Universidad de La Habana
Facultad de Matemática-Computación

MATCOM
Facultad de Matemática - Computación

Buscador Web

TRABAJO DE DIPLOMA

Autor: José Albert Cruz Almaguer

Tutor: Ing. William Azcuy Morales

UCI
Universidad de las Ciencias Informáticas

Ciudad de La Habana, julio de 2004

Agradecimientos

No es nada fácil expresar agradecimientos en un trabajo que aunque pequeño en extensión es de inmenso valor pues simboliza la culminación del esfuerzo de muchos años. La relación tiene necesariamente que comenzar por mi madre sin la que indudablemente no habría podido llegar hasta aquí y de quién siento además de amor una profunda admiración. Además debo agradecer a la familia Ruiz Miyares, en especial al Dr. Leonel Ruiz Miyares con quienes obtuve no sólo conocimientos académicos sino también un ejemplo excelente a seguir en cuanto a lo que vale la perseverancia y el sacrificio para lograr las más importantes metas en la vida; y en general a todos mis compañeros de estudio y trabajo con los que he comenzado a recorrer el largo camino del perfeccionamiento humano.

Gracias,

José Albert.

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor del presente trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) y a la Facultad de Matemática y Computación de la Universidad de La Habana para que hagan el uso que consideren necesario con el mismo.

Para que así conste firmo la presente a los 25 días del mes de junio de 2004.

_____ _____

Firma del Autor Firma del Tutor

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado Buscador Web, fue realizado en la Universidad de las Ciencias Informáticas. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

□ Totalmente

□ Parcialmente en un ____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes (cuantificar):

_____

_____

_____

_____

_____

_____

_____

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a _____.

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año _____

_____ _____

Representante de la entidad Cargo

_____

Firma Cuño

Resumen

Gestionar los documentos existentes en sitios web de forma manual es, dada la cantidad de información que representan y la velocidad a la que sufren modificaciones, impracticable; los sistemas automatizados de recuperación de información surgen como la muy necesaria solución a tal problema. El presente trabajo trata este complejo tema, particularmente enfocado al diseño e implementación de un buscador web; y más específicamente aún se plantea el desarrollo del módulo encargado de la interpretación de la consulta con la que el usuario interrogará al sistema, y el establecimiento de un orden, según su importancia, a los documentos que tengan algún grado de relación con los intereses que los navegantes hayan expresado mediante dicha consulta. Para ello asumirá la existencia de un índice (o base de datos) en el que estará, debidamente estructurada, la información existente en la Web.

Este trabajo se enmarca dentro del Programa de Informatización de La Universidad de las Ciencias Informáticas, centro de avanzada en el uso de las Tecnologías de la Información y las Comunicaciones, para el cuál resulta muy necesario un mecanismo de este tipo.

Introducción

Desde que en 1990 Tim Berners-Lee creara la WWW, permitiendo nacer a la Internet, no ha habido un día en que esta haya dejado de crecer, convirtiéndose con el decursar del tiempo en una enorme biblioteca virtual contenedora de una buena parte del saber humano. Sin embargo, de nada sirve tanto conocimiento sin un mecanismo que permita localizar dentro de ese mar de información, de manera rápida y efectiva, aquella que en un momento dado interese.

Los sistemas de recuperación de información constituyen el mecanismo que, de forma automática y rápida, están destinados a solventar este problema. En general pueden ser usados para localizar no sólo información textual, sino también imágenes, sonidos y videos. Dentro del ámbito de la Web son clasificados en dos grandes grupos: directorios temáticos, y motores de búsqueda o buscadores; el presente trabajo está dirigido particularmente hacia este último.

Actualmente esta área es de las más activas en cuanto a investigación y desarrollo en el mundo, destacándose dentro de los directorios: Yahoo, y dentro de los motores de búsqueda: MSN de Microsoft, próximo a ser actualizado, y particularmente, Google, el buscador de mayor éxito; manteniendo todos una tendencia hacia la integración de servicios: ej. el reciente Google Froogle destinado a permitir la localización de tiendas virtuales.

Nuestro país se encuentra inmerso en un profundo proceso de informatización de la sociedad, la Universidad de las Ciencias Informáticas constituye una institución abanderada en este sentido, y a corto plazo se proyecta como la más desarrollada en cuanto al uso de las Tecnologías de la Información; en el marco de esta realidad el contar con una herramienta propia de búsqueda en la Web sería de un gran valor, por lo cual se ha concebido el presente proyecto encaminado a lograr ese objetivo. Para ello se deberá hacer un profundo estudio de las metodologías de recuperación de información, particularmente de las diferentes técnicas de organización de la información así como de asignación de relevancia a documentos.

Objetivos

1 Estudio de técnicas de organización e indexación de la información.

2 Desarrollo de un lenguaje de consulta que provea al usuario de un mecanismo de expresión de sus necesidades informativas.

2.1 Generación de una representación de dicha consulta (forma interna) que posibilite la búsqueda de los documentos.

3 Implementación de algoritmos que den la medida de cercanía entre la consulta y las páginas relacionadas.

4 Estudio e implementación de algoritmos para establecer un orden de relevancia entre documentos.

Estructuración del contenido

El presente trabajo está formado por una introducción y dos capítulos; el primero: *Fundamentación Teórica*, está orientado a dar una panorámica del estado actual de los mecanismos de localización de información en la Web, así como de las principales áreas del conocimiento implicadas en los sistemas de recuperación de información, para ello se analizarán los principales modelos de RI existentes, métodos de organización de los datos, características de los lenguajes de consulta y de los criterios de asignación de relevancia. El segundo por su parte: *Caracterización, Análisis y Diseño del Sistema*, se encarga primeramente del planteamiento de los requisitos funcionales; seguido del análisis y diseño realizados así como de los detalles del funcionamiento del subsistema implementado: analizador de consultas y recuperador de documentos, todo ello en correspondencia con lo planteado en el primer capítulo. Al final se presentan las conclusiones obtenidas mediante la realización del trabajo y se hace una serie de recomendaciones en aras de completar y perfeccionar el SRI en el futuro.

Capítulo 1. Fundamentación teórica

1.1 Estado del arte

Continuamente se invierten grandes recursos en la investigación así como en el desarrollo de sistemas de recuperación de información con mira en la Web, por lo general las personas que usan la Internet no saben de antemano dónde se encuentra la información que necesitan, y aún aquellas que conozcan adonde deben dirigirse es muy difícil que se conformen con lo existente en ese lugar pues con seguridad existirá en algún otro punto del ciberespacio datos de mucha pertinencia para sus intereses.

1.1.1 Directorios y buscadores

Las herramientas para localizar información en la World Wide Web son:

- Los directorios temáticos.
- Los buscadores o motores de búsqueda.

La información que contienen los directorios normalmente es recogida y organizada por expertos de forma manual en una estructura jerárquica. Para un usuario usarlos deberá ir navegando de categoría en categoría hasta encontrar lo que más se adecue a sus necesidades.

Los directorios más comunes son aquellos que ofrecen una navegación por temas como Yahoo *(Yet Another Hierarchical Officious Oracle)* [http://www.yahoo.com]. Sin embargo, también existen directorios que permiten una navegación geográfica como, por ejemplo, Virtual Tourist [http://www.vtourist.com/vt]

Los motores de búsqueda por su parte presentan una estructura constituida por:

- Robot o araña (spider): programa que cruza la WWW moviéndose de un documento a otro, descendiendo progresivamente a través de los enlaces.
- Un programa de indexación que organiza de forma conveniente la información de los millones de páginas web ubicadas en servidores conectados a la red.
- Una interfaz mediante la que los usuarios satisfacen sus necesidades de información consultando enormes bases de datos que son alimentadas por los sistemas anteriores.

Como ejemplos de buscadores se pueden encontrar: Google [http://www.google.com], Alltheweb [http://www.alltheweb.com]. Independientemente a estas distinciones existen servicios con características de directorio que ofrecen la posibilidad de realizar búsquedas en sus estructuras jerárquicas (por ejemplo el directorio Yahoo! con su servicio *Yahoo Search*) y servicios de búsqueda que han incorporado facilidades de directorio (por ejemplo, Google a través del *Google Directory* [http://directory.google.com]) si bien por lo general estos servicios mantienen estructuras de información independientes en un caso y en el otro.

	Descubrimiento de recursos	Representación del contenido del documento	Representación de la consulta	Presentación de los Resultados
Directorios	La realizan personas	Clasificación manual	Implícita (mediante navegación por las categorías)	Páginas creadas previamente a la consulta. Poco exhaustivos y muy precisos.
Buscadores	Principalmente de forma automática mediante robots.	Indexación Automática	Explícita (mediante palabras clave o conceptos, operadores, delimitadores, etc.)	Páginas creadas de forma dinámica para cada consulta. Muy exhaustivos y poco precisos

Tabla 1. Características principales de los Directorios y Buscadores.

Es necesario señalar que existen metabuscadores, sistemas que utilizan a otros buscadores, no disponen de spider propia ni de índice alguno, y de los resultados devueltos por estos analizan cuáles se adecuan más a las necesidades del usuario, dándolos por respuesta.

El objetivo de los buscadores es presentar al usuario dónde encontrar documentos relevantes a un tema buscado. Para ello se tienen que enfrentar a grandes problemas como son:

- La información que se encuentra en Internet está desordenada.
- La información en Internet cambia continuamente.
- La información en Internet es redundante.
- La gran cantidad de datos que deben manejar.
- ¿Cómo representar la información para su procesamiento?
- La precisión y la relevancia de los documentos recuperados.

Cada una de estas problemáticas tiene una influencia mayor o menor en los sistemas hasta ahora desarrollados, pero de todas, la última, es la que ha resultado decisiva.

Todo esto ha provocado que los diferentes buscadores hayan desarrollado numerosas técnicas, utilidades, interfaces de usuario y algoritmos (la mayoría de carácter privado), con el objetivo de solucionar la mayoría de los problemas citados, con la particularidad de que cada uno intentará dar los mejores resultados y en el menor tiempo posible según la información que abarque en su propia base de datos y de acuerdo con los criterios utilizados en las distintas técnicas.

1.1.2 Principales motores de búsqueda en Internet

1.1.2.1 Caso de estudio: Google

De todos los motores existentes el de más éxito por el momento es Google, con 4 000 millones de páginas indexadas hasta el momento, este buscador basa su funcionamiento en una red de miles de máquinas de bajo costo que trabajan en paralelo. Muestra muchos resultados, entre los que se encuentran: el título del documento, fragmentos que contengan los términos de la consulta, la URL de la página, la descripción, el tamaño del fichero y una opción llamada *Páginas similares* con la que se pueden localizar documentos relacionados.

Google fue fundado en septiembre de 1998 por Larry Page y Sergey Brin, dos estudiantes de doctorado de Stanford [01]. El nombre proviene de un juego de palabras con el término "googol", acuñado por Milton Sirotta, sobrino del matemático norteamericano Edward Kasner, para referirse al número representado por un 1 seguido de 100 ceros. El uso del término refleja la misión de la compañía de organizar la inmensa cantidad de información disponible en la web y en el mundo.

En este buscador la principal causa de éxito (según sus propios creadores) es PageRank [02], un sistema de clasificación que es el encargado de establecer un orden de relevancia entre las páginas web. Este algoritmo usa los enlaces existentes entre las páginas como base para calcular el valor o relevancia de ellas, Google interpreta un vínculo desde la página A hacia la página B como un voto de A por la página B, a su vez analiza la relevancia de la página que emite el voto. Los votos emitidos por páginas que son en sí mismas "importantes" pesan más y ayudan a convertir a otras páginas también en "importantes".

1.1.2.2 Caso de estudio: Alltheweb

AlltheWeb apareció en agosto de 1999 como fruto del trabajo de los estudiantes y profesores de la Universidad Noruega de Ciencia y Tecnología [03]. Inicialmente su nombre fue FAST (Fast Search & Transfer), pero al cambiar su diseño en el verano del

2001 el nombre Alltheweb se convirtió en el definitivo. Dispone de una de las bases de datos más importantes en Internet, siendo una de las que se actualiza más frecuentemente (aproximadamente un 30% de sus registros se reindexan semanalmente). Se utiliza en Lycos, Scirus y otros motores de búsqueda, muchos de ellos europeos. Sus otras bases de datos (noticias, imágenes, audio, video y FTP) también son de las más amplias de la Red. El buscador funciona bajo servidores Dell PowerEdge 4300 y sistemas de almacenamiento PowerVault, con el sistema operativo FreeBSD.

Actualmente constituye un serio competidor de Google, con más de 3 000 millones de páginas web indexadas. Su principal ventaja sobre este es que actualiza sus bases de datos con una periodicidad semanal, mientras Google lo hace mensualmente.

1.1.3 Criterios para evaluar un SRI

Un Sistema de Recuperación de Información es evaluado generalmente por los siguientes criterios: *eficacia en la ejecución*, *efectividad en el almacenamiento*, *efectividad en la recuperación* y la riqueza de funcionalidades que ofrezca al usuario. La relativa importancia de estos factores debe ser decidida por el diseñador del sistema seleccionando en función de ello las estructuras de datos y algoritmos adecuados para su implementación.

La *eficacia en la ejecución* es medida por el tiempo que se toma un sistema o parte de él para realizar una operación (el tiempo entre el pedido y la respuesta). Este parámetro posee una importancia decisiva pues no son muchos los usuarios que están dispuestos a esperar siquiera varios segundos por un resultado.

La *eficiencia del almacenamiento* es dada por el número de bytes que se precisan para almacenar los datos. Como medida del exceso de espacio se usa la razón existente entre el tamaño del índice de los ficheros además del tamaño de los archivos del documento y el tamaño de los archivos del documento.

$$Exceso\ de\ espacio = \frac{tamaño_{índice} + tamaño_{documentos}}{tamaño_{documentos}}$$

Otra característica de enorme importancia es *la efectividad en la recuperación* basada en la relevancia que para los intereses del usuario tengan los documentos recuperados, excluyendo las características inherentemente subjetivas del problema se han propuesto básicamente dos medidas de dicha efectividad: relevancia y precisión.

La *relevancia* (*recall*) es la proporción entre el número de los documentos relevantes recuperados en una búsqueda dada y el número de documentos relevantes realmente existentes para esa búsqueda en la base de datos. Este denominador es por lo general desconocido y debe ser estimado por muestreo u otros métodos similares.

$$Recall = \frac{Cantidad\ de\ documentos\ relevantes\ recuperados}{Cantidad\ de\ documentos\ relevantes}$$

La precisión por su parte es la proporción entre el número de documentos relevantes recuperados y el número total de documentos recuperados.

$$Precisión = \frac{Cantidad\ de\ documentos\ relevantes\ recuperados}{Cantidad\ de\ documentos\ recuperados}$$

El rango de valores de ambas medidas se encuentra entre 0 y 1.

1.2 Modelos conceptuales de recuperación de información

Un modelo de Recuperación de Información queda definido cuando se fijan en un sistema en particular lo siguiente: cómo se obtienen las representaciones de los documentos y de la consulta, la estrategia para evaluar la relevancia de un documento respecto a una consulta, los métodos para establecer la importancia (orden) de los documentos recuperados y los mecanismos que permiten una retroalimentación por parte del usuario para mejorar la consulta.

Habiendo visto esto, conceptualmente un modelo de recuperación de información es una tupla $< D, Q, F, R(q_i, d_j) >$, donde:

- *D*: Representación lógica de los documentos.
- *Q*: Representación lógica de los requerimientos de información (*queries* o consultas).
- *F*: Marco para modelar documentos, consultas y sus relaciones.
- $R(q_i, d_j)$: Función de *Ranking*.

Ejemplos de modelos de recuperación de información son:

- Modelo booleano.
- Modelo vectorial.

1.2.1 Modelo booleano

En este modelo los documentos se representan por un conjunto de términos de indexación, cada uno de los cuales a su vez se trata como una variable booleana que tomaría valor **verdadero** si la característica asociada está presente en el documento y **falso** en caso contrario. En este modelo, las consultas de usuario consisten en expresiones booleanas convencionales en función de palabras claves conectadas mediante los operadores lógicos AND, OR y NOT.

En los sistemas de Recuperación de Información más simples el usuario introduce directamente las expresiones; en otros sistemas más sofisticados, el usuario emplea expresiones en lenguaje natural, las cuales el sistema convierte a expresiones booleanas.

La semejanza entre un documento y una consulta es una función booleana que devuelve 1 si el documento satisface la consulta y 0 si no.

Las técnicas más usadas en la implementación de este modelo son [04]:

Vectores de bits: enfoque según el cual cada documento se representa por una secuencia de bits, cada uno asociado a una palabra. Estos bits valdrán 1 (VERDADERO) si la

palabra que le corresponde está en el documento y un 0 (FALSO), si no. Este enfoque es muy apropiado cuando se emplean pocos elementos.

Hashing: técnica no tan rápida como la anterior, que emplea el espacio de forma más eficiente, pues mientras que en la anterior se obliga a que todos los vectores tengan la misma longitud, y además que ésta sea invariable, el hashing permite emplear conjuntos arbitrarios.

1.2.2 Modelo vectorial

Este modelo está basado en que cada documento de la colección está representado por un vector *n*-dimensional, donde *n* es la cardinalidad del conjunto de términos de indexación elegidos para la colección de documentos, en el que cada componente representa el peso del término previamente asociado a esa dimensión. Este peso representa un estimado de la utilidad del término como descriptor del documento, es decir, de la utilidad para distinguir ese documento del resto de los presentes en la colección [05] (ej. un término recibe un peso igual a 0 en los documentos en los que no aparece). Los pesos pueden reflejar diferentes medidas, por ejemplo, la frecuencia de aparición de un término en un conjunto dado (técnica conocida como *TF*, de uso generalizado). Es bueno señalar que una representación como ésta, basada sólo en la existencia de los términos, sacrifica información sobre el orden en que los términos ocurren, información sintáctica, etc.

Las consultas formuladas por los usuarios, al estar compuestas también por términos, son representadas de la misma forma que los documentos; posibilitando el cálculo de la semejanza entre una consulta y un documento a partir de la obtención del producto escalar de ambos vectores:

$$sem(d_i, q_j) = \frac{d_i \cdot q_j}{\|d_i\| \|q_j\|} = \frac{\sum_{k=1}^{n} (w_k^i \cdot w_k^j)}{\sqrt{\sum_{k=1}^{n} (w_k^i)^2 \cdot \sum_{k=1}^{n} (w_k^j)^2}}$$

donde $d_i = (w_1^i, w_2^i, ..., w_n^i)$ es la representación del documento y $q_j = (w_1^j, w_2^j, ..., w_n^j)$ la de la consulta.

El modelo vectorial es el que más rendimiento en cuanto a eficacia está produciendo hoy en día. Si bien, su inconveniente es que requiere un esfuerzo de procesamiento muy superior al resto de los modelos.

1.3 Métodos de organización de la información

Un elemento de mucha importancia a tener en cuenta a la hora de diseñar un sistema de Recuperación de Información es sin duda la selección de la estructura de ficheros ha usar como base de datos subyacente. Y esta decisión no es independiente del modelo conceptual considerado, existe una relación de dependencia entre la mayoría de los modelos conceptuales y el uso de una organización de la información determinada.

A la hora de organizar la información se puede indexar los documentos: darles una estructura particular en dependencia de las necesidades o bien no indexarlos: almacenarlos como ficheros de texto plano, por lo general este último enfoque dadas sus excesivas exigencias de memoria, los costos que plantea para la actualización y la pobreza de posibilidades de elección de modelos conceptuales (tan sólo el booleano) hacen que sea muy poco usado.

La idoneidad de indexar la información depende, entre otros factores, de la frecuencia de su actualización y el volumen de la misma. Así, una colección de documentos que varía muy frecuentemente no parece adecuada para ser indexada, ya que el esfuerzo invertido en la indexación de la colección de documentos no amortizaría el tiempo ganado a la hora de resolver las consultas. Sin embargo, en el caso de que el tamaño de la colección de documentos sea grande (la generalidad de los casos), el único modo de responder a las consultas en un tiempo razonable será disponiendo de un índice. Por esta razón la mejor opción es la de indexar los documentos usando técnicas de actualización de índices o alguna solución de compromiso en los casos en que la tasa de actualización sea muy alta.

Cuando se realiza la indexación de los documentos, no todas las palabras o términos que los componen se incluyen en los índices, sólo aquellas partes de los documentos que se consideren relevantes se incluirán dentro del conjunto de los *elementos de indexación*. Además, hay que considerar que dichos elementos pueden sufrir una serie de transformaciones antes de incluirse en el índice.

Algunos de los principales métodos de organización se enumeran a continuación.

Ficheros planos

Es la forma más primitiva de organización de la información. En el enfoque de un fichero plano, uno o más documentos se almacenan en un fichero (generalmente en formato de texto ASCII); trayendo como consecuencia que las búsquedas sobre estos ficheros planos se lleven a cabo, generalmente, por medio de la localización de patrones de texto, en el que un algoritmo de búsqueda rápido recorre los documentos en tiempo real cada vez que se tenga que resolver una consulta. Tiene como ventaja que es muy fácil de implementar, sin embargo es difícil de actualizar y el acceso aleatorio es lento. Actualmente son usados paralelamente a otros métodos con el fin de permitir a los usuarios hacer búsquedas de expresiones de la forma: "la casa de Bernarda", situaciones que para los otros modelos resultan demasiado costosas o hasta imposibles de resolver.

Ficheros inversos o invertidos

La utilización de la técnica del fichero invertido es un elemento clásico en los sistemas de recuperación de información textual. Dada la gran cantidad de información contenida en los documentos textuales, ya sea un fichero Word, Adobe o una página Web, los procedimientos clásicos de búsqueda secuencial o de ficheros indexados no son capaces de responder de manera adecuada a los requerimientos necesarios de velocidad y exactitud en la respuesta para satisfacer al usuario. Por esta razón, estos sistemas usan una especialización de los ficheros indexados conocida como fichero invertido.

Un fichero invertido es un tipo de fichero indexado, donde la estructura de cada ítem o entrada del fichero generalmente tiene la forma:

Término índice	Identificadores de Documentos	Identificador de Campo

Un *término índice* es una palabra clave, frase (en general un término) con significado semántico capaz de describir a, al menos, un documento de la colección de documentos. El fichero invertido contiene a dichos términos índices ordenados de forma alfabética.

El identificador de documento es único para cada documento. Cada término índice tiene asociado una lista de los identificadores de los documentos que lo contienen. Por otra parte el identificador de campo es el utilizado para almacenar información útil dependiendo de los requerimientos del sistema de recuperación de información. Por ejemplo, el identificador de campo puede indicar en cual parte del documento (título, índice, etc.) aparece el término índice y cuantas veces se repite.

Algunos sistemas incluyen también información acerca de la localización del término en el documento, por ejemplo, el párrafo o frase en que el término aparece dentro del documento. Si se utiliza el modelo conceptual de espacio vectorial, para cada identificador de documento asociado a un término índice puede incluirse su peso en el documento, ya sea utilizando el *TF*, o cualquier otra técnica de pesado de los términos.

Este modelo permite que en el momento de realizar una búsqueda o consulta, el sistema de recuperación de información no tenga que leer todos y cada uno de los documentos; en su lugar, buscaría en el fichero invertido las ocurrencias de los términos buscados, dando por resultado los documentos en los cuales éstos aparecen.

La construcción de un fichero de estas características comprende, a grandes rasgos, los siguientes pasos:

1 Formar una lista de los términos que aparecen en el texto, junto con su localización en el mismo.

2 Invertir la lista anterior, es decir, formar una lista de los términos ordenados lexicográficamente y por orden de aparición con las localizaciones asociadas a los términos.

3 Opcionalmente, procesar posteriormente el fichero invertido, añadiendo pesos a los términos, reorganizándolos o comprimiéndolos.

Esta forma de organización es conveniente para los modelos conceptuales booleano y vectorial, permite lograr buenas velocidades de respuesta y la elaboración de algoritmos de asignación de relevancia complejos, razones que lo convierten en una muy buena opción.

Redes o grafos

Los grafos o redes son colecciones ordenadas de nodos conectados por arcos. Se usan para representar documentos de las más diversas formas. En esta clasificación entran las denominadas redes semánticas, las cuales se encargan de representar las relaciones semánticas presentes entre los términos de los documentos, información que por lo general se pierde en los demás modelos.

Este modelo desde el punto de vista teórico es muy atractivo, pero la necesidad de intervención humana en la indexación de las colecciones de documentos lo hacen inviable en la práctica.

1.3.1 Métodos de indexación

1.3.1.1 Concepto de indexación

La indexación, es la operación destinada a representar los resultados del análisis del contenido de un documento o de una parte del mismo, mediante elementos (*términos de indexación*) de un lenguaje documental o natural, orientados a facilitar la posterior recuperación de los documentos indexados.

Para hacer la indexación existen diversas alternativas [06]:

1 Indexación manual por vocabulario controlado: consistente en la asignación por expertos de los términos de indexación a partir de un conjunto finito de los mismos.

2 Indexación automática con vocabulario controlado: automáticamente se escogen los términos de los existentes en un conjunto finito.

3 Indexación libre: son los propios autores de los textos los que asignan los términos de indexación pensando en los términos que utilizarían los usuarios si buscasen información sobre los temas de sus textos.

4 Indexación a través del lenguaje natural: los términos de indexación los elige la computadora a partir de un análisis de los textos, aplicando sofisticadas técnicas de procesamiento del lenguaje natural. Este último tipo aparece hoy en día debido a la gran variedad de textos disponibles en formato electrónico y es el centro de atención en las investigaciones actuales en Recuperación de Información.

El método más empleado en este tipo de indexación es tomar como índices las palabras del texto. La gran dificultad de este enfoque es que al usarlo, se obtienen decenas miles de características, cantidad más allá de la que se podría considerar como óptima. Un efecto perjudicial es la ambigüedad en sus significados, entendiéndola como la existencia de dos o más significados para una misma palabra o expresión lingüística. Esto se puede ver como la disyunción entre dos o más conceptos no relacionados. La ambigüedad podemos evadirla por las representaciones de los textos, puesto que es poco probable que todos los significados sean de interés en un mismo texto. Por ejemplo: *"planta"* tiene varios significados y es poco probable que a un usuario que quiera recuperar documentos en los que la palabra tiene uno de esos significados (por ejemplo: factoría), le agrade que aparezcan documentos en los que tiene los otros significados (por ejemplo: vegetal o parte inferior del pie).

A diferencia de los directorios, cuya indexación es intelectual o manual, la mayoría de los buscadores de la Web realizan una indexación automática.

1.3.1.2 Niveles de indexación

Los buscadores pueden optar por diferentes niveles de abstracción a la hora de indexar los documentos en sus bases de datos.

Indexación en el nivel submorfológico: sin análisis morfológico, sintáctico ni semántico, ofrece un método muy flexible para la recuperación; la información se indexa como patrones de bits (*bit patterns*) de manera que el texto, el sonido y las imágenes en movimiento, pueden indexarse y recuperarse de la misma manera.

Indexación por palabra clave: es la forma más común de indexación de textos en la Web (esencialmente morfológico y estadístico). Se crean índices inversos de raíces y palabras claves, direcciones, ubicación y frecuencia de apariciones.

Indexación por conceptos: Existen varios procedimientos para construir bases de datos basadas en conceptos, algunas de ellas muy complejas y basadas en sofisticadas teorías lingüísticas y de Inteligencia Artificial. En otros casos, a partir de análisis estadísticos, el buscador determina qué conceptos aparecen juntos o relacionados en textos que se centran en un tema concreto. Mediante este sistema se pueden recuperar recursos que tratan un tema dado, incluso aunque las palabras incluidas en el documento no coincidan formalmente con las de la pregunta.

Indexación por hiperenlaces: La Web, como conjunto de páginas HTML relacionadas mediante enlaces o hipervínculos, puede ser interpretado como un grafo, en el que cada página constituye un nodo y los enlaces entre ellas arcos. Dado que los enlaces o hipervínculos parten de páginas o nodos concretos y apuntan a otra página concreta, se puede hablar de grafo dirigido.

1.4 Modelos de búsqueda

Las necesidades de información del usuario son expresadas a través de una ecuación de búsqueda, que en el caso más simple consta de palabras sueltas, denominadas *palabras clave* y, en otros, puede ser una complicada combinación de operadores, palabras clave y paréntesis. Algunos buscadores permiten expresar las consultas en lenguaje natural

siendo el motor de búsqueda el responsable de traducir dicha expresión a un formato estructurado sobre el cuál aplicar los algoritmos correspondientes, servicios de este tipo son dados por ask.com.

Independientemente a la forma en que el usuario exprese la consulta, ésta será analizada por el buscador y será traducida a una representación interna que permita compararla con los términos recogidos en la base de datos y seleccionar las direcciones URL que sean más relevantes

En general los buscadores ofrecen la posibilidad de realizar dos tipos de búsquedas: una simple, en la cual se especifican tan solo las palabras claves a buscar, y otra avanzada en la que el usuario puede, usando operadores, construir expresiones bastante complejas, para la confección de estas últimas normalmente se provee al sistema de una interfaz amigable que no le exija a los usuarios el tener que memorizar los operadores, ni la sintaxis.

Las búsquedas avanzadas suelen ofrecer la posibilidad de utilizar distintos tipos de operadores:

- Operadores booleanos.
- Operadores posicionales.
- Operadores de existencia.
- Operadores de truncamiento.
- Operadores de límite/comparación.

1.4.1 Operadores booleanos.

Los operadores booleanos son muy utilizados en los sistemas de recuperación de información. Los tres operadores básicos son: operador suma/unión (OR), operador producto/intersección (AND) y el operador resta/negación (NOT). Dichos operadores pueden combinarse, dando como resultado operaciones más complejas.

Para simplificar su uso a los usuarios no familiarizados con este tipo de operadores, algunos servicios de búsqueda utilizan frases del *estilo 'Buscar resultados con todas las palabras*' en lugar del operador AND y '*Buscar resultados con alguna de las palabras*' en lugar del operador OR.

1.4.2 Operadores posicionales.

Los operadores posicionales tienen por objetivo superar determinadas limitaciones de los operadores booleanos. Ellos toman como punto de partida la valoración del término dentro del contexto en el que se encuentre; se dividen en dos grupos:

Posicionales absolutos. Son operadores que permiten buscar un término en un lugar específico del documento como, por ejemplo, el título.

Posicionales relativos o de proximidad. Son operadores para establecer la posición de un término respecto a otro. Se pueden buscar palabras que estén juntas, separadas por varias palabras o caracteres, que se encuentren en la misma frase o párrafo e incluso si se debe o no respetar el orden en que se han introducido los términos. Un operador posicional muy común es NEAR.

1.4.3 Operadores de existencia.

Estos operadores tienen como fin forzar la presencia de determinados términos en los documentos recuperados, caso en el por lo general se usa el operador '+', o por el contrario obligar la exclusión, situación en la que se suele usar el operador '-'.

1.4.4 Operadores de truncamiento.

Pueden darse situaciones en las cuales sea necesario utilizar no un término simple, sino también sus derivados, fijados por prefijación o sufijación, mínimas variantes léxicas, etc. Los operadores de truncamiento facilitan este tipo de búsqueda. Se trata de operadores (normalmente símbolos como *, $), cuya presencia puede sustituir a un carácter o a un conjunto de caracteres, situados a la izquierda, dentro o a la derecha del término especificado.

1.4.5 Operadores de límite/comparación.

Los operadores de límite/comparación especifican el rango de búsqueda fijando las cotas para la misma, cotas que pueden ser tanto numéricas como alfabéticas, dichos operadores suelen ser del tipo: "*mayor que*", "*menor o igual que*" o combinaciones de

éstos. Dichos operadores están destinados a documentos contenedores de información numérica.

1.5 Respuestas de los buscadores

Los resultados que los buscadores muestran presentan en general características comunes, los listados de respuestas suelen incluir los siguientes elementos:

- Título de la página Web en forma de hipervínculo. Se toma de las etiquetas <TITLE>, donde teóricamente el constructor de la página puso una primera aproximación del contenido. La pulsación sobre el título lleva al documento original.
- Nivel de pertinencia o adecuación a la consulta planteada, es decir, el valor de relevancia del documento respecto a la consulta. Este indicador es calculado según criterios propios a cada buscador, razón por la que puede no estar de acuerdo con el valor que le daría el usuario. Normalmente se expresa en por cientos.
- La URL en la que puede localizarse el documento.
- Tamaño del documento en bytes.
- Fecha de la última actualización.
- Idioma.
- Categoría temática en la que se ha incluido, sólo si el servicio posee directorio.
- Los términos de la búsqueda presentes en la página Web, así como los fragmentos de esta en los que aparecen.
- Un breve resumen, creado (dependiendo de cada motor de búsqueda) usando las etiquetas <META>, las primeras frases de la página web, los encabezamientos interiores del mismo u otros criterios.

En algunos motores de búsqueda, se acompañan de enlaces de *tipo* "*Más como éste...*". Si se trata de un documento especialmente útil para el usuario, la pulsación de este enlace le permitirá obtener un nuevo listado con otros de contenido muy similar.

1.6 Criterios de asignación de relevancia

Los buscadores utilizan un algoritmo, cuyos detalles son siempre guardados con recelo, para darle un orden a los sitios que devolverán como respuesta a una petición formulada por el usuario; estos algoritmos constituyen la piedra angular de la eficacia de un buscador, y determinan en gran medida su éxito. Los buscadores en su gran mayoría utilizan una combinación de diferentes indicadores para determinar la relevancia de los documentos recuperados. La ponderación de estos indicadores varía de un sistema a otro, ejemplos de ellos son:

- Frecuencia de la palabra o frase de la consulta en el documento. Generalmente, se da prioridad en el ranking a las páginas que contienen un gran número de veces las palabras claves de la consulta.
- Longitud del documento. Si es corto y contiene repetidamente los términos de la búsqueda, tiene prioridad sobre otro más extenso que también repita las palabras con frecuencia.
- Grado de presencia del conjunto de las palabras o frases de la consulta en el documento recuperado.
- Proximidad entre sí de las palabras clave de una ecuación compleja en el documento recuperado.
- Presencia de las palabras o frases de la consulta en el principio del texto, en el título y/o en los encabezamientos.
- Presencia de las palabras de la ecuación de búsqueda en las etiquetas META, utilizadas para poner información que describa al documento.

- Grado de "popularidad" del documento, es decir, si ese recurso es muy citado en otras páginas Web, criterio que, particularmente, ha constituido la base del éxito de Google.
- En el caso de los directorios, las categorías situadas en las ramas superiores del árbol jerárquico, que corresponden a encabezamientos más generales, se consideran más relevantes que las subordinadas.

1.7 Microsoft .NET

En la implementación del sistema será utilizada la plataforma .NET de Microsoft, tecnología que dado el nivel de integración y facilidades de desarrollo que posee resulta muy conveniente.

Microsoft .NET es un conjunto de nuevas tecnologías que persiguen obtener una plataforma sencilla y potente para distribuir el software en forma de servicios que puedan ser suministrados remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados [07].

1.7.1 Common Language Runtime (CLR)

El Common Language Runtime (CLR) es el núcleo de la plataforma .NET. Constituye el motor encargado de gestionar la ejecución de las aplicaciones para ella desarrolladas, a las que ofrece numerosos servicios que simplifican su desarrollo y favorecen su fiabilidad y seguridad.

Presenta diversas características, de las cuales las fundamentales son:

Modelo de programación consistente y sencillo: A todos los servicios y facilidades ofrecidos por el CLR se accede de la misma forma: a través de un modelo de programación orientado a objetos. Con él desaparecen muchos elementos complejos

incluidos en los sistemas operativos actuales (registro de Windows, GUIDs, HRESULTS, etc.).

Ejecución multiplataforma: El CLR actúa como una máquina virtual, encargándose de ejecutar las aplicaciones diseñadas para la plataforma .NET. Por lo que, cualquier plataforma para la que exista una versión del CLR podrá ejecutar cualquier aplicación .NET.

Integración de lenguajes: Desde cualquier lenguaje para el que exista un compilador que genere código para la plataforma .NET es posible utilizar código escrito en otro lenguaje para el que también exista un compilador de .NET.

1.7.2 Active Server Page .NET

La tecnología ASP .NET es el subsistema destinado al desarrollo de aplicaciones para la Web dentro de la plataforma .NET. Constituye un marco de trabajo sin precedentes en cuanto a facilidades de desarrollo y posibilidades de escalabilidad de aplicaciones empresariales, permite además implementar soluciones informáticas para una diversidad de dispositivos electrónicos y dar en general cualquier servicio de información a través de la Internet, en fin ha sido creada con el ojo puesto en sacarle el mayor provecho a las potencialidades alcanzadas por la tecnología [08].

1.7.3 ADO .NET

Microsoft ADO .NET es la opción dentro del .NET framework para el acceso a bases de datos. Ofrece un mecanismo de acceso a diferentes gestores de bases de datos y lo hace sobre un modelo escalable. Aunque ADO.NET conserva algunos de los conceptos del anterior ADO se ha mejorado considerablemente para permitir el acceso a información estructurada procedente de distintas fuentes a través de un consistente modelo de programación estandarizado.

Presenta además un modelo desconectado de acceso a los datos que le permite trabajar aún cuando se haya perdido la conexión con el origen de datos.

Capítulo 2. Caracterización, análisis y diseño del sistema

Problema y situación problémica

El sistema a desarrollar deberá dar solución al problema de extraer de un conjunto grande de documentos información útil a las necesidades de un usuario. Lograr semejante acometido, comprometido desde el inicio con la subjetividad del que busca, no es nada fácil; si además, se tiene en cuenta la heterogeneidad reinante entre los consumidores del servicio a prestar, en cuanto a gustos, intereses y formación intelectual la situación se hace torna aún más compleja.

En el marco de esta situación proponemos un sistema que implemente un modelo de Recuperación de Información eficiente, en cuanto a tiempo de ejecución, y por ende rápido, basado en un índice con estructura de fichero invertido y en la metodología probabilística-estadística del modelo vectorial como lógica de recuperación. Como lenguaje de consulta se implementará uno lo bastante simple como para no convertirse en obstáculo a la hora de elegir entre este sistema y algún otro disponible, pero a la vez capaz de elaborar consultas lo suficientemente complejas como para lograr resultados exitosos en la mayoría de los casos.

Información que se maneja

El SRI, con el fin de lograr su acometido de dar respuestas correctas a las consultas formuladas por los usuarios, manipulará eficientemente a través de un índice (anteriormente definido), la colección de documentos presentes en la Web; para lograrlo descompondrá dicha información en un conjunto de términos de indexación previamente definido y extraerá las posiciones de dichos documentos en las que éstos aparezcan. El modelo conceptual correspondiente se muestra a continuación.

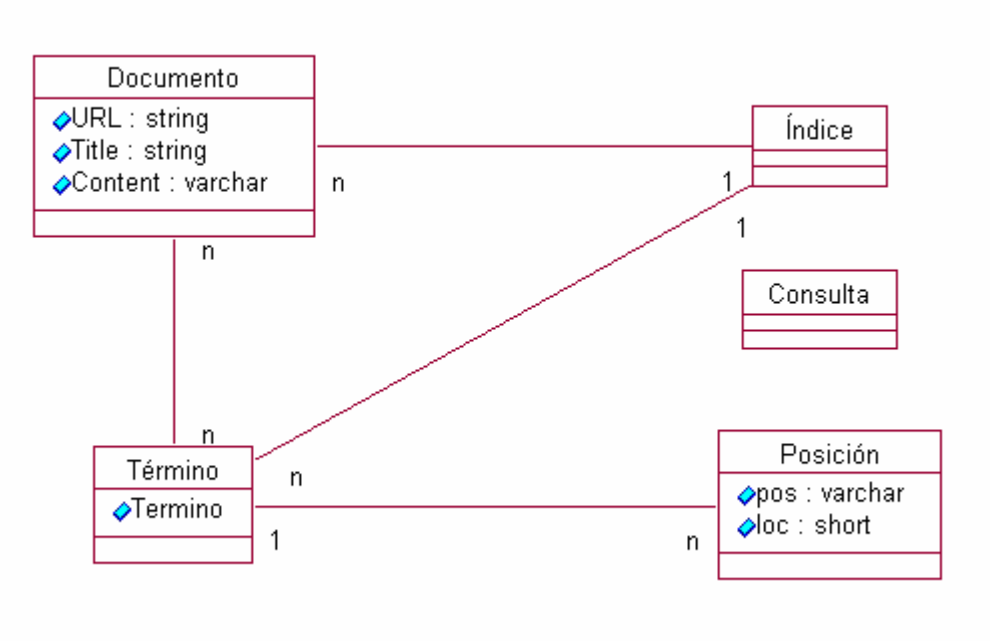


Fig. 2.1 Modelo conceptual

2.1 Análisis de requisitos funcionales

Se deberá implementar de un Sistema de Recuperación de Información (SRI), el módulo encargado de la realización de las consultas; el cuál proveerá al usuario de un mecanismo de expresión de sus necesidades informativas (lenguaje de consulta), en correspondencia a estas dará una relación de referencias a páginas web ordenadas según su relevancia (R1).

Haciendo uso de uno de los modelos de recuperación de información estudiados el sistema debe proveer un marco para la representación de documentos, consultas y sus relaciones. En función de ello estará provisto de un lenguaje de consulta para la elaboración de las solicitudes que incluya los operadores más importantes.

2.2 Definición de los casos de uso

Identificación de los actores

Nombre del actor	Descripción

Navegante	Es el único cliente del sistema. Elabora las consultas que le satisfacerán sus necesidades de información y obtiene los resultados.

Casos de uso

En el subsistema a realizar sólo está presente el caso de uso *Realizar consulta*.

CU-01	*Realizar consulta*
Actor	Navegante
Descripción	El usuario elaborará una consulta a partir de sus necesidades informativas y la entregará al sistema.
Referencia	R1

Diagrama de los casos de uso

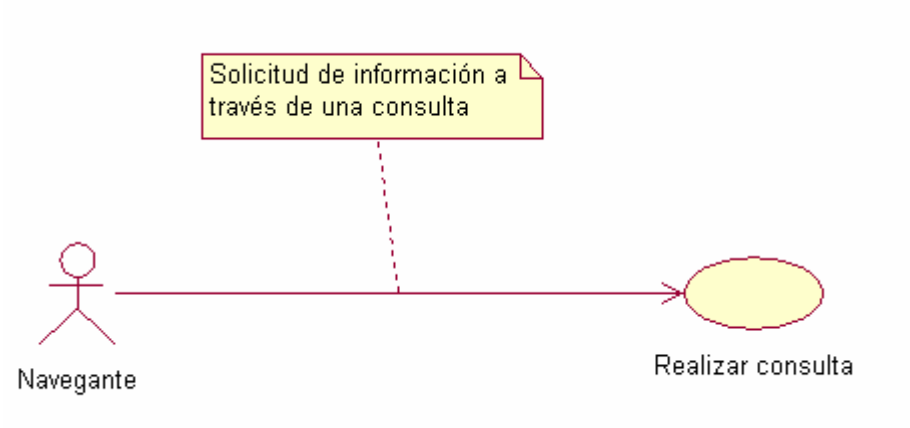


Fig. 2.2 Caso de uso *Realizar consulta*.

La expansión de los casos de uso se encuentra en el Anexo 1.

2.3 Arquitectura del sistema

Se seleccionó una arquitectura centralizada por considerase que es satisfactoria dado el volumen de datos que se manejarán. Sus componentes principales son: un índice, un

motor de búsqueda y una interfaz, la gráfica siguiente muestra la esquematización de dicha arquitectura.

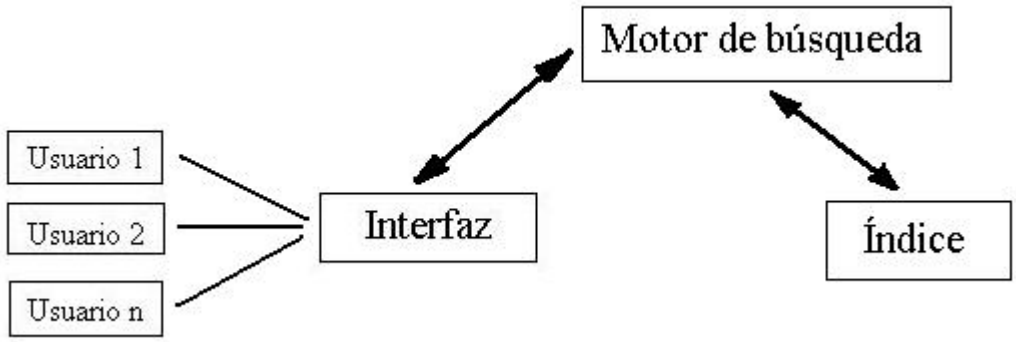


Fig 2.1 Arquitectura general del sistema

2.3.1 Análisis y diseño del sistema

Modelo de diseño

Diagrama de interacción

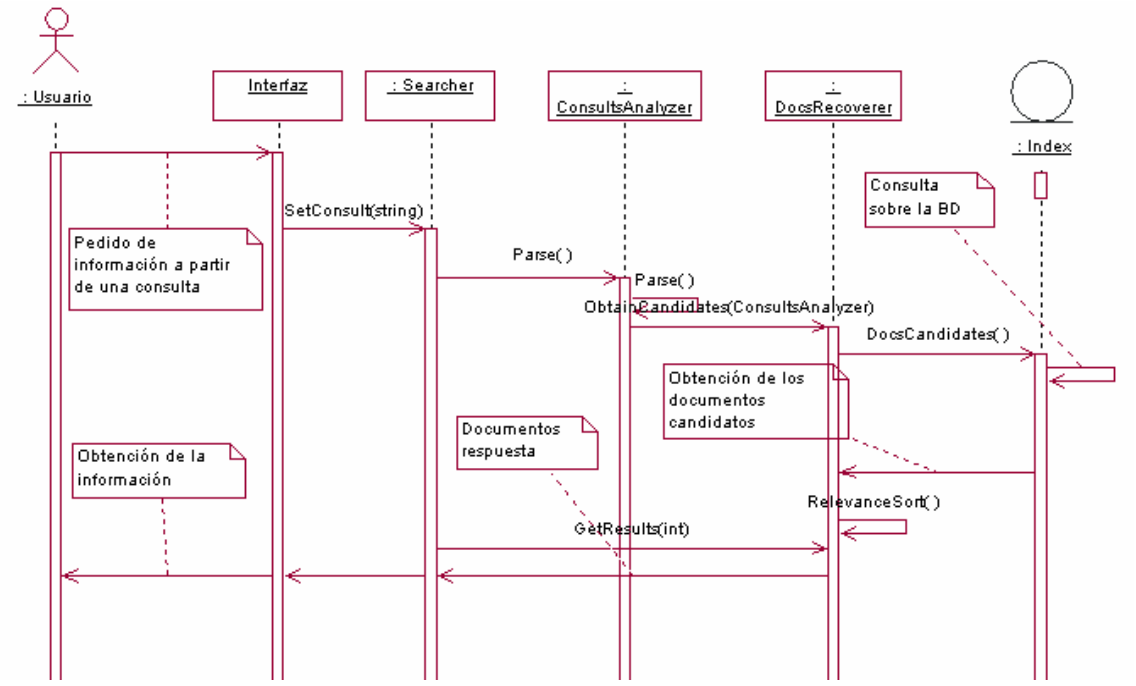
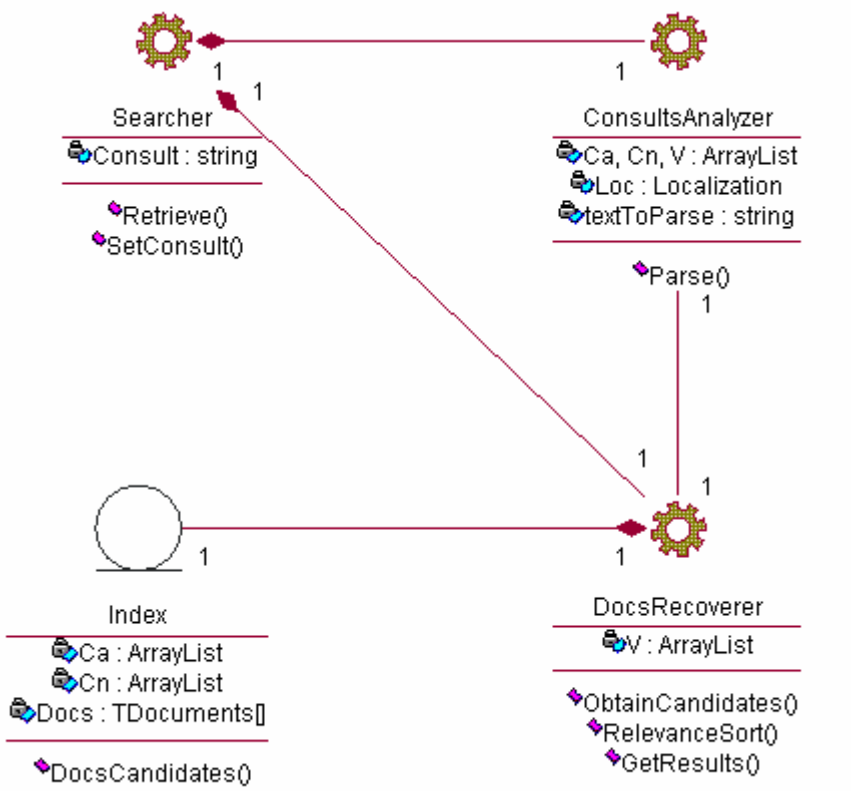


Diagrama de clases



2.3.2 Descripción de las clases

Nombre: Searcher	
Tipo de clase: controladora	
Atributo	**Tipo**
Consult	String
Responsabilidades:	
Nombre:	**Descripción:**
SetConsult(string): void	Operación encargada de la inserción de la consulta a analizar.
Retrieve(): Results[]	Obtiene grupos de documentos respuesta.

Nombre: ConsultsAnalyzer	
Tipo de clase: controladora	
Atributo	**Tipo**

Ca	ArrayList
Cn	ArrayList
V	ArrayList
Loc	Localization
textToParse	string
Responsabilidades:	
Nombre:	**Descripción:**
Parse(): void	Realización de los análisis lexicográficos y sintácticos, así como la generación de la forma interna de la consulta.

Nombre: DocsRecoverer	
Tipo de clase: controladora	
Atributo	**Tipo**
V	ArrayList
Responsabilidades:	
Nombre:	**Descripción:**
ObtainCandidates(Analyzer): void	Se encarga de seleccionar en el índice todos aquellos documentos que tengan alguna relación con la consulta formulada por el usuario.
RelevanceSort(): void	Encargada de dar a los documentos un orden teniendo en cuenta el grado de relevancia de cada uno de los candidatos.
GetResults(int): Results[]	Devuelve grupos (arreglos) de documentos respuesta.

Nombre: Index	
Tipo de clase: entidad	
Atributo	**Tipo**
Ca	ArrayList
Cn	ArrayList
Docs	TDocuments[]
Responsabilidades:	

Nombre:	Descripción:
DocsCandidates(): void	Realiza el acceso a la BD, con el fin de seleccionar los documentos.

2.3.3 Sistema de almacenamiento

Este sistema es el formado por la base de datos en la que se almacena el índice del buscador. Su diseño se corresponde al método de organización de ficheros invertidos, el cual fue escogido por las ventajas ya analizadas dadas, fundamentalmente, por las altas velocidades de acceso que mediante el pueden lograrse. Esta estructura es utilizada en la actualidad en SRI comerciales.

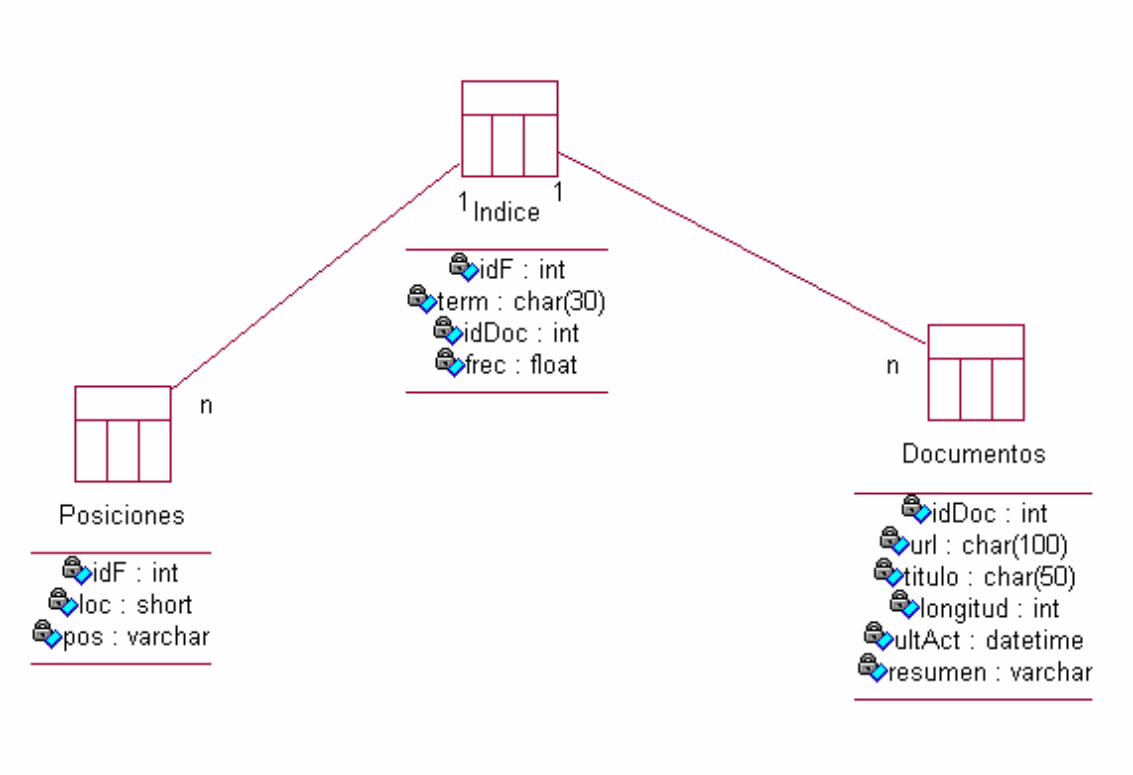


Fig 2.2 Diagrama entidad relación

Descripción de las tablas

Nombre: Indice
Descripción: Tabla encargada del almacenamiento de los términos de indexación, definidora del índice del motor de búsqueda (fichero invertido).

Atributo	Tipo	Descripción
idF	int	Constituye el campo llave.
term	char(30)	Campo para almacenar los términos.
idDoc	int	Identificador del documento en que está presente.
frec	float	Frecuencia absoluta en la que se encuentra ése término en el documento.

Nombre: Posiciones		
Descripción: Destinada a almacenar las posiciones en que se encuentran los términos en los documentos.		
Atributo	**Tipo**	**Descripción**
idF	int	Entrada del fichero invertido.
loc	short	Parte del documento en la que se encuentra el término (título, URL, etc.).
pos	varchar	Posiciones ocupadas por el término en el documento.

Nombre: Documentos		
Descripción: Tabla en la que tienen todos los datos de los documentos recuperados por la *spider*.		
Atributo	**Tipo**	**Descripción**
idDoc	int	Campo llave.
url	char(100)	URL en la que se encuentra el documento.
titulo	char(50)	Título del documento.
longitud	int	Cantidad de términos presentes en el documento.
ultAct	datetime	Fecha de la última actualización sufrida por el documento.
resumen	varchar	El conjunto de términos más relevantes presentes en el texto del documento.

2.3.4 Motor de búsqueda

Es el subsistema encargado de procesar las solicitudes de información elaborada por los usuarios a través de las consultas, dándoles las correspondientes respuestas. Está formado por dos módulos: uno *analizador de consultas*, y otro *recuperador de documentos*; cuenta además con una interfaz para la interacción con los usuarios.

En el *analizador* se efectúa el análisis léxico y sintáctico de la consulta elaborada por los usuarios, obteniéndose una representación adecuada de ella (forma interna), la que es pasada al *recuperador de documentos*, módulo encargado de: seleccionar un conjunto de documentos "candidatos" a ser devueltos como respuesta, y evaluar (medir) la relevancia que ellos presenten de acuerdo a la consulta para en base a dicho valor establecer un orden en los documentos recuperados que se le mostrarán al usuario.

El modelo de recuperación seleccionado es el vectorial, particularmente diseñando, para el pesado de los términos, una variante del *TF* que combina la frecuencia de aparición de los términos con la distancia (entendiéndola como la cantidad de términos existente entre ellos) que los separe entre otras cosas.

2.4 Funcionamiento del sistema

2.4.1 Analizador de consultas

El analizador de consultas es el módulo encargado del análisis lexicográfico y sintáctico de la consulta elaborada por el usuario, así como de la generación de la forma interna correspondiente.

El lenguaje de consulta está formado por operadores booleanos, posicionales y de existencia; la descripción de los mismos ya fue realizada en el epígrafe (1.4), los detalles de implementación serán dados a continuación en el epígrafe.

Seguidamente serán detallados los operadores del lenguaje de consulta, en cada caso se dará la especificación sintáctica (utilizando por notación la Forma Normal de Backus) [9] así como la especificación semántica.

Operadores booleanos

Los operadores incluidos en el lenguaje son: *or* (para indicar necesidad de documentos en los que aparezcan alguno de los términos), *and* (interés en los documentos en los que estén todos los términos) y *not* (exclusión de los términos). En el caso del operador *and* es bueno destacar su carácter no restrictivo: dada una expresión en la forma: pal1 *and* pal2 *and* pal3 *and* pal4, no sólo son relevantes los documentos en los que estén presentes las cuatro palabras, sino también aquellos en los que haya un subconjunto de ellas.

Operadores posicionales

Los operadores posicionales absolutos implementados en el lenguaje están destinados a indicar búsquedas por los campos *título*, *url*, *texto* y *keyword* de los documentos.

Sintaxis:

op ::= campo: (conjunto_de_términos)

campo → 'title' | 'url' | 'text' | 'keyword'

Su semántica consiste en obtener los documentos que posean en el campo especificado alguno de los términos indicados. Es bueno señalar que en una consulta dada se podrá buscar solamente por un campo.

Operadores de existencia

Los operadores de existencia son, tal y como se había mencionado anteriormente: '+' y '-'.

Sintaxis:

op ::= + (lista_de_palabras_claves)

Obliga la presencia de cada una de las palabras afectadas en los documentos recuperados.

op::= - (lista_de_palabras_claves)

Obliga a que los documentos recuperados no contengan las palabras especificadas.

2.4.1.1 Análisis lexicográfico y sintáctico

En la realización de los análisis lexicográfico y sintáctico se ha utilizado el generador freeware de tablas LALR(1) *GOLD Parser* [10], en conjunción con las librerías open-source golddotnet [11] consistentes en el motor de ejecución encargado de interpretar las tablas producidas por *GOLD Parser*.

2.4.1.2 Gramática del lenguaje de consulta

El lenguaje de consulta utilizado es libre de contexto o de tipo II, y por lo tanto puede ser generado también por una gramática (G) de tipo II definida de la forma siguiente:

G = <{I, L, S, T, K, F}, Σ , $\mathcal{P}$, I> donde:

El alfabeto queda definido como Σ ={ε, or, and, not, término, campo, op, (,), +, -}, en el que: ε constituye la cadena vacía, y los tokens quedan especificados como sigue:

or ← [Oo][Rr]

and ← [Aa][Nn][Dd]

not ← [Nn][Oo][Tt]

término ← [A..Za..z0..9ÁÉÍÓÚáéíóú.,?%]+

op ← [Uu][Rr][Ll][:] | [Tt][Ii][Tt][Ll][Ee][:] | [Tt][Ee][Xx][Tt][:]

| [Kk][Ee][Yy][Ww][Oo][Rr][Dd][:]

Las producciones son las siguientes:

$\mathcal{P}$ = {

I → L S

L → op | ε

S → T | T **or** S | T **and** S

T → K F

K → **not** | + | - | ε

F → **término** F | **término** | (S)

}

Nótese la posibilidad de escribir términos de forma consecutiva, en estos casos se interpretará el interés de que sean tratados como si estuvieran relacionados por el operador AND; esto ha sido diseñado así sobre el precepto de que éste operador es el más usado.

2.4.1.3 Forma interna

El objetivo del lenguaje de consulta es proveer al usuario de un mecanismo de expresión de sus necesidades que le resulte cómodo, pero además, para que sea funcional debe permitir extraer de una expresión dada una representación estructurada o *forma interna* en función de la cual se desarrollarán los algoritmos de obtención de documentos candidatos a ser recuperados y de asignación de relevancia. Dicha forma interna está formada por:

- El conjunto de los términos que pueden aparecer en los documentos recuperados C^a. En este caso se encuentran los términos no afectados por el operador '-' o aquellos no influidos por un número impar de operadores *not.*
- El conjunto de términos que están obligados a estar presente en los documentos respuesta C^+.
- El conjunto de términos que no pueden aparecer en las respuestas C^-.
- El campo sobre el que se hará la búsqueda: título, URL, etc.
- Conjunto de vectores de términos pertenecientes a C^a resultado del tratamiento de una consulta como fórmula del cálculo proposicional, y su transformación a la *forma normal disyuntiva* [12] correspondiente.

Toda esta información es generada por el analizador y posteriormente utilizada por el sistema en la selección de los documentos respuesta.

2.4.2 Recuperador de documentos

Este subsistema es el de mayor complejidad y peso en el resultado final del proceso de recuperación. Él debe primeramente, a partir de la representación que de la consulta haga el *analizador* anteriormente descrito, obtener el conjunto de los documentos candidatos a ser devueltos al usuario; y a continuación calcular, para cada uno de estos documentos la medida de cuán relevante resultan a la consulta, con el objetivo de establecer un orden de importancia en dicho conjunto.

2.4.2.1 Selección de los posibles documentos respuesta

Este proceso se realiza a partir de los conjuntos descritos en el apartado anterior, los cuales constituyen la forma interna generada a partir de la consulta formulada por el usuario.

En general el procedimiento a seguir consiste primeramente en la obtención de todos los documentos que contengan al menos un término de los contenidos en C^a, los cuales constituyen en cualquier caso los únicos con algún sentido a ser devueltos.

En el caso que $C^- \neq \varnothing$ se deberán buscar todos aquellos documentos en los que estén presentes algunas de las palabras pertenecientes a dicho conjunto con el fin de excluirlos, denotemos dicho conjunto con $\mathcal{N}$. Por otra parte si $C^+ \neq \varnothing$ se deberá tener presente la necesidad de que todos los elementos contenidos en dicho conjunto deberán aparecer obligatoriamente en los documentos.

Existen múltiples estrategias a seguir para obtener estos conjuntos de documentos, sin embargo, dada la necesidad de lograr tiempos de respuesta muy cortos, la alternativa más lógica, dada la representación de los ficheros invertidos, es a través de consultas del lenguaje SQL. Por ejemplo en el caso de los documentos generados a partir de C^a (tomando C^a como $\{t_1, t_{2, \ldots,} t_n\}$) se podrían obtener de una consulta con la forma: SELECT *IdDoc* FROM *Indice* WHERE *term* in $[t_1, t_{2, \ldots,} t_n]$, y así para los demás conjuntos que tienen en cuenta la presencia de los términos en el campo particular sobre el que se esté desarrollando la búsqueda, es decir: título, texto, keyword o URL, sean estos tres últimos conjuntos denotados por $\mathcal{I}$, $\mathcal{T}$, $\mathcal{K}$ y $\mathcal{U}$ respectivamente, y $\mathcal{S} = \mathcal{I} \cup \mathcal{T} \cup \mathcal{K} \cup \mathcal{U}$.

Se puede concluir que si tomamos a $\mathcal{A}$ como el conjunto de todos los documentos con algún grado de relación con la consulta (los generados a partir de C^a) el conjunto buscado de los documentos candidatos, $\mathcal{D}$, queda definido por:

- $(\mathcal{A} - \mathcal{N}) \cap \mathcal{S}$, si $\mathcal{S} \neq \varnothing$.
- $(\mathcal{A} - \mathcal{N})$ en otro caso.

2.4.2.2 Cálculo de relevancia.

El procedimiento de más peso en la correctitud del proceso de búsqueda es realizado durante el ordenamiento, según el grado de importancia, de los posibles documentos a devolver como respuesta a la solicitud de un usuario. Éste debe tener en cuenta los

siguientes elementos referentes a los términos y los documentos en los que ellos se encuentren:

- Frecuencia de aparición del término.
- Grado de presencia del conjunto de términos.
- Cercanía entre los términos.
- Longitud del documento.
- Presencia de los términos en partes específicas de los documentos: título, URL, meta-etiquetas keywords, etc.

Para lograr el anterior acometido se procede, una vez obtenido el conjunto de documentos candidatos, a calcular la medida de relevancia de cada documento.

En función de esto definamos una función de semejanza como dicha medida, teniendo en cuenta para ello el modelo de recuperación elegido para la implementación del SRI y la representación de las consultas anteriormente vista.

Sea $\mathcal{V}$ ($\mathcal{V} \subset \rho(T)$, donde T es la colección de términos de indexación y ρ es el conjunto potencia), un conjunto de vectores de términos generados a partir de una cadena del lenguaje de consulta anteriormente especificado, según las reglas de transformación de una fórmula del cálculo proposicional en *forma normal disyuntiva* y con la conveniente supresión de los términos pertenecientes a C^-. Por ejemplo:

Una consulta de la forma: (*pal1 pal2 pal3* OR *pal4*) AND *pal5*, daría por resultado el conjunto: $\mathcal{V}$ = {(*pal4, pal5*), (*pal1, pal2, pal3, pal5*)}.

Definamos las siguientes funciones:

f: T × $\mathcal{D} \rightarrow$ R+, tal que *f*(*t*, *d*) da la frecuencia del término *t* en el documento *d*.

w: T × $\mathcal{D} \rightarrow$ R+, *w(t, d)* está encargada de asociar un peso en *d* al término *t* en correspondencia con la parte en el documento en que se encuentre éste: título, URL, keyword, etc., en este caso se ha asumido:

w(*t*, *d*) = 10 si *t* aparece en la URL de *d*.

$w(t, d) = 8$ si t aparece en el título de d.

$w(t, d) = 6$ si t aparece entre las keywords de d.

$w(t, d) = 3$ si t aparece en el texto.

En los casos en que la búsqueda que se esté realizando sea por alguno de estos campos en particular, el valor del peso correspondiente será incrementado.

$l: \mathcal{D} \rightarrow$ N, $l(d)$ calcula la longitud del documento d (cantidad de términos)

$g: \mathcal{V} \times \mathcal{D} \rightarrow \mathcal{L}$, siendo $\mathcal{L}$ un conjunto formado por sucesiones de números enteros ordenados. Esta función, $g(v, d)$, tiene por objetivo generar la relación de posiciones ocupadas por los términos que conforman v, y que están presentes en d.

A partir de la lista de posiciones de términos, se puede pensar, con el fin de caracterizar el documento a partir de las distancias existentes entre ellos (entendiendo dichas distancias como la cantidad de términos), en hacer una clasificación de k clases de *cercanías* a tener en cuenta; asociándoles k pesos a dichas clases. (1)

Dadas estas premisas el cálculo de la medida de la relevancia quedaría dada por:

$r: \mathcal{D} \times \mathcal{L} \rightarrow [0..1]$, según:

$r_{(d, l)} = r_{(d, [l1, l2, \ldots, ln])}$

$$= 0.5 * \sum_{i=1}^{m} \frac{f(t,d) + w(t,d)}{m(l(d) + w(t,d))} + 0.4 * ((\sum_{j=1}^{k} \frac{\alpha_j C_j}{n} - 0.1) * 5) + 0.1 * \omega(l) \qquad (2)$$

donde:

m es la cantidad de términos usados para obtener l

$\omega(l)$ es de los m términos posibles, la cantidad real de ellos presentes en d.

C_j es la cantidad de pares de términos localizados en d a la distancia correspondiente a la clase j -- según (1) --, y

α_j es el peso correspondiente a dicha clase.

Un ejemplo de posible particionamiento del espacio de distancias sería:

Clases C_i (K=5)	Pesos α_i	Rango de distancias	
C1 = 4	0.30	1	4
C2 = 5	0.25	5	14
C3 = 1	0.20	15	26
C4 = 0	0.15	27	39
C5 = 0	0.10	40 en adelante	

Consistente en darle una importancia de 0.30 a la cantidad de términos presentes a una distancia entre 1 y 4, 0.25 a los que estuvieran entre 5 y 14, y así sucesivamente. En la tabla se ha expresado además, un caso particular de una consulta, en la que ocurre la existencia de cuatro términos a una distancia menor a tres, cinco en el intervalo 5 – 14 y uno en la clase 3.

Mediante la función (2) se expresa el realce de la importancia del primer término (mediante su multiplicación por 0.5) consistente en una variante de la técnica *TF* anteriormente mencionada; seguidamente se pesa mediante 0.4 el término relacionado con el análisis de las distancias entre los términos, y, finalmente se le concede a la cantidad de términos existentes en el documento una importancia de 0.1.

Conclusiones

Con la realización de este trabajo se da cumplimiento a los objetivos planteados, pues se lograron los siguientes resultados relacionados con la recuperación de documentos existentes en un índice:

- Se diseñó e implementó un lenguaje de consulta que permite el uso de operadores booleanos, posicionales absolutos y de existencia; los cuales permiten al usuario, en unión con los paréntesis, interrogar al sistema según sus intereses.
- Se hizo una adecuada interpretación de la posible información a expresar a través de dicho lenguaje.
- Se desarrolló un algoritmo de cálculo de relevancia a documentos, que tiene en cuenta la frecuencia de aparición de los términos, la posición ocupada por los mismos, la longitud de los documentos, el número de términos de los presentes en la consulta que estén en el documento y las distancias entre sí a las que estén éstos.

Recomendaciones

Este proyecto constituye un subsistema de un motor de búsqueda, al cual, además del completamiento del SRI se recomienda:

- La incorporación de nuevos operadores al lenguaje de consulta que permitan la elaboración de ecuaciones de búsqueda más complejas.
- La añadidura de un directorio temático como complemento, a la usanza de los SRI comerciales, auxiliado de técnicas de clasificación automática de documentos que pudieran ayudar además en el proceso de asignación de relevancia.
- La realización de pruebas con diferentes criterios a la hora de asignar pesos en el proceso de asignación de relevancia, así como en la selección de rangos de las diferentes clases de cercanías con vistas a lograr la configuración óptima.

Referencia Bibliográfica

[01] Ficha técnica de Google
http://www.infobuscadores.com/google_02.htm (18/03/04).

[02] ¿Por qué usar Google?
http://www.google.com/intl/es/why_use.html (19/03/04).

[03] Ficha técnica de Alltheweb
http://www.infobuscadores.com/alltheweb_02.htm (18/03/04).

[04] Frakes, W.B.; Baeza-Yates, R. *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall. Englewood Cliffs, N.J. 1992.

[05] Greengrass, Ed. *Information Retrieval: a Survey*. November, 2000.

[06] Willet, P. "*Recent trends in hierarchical document clustering: a critical review*". Information Processing & Management 24, 1988.

[07] González Seco, José Antonio *El lenguaje de programación C#,* 2001.

[08] Parihar, Mridula. *La Biblia de ASP .NET*. Anaya Multimedia. 2002.

[09] Alfred V. Aho and Jeffrey D. Ullman. *The Theory of Parsing, Translation and Compiling*. Prentice-Hall, 1973.

[10] Sitio de GOLD Parser
http://www.devincook.com/GOLDParser/

[11] Sitio de GoldDot
http://golddotnet.sourceforge.net/golddot.php

[12] García Garrido, Luciano. *Introducción a la Teoría de Conjuntos y a la Lógica.* 2002.

Bibliografía

1. Alfred V. Aho and Jeffrey D. Ullman. *The Theory of Parsing, Translation and Compiling*. Prentice-Hall, 1973.
2. Brin, Sergey; Page, Lawrence. The Anatomy of a Large-Scale Hypertextual Web Search Engine. Stanford University, 1998.

   http://www.n3labs.com/pdf/brin98anatomy.html(25/3/2004)

3. Charte, Ojeda Francisco. *Programación con Visual C# .NET.* Anaya Multimedia. 2002.
4. Frakes, W.B.; Baeza-Yates, R. *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall. Englewood Cliffs, N.J. 1992.
5. García Garrido, Luciano. *Introducción a la Teoría de Conjuntos y a la Lógica.* 2002.
6. González Seco, José Antonio *El lenguaje de programación C#,* 2001.
7. Greengrass, Ed. *Information Retrieval: a Survey*. November, 2000.
8. Hammer, J.; Garcia-Molina, H. Extracting Semistructured Information from the Web. Stanford University, 1997.

   http://oak.cs.ucla.edu/~cho/papers/cho-extract.pdf (27/3/2004)

9. Mendelzon, Alberto. Querying the World Wide Web. University of Toronto, 1997.

   http://master.cpe.ku.ac.th/~wkitsana/Courses/204551_Advanced_Database_

   Syste/websql.pdf (15/2/2004)

10. Parihar, Mridula. *La Biblia de ASP .NET*. Anaya Multimedia. 2002.
11. Willet, P. "*Recent trends in hierarchical document clustering: a critical review*". Information Processing & Management 24, 1988.

Anexos

Anexo 1. Expansión de los casos de uso.

Caso de uso: CU-01	
Actor: Navegante	
Descripción: El usuario elaborará una consulta a partir de sus necesidades informativas y la entregará al sistema.	
ACTOR	RESPUESTA DEL SISTEMA
1. El caso de uso comienza cuando el usuario solicita la búsqueda a través del botón existente para ello, luego de haber elaborado la consulta en el cuadro de edición habilitado a tal efecto.	
	2. El sistema traduce la consulta en una expresión adecuada para ser utilizada en la recuperación de los documentos.
	3. El sistema obtiene todos aquellos documentos que se ajustan a la especificación del usuario.
	4. El sistema ordena los documentos obtenidos aplicando criterios de asignación de relevancia.

	5. El sistema muestra en el browser los resultados de la consulta en el orden previamente obtenido.

Glosario de términos

Buscador web: SRI orientado hacia la recuperación de documentos web, de funcionamiento totalmente automático, conformado básicamente por una spider, un indexador y un mecanismo de formulación de consultas.

Consulta: operación realizada por un usuario sobre un SRI, consistente en la formulación de una expresión que pude incluir operadores y que tiene por objetivo expresar una necesidad de información.

Directorio web: herramienta destinada a localizar información en el web, de construcción manual, organizada siguiendo un principio jerárquico de clasificación de documentos en categorías previamente definidas.

Forma interna: representación que de una expresión de un lenguaje dado se hace con el fin de facilitar su tratamiento computacional.

Grafo: estructura para la representación de información, conformada por colecciones de nodos entre los que existen relaciones (arcos).

Indexación: operación realizada por un SRI destinada a dar una representación adecuada a los documentos, a partir de un conjunto de términos de indexación, con el fin de facilitar la recuperación de estos.

***TF*:** del inglés Term Frequency, técnica de pesado de términos de indexación basada en la cantidad de veces que éstos aparezcan en los documentos.

LALR(1): algoritmo de análisis sintáctico (parsing) que se auxilia de una tabla para la realización del proceso, dada su velocidad y expresividad es el más usado en la implementación de los lenguajes de programación comerciales.

Palabras clave: dentro de un conjunto de palabras, aquellas de mayor valor semántico. En el caso de los documentos web son normalmente incluidas en las meta-etiquetas keywords.

Recuperación de información: en inglés Information Retrieval (IR): área de investigación de la Inteligencia Artificial encargada del estudio de técnicas de acceso rápido y automático a grandes volúmenes de información de cualquier tipo.

Spider: también conocido por robot, es el programa encargado de recorrer la web seleccionando la información con la que se alimentan las bases de datos que sirven para dar respuestas a las consultas que los usuarios le formulan al motor de búsqueda.

Término de indexación: constituye la unidad de información a manejar en RI, puede estar dado por palabras (el más común), frases, conceptos, etc.