de los métodos se empleará en la construcción del motor de búsqueda, debemos hacer un análisis previo de los métodos expuestos, con la salvedad de que cualquiera de ellos puede utilizarse en el proceso de desarrollo.

### 3.5.1 Método 1. Cuadros de lista de selección múltiples

Este método convierte los datos seleccionados de las listas, en una sola consulta, debido a que el método permite restringir el tema, título, categoría, etc. de la consulta; a simple vista, podría decirse que es un método eficiente pero, no permite flexibilidad al usuario para la construcción de la consulta, se puede decir que es un "método cerrado", otro problema que se encuentra en este método, es el tiempo que se emplea en estar seleccionando temas y/o títulos de las listas para ir conformando la consulta. Una de las ventajas que se le puede encontrar en este método es cuando se llega al final de la construcción de la consulta, ésta es, una consulta bastante restringida por lo que la calidad de los resultados devueltos es muy buena. Este método ya es muy poco utilizado por los metabuscadores.

### 3.5.2 Método 2. Consultas por lotes

Este método, por medio de un archivo por lotes, permite al motor de búsqueda enviar varias consultas con palabras no congruentes; cada una de las consultas deberá devolver resultados diferentes en conceptos de búsqueda, debido a que los mismos serán almacenados en diferentes listas de resultados.

Al finalizar la ejecución del archivo de cada una de las sentencias del archivo por lotes, se muestra en el navegador por medio del motor de búsquedas los resultados obtenidos, todo esto, por medio de enlaces a las páginas encontradas en el directorio *Web* del motor de búsqueda.

La característica básica de este método es la rapidez de presentación de los resultados, debido a que por cada sentencia ejecutada se crea una lista de enlaces (links) hacia las direcciones de las páginas que muestran los datos obtenidos, en la mayoría de los navegadores, los motores de búsqueda presentan una lista menor o igual a diez enlaces, si la cantidad de enlaces obtenidos sobrepasara esta cantidad, entonces se crea una lista de páginas múltiples que son herederas de la página de la consulta principal.

Cada una de las sentencias del archivo por lotes se va ejecutando y al finalizar genera una lista de los resultados obtenidos, los cuales a la vista del usuario se convierten en enlaces a las páginas.

#### 3.5.2.1 Revisión de una tabla

Para examinar una tabla en una aplicación frontal (como en Internet), en donde se requiere visualizar los datos, pero, que al momento en que ya se hayan obtenido, la computadora deje de utilizarlos en memoria, se debe de añadir las palabras clave: "*for browse*", al final de la instrucción select enviada al SQL. La cláusula "*for Browse*" sólo puede utilizarse para visualizar una tabla y no para utilizarla donde exista el operador UNION.

Por ejemplo:

Select (Sentencia en servidor de aplicaciones)

...

for browse

La utilización de este método resulta beneficiosa cuando se requieren resultados de manera rápida, para ello se debe de tomar en cuenta que la información que se busca debe ser de fácil acceso.

### 3.5.3 Método 3. Consultas múltiples

Este método de selección de información para resultados de consultas en Internet por medio de motores de búsqueda, es muy utilizado cuando se requiere de información que sea precisa y de difícil selección, el método de consultas múltiples funciona obteniendo las palabras claves del navegador y separándolas en varias consultas ejecutando una por una y mostrando los resultados finales hasta que la última sentencia sea ejecutada, los resultados finales se muestran cuando se realiza el análisis de qué páginas poseen más peso, más repeticiones de palabras clave o las que poseen el título que tenga más parecido a la palabra clave incluida en una sentencia. Se puede decir, que es el método que muestra información más precisa, debido a que por la forma en que está diseñado el proceso que muestra los datos permite formar listas de enlaces a páginas en donde los enlaces son mostrados de mayor a menor con relación a la prioridad que se les da a los mejores resultados, tomando en cuenta los criterios antes mencionados.

### 3.6 Presentación de resultados en archivos HTML

Los archivos de extensión HTML contienen un número de palabras clave que controlan la construcción del documento HTML de salida. Estos archivos, permiten construir de manera fácil diferentes formatos de salida para la visualización de la información, debido a su versatilidad para manejar información en grandes cantidades. Estos archivos, como cualquier lenguaje de programación posee palabras reservadas. Se hará referencia a las más importantes por tener relación con la base de datos donde se consulta la información.

Las palabras clave <%begindetail%>, <%enddetail%> rodean una sección del archivo de extensión HTML donde se combinarán los archivos de salida de la base de datos. Dentro de la sección, los nombres de columna delimitados por <% y %> o <!--%%-->se utilizan para marcar la posición de los datos devueltos por la consulta. Por ejemplo:

```
<%begindetail%>
<%campo_A%>:<%campo_B%>
<%enddetail%>
```

Presentará las columnas de datos que contengan Campo_A y Campo_B. Es posible hacer referencia a cualquier columna de este modo.

Una consideración muy importante de un archivo salida de resultados de base de datos es de que si la consulta no ha devuelto ningún registro, la sección <%begindetail%> se saltará. Por cada instrucción SQL que genere un conjunto de resultados (por ejemplo, SELECT), debería existir una sección <%begindetail%> <%enddetail%> correspondiente en el archivo .htx.

### 3.7 Sentencias <%if%>, <%else%>, <%endif%>

Los archivos de extensión pueden contener lógica condicional con una instrucción if-then-else para controlar la construcción de la página *Web*. Por ejemplo, una utilización habitual es insertar una condición para mostrar los resultados de la consulta en la primera fila, dentro de una sección <%begindetail%>; en el caso de que la consulta no devuelva ningún registro, se mostrará el texto "No hay ningún registro recuperado de %Campo_A%. Mediante la utilización de la instrucción <%if%> y una variable incorporada denominada "CurrentRecord" es posible personalizar el resultado para que se imprima el mensaje de error cuando no se devuelva ningún registro.

# 4. ANÁLISIS DE NUEVAS TÉCNICAS

El desarrollo de técnicas para la búsqueda de información en Internet va en aumento, la mayoría de desarrolladores *Web* involucran dentro de sus proyectos programas que les ayuden a obtener la información adecuada, basta con una simple conexión remota hacia un servidor de páginas o más bien dicho un servidor de Internet para que un programa desarrollado en lenguaje *Web* se conecte y así poder ejecutar consultas de información. La creciente población de programadores ha hecho cada día más fácil la creación de técnicas precisas para involucrarse en el área de la *Web*, estas técnicas han facilitado la creación de Intranets con potentes conexiones hacia WAN's, creación de portales con variedad de servicios, así como numerosos sitios que brindan los servicios de correo electrónico (*e-mail*), compra-venta de artículos y/o servicios (*e-commerce*), almacenamiento de información de tipo texto, sonido, imagen, a los cuales se les llama buzones, chat, envío y recepción de mensajes por medio de páginas, entre otros. Entre todos los servicios que se pueden acceder a través de la *Web* se puede encontrar búsqueda de información a través de los famosos motores de búsqueda (en inglés *Search Engines*) los cuales han sido desarrollados aplicando técnicas conocidas para la búsqueda de información, entre algunas de estas técnicas, se pueden mencionar búsquedas de palabras a través de árboles n-arios, también con la utilización de las estructuras de datos conocidas como tablas de HASH en la representación de índices, pero realmente las técnicas más utilizadas son las técnicas que emplean las estructuras de árbol, porque las búsquedas de palabras y de términos léxicos son analizados de una mejor manera.

Para definir algunas técnicas, es necesario definir antes los tipos de motores de búsqueda más conocidos.

## 4.1 Tipos de buscadores

Existen muchos tipos de buscadores, cada cual con sus ventajas y sus desventajas. El siguiente es un intento por clasificar los diferentes tipos existentes en la WWW.

### 4.1.1 Buscadores automáticos

Son aquellos cuya base de datos se construye automáticamente a medida que sus "robots' buscan información nueva en la red, incluso, buscando páginas dentro de sitios. Por ello, suelen estar muy actualizados. Están constituidos por tres partes: una base de datos, un motor de búsqueda y los robots. La desventaja del buscador automático es que acopia información sin agruparla por tema, y además, al darle más importancia a los sitios o páginas con mayor cantidad de ocurrencias, no puede valorar su relevancia.

Los buscadores automáticos se pueden identificar porque presentan un formulario donde el usuario introducirá la palabra buscada. Ejemplo: *Google.*

### 4.1.2 Buscadores temáticos

Son aquellos que han organizado previamente la información según el tema en categorías y subcategorías, lo cual permite restringir el campo de búsqueda a lo que a uno le interesa. No son automáticos en el sentido de que su base de datos se construye con sitios que son añadidos manualmente (por ejemplo, añadiendo sitios sugeridos por los mismos usuarios mediante la opción 'añadir *Web*').

Tal es la ventaja de estos buscadores al ser categorizados manualmente, ya que la información queda mejor organizada. Sin embargo, tienen una desventaja sobre los automáticos y es que las personas que construyen el sistema de categorías temáticas son más lentas que los robots y suelen estar menos actualizadas.

Los buscadores temáticos se pueden identificar porque en su página de inicio presentan una serie de categorías temáticas, como por ejemplo "arte y cultura", "educación", "salud", etc., y porque en los resultados presentan mayormente sitios y no sus páginas interiores. Ejemplo: *Google, Yahoo*, etc.

### 4.1.3 Buscadores generales

Buscan todo tipo de información, independientemente del tema tratado. Aquí coexistirán, por ejemplo, sitios religiosos con sitios pornográficos, o sitios de organismos gubernamentales con sitios personales. Ejemplos: *Google*, Altavista, etc.

### 4.1.4 Buscadores especializados

Buscan información en determinada área temática. Hay buscadores especializados en psicología, en informática, en pornografía, etc. Algunos sitios como www.telepolis.com, producen periódicamente los llamados 'monográficos', un listado organizado de sitios referidos a un determinado tema. Ejemplo: Psicobank (especializado en psicología).

### 4.1.5 Buscadores de buscadores

Son buscadores que ofrecen listas de buscadores. En algunos casos, los buscadores son agrupados por país, o por temática, o por idioma, etc. El buscador de buscadores sería como un buscador especializado, pero en buscadores. Ejemplos: Buscopio, etc.

### 4.1.6 Buscadores en buscadores

Son buscadores que no tienen una base propia de datos, y lo único que hacen es enviar una orden de búsqueda a una serie de buscadores previamente establecidos, siendo los resultados presentados en forma unificada. La desventaja de los metabuscadores es que arrojan mucha cantidad de información (con lo que se tarda más en buscarla) y quedan excluidos los enlaces menos repetidos, con lo que puede perderse información útil. Obviamente, la gran ventaja de los metabuscadores es que nos evitan tener que buscar información de buscador en buscador: ellos lo harán por nosotros. Ejemplos: *Metacrawler, Mamma*, etc.

### 4.1.7 Buscadores en catálogos

En lugar de buscar sitios *Web*, estos buscadores buscan libros o revistas impresas. Estos buscan dentro de catálogos y los encontraremos en sitios *Web* de bibliotecas públicas o privadas, o en sitios *Web* de editoriales o librerías. La mayor parte del material encontrado deberá ser consultado fuera de Internet (visitando una biblioteca o comprando un libro en una editorial), aunque existen sitios como www.elaleph.com, que permiten bajar algunos libros gratuitamente para leerlos en la PC.

Una manera de buscar bibliotecas virtuales es introducir, en cualquier buscador automático, las palabras 'biblioteca virtual'. Ejemplos: www.cervantesvirtual.com, www.biblioteca.org.ar, etc.

### 4.1.8 Buscadores internos

Son los que buscan información dentro de un sitio o bien dentro de una página del sitio. Con respecto a los primeros, algunos sitios, especialmente los grandes, han incorporado una herramienta de búsqueda dentro de su propio sitio; y respecto de los segundos, los navegadores incluyen una herramienta para buscar una palabra dentro de una página específica (la que en ese momento esté presentada en pantalla).

Como el lector habrá advertido, estos diferentes tipos de buscador pueden agruparse de acuerdo a criterios definidos. Por ejemplo: según que el banco de datos se construya automática o manualmente, se tendrán los buscadores automáticos y los temáticos; o según que busquen información sobre cualquier tema o sobre uno específico, se tendrán buscadores generales y especializados.

Un mismo buscador puede ser ubicado en varios tipos. Por ejemplo:

- Buscadores automáticos y temáticos
- Buscadores automáticos y especializados
- Buscadores temáticos y especializados

### 4.1.9 Buscadores automáticos y temáticos

Reúnen las ventajas de ambos tipos de buscadores y compensan sus desventajas. El *Yahoo*, por ejemplo, combina en una misma página un buscador temático propio con el buscador automático del *Google*. El *Google* cuenta con un buscador automático y otro temático, pero en diferentes páginas.

### 4.1.10 Buscadores automáticos y especializados

*Psicobank* es un buscador que reúne ambas características, por cuanto, cuenta con un robot de búsqueda automática en la red, al mismo tiempo que se especializa en psicología.

### 4.1.11 Buscadores temáticos y especializados

Por ejemplo un sitio de psicología que incluye una gran cantidad de enlaces a páginas de la misma temática puede ser considerado un buscador temático porque indexa o enlista sitios en forma manual y también un buscador especializado porque incluye solamente sitios de psicología.

Después de haber definido los tipos de motores de búsqueda se dará una breve explicación de algunas técnicas empleadas:

## 4.2 Selección de Información a través de SQL - Estructuras de datos

La capacidad y potente rendimiento del *Structured Query Language* ha sido utilizado como una herramienta infaltable en la construcción de los motores de búsqueda, debido a que es considerado como una de las piedras fundamentales de este tipo de programas, el flujo de información que está contenido en las bases de datos en Internet es manipulada por esta herramienta, pero se debe de considerar, que la cantidad de peticiones de búsqueda de información en un motor de búsqueda es excesivamente grande, por lo cual este lenguaje de manipulación de datos puede en algún momento saturarse; las estructuras de datos como auxiliares de este proceso permiten manipular este flujo de datos a través de las conexiones con la base de datos y la interface con el usuario, lo cual hace que el motor de búsqueda y el precompilador de búsquedas propio de la base de datos pueda tener más capacidad de respuesta ante las constantes peticiones.

Las estructuras de datos más empleadas para la presentación de datos en la página del buscador son los árboles binarios, debido a que estos TADS (Tipos Abstractos de Datos) permiten la manipulación de cualquier tipo de herencia de información que resulte en búsquedas que se consideran atrasadas y al mismo tiempo presenta la información en forma ordenada, pero demora un poco más en hacerlo, de cualquier modo estas estructuras alimentan las páginas de los usuarios finales.

Las listas dinámicas también son utilizadas juntamente con una estructura de datos variante de las tablas de HASH con la diferencia entre la manipulación de una y otra estructura, debido a que la información en la estructura similar a las tablas de HASH permanecerá ordenada con la carga de información inicial, pero cuando llegue información extra, ésta deberá realizar una actualización de sus índices, lo cual, la hace más lenta, en tanto que las listas dinámicas almacenan su información conforme ingresan.

## 4.3 Listas encadenadas

Regularmente, una estructura de datos muy utilizada para manejar los índices que se emplean en tablas de datos son las listas enlazadas de forma múltiple, éstas proporcionan al programador una verdadera estructura de datos versátil, la cual pueden manipular dependiendo de las características de los datos que se estén utilizando, este tipo de dato o estructura de dato, se puede implementar al momento de la creación de una página de resultados a partir de una búsqueda ejecutada; sin embargo, una página de resultados debe contener enlaces a información que se encuentre actualizada, de tal forma, los índices ayudan al proceso de búsqueda de información; debido a que al momento en que los índices se encuentren actualizados éstos mostrarán únicamente la información que realmente éste disponible, el proceso permitirá realizar consultas de forma rápida por la manipulación de una estructura de dato de este tipo, debido a que resulta muy sencillo.

## 4.4 Ventajas y desventajas de las técnicas utilizadas

Las técnicas actuales de búsqueda de información basan su potencial en la capacidad que tienen sus manejadores de bases de datos, sus proveedores del servicio de Internet en la mayoría de los casos, sin utilizar técnicas especiales de búsqueda de información más allá de las ya conocidas, de tal forma, que las desventajas existentes en su mayoría, más que las ventajas hacen un desbalance que fácilmente se puede determinar de la siguiente forma.

### 4.4.1 Buscadores automáticos y temáticos

Estos tipos de buscadores poseen gran habilidad para obtener información, pero en la mayoría de los casos desactualizada, realmente, es una debilidad que poseen los actuales motores de búsqueda y que podría ser cubierta y explotada al máximo en un futuro cercano, lo anterior, tomando en cuenta que la necesidad de información actualizada es de suma importancia; si realizamos el ejemplo con una agencia noticiosa y suponemos que este tipo de empresa necesita información sobre “la clonación” en ningún momento le serviría recabar información sobre este tema de un articulo que fue publicado al inicio de la década de los años noventa, realmente, esta empresa necesita información sobre el tema, de artículos publicados en días anteriores, es más talvez de artículos que han sido publicados en el mismo día. Es por ello que es de vital importancia que los enlaces a los índices de páginas se encuentren actualizados.

### 4.4.2 Buscadores de clasificación

Realizan la búsqueda de la información con gran precisión, pero el proceso de clasificar un tema o título es demasiado lento, lo que lo hace ser una desventaja muy marcada, esto, debido al potencial encontrado en estos momentos en el hardware disponible y que permite realizar millones de operaciones en milésimas de segundos. Si nos referimos a un portal de una institución noticiosa que cuenta con un motor de búsqueda propio, como el caso anterior y el centro de la búsqueda de información es “un desastre natural ocurrido en el océano Pacífico”, esta institución presta el servicio de actualización de información con una frecuencia de 5 minutos, entonces el problema de la demora en la búsqueda de la información podría ser una desventaja que afecte seriamente a esta empresa por no prestar el servicio como realmente se especifica.

Realmente, en el análisis del último caso, tiene más peso las desventajas que las ventajas, siendo las últimas no tomadas en cuenta si lo vemos desde el punto de vista de la optimización de las consultas, pero, por mencionar algunas de ellas, como el inicio de la centralización de la información mundial, que realmente fue el punto de partida para el desarrollo de aplicaciones *Web*, debido a que basaron todo su entusiasmo y empeño en la perfección de tan beneficioso proyecto, que a nuestros días sigue en un constante crecimiento, desarrollo y que promete ser una de las empresas más fuertes en el área del *Web*, por su facilidad para las comunicaciones y el traslado de información en cualquier formato.

## 4.5 Proposición de nueva técnica

### 4.5.1 Análisis de nueva técnica

El flujo de información que se presenta hoy día en la *Web* se debe de considerar como un elemento, el cual la sociedad está dispuesto a utilizarlo para mejorar sus tareas diarias, tanto académicas como laborales, este elemento de información ha sufrido una de las evoluciones más grandes y rápidas, con el constante crecimiento de los elementos de hardware y software de la prominente industria de las computadoras, la información ha pasado a formar parte del elemento más importante que se encuentra intrínseco en los dos elementos anteriormente mencionados, sin los procesos de clasificación, identificación, investigación de la información ningún proyecto de desarrollo de propuestas a cualquier nivel se hubiese podido realizar, y es más, en el futuro todas las empresas, organizaciones, instituciones, gobiernos que basan su funcionalidad y producción en la investigación de nuevos métodos, nuevas técnicas, encontrarán una gran barrera en su camino si no poseen herramientas que les ayuden a la investigación, control y clasificación de la información.

Durante la década de los años noventa se mencionó mucho sobre el elemento información, pero muy poco se dijo sobre la importancia que en un futuro pudiera tener el acceso a toda la información existente desde la comodidad de la casa o inclusive llevar consigo toda la información a través de un ordenador portátil y una conexión remota, antes todo este tipo de ideas eran únicamente eso, sólo ideas, pero ahora es una realidad en la cual nos encontramos, lo cual, brinda excelentes beneficios a todos los investigadores, pero, ahora necesitamos concentrar toda esa información en un lugar que sea capaz de organizarla y clasificarla para que el acceso a ella sea posible en el menor tiempo.

Para poder decidir realmente el camino que se debe tomar en función de lo que desea alcanzar y diseñar, es necesario que se examinen detenidamente los diferentes tipos de motores de búsqueda definidos anteriormente.

Se puede concluir que uno de los objetivos principales de la construcción de un motor de búsqueda, es agilizar la búsqueda de la información y así el tiempo de respuesta hacia el usuario, actualmente, existen herramientas que realizan el mismo proceso, pero con alguna deficiencia entre las que podemos mencionar rápidamente la lentitud del tiempo de respuesta, información no actualizada etc., para poder desarrollar una herramienta que sea cien por ciento funcional y con ventajas, como por ejemplo: más rapidez y exactitud debemos de analizar y proponer un esquema que nos ayude a mejorar esas situaciones, pudimos apreciar que entre los diferentes tipos de motores de búsqueda existen ventajas y desventajas entre unos y otros, es por ello que para unificar las ventajas y disminuir las desventajas debemos de presentar un modelo que se ajuste para cubrir esas deficiencias existentes, entre los cambios a implementar, se pueden mencionar los siguientes:

- Interfaz con el usuario simple pero intuitiva
- Unificación de métodos existentes
- Método de selección de información para agregarla en la base de datos
- Presentación de resultados al usuario
- El servidor *Web*
- El conector de base de datos en Internet
- Administración de información (utilización de índices, replicación, distribución, servidores clasificados)

#### 4.5.1.1 Interfaz con el usuario simple pero intuitiva

El objetivo de la construcción de una interfaz simple es de que el usuario pueda navegar en la búsqueda de información de una manera fácil, sin enfrentar obstáculos, como los que actualmente enfrentamos, la publicidad dentro de los navegadores es uno de las barreras más comunes, debido a que existen buscadores que contienen en su interfase de comunicación con el usuario, una serie de elementos que obstaculizan de gran forma la obtención de la información requerida, es por ello que la interfase del motor de búsqueda debe ser sencilla, pero, al mismo tiempo que le ayude al usuario a introducir y redactar de la mejor manera la consulta de requerimiento de información. Muchos de los motores de búsqueda almacenan en sus índices información de tipo publicidad, la cual nos lleva a sitios que únicamente nos ofrecen la venta de x o y determinado producto y/o servicio. Una consideración muy importante es la de implementar el motor de búsqueda de tal forma que el usuario pueda llegar a restringir y redactar su consulta de selección de la manera óptima.

**Figura 1. Diseño de la interfase del usuario**

#### 4.5.1.2 Unificación de métodos existentes

La clasificación de los diferentes tipos de motores de búsqueda sirvió para aclarar cuáles son los elementos que debemos tomar y cuáles se deben mejorar, esto, con el único objetivo de construir un motor de búsqueda que sea robusto ante flujo de información grande, como se pudo observar un motor de búsqueda automático, casi siempre, estará actualizado, lo cual nos garantiza que la información que nos devuelva es reciente y confiable pero su gran deficiencia es de que no clasifica la información, la almacena en el orden como los robots la van encontrando en la red, por su estructura lo hace un motor de búsqueda básicamente rápido pero se corre el riesgo de que la información que devuelva sea errónea, al contrario si se desea que la información que se obtenga sea precisa en un alto porcentaje se puede utilizar un motor de búsqueda temático, el cual nos ayuda a restringir y al mismo a tiempo a clasificar nuestra consulta de selección, debido a que nos guía por medio de "etiquetas", pero el problema de este tipo de motor de búsqueda es el tiempo que empleamos en realizar una consulta para la obtención de información, así se puede seguir mencionando las ventajas y desventajas de los distintos tipos de motor de búsqueda que existen (que de hecho se mencionarán más adelante), la idea es tratar de complementar todas estas ventajas y unirlas en un solo motor de búsqueda, el cual tenga a su vez, calidad en la información obtenida, rapidez para encontrarla, redacción de la consulta fácil y rápida para ello se debe implementar una interfase fácil de navegar, facilidad para el traslado de información de la *Web* hacia cualquier dispositivo de almacenamiento, entre otras.

#### 4.5.1.3 Método de selección de información

Este proceso de selección de información es aquel que se lleva a cabo para introducir páginas de información a la base de datos, este proceso rastrea periódicamente todas las páginas nuevas y las incluye en su base de datos (robots), clasificándola, según su definición primaria, básicamente esta clasificación determina el grupo en el cual aparecerá la página cuando se ejecute una búsqueda, existe información que es importante y que se introduce conjuntamente cuando se da de alta la(s) página(s), ésta puede ser, la clasificación de la información, el título de la página, a quién va dirigida la información que se incluye en esa página, palabra(s) principal(es) (*metatags*), grupo primaria en donde se desea que aparezca la página por ejemplo, computadoras, agricultura, deportes, etc., también podemos mencionar el peso de los *metatags*, es decir, la cantidad de veces que se repite en la página la palabra clave.

#### 4.5.1.4 Presentación de resultados ordenados

Al ejecutar una consulta, lo que se desea es que cuando se presente la información muestre los mejores resultados obtenidos, como ya sabemos, esto se logra en un alto grado por medio de la buena redacción de la consulta, también deseamos que muestre los resultados ordenados por prioridad de mayor a menor, es decir, que muestre primero los resultados que mayor grado de coincidencia tenga con la información solicitada.

Otro aspecto que nos importa muchas veces o en la mayoría de veces, es que no nos muestre enlaces a páginas de publicidad; existen muchos motores de búsqueda que subsisten por medio de la publicidad, debido a que es por ese medio que obtienen recursos para mantener el proyecto, pero también existen otros motores de búsqueda que son subsidiados por organizaciones internacionales, universidades entre otras, que suelen ser los más efectivos.

Las definiciones de los distintos tipos de motores de búsqueda que existen, así como los cambios que se han mencionado en relación a lo actual nos guía a la proposición de un modelo de motor de búsqueda que contenga elementos que sean capaces de sobrepasar los límites y/o barreras que actualmente existen en relación a esta herramienta que, hoy por hoy, es una de las principales en la obtención de información de cualquier tipo, clasificación, etc.

## 4.6 El servidor *WEB*

Para poder emplear todos los servicios de Internet necesitamos utilizar un servidor que sea capaz de prestar a nuestra aplicación los recursos tales como servicio de URL, HTTP, correo electrónico, almacenamiento de información, servicios de WWW, etc. La primera versión de la WWW se presentó sobre una máquina NeXT y tuvo la capacidad de inspeccionar y transmitir documentos de hipertexto. El hipertexto, se refiere al texto que contiene vínculos (*hyperlink*) a otros documentos. Dichos documentos pueden estar en la misma computadora o en cualquier otra que se encuentre conectada a la red, sin importar su situación geográfica.

Un vínculo (*hyperlink*) se puede definir como "*Get the address asociated with this link and go there*", obtiene la dirección asociada a este vínculo y ve a ella. La *World Wide Web*, se define oficialmente como una "iniciativa global de recuperación de información hipermedia con acceso universal al inmenso conjunto de documentos en Internet". Lo que el proyecto *World Wide Web* ha hecho, es proveer a los usuarios de las redes de computadoras el acceso a la información a través de un medio uniforme de manera simplificada. Lo anterior, significa que, después de varios intentos, en Internet surge un programa de fácil manejo que puede obtener información de cualquier computadora conectada a la red.

Las primeras visiones de los sistemas como la WWW tuvieron como meta el adelanto de la ciencia y la educación, aunque el proyecto *World Wide Web* tiene la potencialidad para generar un impacto importante en el comercio, la política y la sociedad.

Hasta hace algunos años el uso de Internet se encontraba en manos de los expertos, dada la cantidad de conceptos y comandos que el usuario debía conocer para poder entrar al mundo cibernético. En los últimos años, los expertos comenzaron a desarrollar sistemas que pudieran ser usados por personas con pocos conocimientos y experiencia en sistemas de cómputo. El servicio HTTP y *World Wide Web* facilitó que las personas con pocos conocimientos sobre computación pudieran utilizar la red, sin realizar un esfuerzo mayor en el aprendizaje de comandos y/o conceptos altamente complejos que le ayudarán a manipular la super red.

El servicio de correo electrónico es un herramienta poderosa, debido a que por medio de él se puede transferir gran cantidad de información, con la simple indicación del destino al cual se quiere llegar, a través de un servidor principal el cual se encarga de administrar la información.

## 4.7 Localizadores uniformes de recursos (URL)

El URL contiene los segmentos de información que un navegador necesita para localizar una página *Web*. Este busca en un URL para encontrar una página principal. La página principal es la página *Web* primaria (index.HTML), que sirve como punto de partida.

HTTP indica al navegador que utilice el protocolo de transferencia de hipertexto para obtener una página *Web*. Si la primera parte del URL fuera FTP, iniciales del protocolo de transferencia de archivos, eso indicaría al navegador que solicite una conexión a un sitio FTP, de tal forma que, la primer parte del URL debe especificar el tipo de protocolo o servicio a utilizar.

El “nombre.dominio” corresponde al nombre de la computadora y al dominio correspondiente, para este caso, el servidor de HTTP se nombra como “nombre” que pertenece al dominio “dominio”, este nombre identifica de manera única una máquina en Internet y una vez que el navegador tiene el URL, éste sabe en donde se encuentra exactamente la página *Web*.

Existen direcciones que incluyen la tilde (~) la cual indica que es el directorio base del usuario. El archivo index.HTML, indica al navegador el nombre del archivo a buscar, la extensión .HTML, indica que es un documento de hipertexto.

En muchos de los servidores *Web*, el index.HTML es el nombre del archivo predeterminado a buscar, incluso si el URL no tiene el nombre de la página principal, éste automáticamente busca y visualiza el archivo index.HTML.

En sistemas Unix como en muchos otros, no es necesario que la extensión de las páginas *Web* sea .HTML, por el número de caracteres de la extensión, éstas pueden ser extensiones HTM.

Para que un motor de búsqueda pueda visualizar una página *Web* necesita el URL o en su defecto una dirección IP, el URL, es la forma de representar direcciones para que un usuario pueda recordar fácilmente una dirección Internet, una dirección Internet, es una dirección IP. Ya hemos hablado de la función de los servidores DNS, que prácticamente tienen la función de asociar direcciones URL a direcciones IP.

### 4.8 El conector de base de datos en Internet (IDC)

*Internet Database Connector* (IDC), es un componente integral de un servidor de Internet, para nuestro caso de estudio, podemos mencionar el *Internet Information Server*. Originalmente, fue diseñado para programadores familiarizados con SQL y con poca experiencia con el lenguaje HTML. El archivo IDC ofrece un mecanismo directo de alto rendimiento para la integración del contenido de una base de datos dentro de una página *Web*, la cual regularmente la realiza el motor de búsqueda.

Una aplicación IDC consiste en dos documentos: uno que contiene la información de la consulta y otro es un archivo HTML estándar con una sintaxis especial para hacer referencia a los resultados de una consulta.

Cada vez que un usuario hace una requisición a un archivo IDC, la consulta asociada con él se ejecuta como un programa DLL/ISAPI y se comunica a la base de datos SQL a través de ODBC.

Puesto que IDC utiliza el lenguaje de consultas SQL, el modelo de programación es familiar a todos los programadores de bases de datos y es significativamente fácil de implantar.

Un beneficio adicional de usar consultas SQL estándar, es la facilidad para insertar información en la base de datos, así como la recuperación de la misma. Más aún, un programador puede elegir usar procedimientos almacenados, incrementando significativamente la eficiencia de la programación y ejecución. Realmente para la construcción de un motor de búsqueda es imprescindible la utilización de un conector de base de datos en Internet.

### 4.8.1 Características y funciones

El IDC, en combinación con el servicio WWW que mencionamos anteriormente y los dispositivos ODBC provistos con el servidor (IIS), posibilita crear páginas *Web* con información contenida en una base de datos, así como también, insertar, actualizar y borrar información en la misma, según los datos proporcionados por el usuario en un formulario HTML con los métodos GET o POST. Además, se pueden ejecutar otros comando SQL sobre la base de datos.

IDC provee al programador conocedor de HTML y SQL, un mayor control para dos cosas: qué información será recuperada de una base de datos y cómo esa información será presentada al usuario.

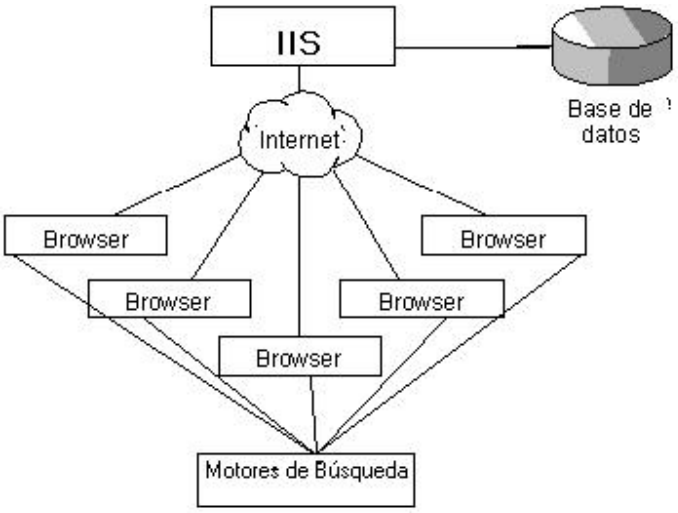
Actualmente, existen algunas herramientas que facilitan al programador generar automáticamente aplicaciones IDC y publicar datos de un servidor SQL, así como de cualquier bases de datos ODBC.

Conceptualmente, el acceso a las bases de datos es realizado por el servidor (IIS) como se muestra en siguiente figura. Los clientes del navegador, en este caso, los motores de búsqueda envían las requisiciones al servidor *Web* a través de HTTP.

El servidor *Web* responde con un documento en formato HTML. El acceso a una base de datos es realizado por el componente IDC del servidor (IIS). El IDC, es un programa DLL/ISAPI (httpodbc.dll) que utiliza ODBC para el acceso a las bases de datos.

Estamos utilizando IIS como servidor estándar para la administración de los servicios *Web*, pero de cualquier otra forma se puede utilizar cualquier otro servidor, como por ejemplo, Apache.

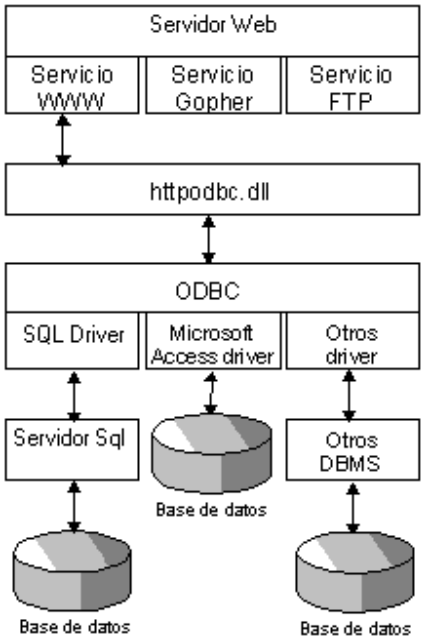
**Figura 2. Acceso a base de datos a través del servidor: IIS**




Fuente: Bases de datos en Internet Forum.

El IDC usa dos tipos de archivos para controlar: cómo la base de datos es accedida y cómo se construye la página *Web* de resultados. En esta parte la construcción de un motor de búsqueda es de vital importancia, debido a que es aquí en donde se pueden definir los procedimientos que agilicen la presentación de información al usuario. Estos archivos son: IDC (.idc) y extensión HTML (.htx).

**Figura 3. Componentes de IDC para conectarse a una base de datos con IIS.**




Fuente: Bases de datos en Internet Forum.

### 4.8.2 Administración de información

Con los aspectos técnicos antes mencionados, podemos acercarnos a la definición de la administración de la información, ésta tiene que ver con todo el proceso de su manipulación, desde que se comienza a buscar hasta que el usuario la tenga en su poder.

### 4.8.3 Creación y utilización de índices

Los índices como ya vimos, son aquellas estructuras de datos que nos ayudan a clasificar, buscar la información de manera sencilla y sobre todo rápida, estos elementos nos pueden ayudar dependiendo del enfoque que se les dé para poseer actualizados nuestros enlaces de información. Podemos tener índices por títulos, por palabras claves, por autor, por tema, por idioma, por región, etc, es decir podemos tener índices por cualquiera de los campos anteriores, dependiendo del enfoque que tenga nuestro buscador.

## 4.9 Ventajas y desventajas

El proceso de optimizar las consultas de información en un motor de búsqueda conlleva a realizar cambios bastante radicales en los motores de búsqueda actuales, siendo uno de ellos, la utilización de índices en la base de datos o en las bases de datos que se utilicen para ejecutar la consulta, este proceso, realmente ayuda de manera considerable a la optimización de consultas siendo una de ellas la ventaja más sobresaliente, lo anterior, nos encaminaría a la presentación de otra ventaja como sería la obtención de información actualizada, debido a que si podemos construir índices con los distintos campos y un algoritmo que sea capaz de manipularlos eficientemente podemos llegar a optimizar en un buen porcentaje la selección de información.

Por otro lado, podemos mencionar las desventajas que encontramos utilizando el proceso anterior de creación de índices, y uno de los más significativos sería que al momento que nuestro robot realice la operación de rastrear la red para verificar la información nueva que existe, debería realizar un proceso similar con cada uno de nuestros índices, debido a que debería ir a actualizar dependiendo de la cantidad de campos que se encuentren indizados.

Otra desventaja de gran importancia, pero que se podría considerar “no tan relevante”, es el proceso de actualización de información, éste debería de tomar una cantidad de tiempo considerable dependiendo de la cantidad de campos indizados que encuentre, el ingreso de información hacia la base de datos, sería lenta por lo anterior expuesto, pero debemos de tomar en cuenta que realmente nuestro enfoque está orientado hacia la optimización de las consultas, de modo que si el proceso de ingreso de información se demora más de lo que comúnmente se realiza, no nos afectaría de manera directa en nuestro proceso de agilización de búsqueda de información.

# 5. CONSTRUCCIÓN MOTOR DE BÚSQUEDA

## 5.1 Definición del nuevo motor de búsqueda

El objetivo fundamental de este proyecto es la definición de las distintas etapas que forman parte de la construcción de un motor de búsqueda, y en este caso en particular los conceptos nuevos que se adhieren a los ya existentes y que servirán no sólo para definir una herramienta de este tipo, sino, un software que sea innovador, confiable, seguro y rápido en la obtención de información de la *World Wide Web*; debido al gran alcance y popularidad que han mostrado los usuarios de la red internacional hacia software de esta naturaleza. Explícitamente podemos identificar, que la mayoría de herramientas que prestan este servicio, cuentan con ciertas debilidades las cuales se van a aprovechar para mostrar las ventajas que se pueden brindar aun en esta área. Un análisis de vital importancia que se debe hacer antes de pasar a definir puramente aspectos conceptuales y técnicos es el análisis FODA, el cual nos mostrará las fortalezas del software a desarrollar, las oportunidades que tienen en el mercado en este momento, las debilidades que deben de tratar de disminuir al máximo y las amenazas con las cuales se debe tener cuidado, debido a que si no se saben guiar pueden representar graves problemas para nuestro proyecto.

## 5.2 Análisis FODA

Fortalezas

1. Conocimiento amplio de las reglas del negocio
2. Identificación de las etapas de desarrollo
3. Conocimientos de herramientas
4. Accesibilidad a herramientas de desarrollo de software específico

Oportunidades

1. Introducir el software al mercado utilizando las innovaciones
2. Competir con software ya existente
3. Introducir el software para uso común

Debilidades

1. Poca experiencia en el desarrollo de este tipo de software

Amenazas

1. Software de este tipo con innovaciones similares
2. Segmento del mercado al cual va dirigida la herramienta con poco interés en utilizarla.

Después de haber definido el análisis FODA, el paso siguiente es especificar y definir a detalle los aspectos básicos y fundamentales del nuevo motor de búsqueda.

El nuevo motor de búsqueda está orientado a buscar información en los sitios de Internet. El novedoso sistema del motor de búsqueda permitirá clasificar la información encontrada a través de diversas formas, las cuales se detallarán más adelante.

Como se sabe Internet es un medio de comunicación, en donde es muy fácil leer información utilizando un navegador y un motor de búsqueda convencional, pero en varias ocasiones resulta muy engorroso buscar información ya que los motores de búsqueda actuales presentan la información confusa, desactualizada o simplemente no la encuentran. En algunos casos el proceso es costoso en tiempo y dinero y suele suceder que la información sí se encuentra disponible pero debido al diseño ya sea de búsqueda, clasificación o presentación de los datos no se pueda acceder. La solución que se propone para construir una herramienta que llene todos esos espacios está basada realmente en varios pilares técnicos que ayudan a cumplir el objetivo.

Se puede definir el motor de búsqueda con la siguiente ecuación:

Nuevo motor de búsqueda = información actualizada + rapidez + funcionalidad

En donde la información actualizada es el elemento central del proceso del motor, la funcionalidad se refiere a la posibilidad de acceso que tengan los usuarios al motor y la versatilidad para interactuar con los mismos, y por último la rapidez que es parte fundamental de la optimización de la consulta de información.

### 5.3 Beneficios del nuevo motor de búsqueda

El motor de búsqueda como software administrador de información en Internet cuenta con las siguientes beneficios y características técnicas.

- El motor de búsqueda contará con una base de datos para clasificar toda la información necesaria para construir su propia bitácora de operaciones y de índices actualizados.

- El motor de búsqueda será compatible con todo tipo de servidor de Internet sin importar su tecnología de hardware o plataforma de operación.

- El motor de búsqueda indexa no solamente enlaces a páginas *Web*, por medio de palabras que hayan servido como *metatags*, sino también a documentos PDF, *Word* y hojas de *Excel*, permitiendo a los usuarios especificar la búsqueda por tipo de archivos.

- El motor de búsqueda es capaz de detectar automáticamente el lenguaje de cada ítem.

- El motor de búsqueda permite que cualquier cantidad de usuarios trabajen de forma concurrente consultando información en las bases de datos de Internet.

- La Integridad de los enlaces es garantizada por el motor de búsqueda a través de su índice, esto quiere decir que el método que realiza el rastreo de sitios, deberá identificar los sitios vigentes y los que han sido dados de baja; con esto se garantiza que nunca habrán punteros internos perdidos hacia enlaces de páginas.

- En la definición de los estándares de desarrollo del motor de búsqueda, se debe de integrar todo el proceso con una estructura que permita unificar las fases de construcción, de tal forma se ha analizado incorporar plantillas de HTML y XML inmerso, todo este proceso técnico se detalla posteriormente.

- Reconoce la gramática de las palabras en el sentido que encontrará las páginas sin importar en qué forma gramatical estén las palabras que busca el usuario.

- Permite desplegar los resultados ordenados por relevancia o fecha, por idioma español o inglés, y se puede filtrar por rangos de fechas.

- La página de los resultados es totalmente configurable a nivel visual, es decir que es posible ordenar los resultados por fecha, por peso de *metatags*, y alfabéticamente, entre otros.

## 5.4 Descripción de las fases de desarrollo

La construcción del motor de búsqueda es un proyecto de desarrollo de software el cual debe estar documentado con todos sus procesos y funcionalidades, la separación de tareas y fases de desarrollo, es importante para delimitar los alcances que se desean obtener así como los logros que al final se obtengan.

Las fases identificadas de desarrollo son:

- Análisis de los procesos principales, reglas, así como la definición de alcances, límites del sistema de búsqueda.

- Diseño de la base de datos, forma de conexión propia del motor de búsqueda con las otras bases de datos, el protocolo que permita la conexión y la herramienta a utilizar.

- Diseño de interfase para la interacción usuario - motor de búsqueda.

- Procesos de creación de índices por parte del motor de búsqueda para optimización de consultas.

- Implementación de método de consulta: “consultas múltiples – consultas por lotes”.

- Proceso de búsqueda por tipo de documento.

- Desarrollo de página de presentación de datos, esto incluye las innovaciones de personalización de presentación de información.

### 5.4.1 Análisis de los procesos principales

El objetivo del motor de búsqueda no se limita a buscar información en varias bases de datos en Internet; como se mencionó en el capítulo uno, el objetivo primordial de este sistema es buscar información en varias bases de datos en el menor tiempo posible y con un porcentaje alto de efectividad.

#### 5.4.1.1 Procesos principales

a. Captura de datos: la captura de datos para el motor de búsqueda representa el punto de partida para dar inicio a todos los procedimientos del sistema. Este proceso se llevará a cabo por medio de la página de presentación del buscador la cual permite el ingreso de la información que el usuario necesita obtener, como se mencionó en el capítulo tres, la información introducida por el usuario será transformada en el sentido que si lo que el usuario introduce es una frase, ésta será descompuesta en su unidad comprensible más pequeña (palabra) para ser analizada por el precompilador de consultas del sistema, esto permitirá que la cantidad de información resultante sea más completa y atinada de acuerdo al sentido de la búsqueda.

   En el capítulo anterior se definió que la página de captura de información para el motor de búsqueda debería ser simple pero intuitiva, es decir que no debía contener distractores por ejemplo de publicidad que desviaran la atención para la redacción de una buena consulta, al contrario la página de inicio debería de tener la menor cantidad de elementos, e inducir al usuario a construir consultas inteligentes en donde se obligue al motor de búsqueda a devolver resultados congruentes con la información solicitada, por ejemplo, si alguien busca información sobre el Síndrome de Inmuno Deficiencia Humana, podría redactar su consulta de la siguiente forma:

- Sida & HIV & Síndrome de Inmuno Deficiencia Humana

o la podría redactar así:

- “Sida” and “HIV” and “Síndrome de Inmuno Deficiencia Humana”

esto dependiendo de la sintaxis definida del motor de búsqueda.

b. Interpretación de la información: el motor de búsqueda posee un algoritmo que permite separar las palabras de una frase, para posteriormente agregarlas a una cláusula se SQL e introducirla a un archivo por lotes para su posterior ejecución en el motor de la base de datos. Este algoritmo comúnmente llamado *scaner* (Rastreador en su traducción al español), también es el encargado de crear los archivos por lotes los cuales se mencionaron en el capítulo tres para ejecutar posteriormente las consultas.

c. Alcance y límites: el motor de búsqueda así como todo sistema de software tiene definidos sus alcances y límites, los cuales servirán para indicar qué es lo que se puede esperar de él. El motor de búsqueda proporcionará toda la información encontrada en su índice de enlaces a sitios que contengan relación con lo solicitado, y también mostrará lo encontrado en las bases de datos visitadas.

#### 5.4.1.2 Diseño de la base de datos

Esta fase permite definir algunas de las tablas principales y los campos que contendrá la base de datos para almacenar la información.

A continuación se muestran los campos que contienen algunas tablas de la base de datos:

Tabla Enlace (
cod_enlace char(20)
nombre char(100)
enlace char(100)
cod_clasificacion char(50)
titulo char(100)
nombre_sitio char(100)
metatag char(200)
peso long
cod_idioma char(30)
consultas long
cod_database char(50))

Tabla Tipo_Clasificacion (
cod_clasificacion char(50)
Descripción char(100)
restriccion char(100) )

Tabla Idioma (
cod_idioma char(20)
Descripción  char(50))

Tabla Error (
cod_error char(50)
Descripción char(100) )

Tabla Database (
cod_database char(50)
Descripción char(100) )

#### 5.4.1.2.1 Descripción de tablas

- Tabla enlace: tabla fundamental de la base de datos, es en esta tabla en donde se lleva mayor parte de información, ésta será consultada todo el tiempo en el que el motor de búsqueda se encuentre en línea, es decir esta tabla es la que mayor movimiento tendrá. La mayor parte de los campos tienen un nombre que describe claramente la información que almacenarán, el campo *metatags* podría dar lugar a confusión, pero es en ese campo en donde se almacenarán las palabras claves que se han utilizado para hacer referencia a este enlace.

Cod_enlace: cada enlace (link) poseerá un código por ejemplo: “msginfobox+medicina”, el cual será único y representa la llave primaria.

Nombre: identifica al enlace por su nombre, por ejemplo: “noticias.com”

Enlace: se refiere a la dirección física del enlace es decir la ubicación que se puede observar a través de HTTP.

Cod_clasificacion: identifica al enlace por la categoría o grupo al cual pertenece, ejemplo “educación”, “tecnología”, etc.

Titulo: muchas veces el título se puede confundir con el nombre, pero un muchos casos las páginas tienen separado el nombre del sitio y el título del sitio.

*Metatags*: como ya se mencionó almacena las palabras clave que se utilizan para hacer referencia a la página.

Peso: campo importante que almacena el peso de la última consulta realizada en relación a un *metatag*.

Cod_idioma: almacena el idioma de la página.

Consultas: almacena la cantidad de veces que ha sido consultada la página.

- Tabla Tipo_Clasificación: esta tabla permite almacenar los distintos tipos de clasificación de sitios en Internet, como por ejemplo, educación, deportes, tecnología, computadoras, etc.

Cod_clasificacion: almacena el código de la clasificación.

Descripción: almacena la descripción del sitio o página.

Restricción: Identifica hacia quien está dirigido el enlace, es decir si es para niños, jóvenes, adultos, etc.

- Tabla idioma: en esta tabla se almacenan los idiomas de las páginas.

  Cod_idioma: almacena el código del idioma por ejemplo: SPA

  Descripción: identifica el código del idioma, por medio de su nombre largo: ESPAÑOL.

- Tabla Error: tabla que lleva el control de los distintos errores que pueden suceder al tratar de acceder a un enlace.

  Cod_error: lleva el control del código de error, “PB”

  Descripción: identifica el código de error, “Página de Baja”

- Tabla Database: tabla que permite llevar el control de las diferentes bases de datos a las cuales se conecta el motor de búsqueda.

  Cod_database: lleva el control de los códigos de las bases de datos.

  Nombre: lleva el control de los nombres de las bases de datos.

Descripción: define a manera de detalle los componentes de la base de datos.

## 5.5 El protocolo

Como se describió en capítulos anteriores, el protocolo universal de comunicaciones el TCP por su siglas en inglés *Transfer Control Protocol* (Protocolo de control de transferencia), proporciona la amplia capacidad de configurar la mayor cantidad de objetos de conexión de base de datos con la seguridad de la estandarización de su funcionamiento en cualquier ambiente.

### 5.5.1 Herramienta

La gran cantidad de lenguajes 4GL proporcionan la facilidad de generar aplicaciones fácilmente, así como también de generar diagramas de flujo, de modelo, proceso de negocio, casos de uso, etc.

### 5.5.2 Servidor de Internet

Parte importante del desarrollo del proyecto es definir el software que permita resolver los servicios de Internet, como lo son WWW., correo electrónico, transferencia de archivos, HTTP, entre otros. Como caso de estudio se puede mencionar el *Internet Information Server* que permite activar estos servicios así como también ISAPI. Otra de las innovaciones que el motor de búsqueda proporcionará es contar con su propio servidor de aplicaciones interno el cual será un sistema distribuido de cliente servidor.

## 5.6 Diseño de interfase interacción usuario - motor de búsqueda

Para que el sistema de búsqueda pueda interactuar con el usuario es necesario una vía de comunicación y esa es la página principal del motor de búsqueda.

Como se definió anteriormente, la introducción al negocio de Internet de XML beneficiará en gran parte a la construcción de esta vía de interacción.

## 5.7 Procesos de creación de índices por el motor de búsqueda

La utilización de índices en la base de datos del motor de búsqueda servirá para optimizar los tiempos de consulta de información, los índices que se crearán estarán orientados a las tablas más utilizadas y que por consiguiente tienen el mayor número de registros, lo cual hace necesario implementar algunos índices. Ahora la pregunta es por qué utilizar mas de un índice en alguna de las tablas del motor de búsqueda? y la respuesta es sencilla, como se mencionó en el capítulo tres, para la utilización de archivos para la construcción de consultas por lotes y múltiples, se hacia necesario la implementación de índices que permitieran incrementar el tiempo de respuesta de las peticiones, pero surge una pregunta muy importante: ¿si se utilizan varios índices en una o en varias tablas, el proceso de inserción de nuevos enlaces a la base de datos producirá un retardo mayor debido a que tiene que actualizar todos los índices para su utilización posterior? , la respuesta a esa pregunta es que sí, en efecto, agregar nuevos enlaces a la base de datos será un proceso más tardado, pero la justificación más importante y de mayor peso es que la relación consulta-inserción es de : 10000 – 1, y es por ello que no se interesa tanto en el proceso de agregar enlaces.

## 5.8 Implementación: consultas múltiples – consultas por lotes

La definición de este método se encuentra ampliamente explicado en el capítulo tres, en este apartado se dará una introducción técnica de la creación de los archivos y su estructura interna.

Encabezado. Nombre del archivo <Consulta "*metatags*">

Cláusula 1.
Select <*metatags*> from
Nombre_conexión.administrador.tabla
Prioridad [multiple,por lotes]

Cláusula 2.
Select <*metatags*> from
Nombre_conexión.administrador.tabla
Prioridad [multiple,por lotes]

Cláusula 3.
Select <*metatags*> from
Nombre_conexión.administrador.tabla
Prioridad [multiple,por lotes]
.
.
Cláusula N.
Select <*metatags*> from
Nombre_conexión.administrador.tabla
Prioridad [multiple,por lotes]
Fin_archivo.

Es posible que un archivo por lotes contenga N cláusulas SQL, en donde se encuentran separados cada uno de los *metatags* que el usuario introdujo. La prioridad que se indica en la parte final de la cláusula es tomada en cuenta por el precompilador de consultas, el cual separa cada una de estas dependiendo si es múltiple o por lotes la consulta. Se realiza de esta forma debido a que el método trata de unificar los dos conceptos.

## 5.9 Proceso de búsqueda por tipo de documento

Una de la innovaciones más representativas de este motor de búsqueda es la capacidad de buscar información por medio de su formato, esta característica del motor de búsqueda permitirá realizar con gran éxito las consultas que se definan como "Documentos de.....". La mayor parte de motores de búsqueda obtienen información sin importar el formato del archivo. Este proceso de búsqueda está señalado en la interfase con el usuario, en donde se indica que se desea hacer una consulta por tipo de documento, simplemente se ingresa el/los *metatag*(s) seguido de paréntesis, dentro de los cuales se describirá el tipo de extensión, por ejemplo.

- Cultura Maya (.doc)

## 5.10 Página de presentación de resultados

La página de presentación de resultados muestra las características básicas para la buena interpretación de la información presentada, en el encabezado existe un espacio que es donde el usuario puede visualizar las palabras claves de la consulta que realizó, también se muestra las estadísticas de la cantidad total de enlaces a documentos encontrados y el tiempo que tardó en realizar la consulta.

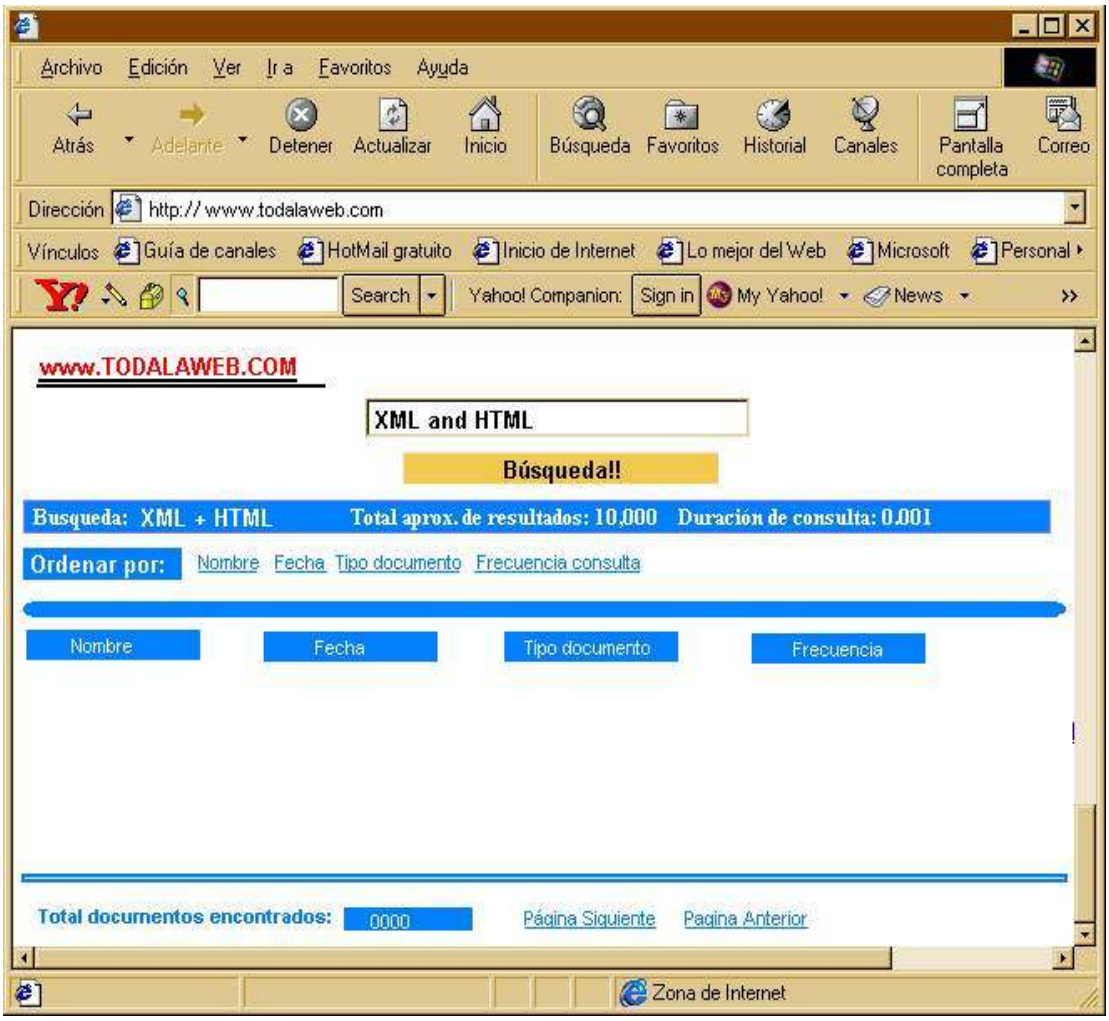
Los enlaces encontrados aparecen en el detalle con las siguientes columnas Nombre, Fecha, Tipo Documento y Frecuencia; en donde:

Nombre: indica el nombre o título del enlace
Fecha: la fecha de publicación del documento.
Tipo documento: indica el formato en el cual está creado el documento.
Frecuencia: indica la popularidad del sitio, es decir cuántas veces ha sido visitado.

En la parte superior a los títulos del detalle se encuentra el texto “Ordenar por” que indica de qué forma se quiere ordenar los enlaces encontrados y presenta las mismas opciones de las columnas del detalle.

La parte del pie de página se encuentran de nuevo la cantidad de documentos encontrados y también aparecen los enlaces “Página siguiente” y “Página anterior”, estos enlaces se habilitan dependiendo de la cantidad de resultados que se encuentren, la capacidad máxima de resultados presentada por página es de quince, esto con el objetivo de que los usuarios puedan visualizar correctamente la información sin tendencia a confusión por excesiva información presentada.

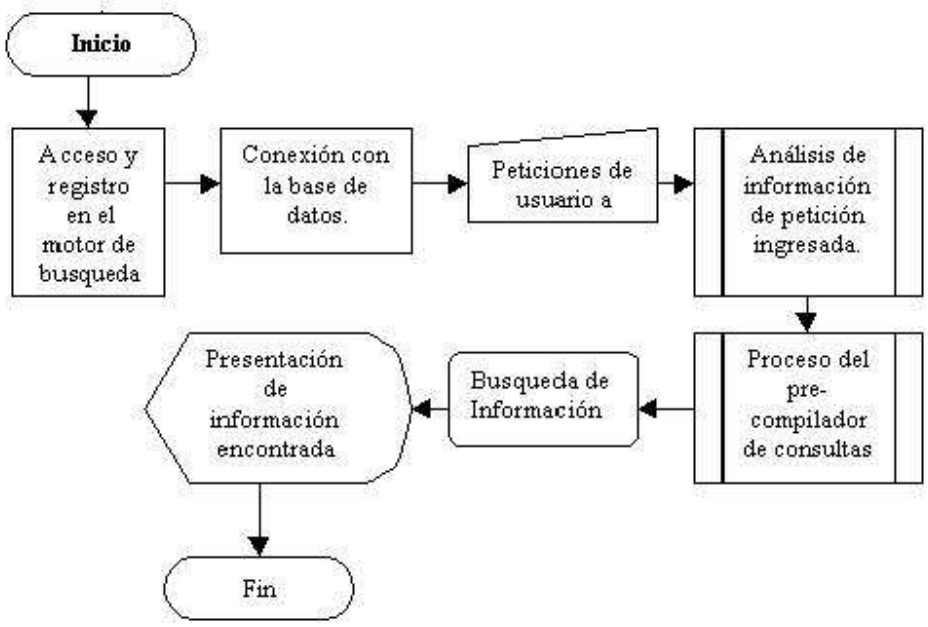
**Figura 4. Página de resultados del motor de búsqueda**



## 5.11 Modelo físico del motor de búsqueda

El objetivo principal del nuevo motor de búsqueda es optimizar las consultas en las bases de datos en Internet, este concepto abarca varias áreas dentro de las cuales existen otro gran número de sub-áreas que ayudan a conformar un diseño físico en el que se puede observar fácilmente el flujo de información, los procesos que se ejecutan, la interacción con los usuarios, las reglas que controlan todo el sistema de búsqueda, y también todos aquellos mini - procesos técnicos que ejecutan funciones específicas para el buen desempeño del sistema.

**Figura 5. Modelo de flujo de información (diagrama general)**



El diagrama anterior muestra la forma en que se lleva a cabo el flujo de información desde que el usuario se conecta con el navegador y realiza la petición de enlace con la WWW, el motor de búsqueda se conecta con la base de datos, el usuario ejecuta una consulta, realiza la petición de información, luego el motor de búsqueda analiza la petición señalada, el precompilador de consultas separa las palabras claves ingresadas, seguidamente se realiza el proceso de consulta de información a las base de datos en Internet, al momento de encontrarse o no la información se muestra una página de resultados, fin del los procesos.

### 5.11.1 Formato de archivos CGI

El formato de los archivos dbCGI es una extensión del formato estándar HTML, el cual soporta las etiquetas especiales <SQL > y </SQL >; así como también, especificaciones de formatos que comienzan con el carácter %.

DbCGI trabaja con ciertos subcomandos SQL para ejecutar acciones sobre base de datos, los cuales son de la forma:

<SQL subcomando>

PARAM1=valor1

PARAM2=valor2

….

PARAMn=Valorn

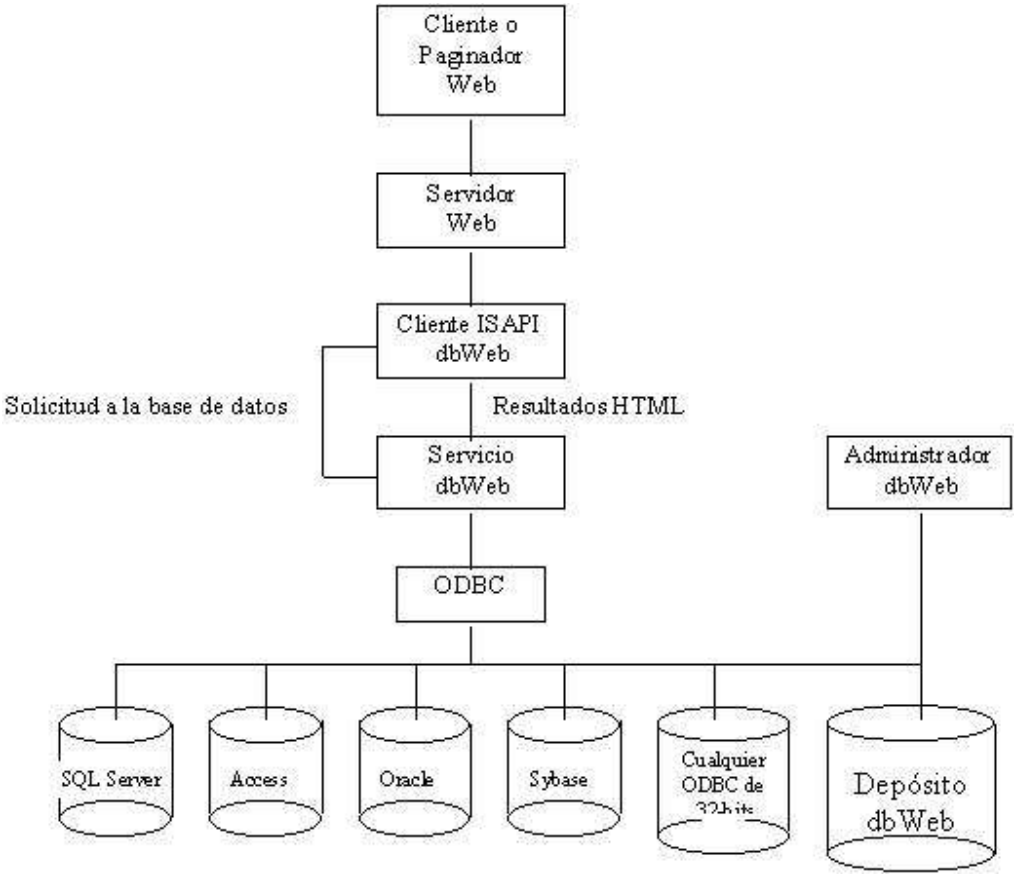
</SQL>

Cada DBMS puede reconocer diferentes parámetros para cada uno de los subcomandos SQL. Si se emplea un parámetro no reconocido por el DBMS en un subcomando, el parámetro será ignorado.

Los subcomandos que reconoce dbCGI son los siguientes:

Init: inicia el DBMS
Ununit: termina el DBMS
Connect: establece la conexión con la base de datos.
Disconnect: termina la conexión con la base de datos.
Query: emite una consulta SQL a la conexión de la base de datos.
Execute: emite una instrucción o comando SQL a la conexión de la base de datos.
Format: da formato a los resultados de una consulta a la base de datos.
Headings: da formato a los encabezados de los resultados de una consulta a la base de datos.
Error: especifica un formato para el despliegue de errores desde la base de datos.
Valarg: Valida argumentos recibidos a través de la requisición HTTP.
Valform: Valida los datos recibidos de variables de formularios HTML.

**Figura 6. Arquitectura del funcionamiento de la interfaz db*Web***




Fuente: Bases de datos en Internet Forum.

### 5.11.2 Servicio db*Web*

En la figura anterior se muestra el servicio db*Web* que es puesto en marcha como un servicio multitarea, utiliza una arquitectura cliente/servidor. Utiliza un programa relativamente pequeño, al que carga dentro de dicho servidor de tal manera que éste abre una ruta de acceso para el servicio db*Web*, que se mantiene en contacto con las fuentes de datos ODBC.

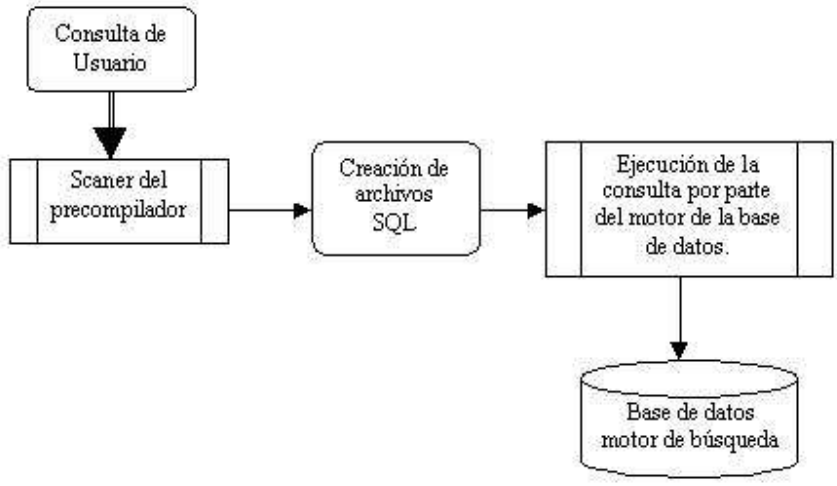
Es decir, el servicio db*Web* debe iniciarse junto con el Servidor *Web*, de tal manera que permanezca cargado en memoria en la espera de requisiciones de datos, enviadas a través de la ruta del navegador *Web* - Servidor HTTP - Cliente ISAPI db*Web*. Este último se encarga de hacer llegar las requisiciones hasta el Servicio db*Web*, el cual a su vez, se pone en contacto con las fuentes de datos ODBC para seleccionar la información que se ha solicitado, enviándola de nuevo al browser a través del servidor *Web*.

**Figura 7. Flujo de información del cliente *Web* hacia la base de datos y viceversa.**


Fuente: Bases de datos en Internet Forum.

El proceso de consulta permite hacer uso del servicio WWW para la petición de información y para devolver los resultados a la página del cliente *Web*. El diagrama muestra de forma general el funcionamiento básico del proceso de solicitar y recibir información.

**Figura 8. Diagrama del precompilador de consultas**




El diagrama de precompilador de consultas representa el proceso mediante el cual se obtiene la petición del usuario y se transforma a una sentencia de SQL en un archivo que contendrá todas las posibles cláusulas con las palabras clave utilizadas. Todas las sentencias finales son trasladas a través de la interfaz db*Web*, la cual a través de ODBC ejecutará las conexiones a las base(s) de dato(s).

## 5.12 Consideraciones técnicas

La incorporación de ciertas herramientas nuevas para el desarrollo del motor de búsqueda es indispensable, y algunas de las consideraciones más importantes que se deben de realizar son:

- El servidor de aplicaciones
- Lenguaje para intercambio de información

El servidor de aplicaciones debe ser la herramienta base para la construcción de todo el motor de búsqueda, todas las especificaciones de desarrollo que se necesitan deben estar relacionadas con el servidor de aplicaciones.

El servidor de aplicaciones es el software que permite reunir todos los componentes de un sistema para que se puedan utilizar desde cualquier cliente en cualquier plataforma y con cualquier tipo de conexión ODBC, API. El servidor de aplicaciones es el concepto que según la mayoría de desarrolladores de aplicaciones tiende a ser la innovación a mediano plazo en la construcción de software cliente - servidor, con tecnología de N capas, las cuales permiten que todo los procesos de requerimientos hacia la base de datos los ejecute el servidor de aplicaciones desde cualquier cliente utilizando para ellos componentes, agentes que básicamente lo que hacen es utilizar una de las conexiones creadas con anterioridad para ejecutar sus procesos sin requerir para ello una nueva conexión que saturaría la red. Los servidores de aplicaciones a menudo se están utilizando para la creación de las nuevas aplicaciones *Web*, en donde se encuentra la versatilidad para manejar estructuras de datos para el intercambio masivo de información sin causar problemas de concurrencia, velocidad, capacidad de almacenamiento. Uno de los servidores de aplicaciones más potentes y que está tomando auge es el EAServer 4.1.1, el cual está basado en un modelo totalmente orientado a objetos en donde la herencia, polimorfismo y encapsulamiento que son las características básicas de este modelo brindan un gran desempeño en la reutilización de código. Paralelamente a la utilización del concepto de servidor de aplicaciones, encontramos la nueva técnica de intercambio de información en Internet a través de XML descrita anteriormente. La combinación de estas dos herramientas pueden llegar a revolucionar en gran parte el intercambio de información a través de la *Web*.

# CONCLUSIONES

1. La tendencia de los usuarios de los motores de búsqueda a disponer de información de alta calidad en el menor tiempo crece a medida que aumentan los sitios *Web* que proporcionan información sobre determinado tema, esto debido que, a mayor número de sitios menor será la capacidad de respuesta de los actuales motores de búsqueda.

2. La tecnología de transferencia de información a través de Internet ha crecido de manera que algunas herramientas utilizadas se han quedado cortas y han dado lugar al surgimiento de herramientas tales como el XML que pretende ser, en un futuro cercano, uno de los lenguajes más potentes de transferencia de datos.

3. Las bases de datos en Internet siguen utilizando los mismos conceptos que ya han sido definidos para las otras bases de datos a diferencia de las interfases y protocolos de comunicación con los motores de búsqueda.

4. La tendencia de todos los desarrolladores *Web* hacia la utilización de un servidor de aplicaciones para la construcción de aplicaciones orientadas a Internet en el ambiente de desarrollo de N capas, va en crecimiento debido a la versatilidad, formas de comunicación, velocidad de procesamiento y reutilización de código que estas herramientas proveen entre otras muchas.

# RECOMENDACIONES

1. Se puede estimar que la mayor parte de los usuarios en Internet, aproximadamente el 95% utilizan el servicio para poder acceder a información por medio de un motor de búsqueda, y que lo que buscan es:

    - Información de alta calidad
    - Tiempos de respuesta mínimos
    - Interfases intuitivas y fáciles de usar
    - Que presten los servicios de transferencia de información entre usuarios

2. Por el lado de los desarrolladores, deberán cambiar algunas técnicas y metodologías de desarrollo, así como también las herramientas actuales, esto significa que:

    - La incursión de XML como herramienta de desarrollo para la transferencia de datos apunta a ser una de las más potentes para este procedimiento en un futuro cercano.
    - La utilización de servidores de aplicaciones para poder centralizar los programas actuales y la información contenida será necesaria.
    - El análisis y desarrollo de protocolos nuevos deberá convertirse en un elemento que ayude a que las nuevas metodologías tengan éxito.

3. Los desarrolladores *Web* deberán acoplarse y prepararse a las nuevas necesidades y requerimientos de los usuarios de Internet en los próximos años.

4. Se conoce por fuentes estadísticas que el crecimiento de información disponible en Internet crece a razón de 3% del total, esto significa que los desarrolladores deben considerar en sus nuevas aplicaciones aspectos de desempeño para que su funcionalidad no se vea afectada.

# BIBLIOGRAFÍA


1. Bases de Datos
   **http://bases.colnodo.org.co/wwwisisnt/**
   Fecha aproximada: octubre 2002

2. Base de datos Web
   **http://www.db*Web*.com**
   Fecha aproximada: octubre 2002

3. Biblioteca de Base de datos
   **http://www.aui.es/biblio/bolet/bole004/boletin.htm**
   Fecha aproximada: octubre 2002

4. Cartelera informativa del cursos Universidad
   **http://www.rau.edu.uy/universidad/bases/ayuda2.htm**
   Fecha aproximada: octubre 2002

5. Documentos Bases de Datos en Internet
   **http://www.cindoc.csic.es/isis/historia.htm**
   Fecha aproximada: octubre 2002

6. Grupos de investigación
   **http://www.uco.es/investiga/grupos/rea/search/descripcion.htm**
   Fecha aproximada: octubre 2002

7. Interfases Webdbms
   **http://www.interface*Web*dbms.com**
   Fecha aproximada: octubre 2002

8. Relational DataBases
**http://www.rds.org.hn/mirador/**
Fecha aproximada: octubre 2002

9. Simon, A. & Company. **Programación en *Web*.** 3ª. edición México: Prentice Hall Hispanoamerica 1998. 224pp.

10. Tanenbaum, Andrew. **Redes de computadoras.** 3ª. edición, México: Prentice Hall Hispanoamerica,1997. 558pp.

11. Villalobos, Jorge. **Diseño y manejo de estructuras de datos en C**. 3ª. edición, Bogotá: McGraw Hill,1996. 208 pp.

12. XML al descubierto
**http://www.xml.com.cgi-bin/docs**
Fecha aproximada: octubre 2002

# APÉNDICES

## 1. Algoritmo de búsqueda. ABSearchEngine.java ++

```
import java.io.*;
class ABSearchEngine extends Thread implements FilenameFilter{
public static PrintStream out=System.out;
public static int maxrunningplugins=1;

public static void main(String[] args){
String q;
if (args.length==0){q="";}else{q=args[0];}
try{
REQManipulation Helper=new REQManipulation(q+"&");
String hole=Helper.REQ("query=");
String Search=Helper.REQ("&");
String[] plgs=ABSearchEngine.getPlugins();
boolean[] plgsCK=new boolean[plgs.length];
Object Plugins[][]=new Object[plgs.length][2];

//Se extraen los Conectores de base de datos que estan activos.
//bases de datos: Access, Oracle, MSSQL, Sybase, etc
int maxplugins=-1;
File CL;
GusPlugin PL;
String FN;
// En la siguiente línea se hace referencia a un Motor de búsqueda en este caso
//www.todala*Web*.com
if (Helper.Text.equals("")){Helper.setText("www.todala*Web*.com=on&");}
while ( Helper.Text.indexOf("=on&")!=-1 ){
   FN = Helper.REQ("=on&");
   CL = new File(FN+".class");
   if (CL.canRead()){
   maxplugins++;
   Plugins[maxplugins][0]=(Object) FN;
   if (maxplugins<=ABSearchEngine.maxrunningplugins){
     for(int i=0;i<plgs.length;i++){if
(plgs[i].equals(CL.toString())){plgsCK[i]=true;}}//end for
     }
```

**continuación …**

```
   }/ si encuentra algún otro conector de base de datos
  }//end while
if
(maxplugins>ABSearchEngine.maxrunningplugins){maxplugins=ABSearchEngin
e.maxrunningplugins;}

//Código para imprimir algunos mensajes en la página de resultados
Helper.setText(Search);
Helper.urlDecode();
ABSearchEngine.out.println("<a name=\"inicio\"></a><h3>Consulta  :
"+Helper.getText()+"</h3>");

//En las siguientes instrucciones se indentifica al motor de búsqueda
ABSearchEngine.out.println("<div align=\"center\"><img
src=\"http://www.todala Web.com/ABSearchEngine/ABSearchEngine.png\" alt=\"[
ABSearchEngine - MetaBuscador]\" border=0></div>");
ABSearchEngine.out.println("<div align=\"center\">  <h5>META BUSCADOR
NO COMERCIAL </a></h5></div>");

ABSearchEngine.out.println("<form action=\"http://www.todala Web.com/cgi-
bin/ABSearchEngine\" method=\"get\" enctype=\"application/x-www-form-
urlencoded\">");
ABSearchEngine.out.println("<div align=\"center\"><input type=\"text\"
name=\"query\" size=60 maxlength=256 value=\""+Helper.getText()+"\">");
ABSearchEngine.out.println("<input type=\"submit\"
value=\"buscar\"></div><br><br>");

//Buscando una plantilla si existe, sino se debe de ingresar el path
File Template=new File("template.html");
if (Template.canRead()){
FileInputStream input=new FileInputStream(Template);
BufferedReader bfr = new BufferedReader(new InputStreamReader(input));
String Line;

while ((Line=bfr.readLine())!=null){ABSearchEngine.out.println(Line);}
//end printing
} //i si encontro algun template

//Imprimi los check buttons para los conectores
  ABSearchEngine.out.print("<center>| ");
  for (int i=0;i<plgs.length;i++){
  Helper.setText(plgs[i]);
```

**continuación ...**

```
  plgs[i]=Helper.REQ(".class");
  if (plgsCK[i]){ABSearchEngine.out.println("<input type=\"checkbox\" checked
name=\""+plgs[i]+"\">"+plgs[i]);}else{ABSearchEngine.out.println("<input
type=\"checkbox\" name=\""+plgs[i]+"\">"+plgs[i]);}
  ABSearchEngine.out.print(" |");
  }//end for
ABSearchEngine.out.println("</center></form><hr>");
if (Search.equals("")){System.exit(0);}


//Generando un Objeto Conector Pedido
boolean[] Parsed=new boolean[maxplugins+1];
for (int i=0;i<=maxplugins;i++){

//cargando el objeto class
 Plugins[i][1]=(Object) Class.forName((String)Plugins[i][0]);

//generando un objeto class (no es necesario guardar el obj class)
    Plugins[i][1]=(Object) (((Class)Plugins[i][1]).newInstance());
    ((GusPlugin)Plugins[i][1]).setSyntax(Search);
    ABSearchEngine.out.println("<!-- <b>Esperado....
"+((GusPlugin)Plugins[i][1]).MyName+" La conexión está siendo
cargada...</b><br> -->");
    ((GusPlugin)Plugins[i][1]).start();
    Parsed[i]=false;
    ABSearchEngine.out.println("<a
href=\"#"+((GusPlugin)Plugins[i][1]).getName()+"\"><b>"+((GusPlugin)Plugins[i][
1]).MyName+"</b></a>");

}//end for

//Procesando los resultados que van siendo devueltos de las cláusulas SELECT
int ok=0;
int oldok=-1;
int timewait=0;
String result[][];
int rlen;
boolean rdy;
String my;
while(ok<=maxplugins){
  for (int i=0;i<=maxplugins;i++){
  sleep(500);
```

**continuación …**

```
// Insólito, sin la sentencia anterior Sleep(500)  la JVM se queda en un Loop
  rdy=((GusPlugin)Plugins[i][1]).ready();
  if ((!Parsed[i])&&(rdy)){
  ((GusPlugin)Plugins[i][1]).parseBuff();

  my=((GusPlugin)Plugins[i][1]).MyName;   //Esto instrucción se utiliza para
depurar.

  result=((GusPlugin)Plugins[i][1]).getResults();
  rlen=((GusPlugin)Plugins[i][1]).resultsLen();
  ABSearchEngine.out.println("<br><a
name=\""+((GusPlugin)Plugins[i][1]).getName()+"\" href=\"#inicio\"><b>From
"+my+":"+rlen+"</b></a></h2><br>");


//Aquí se toman en cuenta los metatags (palabras claves)

     for(int k=0;k<rlen;k++){
                  ABSearchEngine.out.println("<b>"+((GusPlugin)Plugins[i][1]).drop
Tags(result[0][k])+"</b><br>"+((GusPlugin)Plugins[i][1]).dropTags(result[1][k]));
       ABSearchEngine.out.println("<br><a href=\""+result[2][k]+"\"
target=_blank>"+result[2][k]+"</a><hr>");
     }//end printn
  Parsed[i]=true;
  ok++;
  }//end if
  }//end parsed
  if (oldok!=ok){
       if (ok!=maxplugins) {
          sleep(500); timewait++;
          }//if not max
          }//if change
  if (timewait>20){ABSearchEngine.out.println("Fin estado de espera.");
System.exit(0);}
}//end while

}catch (Exception e){e.printStackTrace(); System.exit(0);}
ABSearchEngine.out.println("<!--  Finalización Normal -->");
System.exit(0);
}//end main

public static String[] getPlugins(){
```

**continuación ...**

```
ABSearchEngine g=new ABSearchEngine();
File list=new File("./");
String[] plgs=list.list((FilenameFilter)g);
  for(int i=0;i<plgs.length;i++){

  }//end for
  return plgs;
}//  Fin getPlugins
public boolean accept(File dir, String txt){
if (txt.equals("ABSearchEngine.class")){return false;}
if (txt.equals("REQManipulation.class")){return false;}
if (txt.equals("GusPlugin.class")){return false;}
if (txt.indexOf(".class")!=-1) {return true;}
return false;
} //end accept [FileFilter]
} //end class
```

**2. Creación de índices en las tablas de la base de datos**

```
CREATE  INDEX idx_enlaceA
ON dbo.enlace
(cod_enlace,enlace)

CREATE  INDEX idx_enlaceB
ON dbo.enlace
(titulo,nombre_sitio)

CREATE  INDEX idx_enlaceC
ON dbo.enlace
(metatag,peso)

CREATE  INDEX idx_enlaceD
ON dbo.enlace
(metatag,peso)

CREATE  INDEX idx_enlaceE
ON dbo.enlace
(cod_idioma,consultas)

CREATE  INDEX idx_enlaceF
ON dbo.clasificacion
(cod_clasificacion)

CREATE  INDEX idx_enlaceG
ON dbo.idioma
(cod_idioma)
```

**Figura 9. Modelo Físico de algunas tablas de la base de datos**