9.

USO DE TÉCNICAS IR PARA LA GENERACIÓN DINÁMICA DE WEB SITES

Alumnos : JAVIER E. BROHME
VIVIANA L. HERNÁNDEZ

Director : Dr. GUSTAVO H. ROSSI

Trabajo de Grado de Licenciatura en Informática
Facultad de Ciencias Exactas
Universidad Nacional de La Plata
Argentina
Septiembre de 1997

INDICE

Desarrollo

Anexo

AGRADECIMIENTOS

En el desarrollo de este trabajo contamos con la ayuda y colaboración de varias personas, con lo cual nos gustaría agradecerles :

En primer lugar queremos agradecer a nuestro director *Gustavo Rossi* y a *Fernando Das Neves* quien actuó como nuestro co-director, guiándonos y ayudándonos con sus consejos y críticas acerca de este trabajo, el cual sin su ayuda no hubiera sido posible.

Un agradecimiento muy especial a *Tooby* , incansable manual interactivo de redes, Linux, C,etc. Además colaboró con Soft, Hard y material bibliográfico. Su colaboración nos permitió resolver muchos de los problemos técnicos que surgieron en el desarrollo.

No podemos dejar de mencionar a *nuestros padres*, quienes nos dieron la oportunidad de estudiar y nos apoyaron en todo momento.

Además agradecemos a *Gusti* por prestarnos material bibliogáfico; a *Majo* por sus consejos acerca de la redacción en Inglés; a *Tincho* por dar a conocer sus trucos que facilitaron la edición de este documento y a *Moniquita* que nos facilitó la impresora y su máquina .

Viviana y Javier

Septiembre 1997

Introducción

CAPITULO 1

INTRODUCCIÓN

INTRODUCCIÓN

Motivaciones y Objetivos

El campo de la informática referente a la Recuperación de Información, al cual nos referiremos a lo largo de la presentación de la tesis como "IR" (Information Retrieval), es muy amplio, trata el almacenamiento automático y la recuperación de documentos de texto . Los sistemas de Recuperación de Información fueron originalmente desarrollados para ayudar en el manejo de la vasta literatura científica que se ha desarrollado desde 1940. Este es aún hoy el uso mas común de los sistemas IR .

Al ser un campo no muy difundido entre la comunidad informática, nos pareció interesante desarrollar una tesis basada en la Recuperación de Información, con el propósito de no sólo investigar y presentar nuestras conclusiones y conocimientos sobre el tema , sino de llevar a cabo un desarrollo donde se pudieran aplicar las técnicas aprendidas y obtener resultados y evaluaciones sobre ellas. También nos pareció interesante la idea que el desarrollo pudiera ser de utilidad, es decir que una vez finalizado llegara a tener algún fin práctico.

El **objetivo** será desarrollar un sistema capaz de extraer información de documentos conocidos como Call For Papers (CFP), almacenar dicha información y a partir de consultas generar en forma dinámica páginas HTML con los resultados obtenidos, con el fin de facilitar a la comunidad de investigadores la presentación y planeamiento de sus trabajos.

Los **documentos CFP** son invitaciones a participar en conferencias enviados por los organizadores (a través de Internet ya sea en forma particular o por medio de listas de interés) , a una comunidad afín al tema de dicha conferencia. El objetivo es difundir el desarrollo de la misma e invitar a participar en ella mediante la presentación de papers.

La gran cantidad de conferencias que se realizan ha provocado la existencia de un gran volumen de información referente a los llamados a conferencias. Con lo cual esta información se ha vuelto difícil de registrar y clasificar. La información sobre CFPs está desorganizada por no contar con una forma rápida y eficaz de clasificación y consulta de los mismos. Los autores de los CFPs muchas veces incluyen un pedido de disculpa en caso que el receptor reciba ese CFP más de una vez.

La presentación del sistema a lo largo del desarrollo de esta tesis será acompañada por gráficos esquemáticos del mismo, los cuales se irán detallando cada vez más a medida que se van profundizando los temas . Estos esquemas vendrán acompañados con un número de nivel el cual representa el grado de detalle asociado. A medida que se incrementan los niveles significa que el esquema presenta un detalle más profundo con respecto al nivel anterior . De esta forma analizando los esquemas a lo largo de esta presentación se puede conocer el funcionamiento del sistema .

El esquema general del sistema es el siguiente :

Nivel 0 *Presentación del sistema*

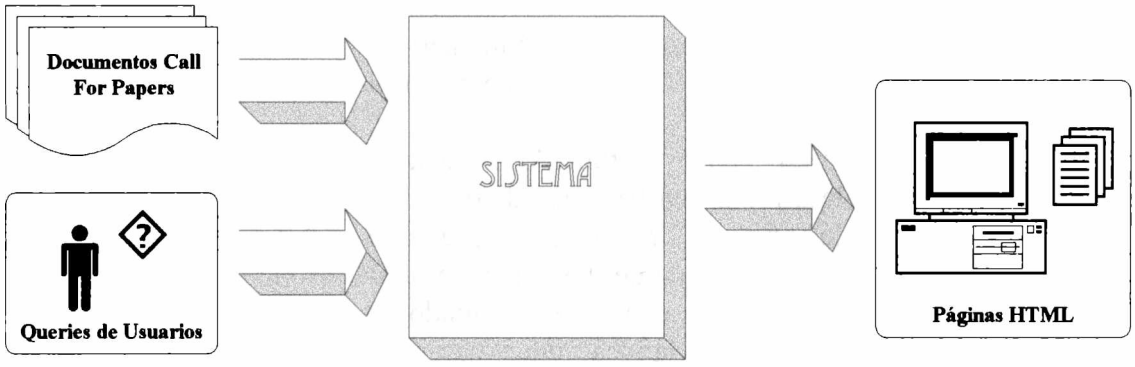


Figura 1.1 *Esquema General del Sistema*

En este esquema se representa la interacción del sistema con la entrada y la salida .

La **entrada** al sistema comprende tanto documentos como queries. Los documentos son del tipo CFP y el sistema tendrá por función, a partir de estos documentos recuperar sus datos relevantes y permitir la consulta de los mismos . Por otro lado se tiene como entrada queries por parte de usuarios, los cuales consistirán en consultas acerca de documentos de tipo CFP, estas consultas se realizan mediante una página HTML creada para tal fin .

La **salida** del sistema comprende páginas HTML creadas dinámicamente. Las mismas presentan los resultados de una consulta ó son páginas especiales con información referente a posibles errores. Esta salida es dirigida al usuario que utiliza el sistema .

ORGANIZACIÓN

La presentación de esta tesis se divide en varios capítulos. A continuación se enumeran cada uno de ellos acompañados con una breve descripción de su contenido.

1.- Introducción

Es el presente capítulo.Comprende la presentación de las motivaciones y objetivos de la propuesta a realizar y los temas que involucra .

2.- Nuestro Enfoque

Describe una presentación conceptual del Sistema de Recuperación de Información de Documentos CFPs y del Sistema de Consulta de Información a desarrollar .

3.- Information Retrieval

Incluye las características destacables de este tema.

3.1 Introducción

Presenta una visión general del problema de la Recuperación de Información, el origen de este problema y las necesidades actuales sobre este campo de la informática .

3.2 Sistemas IR

Incluye la definición y características de los sistemas de Recuperación de Información,clasificación,visión esquemática y funcional .

3.3 Análisis de Contenidos de Documentos

Detalla el análisis que se realiza a los documentos de manera tal de obtener una representación de los mismos. Comprende la base principal de la Recuperación de Información.

3.3.1 Representación de Documentos

Se presenta la idea de representación de documentos .

3.3.2 Análisis de texto automático

Incluye un análisis detallado de las técnicas de Recuperación de Información. Las mismas facilitan la generación de las representaciones de los documentos.

3.4 Estrategias de Búsqueda

Presenta diferentes estrategias de búsqueda para recuperar información .

3.5 Evaluación de los Sistemas IR

Presenta las diferentes medidas de evaluación de los sistemas de Recuperación de Información. Estas medidas permiten cuantificar los resultados de una evaluación de un sistema IR.

4.- Internet y World Wide Web

Incluye las características destacables de este tema.

4.1 Internet

Presenta las características básicas de Internet, de manera tal de conocer sus orígenes y organización general. Incluye una revisión básica de los servicios que brinda, poniendo énfasis en el servicio de correo electrónico (*e-mail*), ya que éste cumple un papel de importancia en el desarrollo del sistema .

4.2 Estructura General de la WWW

Presenta las características básicas y arquitectura de la WWW. Permite tener una visión global de la misma.

4.3 Búsqueda y Recuperación de Información en la WWW

Se presentan las herramientas existentes que facilitan la búsqueda y la Recuperación de Información en WWW. Se presentan los distintos buscadores, analizando las características de los mismos, sus ventajas y desventajas .

4.4 Estructura de las Aplicaciones en la WWW

Incluye una presentación del lenguaje utilizado en la creación de páginas en WWW, llamado HTML (HyperText Markup Languaje) . Se describen las características básicas del mismo de manera tal de comprender su filosofía y la creación de páginas. También se analiza la extensión de la capacidad del servidor a través de CGI (Common Gateway Interface). Se presentan las características básicas de CGI.

5.- Sistema IR de Documentos CFPs

En este capítulo se presenta el sistema desarrollado, el cual procesa documentos CFPs recuperando información a partir de ellos.

5.1 Introducción

Definición general de los objetivos y ventajas del sistema .

5.2 Definición Conceptual del Sistema IR de Documentos CFPs

Detalles del proceso de recuperación. Se analiza cómo actúan en conjunto, reglas heurísticas y técnicas IR en este proceso. Se analiza el concepto del motor de recuperación de información de los documentos CFPs.

5.3 Arquitectura del Sistema IR de Documentos CFPs

Presentación de los módulos que componen el sistema y la interacción entre ellos.

6.- Sistema de Consulta de Documentos CFPs

En este capítulo se presenta el sistema desarrollado, el cual permite consultas sobre un conjunto de documentos CFPs, presentando los resultados de la misma a través de páginas HTML construidas dinámicamente.

6.1 Introducción

Definición general de los objetivos del sistema .

6.2 Definición Conceptual del Sistema de Consulta de Documentos CFPs

Detalles del proceso que permite realizar las consultas sobre la colección de documentos y del proceso de construcción de páginas HTML dinámicas. Incluye la descripción de las opciones que brinda la consulta y el formato de presentación de los resultados al usuario .

6.3 Arquitectura del Sistema de Consulta de Documentos CFPs

Comprende detalles referidos a la estructura interna del proceso de consulta y muestra de datos (programa CGI). Se brindan detalles acerca del proceso de consulta por parte de un usuario y la generación dinámica de páginas HTML.

7.- Evaluación del Sistema IR de Documentos CFPs

En este capítulo se incluye un análisis de los resultados obtenidos del sistema, cuantificados según medidas de evaluación de los sistemas IR.

8.- Conclusiones

Incluye las conclusiones a las que llegamos luego de realizar este desarrollo, los inconvenientes que surgieron a lo largo del mismo y una visión de posibles ampliaciones a realizar .

9.- Glosario

Incluye definiciones de términos específicos que se mencionan en la presentación de la tesis .

10.- Referencias Bibliográficas

Presentación detallada de la bibliografía utilizada .

Nuestro Enfoque

Capitulo 2

NUESTRO ENFOQUE

Presentación de los sistemas desarrollados y la interacción entre ellos

Tras analizar una gran variedad de CFPs de diferentes áreas temáticas y procedencias, pudimos establecer que un CFP en el general de los casos, incluye ciertos datos como el título, fecha, lugar de realización de la conferencia (ciudad y/o país), el tema a tratar y las características y/o limitaciones que deben tener los trabajos (papers) a presentar. Incluyen también otro tipo de datos como el comité organizador, fecha de vencimiento del recibo de papers , fecha de notificación de la aceptación o rechazo de los mismos y direcciones donde se puede obtener más información acerca de la conferencia (URLs). Además encontramos que no siempre un CFP hace referencia en forma unívoca a una conferencia, en algunos casos también se hace referencia a una conferencia asociada .

Debido a que el idioma Inglés es el más utilizado en la Web y la mayoría de los CFPs analizados (mas del 95 %) estaban escritos en este idioma, el sistema analiza documentos de cualquier procedencia y área temática siempre que estén escritos en Inglés. En caso en que estén escritos en otro idioma, el sistema puede llegar a recuperar ciertos datos pero los mismos podrían no ser confiables ya que no está preparado para ésto .

El ***Sistema IR de Documentos CFPs*** tiene como objetivo la recuperación de la información contenida en estos documentos CFPs *en forma automática.* Aunque una manera de organizar esta información podría haber sido en forma manual a través de una persona a cargo, a menudo ésto no es adecuado sea por la falta posible de personal dedicado a esta tarea o porque se podrían usar criterios personales en el filtrado de la información. Todo esto lleva a la idea de automatizar la recuperación de la información de los CFPs. Además actualmente según nuestro conocimiento,no existe una herramienta que obtenga los datos en ésta forma .

Este sistema recupera términos con un significado asociado y trabaja con una muestra incremental de documentos la cual va creciendo en el tiempo. Esta muestra a priori no es representativa de algún dominio y además el objetivo del sistema no es enfocar un dominio en particular .

La recuperación de información se lleva a cabo a través de un proceso que utiliza reglas heurísticas complementadas con técnicas de recuperación de información. Las reglas heurísticas las desarrollamos a partir del análisis de documentos CFPs y permiten recuperar información utilizando *pattern matching* y *recuperación basada en contexto* .

Para poder realizar la consulta de los datos recuperados desarrollamos un ***Sistema de Consulta de Documentos CFP***. En este punto interviene el otro tema de estudio que es Internet y World Wide Web, debido que para realizar las consultas sobre esta información desarrollamos una interfase en WWW.

Utilizamos WWW como medio de publicación ya que es un medio que nos permite proveer información potencialmente útil en forma tal que pueda ser accedida por diferentes usuarios trabajando en distintas plataformas y en lugares distantes. También facilita a los usuarios poder acceder a información almacenada sin requerir conocimiento acerca de los mecanismos subyacentes de implementación de tales accesos .

Este sistema permite consultas según diferentes criterios simples o combinados. A partir de estas consultas generamos en forma dinámica páginas HTML con los resultados de las mismas. La idea de "páginas HTML dinámicas" implica que en vez de crear páginas HTML que contienen texto, gráficos o sonido estático que nunca cambia de una sesión a otra, las páginas son creadas al momento por el servidor (según los requerimientos del usuario) sin almacenarse en un archivo .

Para comprender mejor ambos sistemas desarrollados, el siguiente esquema representa la interacción entre ellos :

Nivel 1 *Interacción de los Sistemas desarrollados*

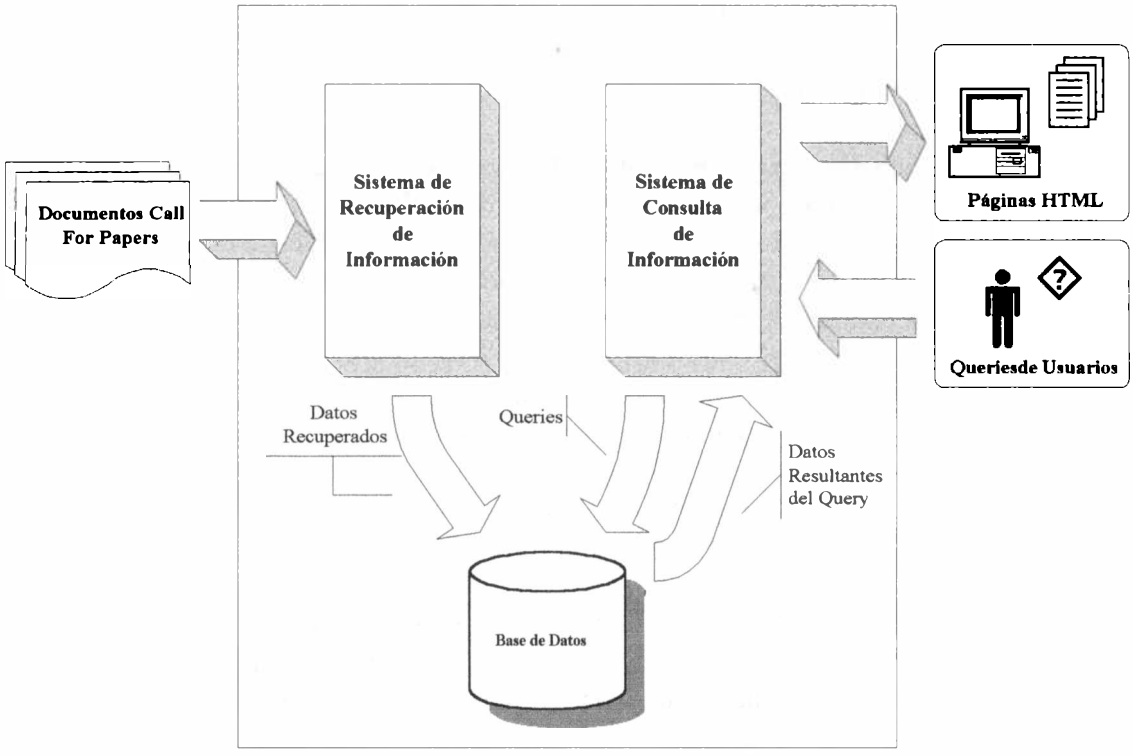


Figura 2.1 *Sistemas de Recuperación y Consulta de Información de Documentos CFPs*

Sistema de Recuperación de Información de Documentos CFPs

Como entrada recibe documentos del tipo CFP. Tiene como función recuperar la información considerada relevante de los mismos y esta información es almacenada en una BDs.

Sistema de Consulta de Información de Documentos CFPS

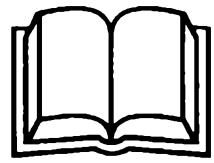
Como entrada recibe *queries* por parte de usuarios solicitando información acerca de CFPs. Este sistema consulta a la BDs obteniendo la información solicitada, en caso que exista. Como salida el sistema presenta al usuario páginas HTML creadas dinámicamente con esta información. En caso en que ocurra algún tipo de error, se presentan páginas HTML con información referente al mismo también creadas dinámicamente.

Podemos observar a partir del esquema presentado en la *Figura 2.1*,que ambos sistemas interactúan entre sí mediante la BDs que contiene la información referente a un conjunto de documentos CFPs .

Con el fin de comprender los detalles del desarrollo de cada sistema, en los capítulos siguientes presentamos en profundidad los temas estudiados destacando los puntos de mayor interés intervinientes en el sistema .

Por último queremos destacar que consideramos que el desarrollo de ambos sistemas además de reflejar los temas estudiados, facilitará a los investigadores un acceso más organizado y simple a los CFPs, que son recurso básico para poder desarrollar la difusión de sus tareas en los ámbitos académico y profesional .

Temas de Estudio



Information Retrieval

CAPITULO 3

Este capítulo incluye :

INTRODUCCIÓN A LA RECUPERACIÓN DE INFORMACIÓN

Introducción a la idea de recuperación de información y discusión sobre los problemas existentes

Para introducirnos en el tema de la Recuperación de Información primero nos tenemos que remontar al origen de la información textual.

El texto es la vía primaria por la cual se almacena el conocimiento de los seres humanos, y después de la palabra es la vía primaria de transmisión. Las técnicas para el almacenamiento y búsqueda de documentos textuales son casi tan antiguas como el mismo lenguaje escrito .

Desde 1940 el problema del almacenamiento y recuperación de información ha atraído la atención de los investigadores. Cada día se tiene mayor cantidad de información, y acceder a ésta información en forma veloz y precisa se vuelve una tarea dificultosa. En ciertos casos se ignora la información relevante, es decir que no se descubre la información que realmente interesa. Esto sucede debido a que se tiene un gran volumen de información y no se cuenta con herramientas adecuadas para su recuperación. Este problema provoca una duplicación de esfuerzo y trabajo.

En principio, el almacenamiento y recuperación de información es simple. Si suponemos que tenemos una biblioteca por ejemplo, que contiene diferentes documentos. Una persona (un usuario de la biblioteca) formula una pregunta (consulta o *query*), para la cual la respuesta es un conjunto de documentos que satisfacen la información que solicita el usuario. ¿Cómo puede el usuario obtener este conjunto de documentos ?. Lo que podría hacer es leer todos los documentos que se encuentran en la biblioteca reteniendo los documentos relevantes para su necesidad y descartando el resto, esto constituiría una "recuperación perfecta". Obviamente esta solución es impracticable. El usuario o no tiene tiempo o no desea leer la colección completa de documentos. Además puede ser físicamente imposible para el usuario realizar esta tarea .

Cuando surgieron las computadoras de alta velocidad muchos pensaron que una computadora podría se capaz de "leer" una colección de documentos y obtener los documentos relevantes. Pronto se llegó a la conclusión que utilizar el texto en lenguaje natural de un documento no sólo causa problemas de almacenamiento y entrada de datos, sino que deja sin resolver el problema intelectual de caracterizar el contenido de un documento, es decir obtener una representación del mismo.

Intentar duplicar el proceso humano de "leer" es realmente un problema importante. En forma más específica, "leer" implica intentar extraer información semántica y sintáctica de un texto y utilizarla para decidir cuándo un documento es relevante o no para un determinado requerimiento de información. La dificultad consiste no sólo en saber cómo extraer la información, sino también cómo usarla para decidir la relevancia de un documento.

La computación ha cambiado la forma de almacenar un texto, buscarlo y recuperarlo y muchos problemas de almacenamiento y recuperación se mejoraron. Sin embargo el problema de una recuperación de información efectiva permanece sin resolver. Consideramos una *recuperación efectiva* cuando se recupera la mayor cantidad de información relevante posible .

Hemos mencionado en varios ocasiones la idea de "relevancia" de un documento. Justamente ésta es la noción central de la recuperación de información. El propósito de una estrategia de recuperación automática es recuperar todos los documentos relevantes y al mismo tiempo la menor cantidad posible de documentos no-relevantes. La *representación de un documento* tiene que ser de tal manera que cuando el documento representado se considera relevante o apropiado para un *query* dado, se pueda recuperar éste documento en respuesta a dicho *query*. Estas representaciones de documentos se obtienen en el momento del proceso de indexación de un conjunto de documentos, su definición depende del tipo de proceso que se realice, es decir si se realiza el índice en forma manual o en forma automática.

Cuando se realiza un índice *manualmente*, se caracteriza o representa a los documentos asociándoles términos índices a los mismos. Estos términos índice se encuentran anticipándose a los términos que un usuario emplearía para recuperar cada documento cuyo contenido se quiere describir. Implícitamente, lo que se hace es construir *queries* para los que cada documento sería relevante.

Cuando se realiza un índice *automáticamente* se asume que, si se aplica el mismo análisis automático a un documento de texto y a un *query* la salida será una representación del contenido del documento de tal forma que, si el documento fuera relevante para un *query* ésto se podría conocer a través de un procedimiento computacional.

SISTEMAS IR

Presentación de los conceptos básicos referentes a los Sistemas de Recuperación de Información

El subcampo de la ciencia de la computación que trata el almacenamiento automático y recuperación de documentos se conoce como "***Information Retrieval***" (IR) .

La definición de un sistema IR se puede ajustar a la dada por Lancaster [vanRijsbergen79] , la cual dice:

'Un Sistema de Recuperación de Información (IR) no informa (es decir, no cambia el conocimiento) al usuario acerca del tema sobre el cual está requiriendo información, sino que informa sobre la existencia o no existencia de los documentos referidos a ese tema, y dónde se encuentran los mismos.'

Los sistemas IR automáticos fueron desarrollados originalmente con el objetivo de ayudar en el manejo de la extensa literatura científica que se ha desarrollado desde la década del '40. Este es aún hoy el uso más común de los sistemas IR.

Actualmente muchas universidades, corporaciones y bibliotecas públicas utilizan sistemas IR para proveer un acceso a libros, publicaciones y otro tipo de documentos. Todo campo que utilice documentos para realizar su trabajo, podría beneficiarse con las técnicas IR.

En este punto nos podemos preguntar qué es lo que realmente hace un sistema IR. La función de un sistema IR es encontrar un *matching* entre *queries* de un usuario y documentos almacenados en una BDs.

Al hablar de documentos nos referimos a objetos de datos usualmente textuales, aunque también podrían contener otros tipos de datos como por ejemplo, gráficos, fotografías, etc. A menudo estos documentos no están almacenados en la BDs en forma completa, en vez de esto el sistema utiliza una representación del mismo. Por ejemplo si consideramos al capítulo de un libro como un documento, se podría tener en el sistema una representación del mismo que incluyera sólo el título, autor y un resumen de dicho capítulo. Esta idea de utilizar representaciones de documentos en un sistema IR generalmente se realiza para incrementar la eficiencia del sistema, ya que de esta manera, se reduce el tiempo de búsqueda y el tamaño de la BDs.

Un sistema IR típico, provee ciertas operaciones básicas tales como agregar, borrar y cambiar documentos en una BDs y proveer una vía por la cual los usuarios puedan buscar documentos por medio de *queries* y examinar los documentos recuperados.

Clasificación de los Sistemas IR

Podemos tener dos clases de sistemas IR, los experimentales y los operacionales[VanRijsbergen79]. La diferencia entre ellos se basa en su evaluación.

➲ Los *Sistemas IR Experimentales* generalmente se desarrollan en un 'laboratorio' y tienen como objetivo realizar estudios e investigaciones acerca de los sistemas IR. Son evaluados comparando los experimentos de recuperación con estándares especialmente construidos para un determinado propósito.

➲ Los *Sistemas IR Operacionales* son sistemas comerciales que cobran por el servicio que proveen. Estos sistemas también llamados 'sistemas del mundo real ' , son evaluados en términos de la 'satisfacción del usuario' y el precio que un usuario pagaría por utilizar sus servicios .

Visión Esquemática de un Sistema IR

Antes de profundizar aún mas sobre los sistemas IR, podemos ilustrarlos de una manera general, diferenciando tres componentes principales[VanRijsbergen79] :

- Entrada
- Proceso
- Salida

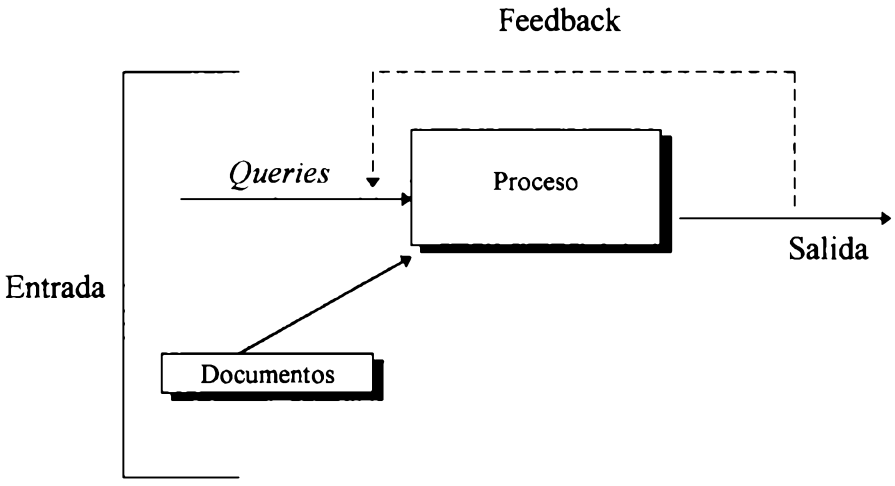


Figura 3.1

Esquema de los componentes de un sistema IR típico

El primer componente (*la entrada*), se refiere a la incorporación de datos al sistema. La *entrada* al sistema comprende tanto a los *queries* que realiza el usuario solicitando información, como a la incorporación de documentos a la BDs. Como se ha mencionado antes, en general los sistemas IR almacenan sólo una representación de los documentos. Por lo tanto el principal problema en este caso es justamente obtener una representación adecuada de cada documento para ser utilizada por el sistema .

Una *representación de un documento* podría ser por ejemplo, una lista de palabras extraídas del mismo consideradas representativas o significativas de dicho documento.

Una alternativa con la que cuentan ciertos sistemas IR *on-line*, es permitir al usuario alterar su pedido de información durante una sesión de búsqueda según la información recuperada, y de esta manera mejorar la recuperación. Este procedimiento se conoce como "*feedback*" .

El segundo componente (*el proceso*), se refiere al *proceso* de recuperación propiamente dicho. Este puede involucrar varias facetas, tal como estructurar la información de una manera apropiada, por ejemplo clasificarla según algún criterio. Incluye también la ejecución de la estrategia de búsqueda en respuesta al *query* del usuario, es decir la función de recuperación.

Por último se hace referencia a la *salida* del sistema. Usualmente comprenderá un conjunto de citas o numeraciones de aquellos documentos que se consideren apropiados para el requerimiento del usuario.

Este esquema permite tener un concepto general de las partes principales que intervienen en un sistema IR típico.

Operaciones sobre un Sistema IR

Un sistema IR incluye operaciones que se realizan sobre los *queries* del usuario y sobre los términos (en general los términos se refieren a las palabras dentro de un texto) del documento[BaezaYates-Frakes92].

Operaciones sobre Queries

Los *queries* son sentencias formales que representan una necesidad de información por parte de un usuario. Estos *queries* actúan como una entrada al sistema IR. Las operaciones sobre los *queries*, dependen del tipo de *query* y de las capacidades del sistema IR. Una operación común sobre *queries* es el *parsing*, el cual consiste en dividir el *query* en sus elementos constituyentes. Por ejemplo si se tiene *queries* booleanos, estos deben ser divididos en los términos y operadores lógicos. Se recupera el conjunto de identificadores de documentos asociado con cada término del *query*, y luego estos conjuntos son combinados de acuerdo a los operadores booleanos que intervienen en el *query*.

Cuando el sistema IR provee la facilidad de *feedback*, se utiliza la información de las búsquedas anteriores para modificar los *queries*. Por ejemplo los términos que se encuentran en documentos relevantes pueden ser agregados al *query* para un nueva búsqueda, y los términos no relevantes pueden ser borrados.

Operaciones sobre Términos

Las operaciones sobre términos permiten normalizar el vocabulario, y estas operaciones favorecen tanto en la indexación de documentos como en el procesamiento de *queries*. Al hablar de indexación se hace referencia al proceso de obtener a partir del texto del documento una representación del mismo.

Existen diferentes operaciones que pueden realizarse sobre los términos tales como:

❒ ***Stoplist*** : Una *stoplist* es una lista de palabras que se consideran que no tienen un valor de índice, es decir que no tienen peso para discriminar información. Esta lista se utiliza para eliminar términos que no serán relevantes para la representación de un documento. Si un término se encuentra en esta lista, es eliminado como posible término índice .

❒ ***Truncation*** : Es una fusión o combinación manual de términos usando caracteres de búsqueda múltiple en la palabra (caracteres comodines), de tal forma que el término puede hacer *matching* con múltiples palabras. El usuario utiliza esta operación en sus pedidos de información.

❒ ***Peso de términos*** : Esta operación le asigna a los términos valores numéricos basados en información acerca de la distribución estadística de los mismos, es decir la frecuencia con la cual los términos ocurren en documentos, colecciones de documentos o subconjunto de colecciones de documentos.

❒ ***Stemming*** : Es una fusión o combinación automática de palabras relacionadas morfológicamente. Usualmente se realiza reduciendo las palabras a su forma raíz. Esta operación se puede utilizar tanto en la indexación de documentos como en el procesamiento de *queries* .

❒ ***Thesaurus*** : Permite otra operación de combinación de términos relacionados. Incluye una lista de términos sinónimos y en ciertos casos las relaciones entre ellos.

Visión Funcional de un Sistema IR Típico

La *Figura 3.2* representa las actividades asociadas a un típico ***sistema IR booleano*** [BaezaYates-Frakes92]. Este tipo de sistema representa la operación estándar que realiza un sistema IR .

En un sistema IR booleano los documentos se representan por un conjunto de palabras claves.

Los *queries* booleanos son palabras claves conectadas por operadores lógicos booleanos (AND,NOT,OR) .

Este esquema permite tener una idea de cómo es el funcionamiento general de un proceso de recuperación de información. Además presenta las operaciones que se pueden hacer tanto sobre los *queries* de los usuarios como sobre los términos del documento.

Se reflejan varias operaciones pero no es necesario que se lleven a cabo todas ellas. El conjunto de operaciones a realizar es opcional. Incluye las operaciones sobre *queries* y términos mencionadas anteriormente .

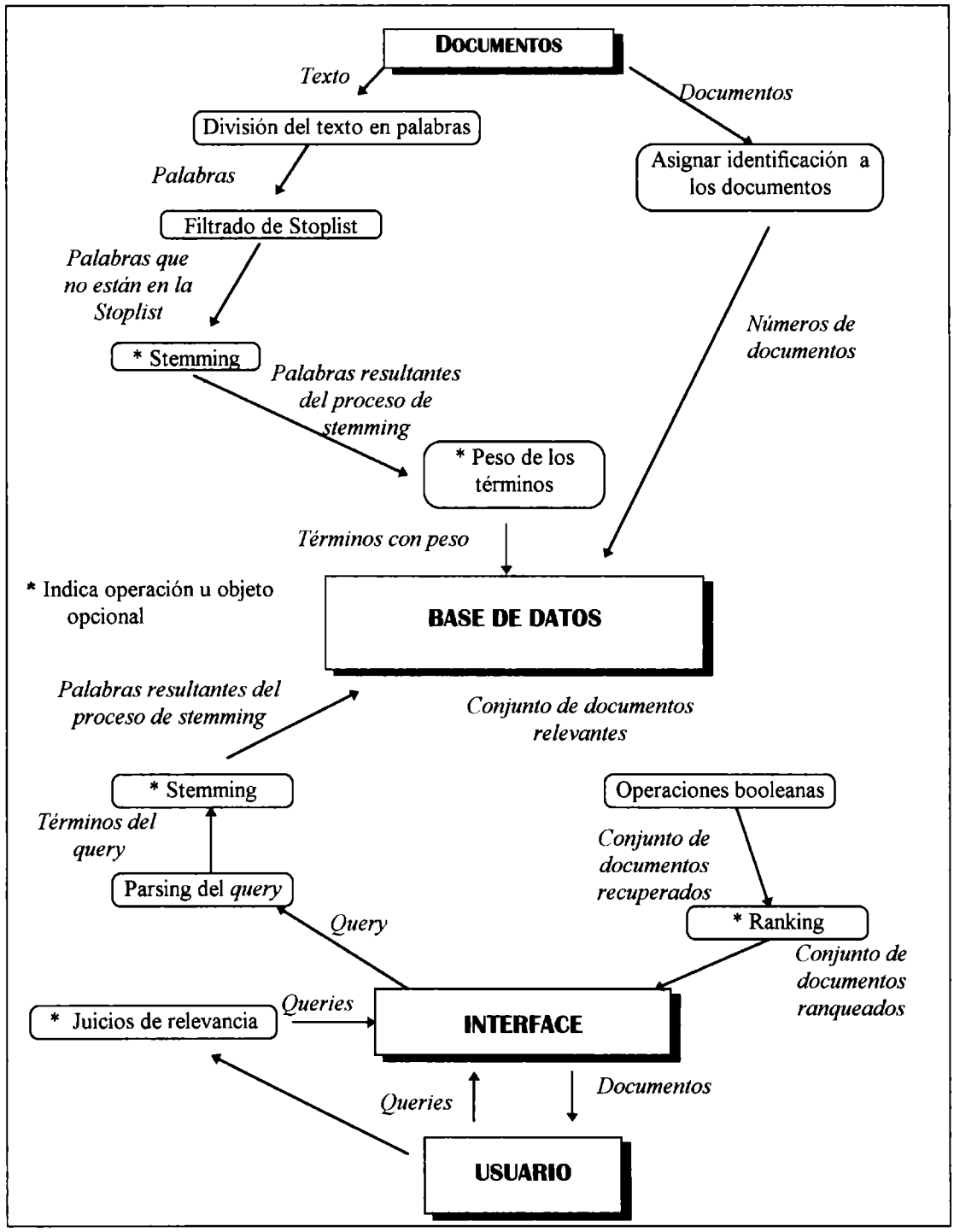


***Figura 3.2** Visión Funcional de un sistema IR booleano*

En el esquema de la *Figura 3.2* se reflejan los dos procesos básicos que se encuentran en un sistema IR, la indexación de los documentos y la recuperación de información a partir de esta colección de documentos .

Indexación de Documentos

Este proceso involucra la construcción de la BDs del sistema IR. Se toman los documentos uno a uno y se procesa su texto. Este texto es dividido en palabras, las cuales son comparadas contra una *stoplist* (lista de palabras que se considera no tienen valor como índice). Aquellas palabras que *no* se encuentren en la *stoplist* son consideradas y se les puede aplicar la técnica de *stemming*. En caso en que se considere un peso para los términos se tiene en cuenta la frecuencia de las palabras en los documentos y se utiliza esta información para luego obtener un ranking de los documentos recuperados. Finalmente se almacenan en la BDs los términos índice y su información asociada tal como los documentos, valores de pesos de los términos, etc.

Recuperación de Información

Cuando se desea recuperar información utilizando este tipo de sistema IR el usuario realiza un pedido a través de un *query*, el cual consiste en un conjunto de palabras claves conectadas por operadores booleanos (AND,NOT,OR) .

Se realiza la operación de *parsing* sobre *query*, la cual divide al *query* en los términos que lo forman y los operadores booleanos. Si en el proceso de indexación se utilizó información sobre la frecuencia de los términos, el conjunto de documentos resultante puede ser ordenado según la probabilidad de relevancia de los mismos. Luego se le presenta al usuario el resultado de la búsqueda.

En algunos sistemas, el usuario puede realizar juicios acerca de la relevancia de los documentos recuperados, y esta información se utiliza para modificar el *query* automáticamente agregando términos de los documentos relevantes y borrando términos de documentos no relevantes. A este proceso se lo conoce como "*feedback* " .

A los términos obtenidos del *query* se le puede aplicar la técnica de *stemming* para obtener la raíz de dicho término y así establecer relaciones morfológicas entre términos .

ANÁLISIS DE CONTENIDOS DE DOCUMENTOS

Análisis de la representación de los documentos y presentación de las Técnicas de Recuperación de Información (Técnicas IR)

1.- Representación de Documentos

Una de las áreas principales de investigación acerca de la Recuperación de Información (IR), es el **Análisis de Contenido de Documentos** .

Esta área tiene que ver con la descripción de los contenidos de los documentos en una forma adecuada para un proceso computacional.

En los primeros tiempos, se almacenaba el documento de texto. Sin embargo no se necesita tener almacenado el texto completo de cada documento en el lenguaje natural en el cual está escrito. En lugar de esto se almacena una **Representación de Documento** , la cual se puede obtener ya sea en forma manual o automática.

Para obtener dicha representación se realiza un proceso de **análisis de texto**. El punto de entrada a este proceso puede ser el documento de texto completo, un resumen del mismo, sólo el título ó quizás sólo una lista de palabras comprendidas en el texto.A partir de esta entrada el proceso de análisis debe obtener una representación de documento en una forma que pueda ser manipulada por una computadora.

Una representación de documento puede estar formada por un conjunto de palabras las cuales permiten representar o caracterizar un documento en la computadora. Esta lista de palabras es lo que suele llamarse "palabras claves" (*Keywords*) y es derivada de cada documento [VanRijsbergen79].

Es común en la literatura IR referirse a los ítems descriptivos extraídos de un documento de texto como *términos índice* o *keywords* .Estos ítems en general, serán la salida producida por un proceso de análisis.

El objetivo es desarrollar un sistema de procesamiento de texto, el cual utilizando métodos computacionales con mínima intervención humana, genere a partir de un texto de entrada (el texto completo, un resumen o el título, por ejemplo) una representación del documento adecuada para el uso en un sistema de recuperación de información automático.

Discriminación y/o Representación

El problema de caracterizar documentos para la recuperación de información puede ser visto según dos ópticas diferentes [VanRijsbergen79].

Una de ellas es caracterizar un documento a través de la representación de sus contenidos, sin tener en cuenta la forma en que pueden ser descriptos los otros documentos. Esto se llama *Representación sin Discriminación* .

Otro caso es considerar la representación de un documento según se pueda discriminar del resto de los documentos en una colección. Esto se llama *Discriminación sin Representación.*

Naturalmente ninguna de estas dos posiciones extremas se asume en la práctica, aunque poder identificar estas dos opciones es útil cuando se analiza el problema de la caracterización de los documentos.

En la práctica la idea sería encontrar una representación que se encuentre entre la *representación* y la *discriminación*. La mayoría de los métodos de indexación automáticos se pueden ver como una mezcla de ambos puntos de vista.

En el caso de utilizar la técnica que remueve del texto palabras con alta frecuencia de ocurrencia (técnica de filtrado de *Stoplist*), se intenta incrementar el nivel de discriminación entre documentos. Sin embargo, hay que tener en cuenta que si se remueven términos que son posibles términos índices, se puede llegar a un estado en el cual los términos que quedan formando parte del texto, no puedan representar adecuadamente los contenidos del documento.

El énfasis sobre la representación de un documento tiene que ver con una *orientación hacia el documento*, es decir la idea es poder representar el tema tratado por el documento.

El énfasis sobre la discriminación tiene que ver una *orientación hacia el query*. La idea es predecir los *queries* que se podrían realizar al sistema IR y con esta información uno podría entonces tratar de caracterizar los documentos de una forma óptima para dichos *queries*.

2.- Análisis Automático de Texto

Como ya se ha adelantado (Pág. 29), la idea es desarrollar un sistema de procesamiento de texto que permita en forma automática generar a partir de un texto de entrada, una representación de documento adecuada para ser utilizada por un Sistema de Recuperación de Información automático.

Este sistema de procesamiento puede aplicar diferentes **técnicas IR** las cuales serán presentadas en detalle más adelante (Pág. 32). Estas se aplican a los términos y representan *niveles de normalización del texto*, por medio de los cuales se puede obtener una representación de documento.

Análisis Léxico

En primer lugar, antes de detallar las operaciones sobre términos debemos introducir la idea del *Análisis Léxico*.

El análisis léxico es el proceso a través del cual se convierte una tira de caracteres de entrada en una tira de palabras o tokens. Los *tokens* son grupos de caracteres con un significado común.

El análisis léxico es el *primer paso* tanto en el indexado automático como en el procesamiento de *queries*.

El **indexado automático** es el proceso que examina ítems de información y genera por medio de un algoritmo, una lista de términos índice. Esta fase produce términos índice *candidatos* que pueden ser procesados aplicándole alguna técnica IR, y eventualmente agregándolos a los índices.

El **procesamiento de *queries*** consiste en analizar los *queries* y compararlos con los índices con el fin de encontrar los ítems relevantes. El análisis léxico de un *query* produce tokens que son analizados y transformados en una representación interna adecuada para compararlos con los índices.

Análisis Léxico para la Indexación Automática

Cuando se va a diseñar un analizador léxico para un sistema de indexación automática, se debe tener en cuenta *cuáles* serán las palabras o tokens considerados en el esquema de indexación. Parecería ser una cuestión fácil de decidir, sin embargo depende del sistema que se va a realizar. Los problemas de decisión básicamente tienen que ver con la aceptación o no de dígitos, signos de puntuación, etc.

No existe una dificultad técnica para resolver este problema, pero en el momento de diseñar el analizador léxico se debe seleccionar una política de análisis a considerar.

Cada política tiene sus ventajas y desventajas asociadas. Por ejemplo si se reconocen números como tokens, se agregan muchos términos con poco valor de discriminación para un índice, sin embargo puede ser una buena política si se apunta a una búsqueda exhaustiva.

La elección del conjunto de tokens que será reconocido por el analizador léxico dependerá de los objetivos del sistema a desarrollar.

Análisis Léxico para el Procesamiento de Queries

Diseñar un analizador léxico para el procesamiento de *queries* es como diseñar un analizador para el proceso de indexación automática. Sin embargo éste depende del diseño que se ha elegido para el analizador léxico del indexado automático, debido a que los términos de búsqueda del *query* deben igualarse (*matching*) a los términos índice.Por lo tanto ambos analizadores deben distinguir los mismos tokens .

Además de esto, el analizador léxico para el procesamiento de *queries* debe también poder distinguir operadores e indicadores de grupos (como paréntesis o corchetes). Debería procesar ciertos caracteres tales como caracteres de control y no permitir caracteres de puntuación. Estos caracteres son tratados como delimitadores en el procesamiento de indexación automática, pero en el procesamiento de *queries* indican ocurrencia de error.

Implementación de un Analizador Léxico

El proceso de análisis léxico para los Sistemas de Recuperación de Información es el mismo que para otro sistema de procesamiento de texto.

Existen tres formas de implementar un analizador léxico :

*Utilizar un generador de analizador léxico, como la herramienta UNIX " *Lex*" para generar un analizador léxico automáticamente.

*Escribir un analizador léxico *ad-hoc* "a mano" .
*Escribir un analizador léxico "a mano" basado en una máquina de estado finita.

El primer método es útil cuando el analizador léxico es complicado. Si el analizador léxico es simple usualmente se utiliza el segundo método.

El tercer método tiene la ventaja que los algoritmos de máquinas de estado finita son extremadamente rápidos. La implementación de una máquina de estado finita se basa en un mecanismo de clasificación de caracteres. Se puede analizar su comportamiento a través de un diagrama de transición. Debe permitir conocer el estado actual en que se encuentra el proceso y contar con una manera de cambiar de un estado a otro según una entrada (*transición*).

TÉCNICAS DE RECUPERACIÓN DE INFORMACIÓN

(TÉCNICAS IR)

Las técnicas IR favorecen al proceso de indexación de documentos por medio del cual se obtiene la representación de los mismos. Además son de utilidad en el procesamiento de *queries* del usuario ya que permiten mejorar en ciertos casos, la precisión y recuperación de información .

En un sistema IR experimental los *queries* pueden ser procesados al mismo tiempo que se procesan los documentos.

En un sistema IR operacional se debe aplicar al *query* el sistema de procesamiento de texto en el momento en que se realiza dicho *query* al sistema IR.

Podemos mencionar tres técnicas IR. Las dos primeras representan operaciones sobre términos mientras que la tercera provee un vocabulario preciso y controlado, el cual permite coordinar la indexación y recuperación de documentos .

❶ STOP LIST

Se ha reconocido desde las primeras investigaciones realizadas sobre la recuperación de información, que las palabras con mayor frecuencia de ocurrencia en general no son útiles como términos índices [BaezaYates-Frakes92].

Si se realizara una búsqueda utilizando alguno de estos términos se recuperarían casi todos los ítems de la BDs sin tener en cuenta su relevancia. Por lo tanto el valor discriminatorio de estos términos es bajo [BaezaYates-Frakes92].

Además estos términos de alta frecuencia ocupan una gran parte del documento, por lo tanto si se eliminan estos términos en el proceso de indexación automático, se ahorra espacio en índices y no daña la efectividad de la recuperación de información.

En consecuencia, las ventajas destacables de ésta técnica comprenden: la eliminación del texto de palabras no significativas las cuales no interferirán en la recuperación de información, y la reducción del tamaño del documento entre un 30 a 50 %.

Se denomina *Stoplist* o *Diccionario Negativo* a una lista de palabras que son filtradas durante el proceso de indexación automática, debido a que no tienen peso como términos índices. A estas palabras se las conoce como *stopwords*. [BaezaYates-Frakes92]

Por lo tanto una forma de mejorar la *performance* de un Sistema de Recuperación de Información es eliminar *stopwords* durante la indexación automática .

Como en el caso del diseño del analizador léxico, no es claro cuáles palabras podrían estar incluidas en una *stoplist*. En general las *stoplist* incluyen las palabras de mas alta frecuencia de ocurrencia. Sin embargo dependiendo del sistema que se esté desarrollando, algunas de estas palabras pueden tener peso como términos índices. Por lo tanto la selección de las palabras que conforman la *stoplist*, depende de las características de la BDs y del sistema.

En el Sistema de Recuperación de Información de Documentos CFPs presentado en el *Capítulo 5*, se encuentra el caso en que la *stoplist* utilizada depende de este sistema (Pág. 97), ya que algunas palabras que tiene una alta frecuencia de ocurrencia en los documentos no pueden estar incluidas en la *stoplist*, debido que son parte necesaria en el proceso de recuperación de información.

La política utilizada para definir la *stoplist* dependerá de la BDs y las características de los usuarios y el proceso de indexación.

En general los Sistemas de Información Comerciales utilizan pocas *stopwords*.

Implementación del filtrado según una StopList

Existen dos formas de filtrar *stopwords* de una tira de caracteres de entrada :

① Examinar la salida del analizador léxico y remover todo término que sea un *stopword*.

② Remover los términos *stopword* como parte del análisis léxico.

① El primer método al filtrar los términos *stopwords* a partir de la salida del analizador léxico, hace que el problema del filtrado de *stoplist* se transforme en el problema estándar de búsqueda en una lista. Se debe buscar en la lista *stoplist* (que incluyen los tokens no significativos) cada token y en caso de encontrarlo en la misma, debe ser removido de la tira de tokens antes de su análisis.

La solución usual a este problema incluye la utilización de árboles binarios de búsqueda, búsqueda binaria sobre un array y hashing. Indudablemente la solución más rápida es la que utiliza *hashing* pero tiene la desventaja que se debe re-examinar cada carácter para generar su valor de *hash* y debe resolver posibles colisiones.

Aunque el método de búsqueda en una lista utilizando *hashing* es muy bueno, probablemente la mejor solución de implementar el filtrado de una *stoplist* sea utilizando el segundo método.

② Este método remueve los términos *stopwords* como parte del análisis léxico. Debido a que de todos modos se debe realizar el análisis léxico, se puede reconocer en este mismo proceso los términos pertenecientes a una *stoplist* casi sin costo extra. Este método podría considerarse el más eficiente.

Además los analizadores léxicos que filtran *stoplist* pueden ser generados automáticamente, lo cual implica que sean más fáciles de construir y tengan menos probabilidades de error que si se escribiera 'a mano 'un algoritmo de filtrado de palabras .

❷ STEMMING

Los algoritmos de *stemming* permiten relacionar términos de búsqueda y términos índice que son similares morfológicamente con el objetivo de mejorar la efectividad de la recuperación y reducir el tamaño de los archivos índice[BaezaYates-Frakes92].

Esta técnica IR permite mejorar la *performance* de un sistema IR ya que el proceso de búsqueda puede encontrar variantes morfológicas de los términos de búsqueda .

Utilizando esta técnica en general se incrementa la recuperación al costo de decrementar la precisión.

Estudios realizados sobre el efecto de la aplicación de esta técnica dan como resultado que en general, el proceso de *stemming* no tiene efecto o tiene un efecto positivo en la *performance* de la recuperación de información .

Si por ejemplo un usuario ingresa como parte del *query* el término *"computation"*, puede ser probable que dicho usuario esté también interesado en términos relacionados como *compute, computer*,etc.

Se utiliza el término ***conflation*** (fusión o combinación) como un término general que representa el proceso de *matching* de variantes morfológicas de un término.

Se puede realizar este proceso de *matching* en forma manual, utilizando alguna clase de expresiones regulares, o automática utilizando programas llamados *stemmers* (algoritmos de *stemming*).

El *stem* de un término representa la raíz del mismo.

Como ya se adelantó al comienzo, la técnica de *stemming* permite reducir el tamaño de los archivos índices debido a que un *stem* simple, típicamente corresponde a varios términos relacionados, entonces si se almacenan como términos índices los *stems* en vez de los términos, se puede alcanzar un factor de compresión de más del 50 % .

La técnica de *stemming* puede aplicarse tanto en el proceso de indexación como en el procesamiento de *queries* .

La ventaja de aplicar la técnica de *stemming* en el proceso de indexación es la eficiencia y la compresión del archivo índice. Es más eficiente debido a que si se cuenta con los *stems* de los términos índice, esta operación no requiere recursos al momento de procesar los *queries* .

La desventaja de aplicar la técnica de *stemming* en el proceso de indexación, es que si sólo se almacenan los *stems* de los términos índices los términos "completos" se pierden. En caso que se necesite dicha información se necesita un almacenamiento adicional para almacenar tanto el término completo como su *stem* asociado.

Algoritmos de *Stemming*

Los métodos de fusión o combinación de variantes morfológicas de términos se ven reflejados en el esquema de la *Figura 3.3*.

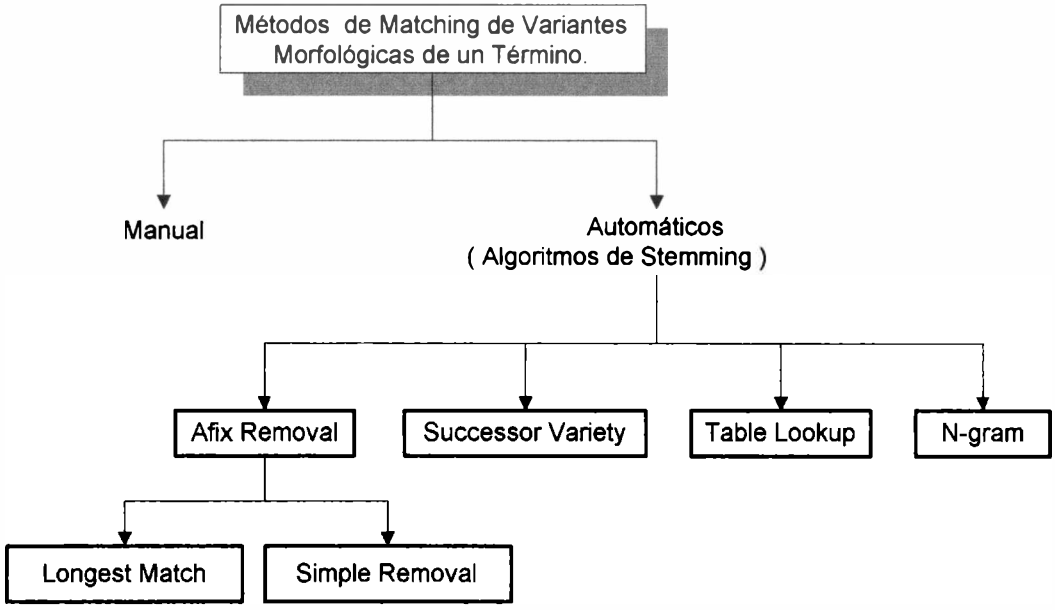


Figura 3.3 *Métodos de Matching de Variantes Morfológicas de Términos*

Existen cuatro métodos automáticos :

2.1 *"Affix Removal"*

Este algoritmo de *stemming* remueve sufijos y/o prefijos de los términos obteniendo así la raíz de la palabra, es decir el *stem* del término. El nombre *stemmer* deriva de este método el cual es el más común.

En ciertos casos también suele transformar el *stem* resultante debido a que puede ocurrir que al remover el sufijo o prefijo del término no sea correcto el *stem* obtenido. Por ejemplo, si queremos remover el sufijo "UAL" sería correcto removerlo de la palabra "FACTUAL" pero no de la palabra "EQUAL". Para evitar remover sufijos erróneamente se utilizan reglas de contexto, con lo cual un sufijo será removido sí y solo sí el contexto es correcto. Que un contexto sea *correcto* puede basarse en :

(1) La longitud del *stem* que se obtiene es mayor que un determinado valor. Por defecto usualmente se considera como tamaño mínimo de *stem*, el tamaño 2.

(2) La terminación del *stem* satisfaga ciertas condiciones. Por ejemplo que no termine con "Q" .

Un ejemplo de un algoritmo "*affix removal*" simple es aquel que remueve los plurales de los términos. Las reglas que intervienen en este tipo de algoritmo serían las siguientes : (Harman 1991)

if (una palabra termina en "ies" pero no termina en "eies" o en "aies")
Then "ies" → "y"
if (una palabra termina en "es" pero no termina en "aes" o en "oes")
Then "es" → "e"
if (una palabra termina en "s" pero no termina en "us" o en "ss")
Then "s" → NULL

En este algoritmo sólo se va a utilizar la primer regla, debido a que la terminación de las palabras considerada incluye a las terminaciones consideradas en las otras reglas.

La mayoría de los algoritmos de *stemming* en uso son algoritmos iterativos del tipo "*longest match* " , una clase de algoritmo de *stemming* que fue desarrollado en primer lugar por Lovin (1968) .

Un algoritmo del tipo "*longest match*" es aquel que remueve (según un conjunto de reglas) de una palabra el string de caracteres de *mayor tamaño posible*. Este proceso es iterativo debido a que se repite hasta que no haya más caracteres que puedan ser removidos. Aún después que todos los caracteres han sido removidos puede ser que los *stems* producidos no sean los correctos.

Por ejemplo si tenemos la palabra "skies", puede reducirse al *stem* "ski" el cual no podrá hacer *matching* con la palabra "sky" .

Existen dos técnicas para manejar este problema:

- ❒ *matching* recodificado
- ❒ *matching* parcial .

❒ La recodificación es una transformación sensitiva al contexto de la forma : AxC → AyC , donde A y C especifican el contexto de la transformación, *x* es el string de entrada e *y* es el string transformado.

En este caso se podría especificar por ejemplo, que si un *stem* termina en "i" siguiendo a una "k", entonces i → y .

❒ En *matching* parcial, sólo se utilizan los *n* primeros caracteres del *stem* para compararlos. Utilizando este método se dice que dos *stems* son equivalentes si ambos comparten los primeros *n* caracteres.

Los algoritmos iterativos del tipo "*longest match*" desarrollados por Lovin, también han sido estudiados por Salton (1968), Dawson (1974) , Porter (1980) y Paice (1990). [BaezaYates-Frakes92]

El algoritmo de Porter es mas compacto que el de Lovin, Salton y Dawson y según los experimentos realizados, la *performance* de recuperación se compara con algoritmos mayores. El algoritmo de Paice también es compacto pero no existe información sobre experimentos realizados con dicho algoritmo, con lo cual se puede considerar al algoritmo de Porter adecuado para representar este tipo de algoritmo de *stemming*. Cabe destacar que este algoritmo está definido para el idioma Inglés.

Algoritmo de Porter

El algoritmo de Porter consiste en un conjunto de reglas del tipo `Condición / Acción` .

Las condiciones pueden pertenecer a tres clases :

.- Condiciones sobre el *stem*.
.- Condiciones sobre el sufijo.
.- Condiciones sobre las reglas.

Condiciones sobre el *stem*

Existen diferentes tipos de condiciones sobre el *stem* :

① La *medida* de un *stem* denotada *m*, se basa en las secuencias alternadas (`vocal - consonante`) que contiene la palabra. Considerando el conjunto de vocales : a, e, i ,o, u y la letra *'y'* cuando se encuentra precedida por una consonante. Las consonantes son todas las letras que no son vocales.

Sea C una secuencia de consonantes y V una secuencia de vocales,la medida *m* , se define como :

$$[C]\ (\ VC\ )^{m}\ [V\ ]$$

El exponente *m* en la ecuación,el cual es la medida, indica el número de secuencia VC. Los corchetes indican una ocurrencia opcional.

Por ejemplo las medidas de los siguientes términos son :

Medida	Ejemplos
m=0	TR , EE , TREE , Y , BY
m=1	TROUBLE , OATS , TREES , IVY
m=2	TROUBLES , PRIVATE , OATEN

② * <X> - El *stem* termina con la letra X .
③ *v* - El *stem* contiene una vocal .

④ *d - El *stem* termina con doble consonante.
⑤ *o - El *stem* termina con una secuencia (consonante-vocal-consonante),donde la consonante final es distinta de ***w***, ***x*** e ***y***.

Condiciones sobre el sufijo

Las condiciones sobre el sufijo tienen el siguiente formato : (`Sufijo_actual == patrón`)

Condiciones sobre las reglas

Las condiciones sobre las reglas tienen el siguiente formato : (Regla que se utiliza)

Acciones

Las acciones son reglas de la forma :

`sufijo_viejo → sufijo_nuevo`

Las reglas están divididas en pasos. Dentro de un paso las reglas se examinan siguiendo una secuencia y sólo se puede aplicar una regla por paso.

Siempre se remueve el sufijo de mayor tamaño debido al orden que tienen las reglas dentro de un paso.

Algoritmo de Porter

El algoritmo de Porter es el siguiente :

```
{
    Paso1a (palabra)
    Paso1b (stem )
    if ( se ha utilizado la segunda o tercer regla del paso1b)
        Paso1b1 (stem)
    Paso1c (stem)
    Paso2  (stem)
    Paso3  (stem)
    Paso4  (stem)
    Paso5a (stem)
    paso5b (stem)
}
```

Las reglas para cada paso de este algoritmo de *stemming* son las siguientes :

Reglas de **Paso1a**

Condiciones	Sufijo	Reemplazo	Ejemplos
NULL	sses	ss	caresses → caress
NULL	ies	i	ponies→poni ties→tie
NULL	ss	ss	carress→carress
NULL	s	NULL	cats → cat

Reglas de **Paso1b**

Condiciones	Sufijo	Reemplazo	Ejemplos
(m > 0)	eed	ee	feed → feed agreed → agree
(*v*)	ed	NULL	plastered → plaster bled → bled
(*v*)	ing	NULL	motoring → motor sing → sing

Reglas de **Paso1b1**

Condiciones	Sufijo	Reemplazo	Ejemplos
NULL	at	ate	conflat(ed) → conflate
NULL	bl	ble	troubl(ing) → trouble
NULL	iz	ize	six(ed) → size
(*d and not (*< L > or *< S > or *< Z >))	NULL	single letter	hopp(ing) → hop tann(ed) → tan fall(ing) → fall hiss(ing) → hiss fizz(ed) →fizz
(m = 1 and *o)	NULL	e	fail(ing) → fail fil(ing) → file

Reglas de **Paso2**

Condiciones	Sufijo	Reemplazo	Ejemplos
m > 0	ational	ate	relational → relate
m > 0	tional	tion	conditional → condition rational → rational
m > 0	enci	ence	valenci → valence
m > 0	anci	ance	hesitanci → hesitance
m > 0	izer	ize	digitizer → digitize
m > 0	abli	able	comformabli → comformable
m > 0	alli	al	radicalli → radical
m > 0	entli	ent	diferentli → diferent
m > 0	eli	e	vileli → vile
m > 0	ousli	ous	analogousli → analogous
m > 0	ization	ize	vietnamization → vietnamize
m > 0	ation	ate	predication → predicate
m > 0	ator	ate	operator → operate
m > 0	alism	al	feudalism → feudal
m > 0	iveness	ive	decisiveness → decisive
m > 0	fulness	ful	hopefulness → hopeful
m > 0	ousness	ous	callousness → callous
m > 0	aliti	al	formaliti → formal
m > 0	iviti	ive	sensitiviti → sensitive
m > 0	biliti	ble	sensibiliti → sensible

Reglas de **Paso1c**

Condiciones	Sufijo	Reemplazo	Ejemplos
(*v*)	y	i	happy → happi sky → sky

Reglas de **Paso3**

Condiciones	Sufijo	Reemplazo	Ejemplos
m > 0	icate	ic	triplicate → triplic
m > 0	ative	NULL	formative → form
m > 0	alize	al	formalize → formal
m > 0	iciti	ic	electriciti → electric
m > 0	ical	ic	electrical → electric
m > 0	ful	NULL	hopeful → hope
m > 0	ness	NULL	goodness → good

Reglas de **Paso4**

Condiciones	Sufijo	Reemplazo	Ejemplos
(m > 1)	al	NULL	revival → reviv
(m > 1)	ance	NULL	allowance → allow
(m > 1)	ence	NULL	inference → infer
(m > 1)	er	NULL	airliner → airlin
(m > 1)	ic	NULL	gyroscopy → gyroscop
(m > 1)	able	NULL	adjustable → adjust
(m > 1)	ible	NULL	defensible → defens
(m > 1)	ant	NULL	irritant → irrit
(m > 1)	ement	NULL	replacement → replace
(m > 1)	ment	NULL	adjustment → adjust
(m > 1)	ent	NULL	dependent → depend
(m > 1) and (*< S > or *< T >))	ion	NULL	adoption → adopt
(m > 1)	ou	NULL	homologou → homolog
(m > 1)	ism	NULL	communism → commun
(m > 1)	ate	NULL	activate → activ
(m > 1)	iti	NULL	angulariti → angular
(m > 1)	ous	NULL	homologous → homolog
(m > 1)	ive	NULL	effective → effect
(m > 1)	ize	NULL	bowdlerize → bowdler

Reglas de **Paso5a**

Condiciones	Sufijo	Reemplazo	Ejemplos
(m > 1)	e	NULL	probate → probat rate → rate
(m = 1 and not *o)	e	NULL	cease → ceas

Reglas de **Paso5b**

Condiciones	Sufijo	Reemplazo	Ejemplos
(m > 1 and *d and *< L >)	NULL	single letter	controll → control roll → roll

2.2 "Succesor Variety"

Estos algoritmos de *stemming* (Hafer and Weiss 1974) utilizan la frecuencia de las secuencias de letras en el cuerpo de un texto como base para la técnica de *stemming*. Se basan en trabajos sobre la lingüística estructural ,donde se intenta determinar palabras y límites morfológicos basados en la distribución de los fonemas en un cuerpo de expresiones.

El método de *stemming* basado en este trabajo utiliza letras en lugar de fonemas y un cuerpo de un texto en lugar de expresiones .

Hafer y Weiss definen formalmente la técnica de la siguiente manera :

Sea α una palabra con longitud n ; α_i es la longitud i prefijo de α. Sea D el cuerpo de las palabras, $D_{\alpha i}$ se define como el subconjunto de D conteniendo aquellos términos cuyas primeras i letras hacen *matching* con α_i exactamente. La "variedad sucesoria" (*Succesor Variety*) de α_i , denotada $S_{\alpha i}$, se define como el número de letras distintas que ocupan posición i+1 de palabras en $D_{\alpha i}$. Una palabra de longitud n tiene n variedades sucesorias $S_{\alpha 1}$, $S_{\alpha 2}$,.....,$S_{\alpha n}$.

Presentado en términos menos formales, la variedad sucesoria de un string es el número de caracteres diferentes que continúan a él en las palabras que se encuentran en el cuerpo de un texto.

Tomemos como ejemplo un texto que contiene las siguientes palabras : {able , axle , accident , ape , about }. Para determinar la variedad sucesoria de la palabra "apple" , se realiza el siguiente proceso:

La primer letra de "apple" es la "a" . Esta letra está seguida en el cuerpo del texto por cuatro caracteres : "b", "x", "c" y "p" . Entonces la variedad sucesoria de "a" es *cuatro* . La próxima variedad sucesoria sería *uno* debido a que solo la "e" sigue a los caracteres "ap" en el cuerpo del texto. Así se realiza el proceso sucesivamente.

Una vez que se ha obtenido las variedades sucesorias para una palabra dada se utiliza esta información para segmentar la palabra.

Hafer y Weiss discuten ciertas formas de realizar esto tales como :

- Utilizar el método "*cutoff*". Se selecciona un valor como grado de admisión (*cutoff*) para variedades sucesorias y se identifica un límite cuando se alcanza este valor. El problema con este método es cómo seleccionar este valor límite. Si es muy pequeño se pueden realizar cortes incorrectos en las palabras, o si es demasiado grande se pueden perder divisiones de palabras correctas.

- Utilizar el método "*peak and plateau*". Se obtiene un segmento después de un carácter cuya variedad sucesoria excede la variedad sucesoria del carácter que lo precede y del carácter que lo sigue. Este método evita la necesidad de seleccionar un valor límite.

● Utilizar el método de "*palabra completa* ". Se obtiene un segmento si este conforma una palabra completa que se encuentra en el cuerpo del texto.

El algoritmo de *stemming* utilizando "*Variedad Sucesoria* " puede reflejarse con el siguiente ejemplo :

Se quiere determinar el *stem* de la palabra READABLE .

Palabra : READABLE
Texto :ABLE, APE, BEATABLE, FIXABLE,READ, READABLE, READING, READS, RED, ROPE, RIPE

Prefijo	Variedad Sucesoria	Letras
R	3	E,I,O
RE	2	A,D
REA	1	D
READ	3	A,I,S
READA	1	B
READAB	1	L
READABL	1	E
READABLE	1	BLANK

Utilizando el método de segmentación de "*palabra completa*", La palabra READABLE será segmentada en "READ" y "ABLE" ya que la palabra READ aparece en el cuerpo del texto. El método "*peak and plateau*" daría el mismo resultado.

Una vez que la palabra ha sido segmentada se debe seleccionar el segmento que será el *stem* .

Hafer y Weiss utilizan la siguiente regla :

```
If ( el primer segmento ocurre en < = 12 palabras en el texto)
        el primer segmento es el stem
   else
        el segundo segmento es el stem
```

El chequeo sobre el número de ocurrencias se basa en la observación que si un segmento ocurre en más de 12 palabras en el texto, es probable que sea un prefijo.

En resumen, el proceso de *stemming* utilizando el método de "Variedad Sucesoria" comprende tres pasos :

➲Determinar la variedad sucesoria para una palabra.

➲Utilizar esta información para dividir la palabra mediante alguno de los métodos descriptos anteriormente.

➲Seleccionar uno de los segmentos como *stem*.

El objetivo de Hafer y Weiss fue desarrollar un algoritmo de *stemming* que requiera poca o ninguna intervención humana. La idea es que aunque el método "*Afix Removal* " es adecuado para el proceso de *stemming*, requiere participación humana debido a que se deben preparar las listas de afijos y las reglas para poder removerlos de una palabra.

2.3 *"N-Gram"*

Este algoritmo de *stemming* combina términos según el número de digramas o n-gramas que ellos comparten.

Adamson y Boreham (1974) definieron un método de combinar términos llamado *método de digrama compartido.*

Un digrama es un par de letras consecutivas. Como también pueden utilizarse trigramas o n-gramas a este método se lo conoce como *método N-Gram.* Aunque se lo trata como un método de *stemming* esto puede resultar confuso debido a que no se obtiene como resultado un *stem.*

En este método, se calculan medidas de asociación entre pares de términos según los digramas únicos que comparten.

Tomemos como ejemplo las palabras "*statistics*" y "*statistical*". Estas palabras pueden ser divididas en los siguientes digramas :

statistics → st ta at ti is st ti ic cs
Digramas únicos = at cs ic is st ta ti

statistical → st ta at ti is st ti ic ca al
Digramas únicos = al at ca ic is st ta ti

Por lo tanto, la palabra "*statistic*" tiene 9 digramas de los cuales 7 son únicos, mientras que la palabra "*statistical*" tienen 10 digramas de los cuales 8 son únicos.

Ambas palabras comparten 6 digramas únicos: {at, ic, is, st, ta y ti }.

Una vez que se han identificado y contado los digramas únicos que comparten ambas palabras, entonces se calcula la medida de similaridad entre ambas palabras.

La medida de similaridad que se utiliza es el *Coeficiente de Dice* , el cual se define de la siguiente manera :

$$S = \frac{2C}{A + B}$$

Donde A= Número de digramas únicos en la primer palabra
B= Número de digramas únicos en la segunda palabra
C= Número de digramas únicos compartidos por ambas palabras ($A \cap B$)

Para el ejemplo dado, el coeficiente de Dice sería: (2 * 6) / (7 + 8) = 80.

Se determina esta medida de similaridad para todos los pares de términos en la BDs, construyendo una matriz de similaridad.

Como el coeficiente de Dice es simétrico, es decir que $S_{ij} = S_{ji}$ entonces se puede utilizar una matriz triangular de la forma :

	Palabra $_1$	**Palabra $_2$**	**Palabra $_3$**	.....	**Palabra $_{n-1}$**
Palabra $_1$					
Palabra $_2$	S_{21}				
Palabra $_3$	S_{31}	S_{32}			
.....	.....	.....	.....		
Palabra $_{n-1}$	S_{n1}	S_{n2}	S_{n3}		$S_{n(n-1)}$

Una vez que se cuenta con la matriz de similaridad, los términos son agrupados según su grado de similitud.

Pruebas realizadas por Adamson y Boreham dieron como resultado que la mayoría de las medidas de similaridad entre pares de términos resultaban 0. Por lo tanto la matriz de similaridad no es muy compacta, con lo cual puede ser apropiado utilizar técnicas para manejar matrices no compactas.

Utilizando como grado de admisión (*cutoff*) de similaridad al valor 0.6,se obtuvo como resultado que 10 de los 11 grupos formados eran correctos. Aún más, casi en ningún caso se obtuvieron asociaciones falsas.

2.4 *"Table Lookup"*

En este caso se almacenan en una tabla *todos* los términos índice y sus correspondientes *stems*, con lo cual el algoritmo de *stemming* para obtener el *stem* de un término, se lleva a cabo buscando en esta tabla.

Por ejemplo :

Término	Stem
engineering	engineer
engineered	engineer
engineer	engineer

Existen ciertos problemas con este método. En primer lugar no existe este tipo de información para el idioma Inglés, es decir que no podríamos asegurarnos de contar con todos los posibles términos índice y sus correspondientes *stems*. Aún si esto fuera posible, no se podrían representar muchos términos que se encuentran en una BDs debido a que son dependientes del dominio, es decir no son términos del Inglés estándar, en cuyo caso necesitaríamos utilizar otra técnica de *stemming*. Otro problema tiene que ver con el tamaño de la tabla debido a que ésta sería muy grande.

Criterios para evaluar los Algoritmos de Stemming

Los criterios que se tienen en cuenta a la hora de evaluar un algoritmo de *stemming* incluyen :

- Correctitud.
- Efectividad en la Recuperación.
- *Performance* en la compresión.

Un algoritmo de *stemming* puede resultar *incorrecto* en dos maneras : *overstemming* y *understemming*.

Overstemming : Es el caso en que a un término se le remueve más contenido de lo correcto. Esto puede provocar que términos no relacionados sean combinados o fusionados como si lo estuvieran. El efecto que tiene este problema sobre la *performance* en un sistema IR, es que se pueden recuperar documentos no relevantes.

Understemming : Es el caso en que a un término se le remueve menos contenido de lo correcto. Esto puede provocar que términos relacionados no puedan ser combinados. El efecto que tiene este problema sobre la *performance* en un sistema IR, es que no se pueden recuperar documentos que son relevantes.

Los algoritmos de *stemming* también pueden ser evaluados según su *eficiencia en la recuperación*, usualmente medida en términos de *recall* y *precisión* (medidas de evaluación de los Sistemas IR presentadas en la *sección de Evaluación de los Sistemas IR* (Pág. 56)), la velocidad, el tamaño, etc.

Finalmente puede evaluarse según la *performance* en la compresión.

Los algoritmos de *stemming* usualmente no son evaluados según la correctitud lingüística, aunque los *stems* que producen usualmente son similares a las raíces morfológicas.

❸ Thesaurus

Definición

Un *thesaurus* comprende una lista de términos, donde cada término puede ser una palabra simple o una frase, junto con las relaciones entre los términos.

El objetivo de un *thesaurus* es proveer *un vocabulario común, preciso y controlado*, el cual asiste en la coordinación de la indexación y recuperación de documentos permitiendo seleccionar los términos más apropiados y reformular la estrategia de búsqueda si así se lo desea. Un *thesaurus* es diseñado para áreas temáticas específicas, en consecuencia es *dependiente del dominio*.[BaezaYates-Frakes92]

El proceso de indexación se refiere al proceso por medio del cual se deriva a partir de un documento una representación resumida del mismo, mientras que el proceso de recuperación es un proceso de búsqueda donde se intenta recuperar los ítems relevantes de un documento .

En el proceso de indexación, mediante un *thesaurus* se pueden seleccionar las entradas del *thesaurus* más apropiadas para representar un documento. En la búsqueda, el usuario puede emplear el *thesaurus* para diseñar la estrategia de búsqueda más apropiada. Si en dicha búsqueda no se recuperan documentos suficientes, se puede utilizar el *thesaurus* para expandir el *query* siguiendo los diferentes links que relacionan términos. De la misma manera, si la búsqueda recupera demasiados ítems se puede utilizar el *thesaurus* para sugerir un vocabulario de búsqueda más específico.

Construcción de un *Thesaurus*

Un *thesaurus* puede construirse en forma manual o automática. No existe un gran apoyo respecto de la construcción automática debido a que se considera difícil la posibilidad de realizar este proceso completamente automático.

Un *thesaurus* construido manualmente es una estructura altamente compleja que presenta gran variedad de relaciones entre términos, incluyendo relaciones jerárquicas, no jerárquicas, equivalentes y asociativas. No es sencillo poder determinar tales relaciones en forma automática. Los métodos automáticos de construcción de *thesaurus* están fuertemente relacionados con las estadísticas.

Una construcción de *thesaurus* manual es altamente conceptual y es una tarea de conocimiento intensiva, en consecuencia resulta una labor extremadamente intensiva.

1.- *Construcción manual de un thesaurus*

En un *thesaurus* manual podemos encontrar dos tipos de relaciones las cuales son semánticas en naturaleza y reflejan las interacciones conceptuales subyacentes entre los términos :

(1) Relación de palabras que se refieren a un mismo tópico
(2) Relación de palabras que se refieren a temas relacionados.

En el primer caso conecta términos que son intersubstituibles, es decir que ubica los términos en clases de equivalencia. Luego se elige un término como representante de cada clase y la lista de estos términos se utiliza como un vocabulario controlado. A su vez el usuario podría seleccionar los términos para expresar el *query*.

En el segundo caso se definen relaciones semánticas entre términos relacionándolos en forma jerárquica.

Las relaciones entre los términos es el aspecto más importante de un *thesaurus* debido a que estas relaciones en el vocabulario es lo más valioso para la recuperación de información. Debemos notar que el valor relativo de estas relaciones para la recuperación no es claro, para identificar estas relaciones se requiere tener conocimiento sobre el dominio para el cual el *thesaurus* está siendo diseñado. La mayoría de las relaciones semánticas son difíciles de identificar utilizando métodos automáticos, especialmente en los casos en que sólo se explotan las relaciones estadísticas entre los términos.

El proceso de construcción manual de un *thesaurus* es bastante complejo aunque se pueden destacar ciertos aspectos .

En primer lugar se deben definir límites al área temática del *thesaurus* (en el caso de la construcción automática los límites están definidos por el área cubierta por la BDs de documentos). Esta definición de límites, incluye identificar el área temática central y las áreas periféricas, ya que en general no todas las áreas de un tema tienen la misma importancia.

A continuación, se particiona el dominio en divisiones o subáreas. Se debe definir el conjunto de términos que corresponde a cada subárea. Para definir esto se puede utilizar una variedad de elementos tal como índices, enciclopedias, libros de texto, artículos, catálogos, así como también cualquier sistema de vocabulario o *thesaurus* existente. También en este paso podrían participar expertos en el área temática tratada o potenciales usuarios del *thesaurus*.

Una vez que se ha identificado el vocabulario inicial, se analiza cada término para obtener el vocabulario relacionado, incluyendo sinónimos, términos más amplios o más restringidos,etc. Estos términos y sus relaciones asociadas se organizan en una estructura que puede ser una estructura jerárquica.

Al organizar el vocabulario pueden surgir problemas como por ejemplo, tener que agregar términos, necesitar nuevos niveles jerárquicos, nuevos sinónimos, reconocer nuevas relaciones entre términos, etc.

Luego de alcanzar una definición inicial del *thesaurus*, se lo debe revisar para chequear la consistencia en lo que respecta a las formas de los términos y las frases.

Típicamente un *thesaurus* provee una organización de los términos en forma jerárquica y alfabética.

El proceso de construcción manual de un *thesaurus* es muy complejo. Una vez que el *thesaurus* ha sido diseñado e implementado para ser utilizado en Sistemas de Recuperación de Información, el problema siguiente es el mantenimiento del mismo. Un *thesaurus* debería reflejar todo cambio en la terminología del área del dominio. Como los documentos viejos deben continuar permaneciendo en el sistema, el *thesaurus* modificado debe también contemplar la información antigua. Las modificaciones típicamente involucran tiempo y personas encargadas de rever y sugerir nuevo vocabulario, así como también nuevas relaciones.

2.- *Construcción automática de un thesaurus*

Mientras que un *thesaurus* construido manualmente se basa en las relaciones semánticas (reconocen sinónimos, relaciones más generales o relaciones más específicas), un *thesaurus* construido automáticamente se basa en las relaciones sintácticas y estadísticas de los términos.

Las relaciones estadísticas entre los términos utilizan la co-ocurrencias de palabras en un documento. A menudo dichas palabras son ítems descriptivos del mismo.

La construcción automática de clases de palabras claves se basa en la siguiente relación: si dos palabras claves *a* y *b* son substituibles una por otra, en el sentido que se puede aceptar un documento que contenga una de las palabras como respuesta a un *query* que contenga la otra, esto implica que ambas tienen el mismo significado o se refieren a un tema o tópico común. Una forma de descubrir dicha relación es estudiando los documentos donde éstas palabras ocurren. Si tienden a ocurrir en los *mismos* documentos se tiene chance que ambas palabras se refieran al mismo tema, y por lo tanto puedan ser sustituidas una por otra.

Se puede destacar tres métodos para construir un *thesaurus* automáticamente :

❖*A partir de una colección de documentos*

En este método se utiliza una colección de documentos como punto de partida en la construcción del *thesaurus*.La idea es aplicar procedimientos estadísticos con el fin de identificar los términos más importantes así como también las relaciones entre ellos. De esta manera se intenta identificar el conocimiento semántico más importante de un *thesaurus*. Este conocimiento será utilizado tanto por el proceso de indexación como por el proceso de búsqueda.

❖*Mezclando thesauri existentes*

Este método es apropiado cuando se cuenta con dos o más *thesaurus* sobre un tema dado, quizás representando diferentes perspectivas del tema y se desea unirlos en uno solo.Este proceso de unión no debe violar la integridad de ningún *thesaurus* .

❖*Thesaurus generado por un usuario*

La idea es que los usuarios de los sistemas IR utilicen las relaciones de los términos en sus estrategias de búsqueda antes de estar éstas incluidas en la definición del *thesaurus*. El objetivo es capturar este conocimiento a partir de la búsqueda del usuario. Se requiere una considerable interacción con los usuarios y se utiliza principalmente metodologías de sistemas expertos.

ESTRATEGIAS DE BÚSQUEDA

Presentación de las estrategias de búsqueda para la Recuperación de Información

El proceso de búsqueda se refiere al proceso por medio del cual se localiza la información requerida.

En el caso de recuperación de documentos, la información obtenida es un subconjunto de documentos potencialmente relevantes para el *query*.

El tipo de búsqueda usual es aquel donde se determina si un ítem se encuentra o no en un determinado contexto. Este tipo de búsqueda es importante cuando se utilizan diccionarios para consultar durante el procesamiento de un texto.

Todas las estrategias de búsqueda se basan en una comparación entre el *query* y los documentos almacenados.

En ciertos casos, se pueden distinguir diferentes estrategias de búsqueda según el lenguaje de *query* utilizado. La naturaleza de este lenguaje a menudo dicta la naturaleza de la estrategia de búsqueda. Por ejemplo, si el lenguaje de *query* permite que las sentencias de búsqueda puedan expresarse en términos de combinaciones lógicas de palabras clave, esto normalmente dicta que la estrategia de búsqueda es booleana. Este tipo de búsqueda obtiene un resultado por medio de comparaciones lógicas (más que numéricas) entre el *query* y los documentos.

Funciones de Matching

Muchas de las estrategias de búsqueda más sofisticadas se basan en una función de *matching*. Una función de *matching* mide la asociación entre un *query* y un documento.

Existen diferentes ejemplos de funciones de *matching*. La más simple está asociada con la estrategia de búsqueda de *matching* simple : $| D \cap Q |$ donde para cada documento D y *query* Q se calcula el tamaño de la intersección entre D y Q, cada uno de ellos representado como un conjunto de palabras clave. A este coeficiente se lo conoce como *coeficiente de matching simple*.

Otro ejemplo de una función de *matching* puede definirse de la siguiente manera : sea M la función de *matching*, D el conjunto de palabras claves que representa un documento y Q el conjunto que representa un *query*, entonces se define M como :

$$M = \frac{2 \, | D \cap Q |}{| D | + | Q |}$$

Esta función es equivalente al coeficiente de Dice presentado en la sección anterior (Pág. 43).

Otra función de *matching* es la que se conoce como " *Cosine Correlation* " . Se asume que un documento y un *query* están representados como vectores numéricos en un espacio *t*-dimensional , es decir se tiene $Q= (q_1,q_2,.....,q_t)$ y $D=(d_1 ,d_2 ,.....,d_t)$, donde q_i y d_i son valores numéricos asociados con la palabra clave $_i$. La función " *Cosine Correlation* " se define como :

$$r = \frac{\sum_{i=1}^{t} q_i \, d_i}{\left(\sum_{i=1}^{t} (q_i)^2 \quad \sum_{i=1}^{t} (d_i)^2 \right)^{1/2}}$$

■ Entre las estrategias de búsqueda, podemos mencionar las siguientes [VanRijsbergen79] :

❶ Búsqueda Booleana

La estrategia de búsqueda booleana recupera aquellos documentos que son "verdaderos" para el *query*. Esta idea tiene sentido solo si los *queries* están expresados por medio de términos índice (o palabras clave) y combinados a través de los conectivos lógicos usuales AND, OR o NOT.

Por ejemplo si se tiene el *query* $Q = (K_1 \text{ AND } K_2) \text{ OR } (K_3 \text{ AND NOT}(K_4))$ entonces, la búsqueda booleana recuperará todos los documentos indexados por K_1 y K_2, así como también todos los documentos indexados por K_3 que no están indexados por K_4 .

Las expresiones booleanas se forman a partir de los *queries* de los usuarios. En algunos sistemas IR el usuario ingresa el *query* directamente, en otros casos, lo ingresa en lenguaje natural y el sistema lo transforma en una expresión booleana. En ambos casos se evalúa esta expresión para determinar los resultados del *query*. Estas expresiones representan un pedido para determinar el conjunto de documentos que contienen un conjunto de palabras claves. Por ejemplo :

Find all documents containing ' information '

Este *query* es evaluado y retorna un conjunto (posiblemente vacío) de documentos que contienen la palabra '*information*'.Sería representado por la expresión booleana :

information

La cual significa " un conjunto de documentos que contienen el patrón '*information*' " .

La mayoría de los *queries* involucran más de un término. Por ejemplo, un usuario podría solicitar la siguiente información :

Find all documents containig 'information' and 'retrieval'

Este *query* sería representado por la expresión booleana :

information and retrieval

Algunos sistemas que utilizan búsqueda booleana, permiten al usuario restringir o generalizar la búsqueda proveyendo acceso a un diccionario estructurado, el cual para toda clave almacena palabras claves relacionadas, ya sea en forma mas precisa o mas general. Si el sistema es interactivo fácilmente se puede reformular la búsqueda utilizando estos términos relacionados.

❷ Búsqueda Serial

Las búsqueda seriales son lentas pero aún se suelen utilizar como parte de grandes sistemas. En este tipo de búsqueda podemos tener una demostración del uso de las funciones de *matching* .

Supongamos que se tiene N documentos $D_{i\ (1 <= i <= N)}$ en el sistema. El conjunto de documentos a recuperar se determina calculando N valores $M(Q,D_i)$, donde M es la función de *matching* definida anteriormente. Este conjunto se puede determinar de la siguiente forma :

Se define un valor mínimo y se recuperan los documentos cuyo valor es mayor que este valor mínimo, y se descarta el resto. Sea T el valor mínimo, entonces el conjunto de documentos a recuperar será : $\{ Di \mid M(Q,D_i) > T \}$.

La principal dificultad con esta estrategia de búsqueda es especificar el valor mínimo. Siempre se selecciona en forma arbitraria debido a que no existe una manera de saber anticipadamente, cuál será el valor que permite obtener la mejor recuperación.

❸ Recuperación Basada en Clusters

La recuperación basada en *clusters* se basa en la idea que aquellos documentos que están fuertemente asociados, tienden a ser relevantes para los mismos requerimientos. La técnica de *clustering* obtiene el conjunto de documentos asociados y los agrupa en un *cluster*.

Si se cuenta con una clasificación jerárquica de documentos la estrategia de búsqueda sería de la siguiente manera: la búsqueda comienza en la raíz del árbol y evalúa una función de *matching* a los nodos descendentes inmediatos. Este patrón se repite en forma descendente en el árbol.

Una regla de decisión dirige la búsqueda. La misma compara los valores de la función de *matching* en cada estado y decide con qué nodo continuar. Además es necesario contar con una regla de terminación, la cual concluye la búsqueda obteniendo la información recuperada. Esta búsqueda es del tipo *top-down.*

También se puede utilizar una estrategia de búsqueda *bottom-up*. En este caso la búsqueda comienza en sus nodos terminales y realiza el proceso "hacia arriba" hasta alcanzar la raíz del árbol. No se necesita una regla de decisión sólo se necesita una regla de terminación, la cual puede ser simplemente un valor límite. Una búsqueda típica buscaría el *cluster* de mayor tamaño que contiene el documento representado por el nodo de comienzo y cuyo tamaño no exceda el valor límite. Una vez que se encuentra este *cluster* se recupera el conjunto de documentos pertenecientes a este *cluster*. Para comenzar la búsqueda referente a un requerimiento, se necesita conocer anticipadamente un nodo terminal que sea adecuado para dicho requerimiento. Puede ser común que el usuario ya conozca algún documento relevante para el pedido y que esté buscando otros documentos similares a él. Entonces se utiliza este documento conocido, como punto de inicio de la búsqueda *bottom-up*.

EVALUACIÓN DE LOS SISTEMAS IR

Presentación de los métodos de evaluación de los Sistemas IR

Para poder analizar el problema de evaluación de los sistemas IR podemos formularnos tres preguntas :

①¿ Por qué se realiza una evaluación ?
② ¿Qué se debe evaluar?
③ ¿Cómo realizar la evaluación ?

① *¿ Por qué se realiza una evaluación ?*

El objetivo de realizar una evaluación a un sistema IR, es obtener una medida de los beneficios (o de las desventajas) de utilizar un sistema de recuperación de información. Los beneficios no sólo se refieren al hecho que el usuario recupere los documentos relevantes, sino también los beneficios que un usuario puede tener al reemplazar su forma tradicional de obtener información por un sistema de recuperación interactivo automático. Además por medio de una evaluación se puede conocer el costo que involucra utilizar este tipo de sistema. Se debe contestar la pregunta ¿ vale la pena ? . Es difícil obtener dicho costo. El costo computacional puede estimarse, pero es difícil estimar el costo que tiene que ver con el esfuerzo personal. La decisión respecto a si vale la pena o no utilizar este tipo de sistema dependerá de cada usuario.

Al evaluar un sistema IR se desea que el usuario pueda tomar decisiones con respecto a (1) si desea utilizar este tipo de sistema (pregunta de aspecto social) ó (2) si vale la pena (pregunta de aspecto económico).

Además, los métodos de evaluación se analizan en forma comparativa para decidir si ciertos cambios en un sistema pueden mejorar la *performance* del mismo.

② *¿Qué se debe evaluar?*

En este caso la idea es obtener una medida que permita representar la capacidad que tiene el sistema de satisfacer al usuario. En 1966 Cleverdon [VanRijsbergen79], presentó una respuesta a esto mediante seis medidas :

1) El *área de cobertura* de la colección involucrando el tema relevante.

2) El *tiempo de retraso*, es decir el intervalo promedio entre el tiempo de búsqueda del pedido de información y el tiempo de presentación de la respuesta.

3) La *forma* de presentación de la salida del sistema.

4) El *esfuerzo* del usuario para obtener las respuestas a sus requerimientos.

5) La *recuperación* (*recall*) del sistema. Es decir la proporción de material *relevante* que es recuperado como respuesta a un requerimiento.

6) La *precisión* del sistema. Es decir la proporción de material recuperado que es relevante.

Las medidas 1) al 4) son fácilmente evaluables.

Las medidas de recuperación y precisión (*recall* y *precision*) son las que representan la eficiencia de un sistema de recuperación de información . En otras palabras, se mide la capacidad del sistema de recuperar documentos relevantes y al mismo tiempo no recuperar documento no - relevantes. Se asume que cuanto más efectivo sea el sistema, en mayor medida se va a satisfacer al usuario. También se asume que las medidas *recall* y *precision* son suficientes para evaluar la eficiencia de un sistema IR.

Una alternativa a estas medidas ha sido la medida "*recall and fall-out*", la cual representa la proporción de documentos no relevantes recuperados. Sin embargo todas las alternativas de medida requieren determinar de alguna manera, la relevancia de los documentos.

Las ventajas de utilizar las medidas *recall* y *precision* es que ambas son :

- El par de medidas más comúnmente usadas.
- Representan cantidades que pueden ser fácilmente comprendidas.

③ ***¿Cómo realizar la evaluación ?***

Se debe tener en cuenta que en general, las técnicas para medir la eficiencia de la recuperación están influenciadas por la estrategia de búsqueda particular que utiliza y por la forma de presentación de la salida.

Por ejemplo si como salida del sistema se presenta un ranking de documentos, entonces se cuenta con el parámetro de la posición del ranking para realizar un control.

Noción de Relevancia

Como se ha adelantado, toda medida de evaluación necesita determinar de alguna manera la relevancia de los documentos.

La *relevancia* es una noción subjetiva. Distintos usuarios pueden diferir acerca de la relevancia o no relevancia de documentos particulares con respecto a ciertas preguntas. Sin embargo esta diferencia no es lo suficientemente grande como para invalidar los experimentos que se realizan con colecciones de documentos donde se dispone de preguntas especiales para testear la relevancia de los documentos. Estas preguntas en general, las formulan los usuarios que tienen interés en la disciplina que se trata en el sistema IR. Los valores de relevancia los define un panel de expertos en dicha disciplina. En estos casos experimentales se cuenta con un conjunto de preguntas donde *se conocen* las respuestas 'correctas'. En general se asume que si un sistema IR da como resultado un buen nivel de recuperación bajo diferentes condiciones *experimentales*, entonces dicho sistema podrá ejecutarse en forma correcta en una situación *operacional* donde la noción de *relevancia* no se conoce con anterioridad.

Medidas de Evaluación

Los sistemas IR puede ser evaluados según diferentes criterios :

- Eficiencia de ejecución.
- Eficiencia en el almacenamiento.
- Eficiencia en la recuperación .
- Características que ofrecen a los usuarios.

Los diseñadores del sistema deben decidir la importancia relativa de estos factores, y la selección apropiada de las estructuras de datos y los algoritmos de implementación dependerán de estas decisiones.

Eficiencia de Ejecución

La eficiencia de ejecución se mide por el tiempo que le toma al sistema o a parte del sistema, ejecutar un proceso computacional. Este criterio ha sido siempre el criterio de evaluación más importante de los sistemas IR debido a que en general, estos sistemas son interactivos y si se tuviera un largo tiempo de recuperación, interferiría con la utilidad del sistema. Los requerimientos no funcionales de los sistemas IR, usualmente especifican un tiempo máximo aceptable para la búsqueda y para las operaciones de mantenimiento de la BDs, tales como el agregado y borrado de documentos .

Eficiencia en el Almacenamiento

La eficiencia en el almacenamiento se mide por el número de bytes que se necesitan para almacenar los datos. Una medida común de este criterio se refiere al *overhead* de espacio ocupado, el cual es el radio del tamaño de los archivos índice más el tamaño de los archivos de documentos, sobre el tamaño de los archivos de documentos.

Eficiencia de la Recuperación

La mayoría de los experimentos sobre sistemas IR se han enfocado sobre la eficiencia en la recuperación. Usualmente esta eficiencia se basa en *juicios de relevancia* . Esto ha sido un problema ya que los juicios de relevancia son subjetivos y poco confiables. Es decir que diferentes juicios asignarán diferentes valores de relevancia a un documento recuperado en respuesta a un *query* dado. Los investigadores de esta área, consideran que el problema de la confiabilidad de los juicios de relevancia no es suficiente para invalidar los experimentos que utilizan juicios de relevancia.

Se han propuesto muchas medidas de la eficiencia de la recuperación de un sistema IR . Las medidas más comúnmente usadas son :

* *Recall* .
* *Precision* .

Recall

Esta medida se define como el radio de los documentos relevantes recuperado para un *query* dado sobre el número de documentos relevantes para dicho *query* en la BDs . Excepto en el caso de tener colecciones de test pequeñas, este denominador generalmente no es conocido y debe ser estimado utilizando ejemplos o por algún otro método. Se puede representar de la siguiente manera :

$$\text{RECALL} = \frac{|A \cap B|}{|A|}$$

Siendo A = Número de documentos relevantes
B = Número de documentos recuperados

Precision

Esta medida se define como el radio del número de documentos relevantes recuperados sobre el número total de documentos recuperados. Se puede representar de la siguiente manera :

$$\text{PRECISION} = \frac{|A \cap B|}{|B|}$$

Siendo A = Número de documentos relevantes
B = Número de documentos recuperados

Ambas medidas toman valores entre 0 y 1 .

Se han desarrollado métodos para evaluar ambas medidas simultáneamente, ya que el general se compara la *performance* de los sistemas IR según estas dos medidas.

Uno de los métodos involucra el uso de grafos *recall - precision* , formado por puntos donde un eje es la medida *recall* y otro eje representa la medida *precision*. La *Figura 3.4* representa un ejemplo de este grafo .

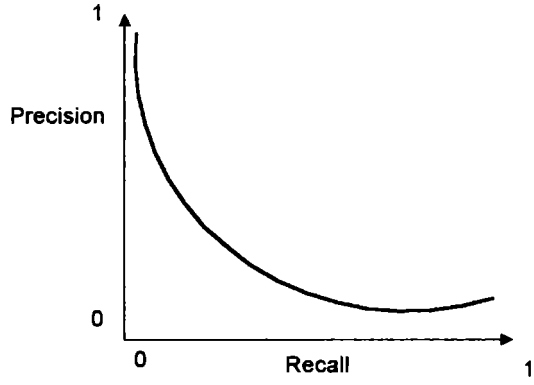


Figura 3.4

Grafo Recall - Precision

Los puntos *recall - precision* muestran que ambas medidas están relacionadas inversamente. Es decir que cuando se incrementa la *precision*, la medida *recall* típicamente disminuye y a la inversa .

Los experimentos sobre IR a menudo utilizan colecciones de test que consisten en una BDs de documentos y un conjunto de *queries* para los cuales se disponen de juicios de relevancia.

INTERNET Y WORLD WIDE WEB

CAPITULO 4

Este capítulo incluye :

INTERNET

Presentación de los aspectos básicos referentes a Internet

¿Qué es INTERNET?

Internet es una red de redes de computadoras de alcance mundial. Se utiliza un protocolo de comunicación común, el *Internet Protocol* (IP). Aquellas redes que utilizan otros protocolos, también están unidas a Internet a través de *gateways* (puertas) que traducen dichos protocolos al formato IP.

Es una red que vincula grandes empresas de comunicaciones que brindan servicios, así como miles de pequeñas redes universitarias, de gobierno y de corporaciones comerciales.

Internet nace en 1969 como un proyecto de la Agencia de Proyectos de Investigación de Defensa Avanzados de EE. UU. (*Defense Advanced Research Proyects Agency* ó DARPA). Su origen se debe a la necesidad de intercambiar información entre los investigadores y científicos militares ubicados en sitios apartados aún en casos en que la red fuese parcialmente destruida (temor fundado especialmente en amenazas nucleares). Esta característica determinó en gran medida su diseño y modo de administración, con lo cual desde sus comienzos Internet fue una red carente de planificación y organización central.

La aplicación más utilizada no fue la que se había definido originariamente (el intercambio de datos de investigaciones vía Telnet), sino fue el correo electrónico o *e-mail* .

Se considera que un host está "conectado" a Internet si puede enviar correo electrónico a cualquier otra computadora dentro de la red.

Servicios que brinda INTERNET

Los servicios que brinda Internet son los siguientes :

①Servicios de recuperación de información (FTP y Gopher)
②Servicios de búsqueda de información (WAIS,Archie,Verónica)
③Servicios de comunicaciones (Correo Electrónico, UseNet, IRC)
④Servicios de información hipertexto/hipermedia (WWW)

① Recuperación de Información

❖ FTP

Uno de los servicios desarrollados para Internet fue FTP (*File Transfer Protocol*). Este servicio brinda a los usuarios la posibilidad de transferir archivos de un lugar a otro. Permite al usuario loguearse a una máquina remota , visualizar los directorios y copiar desde y hacia el servidor .

❖ **Gopher**

Es un sistema de BDs distribuidas al cual se accede mediante un menú. Los sitios en Internet que distribuyen información por este sistema, utilizan servidores Gopher los cuales permiten a los clientes Gopher visualizar y bajar archivos y directorios. La funcionalidad de Gopher es similar a FTP, pero Gopher permite acceder a servicios adicionales para la visualización y bajada de archivos. Es fácil de usar y permite incluir textos descriptivos de los contenidos de los archivos.

② Búsqueda de Información

❖ **WAIS**

El servicio WAIS (*Wide Area Information Server*) es un sistema que permite búsquedas indexadas por contenido, es decir permite encontrar artículos que contengan cierto grupo de palabras atravesando todos los archivos de Internet. El servidor mantiene índices globales de todos los documentos, o al menos de aquellos conocidos por él , lo que permite efectuar una búsqueda con gran detalle.

❖ **Archie**

Este servicio permite localizar archivos que se pueden bajar mediante el servicio de FTP. Indexa los nombres de los archivos FTP alrededor del mundo. Para utilizar este servicio hay que contar con un cliente Archie o poseer acceso a un servidor Archie vía Telnet. Es similar a WAIS, pero la búsqueda se realiza según los nombres de los archivos y su ubicación.

❖ **Verónica**

Este servicio es como si fuera la versión Gopher del servicio Archie. Permite incluir textos descriptivos para cada archivo que se publica. La búsqueda se orienta mas a hacia los textos que hacia el nombre de archivo. No sólo se busca por nombre de archivo, sino también se puede utilizar información por tópicos .

③ Comunicaciones

❖ **Correo Electrónico**

El correo electrónico fue uno de los primeros servicios desarrollados para Internet. Permite el envío de mensajes de una computadora a otra, permitiendo comunicar a las personas en forma rápida y a través de grandes distancias. Las direcciones tienen la forma : *nombre@host.dominio* .

* ***Formato***

El formato de los mensajes para el correo electrónico se rige por la norma RFC 822. El protocolo de transporte se lo conoce como SMTP (*Simple Mail Transfer Protocol*). Se encuentran dos divisiones, el encabezado (sobre) y el contenido, aunque es una división semántica ya que el mensaje es una cadena ASCII . Estas partes están divididas por una línea en blanco.

Encabezado

El encabezado contiene un conjunto variable de campos que indican al sistema quien es el remitente, de qué máquina proviene el mensaje, cual fue la ruta seguida, fecha de envío, destinatario, etc. El destinatario de un mensaje puede ser una única persona, un grupo de personas o algún tipo de sistema que analice sintácticamente el mensaje y ejecute un procedimiento en consecuencia. Del mismo modo, el remitente puede estar constituido por una única persona, un grupo o un sistema.

Los campos del encabezado tienen el siguiente formato : *nombre_del_campo : contenido_del_campo*. Los dos puntos (:) actúan como delimitador . El contenido del campo puede ser estructurado o no. Entre los campos más importantes encontramos :

- Campos del Emisor :
  - FROM : Indica el autor del mensaje (persona).
  - SENDER: Indica quien envía el mensaje (persona, sistema , proceso).
  - REPLY TO: Indica a quien se le debe enviar la respuesta del mensaje.

- Campos del Receptor :
  - TO : Identifica el receptor primario.
  - CC : Identifica los receptores secundarios.

MESSAGE-ID : Identificador del mensaje.

- Campos del Recorrido :
  - RETURN-PATH: Contiene el recorrido de vuelta del mensaje.
  - RECEIVED: Existirá uno de estos campos por cada punto que haya tocado el mensaje.

- Campos de Referencia :
  - SUBJECT : Indica el motivo del mensaje .
  - COMMENTS : Comentarios .
  - ENCRYPTED: Indica la naturaleza del encriptado del mensaje.

- Campos definidos por el usuario :

El usuario puede especificar otros campos de la forma : X-no*mbre_campo* , donde *X* indica que es un campo extra y *nombre_campo* se refiere al nombre definido por el usuario.

Mensaje

El mensaje propiamente dicho, puede contener texto (ASCII de 7 u 8 bits), o el código binario codificado en forma tal que permite su traslado libremente a través de Internet y volver a su forma original en la computadora del destinatario. También el mensaje puede contener un juego de instrucciones con la sintaxis adecuada para ser interpretada y ejecutado en la máquina destino, la que entonces devuelve la salida nuevamente en forma de mensaje, al remitente.

Suponiendo que el mensaje contenga un archivo que debe ser interpretado por algún utilitario, sencillamente se ubica éste archivo en el directorio más conveniente y se ejecuta el producto necesario.

*** *Proceso***

A medida que los mensajes van transitando en la red, se van agregando líneas de cabecera conocidas como *Received* (recibido), en éstas se indica el vecino inmediato que pasó el mensaje, el protocolo que se utilizó en la entrega , la fecha y la hora. Por lo tanto cuando el mensaje llega a destino, se puede reconstruir el camino realizado por el mensaje.

*** *Mailing Lists***

Estos listados de distribución de correo permite que un grupo de personas con intereses comunes compartan información. En la actualidad existen programas que administran estos listados de direcciones de los usuarios pertenecientes a un grupo de interés. Esta función de administrador la realiza el servidor de correo, el cual distribuye los mensajes a todos los miembros del grupo

❖ Internet Newsgroups (UseNet)

Los *newsgroups* o grupos de noticias reúnen a un conjunto de usuarios que tratan temas de interés específicos. UseNet (abreviatura de *Users' Network*) está formada por un grupo de computadoras que almacenan información generada para los *newsgroups*. Las noticias se distribuyen enviando los mensajes individuales (artículos) de una computadora local a todas las computadoras que forman parte de UseNet. Estas computadoras se conocen como servidores de noticias (*news*) y utilizan su propio protocolo NNTP (*Network News Transport Protocolo*).

❖ Internet Relay Chat (IRC)

Este servicio permite que muchas personas "charlen simultáneamente". Es una aplicación cliente/servidor. Si un usuario quiere conversar con otro, debe correr un cliente IRC y conectarse a un servidor IRC. Permite mantener conferencias en tiempo real.

④ Información Hipertexto/Hipermedia

❖ World Wide Web

Permite que el acceso a Internet sea sencillo. Mediante *links* en un documento se puede acceder a otro, el cual puede estar ubicado en otro servidor en cualquier parte del mundo. Esta herramienta se encuentra detallada con más profundidad en la *sección Estructura General de la WWW* (Pág 63).

ESTRUCTURA GENERAL DE LA WORLD WIDE WEB

Presentación de los aspectos básicos de la WWW

Definición

Debido al gran éxito que ha tenido la WWW en relación con Internet, muchas veces se refiere a ésta como equivalente a Internet. Sin embargo la Web es un sistema muy distinto. En primer lugar, la Web no es una red sino un sistema de aplicación (un conjunto de programas de software). En segundo lugar, la Web puede ser utilizada en diferentes clases de redes y a su vez podría ser usada en una computadora que no estuviera conectada a ninguna red. Podemos definir a la Web como sigue :

> La WWW es un sistema de comunicación e información de hipertextos popularmente utilizado en la red de computadoras Internet con una operación de comunicación de datos basada en el modelo cliente/servidor. Los clientes Web (*browsers*) puede acceder a través de diferentes protocolos y a información hipermedia (contando el *browser* con aplicaciones de ayuda) utilizando un esquema de direccionamiento.
> [December-Ginsburg95]

En la *Figura 4.1* se puede observar la organización técnica de la Web según esta definición .

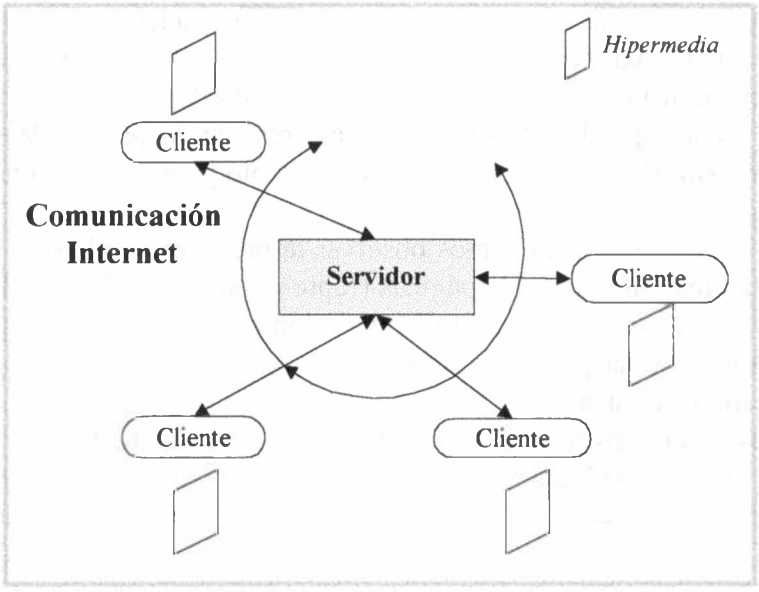


Figura 4.1
Organización Técnica de la Web

La Web provee lo siguiente :

✷ Un medio para proveer información potencialmente útil en forma tal que pueda ser accedida por usuarios distribuidos y muchas veces distantes.

✷ Un medio para que los usuarios puedan acceder a información almacenada en sitios distribuidos sin requerir conocimiento acerca de los mecanismos subyacentes de implementación de tales accesos.

✷ Un medio para estructurar información de forma tal que la misma pueda ser descubierta, recuperada y visualizada por aquellos que la encuentren de utilidad.

La Web fue desarrollada originariamente en el Laboratorio de Física de Partículas Europeo de Génova (*European Particle Physics Laboratory*) como una iniciativa del CERN en Suiza, con el fin de proveer un medio de intercomunicación entre los científicos pertenecientes al área de Física de Partículas aprovechando la red ya existente: Internet, y así unificar los medios de acceso a una gran cantidad de información disponible en línea. El proyecto fue llevado a cabo por Tim Berners-Lee, introduciéndose por primera vez a finales del año 1991 en el CERN Computer Newsletter. Este sistema presentaba similaridades con el sistema WAIS para búsqueda de archivos y datos, sin embargo la principal diferencia fue que la WWW incorporaba el hipertexto como su aspecto central.

Hipertextos

La WWW se basa en hipertextos. Esto significa que la información que se encuentra en la Web no necesariamente es lineal. Permiten no sólo representar piezas de información o conocimiento sino también estructurarlo u organizarlo mejor. No siempre las estructuras jerárquicas (árboles) son adecuadas para ciertos tipos de texto. En realidad el uso de grafos, parece más adecuado. Por lo tanto vemos que, en términos matemáticos la Web es un grafo dirigido donde los nodos (las páginas de hipertexto) se conectan con aristas (los *links* de hipertexto).

El usuario puede seleccionar ciertas áreas de las páginas Web, las cuales se conocen como *anchors*, y recuperar otro documento y visualizarlo en la interface Web (o *browser*).

En la *Figura 4.2* podemos observar la organización básica de un hipertexto. Se observan los *links* entre las páginas (representados por flechas) los cuales conectan un *anchor* en una página del hipertexto con otra página o con una ubicación específica dentro de una página. Estos *anchors* a menudo se presentan como un texto intensificado o subrayado .

Las características del hipertexto es que el texto no es necesariamente lineal y además involucra la noción de información *ilimitada*, ya que se puede acceder a información escrita por otros autores, con lo cual no se limita a un trabajo simple.

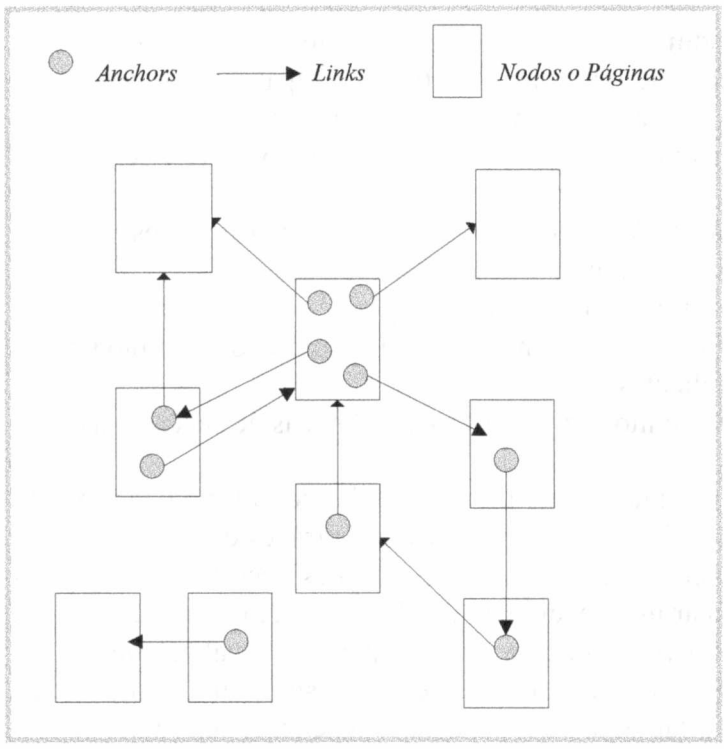


Figura 4.2

La organización del hipertexto

La WWW es un sistema de Comunicación e Información

La Web permite tanto la distribución como la recolección de información . Por lo tanto no es un sistema de distribución de información en un solo sentido, sino que también permite comunicación interactiva. Utilizando la capacidad de *Forms* de HTML (Pág 77) con programación *gateway* se puede desarrollar un sistema que sea manipulado por el usuario.

La WWW es utilizada en la red de computadoras Internet

El software de la Web no necesita una red o el uso de los protocolos de Internet para la transmisión de datos. Sin embargo su uso más popular está asociado a la red Internet, permitiendo el acceso a la información.

La WWW utiliza operaciones de comunicación de datos según el modelo cliente/servidor

Un modelo cliente/servidor para un sistema de red de computadoras involucra tres componentes : el cliente, el servidor y la red .

Un **cliente** (*browser*) es el software de aplicación que a menudo corre en la computadora del usuario y permite visualizar la información. Este software puede estar adaptado según el sistema de hardware del usuario, es decir que el cliente tiene la libertad de presentar la información en el mejor modo disponible según las características operativas. Actúa como una interface desde este sistema y la información provista por el servidor.

Un **servidor** es el software de aplicación que a menudo corre en la computadora del proveedor de información y tiene la capacidad de proveer la información solicitada, ya sea mediante el envío de los documentos reales o a través de la generación de hipertextos virtuales generados al momento en respuesta a los requerimientos recibidos.

Un servidor Web cumple cuatro funciones principales :

- Controlar el acceso a sus recursos.
- Proveer páginas HTML.
- Correr programas de enlace como por ejemplo CGI, y transmitir las salidas correspondientes.
- Monitorear y registrar estadísticas de acceso y uso.

El usuario puede iniciar un pedido de información o una acción a través del software del cliente. Este pedido viaja a través de la red hacia el servidor. El servidor lo interpreta y lleva a cabo las acciones necesarias, las cuales pueden incluir buscar o modificar información en una BDs. Se retornan al cliente los resultados de la transacción requerida (si existe alguno), para su visualización.

Toda comunicación cliente/servidor sigue un conjunto de reglas o protocolos, los cuales están definidos por el sistema cliente/servidor.

La *Figura 4.3* refleja el pedido de un usuario a un servidor y la transmisión de información del servidor al cliente .

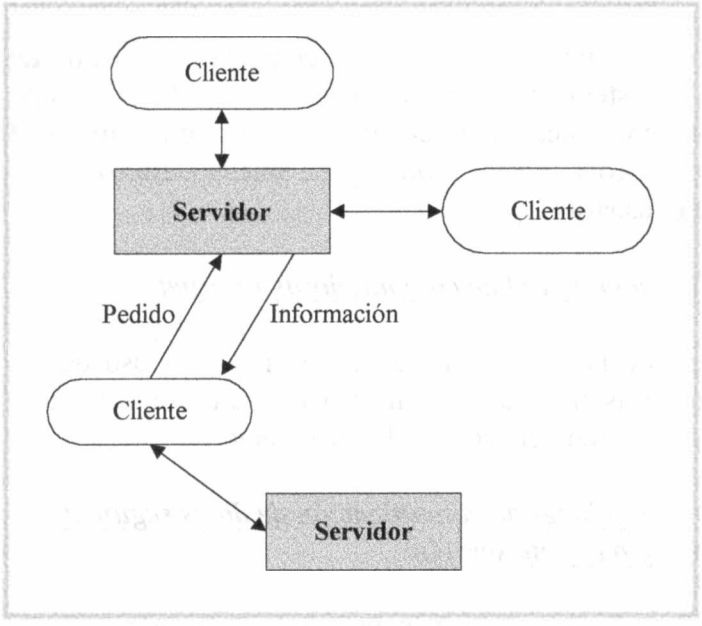


Figura 4.3

Modelo de Comunicación de Datos cliente/servidor

Un cliente puede acceder a muchos servidores empleando protocolos que ambos entiendan.

La forma distribuida de las actividades de pedido y servicio de información, permite muchas ventajas. Como se puede adaptar el software del cliente para una computadora en particular, el servidor no se debe "preocupar" acerca de las particularidades de hardware del software cliente. Por ejemplo, un cliente Web (un *browser*), puede desarrollarse para computadoras Macintosh y pueden acceder a cualquier servidor Web.

Este mismo servidor Web puede ser accedido por un *browser* Web escrito para una *workstation* UNIX que corre bajo el sistema X - Windows. Esto permite desarrollar la información en forma sencilla debido a que existe una clara división entre las obligaciones del servidor y del cliente. No se necesita desarrollar versiones diferentes de información para las diferentes plataformas de hardware debido a que estas adaptaciones se tratan en el software del cliente en cada plataforma.

Los clientes Web (browsers) pueden acceder a distintos protocolos de comunicación

Los *browsers* Web pueden acceder a diferentes servidores que proveen información utilizando un conjunto de reglas para la comunicación (protocolo). En general los protocolos más utilizados son :

- HTTP (*HyperText Transfer Protocol*) : Este es el protocolo "nativo" de la Web, diseñado específicamente para transmitir hipertextos a través de una red. Utiliza una conexión por cada requerimiento de documento y envía un identificador del documento el cual puede estar acompañado de palabras de búsqueda. El servidor responde enviando un documento hipertexto o un documento "simple" (*plaintext*). A medida que llegan los bytes enviados por el server, el *browser* los va mostrando al usuario.
- FTP (*File Transfer Protocol*) : Fue diseñado para permitir a los usuarios transferir archivos binarios o de texto entre computadoras a través de una red.
- Gopher Fue diseñado para compartir información utilizando un sistema de menúes.
- News (*Network News Transfer Protocol, NNTP*) : Se utiliza para la distribución de noticias Usenet .
- Telnet : Se utiliza para conectarse a una computadora (posiblemente remota).

Por lo tanto, un *browser* Web actúa como un cliente Gopher cuando accede a un servidor Gopher o como un cliente News cuando accede a un servidor Usenet.

URL (*Uniform Resource Locator*)

Se conoce como URL a la dirección que hace referencia a recursos de Internet (texto, imágenes, sonido, páginas y documentos en la WWW, así como también recursos de gopher, mail, usenet, etc). Consiste de un string de caracteres que identifican unívocamente a un recurso. Los URLs se utilizan dentro de un documento HTML para crear *links* a los recursos de Internet. En el caso en que el usuario seleccione uno de estos *links*, el recurso al cual hace referencia será recuperado a través de la red y mostrado por el *browser* Web.

Un URL puede estar formado por los siguientes elementos:

protocolo:	//host	:puerto	/camino	;parámetros	?query	#fragmento

➲ El protocolo de intercambio entre el cliente y el servidor : http, ftp, gopher,news, etc .

➲ Número IP o el nombre de la máquina host.

➲ El árbol de directorios (el camino) que conduce al documento y el nombre del archivo.

Menos frecuentemente, esta dirección puede contar con lo siguiente :

➲ Número de puerto TCP que utiliza el protocolo del servidor.

➲ Información de autentificación (*username* y *password*).

➲ Parámetros que se pasarán a un programa en la llamada de un enlace ejecutable en caso que se necesiten.

➲ Un query para un programa CGI.

➲ Un fragmento, el cual es una referencia a un subconjunto del objeto referenciado. En este caso el *browser* Web visualizará una parte específica dentro de archivo referenciado .

El formato y la disposición de estos elementos varían según los protocolos que se utilicen.

◆ La *sintaxis mínima* utilizada para representar un URL es : *protocolo://nombre_del_servidor*

Cuando no se especifica un nombre de archivo, se acude al archivo predeterminado del servidor, habitualmente la *home page* .

◆ La sintaxis que se encuentra habitualmente es:

protocolo://nombre_del_servidor/directorio/subdirectorio/nombre_del_documento

◆ La sintaxis completa es :

protocolo://username;password@nombre_del_servidor:puerto/subdirectorio/ nombre_del_documento?argumentos

Los clientes Web pueden acceder a información Hipermedia

Hipermedia es un hipertexto que no necesariamente contiene sólo texto [December-Ginsburg95]. Una hipermedia puede incluir gráficos, fotos,video y sonido (*multimedia*) .

Debido a que los hipertextos de la Web incluyen *links* con diferentes protocolos y comunicación a través de una red, podríamos decir que la Web es una hipermedia en red, es decir una hipermedia que no está restringida a un servidor de información simple [December-Ginsburg95].

La *Figura 4.4* representa las relaciones en una hipermedia en red, reflejando los posibles *links* desde una página de un hipertexto hacia servidores con distintos protocolos, así como *links* a documentos ya sea de texto, sonido gráficos o video.

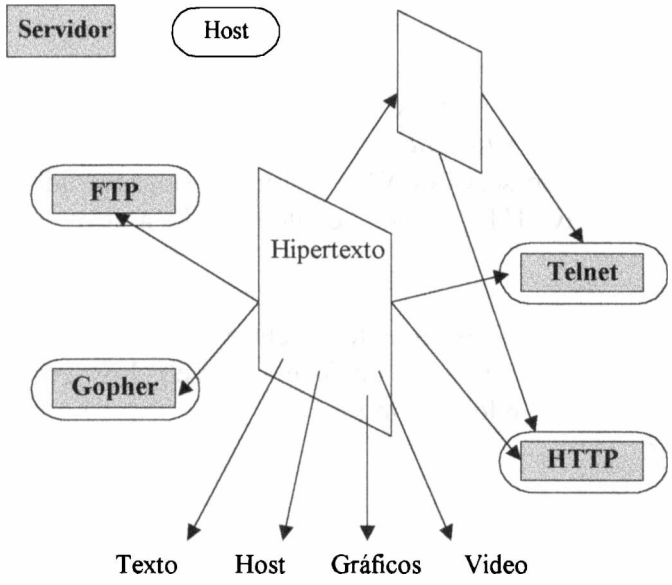


Figura 4.4

La organización de la Web vista como una hipermedia en red

El acceso a Hipermedias se facilita por medio de aplicaciones de ayuda

El *browser* Web para visualizar una información multimedia, invoca un software que está incluido en estas aplicaciones de ayuda (*Helper Applications*). Una aplicación *Helper* es un programa que permite ver un tipo de contenido específico y que es ejecutada por el *browser* como un proceso separado. Por ejemplo para que un usuario pueda ver un video, el servidor Web debe tener instalado el software apropiado para visualizar un video. La gran ventaja es que prácticamente cualquier formato puede ser visualizado ya que el *browser* sólo actúa de mediador, determinando el tipo de contenido y pasando a la aplicación, la tarea de interpretarlo.

Como *conclusión* se puede definir a la Web de la siguiente manera :

La Web se utiliza para la distribución global de información.
Podemos definirla como :

Web = Hipertexto + Multimedia + Red

Hipertexto es la base para la asociación de los links
Multimedia presenta datos e información en múltiples formatos y sentidos
La red es la esencia del alcance global

La Web dentro de INTERNET

La Web es una aplicación que utiliza las herramientas de Internet para la comunicación y el transporte de información. Su poder reside en que permite acceder a los recursos de Internet a través de un sistema de hipertexto.

Desde el punto de vista de un usuario, la Web consiste en un conjunto de recursos de Internet a los cuales se puede acceder a través de un *browser* Web. Conecta estos recursos por medio de hipertextos creados utilizando el lenguaje HTML(presentado en la *sección Estructura de las Aplicaciones en la WWW*(Pág 75)). Estos archivos están ubicados en un servidor Web y pueden ser accedidos por los *browsers* Web (clientes). Un archivo HTML puede contener links a otros recursos de Internet.

La *Figura 4.5* representa las conexiones entre un documento HTML y otros recursos de Internet, y la relación entre un *browser* Web, los servidores de información y los archivos ubicados en los servidores.

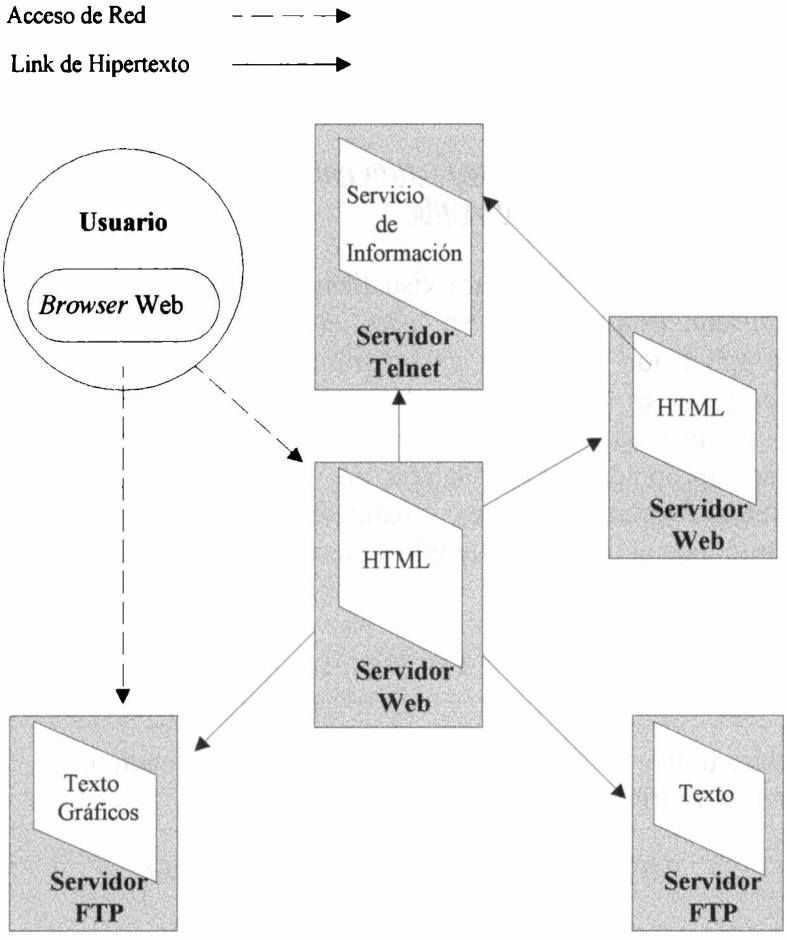


Figura 4.5

La Web dentro de Internet

BÚSQUEDA Y RECUPERACIÓN DE INFORMACIÓN EN LA WWW

Presentación de los métodos de búsqueda de información en la WWW

A través de Internet se puede acceder a gran variedad de información, sin embargo esta información no se encuentra organizada, con lo cual una de los problemas es poder encontrarla. Para solucionar esto surgieron los *buscadores.*

Un buscador es un conjunto de programas instalados en un servidor conectado a Internet. En general está compuesto por :

* *programas robot*, los cuales recolectan la mayor cantidad de información de todo el mundo y la almacenan en grandes BDs.
* *un equipo de expertos*, quienes organizan la información y la catalogan por temas (categorías) o palabras clave.
* *un servidor Web*, el cual a través de un programa que examina la BDs del buscador, recibe los pedidos de información del usuario y le devuelve los resultados.

Existen diferentes maneras de buscar y recuperar información :

① Búsqueda temática.
② Búsqueda por palabra clave.
③ Búsqueda por espacio geográfico.
④ Búsqueda por espacio de información.
⑤ Búsqueda por personas.

① Búsqueda Temática

Existen muchas situaciones en las cuales un usuario desea obtener información acerca de un tema, sin tener una idea precisa de los tópicos específicos. En estos casos puede ser que el usuario no desee utilizar un sistema de búsqueda por palabra clave debido a que no conoce un conjunto de palabras que puedan representar dicho tema. Con lo cual, el objetivo será encontrar información categorizada y organizada según temas y subtemas. De esta forma el usuario puede encontrar información descriptiva general acerca de un tema y luego refinar la búsqueda para encontrar tópicos más específicos.

Actualmente en la Web existen varios recursos de información orientados en forma temática. Podemos mencionar entre otros :

✱ *The WWW Virtual Library* : Fue creado por el CERN. Permite obtener la información según diferentes categorías y está mantenida por varias personas.

✱ *Yahoo* : Es una gran colección de *links* Web escrita y mantenida por David Filo y Jerry Yang.

② Búsqueda por Palabra Clave

Esta técnica de búsqueda por palabra clave, es una buena estrategia en el caso en que el objetivo del usuario es encontrar cierta información, pero no necesariamente la información relacionada que pudiera encontrarse a través de un búsqueda temática.

En general a las herramientas de búsqueda por palabra clave en la Web se las conoce como *spider*, *robots* ó *wanderes*. Constituyen una clase de programas que recorren la Web y coleccionan información acerca de lo que van encontrando, creando una lista de URLs para una búsqueda posterior. Otros *spiders*, *observan* los documentos HTML buscando URLs y palabras claves en los campos del título o en otras partes de los documentos. Podemos mencionar entre otros :

✸ *Lycos* : Permite localizar documentos que contienen referencias a palabras clave y examinar los documentos. El usuario puede determinar si un documento le parece importante, sin tener que recuperarlo. Utiliza un esquema probabilístico para pasar de un servidor a otro en la Web. Lycos comienza con un URL y recolecta información del recurso, incluyendo títulos y encabezados, las 100 palabras de mayor *peso* (el mismo se obtiene utilizando un algoritmo que considera la ubicación de la palabra y la frecuencia de ocurrencia, entre otros factores), las primeras 20 líneas , el tamaño en bytes y el número de palabras. Aunque las primeras herramientas de búsqueda (*spiders*) afectaban al servidor debido a la cantidad de accesos, Lycos utiliza un comportamiento de búsqueda *random* para evitar acceder al mismo servidor repetidamente en un corto período de tiempo. Desarrollado en la Universidad *Carnegie Mellon.*

✸ *WebCrawler* : Encuentra referencias a URLs en la Web y construye la BDs que va estar disponible para la búsqueda. Construye *índices* de los contenidos de los documentos que encuentra además de la información referida a los URLs , texto principal y títulos. Retorna una lista de *links* que hacen *matching* con las palabras claves introducidas por el usuario. Fue desarrollado por *Brian Pinkerton.*

③ Búsqueda por Espacio Geográfico

Esta técnica de búsqueda es útil cuando un usuario está buscando un servidor en particular, en cuyo caso lo más probables es que conozca su ubicación geográfica. Existen varias aplicaciones Web que permiten al usuario encontrar recursos Web presentados a través de un mapa geográfico o de listas de servidores ordenados en forma geográfica. Podemos mencionar entre otros :

✸ *The Virtual Tourist I* : Este servidor, desarrollado por *Brandon Plewe*, actúa como una interface visual de la distribución geográfica de la WWW y de otros servidores de red. Si el usuario selecciona una región puede obtener más información acerca de dicha región.

✸ *CityLink* : Ofrece información acerca de localidades en la Web utilizando como interface mapas donde el usuario puede seleccionar diferente regiones. Se refiere a las ciudades de EE.UU. y como un servicio ofrece material acerca de dichas ciudades. En este caso se enfoca la información turística.

④ Búsqueda por Espacio de Información

Un *espacio de información* es el conjunto de toda la información que se encuentra disponible en los *servidores* de un determinado protocolo. Por ejemplo, el *espacio Gopher* consiste de toda la información y los archivos que se encuentran disponibles a través de los servidores Gopher.

Cada espacio de información presenta sus datos en su propio formato y está definido como el conjunto de toda la información que se encuentra en todos los servidores de ese tipo.

Para buscar información en este caso, el usuario recuperará una lista muy extensa de todos los servidores del tipo buscado. Podemos mencionar entre otros :

✸ *WAIS Space* : WAIS, es un sistema que permite recuperar información basándose en índices de documentos. WAIS Inc. mantiene un directorio de servidores, con todas las BDs WAIS accesibles en el mundo. Esta lista hace referencia a los nombres de las BDs, en vez de los nombre de los servidores o su ubicación geográfica .Un usuario puede buscar una BDs en particular que contenga el tema o las palabras claves buscadas.

✸ *Web Space* : La lista de servidores Web en CERN, está organizada en forma geográfica, pero también puede estar organizada según los nombre de las máquinas. *Matthew Gray* ha creado un robot Web (llamado *World Wide Web Wandered*) para *viajar* a través de la Web y crear una BDs de sitios WWW. Esta lista la información está organizada por nombre de servidor. Permite encontrar servidores particulares según su nombre de dominio.

⑤ Búsqueda por Personas

Aunque los métodos de búsqueda por palabra clave o temática pueden encontrar personas en particular, en ciertos casos es útil poder buscar una persona dentro de directorios de *home pages* .

Existen directorios al estilo de una agenda telefónica y colecciones de *home pages* que permiten buscar personas en la Web. Podemos mencionar entre otros :

✸ *NetFind* : Desarrollado por la Universidad *Nova*. Como un robot Web, NetFind busca en el espacio de directorio de servidores y retorna una lista con la información que hizo *matching* con el patrón de búsqueda ingresado por el usuario.

✸ *Who's Who on the Internet* : A través de un servicio de búsqueda por palabra clave, permite obtener una lista de páginas *home pages* con una breve descripción de cada página. La búsqueda se realiza sobre una BDs donde se registran *home pages* . Es parte de la *WWW Virtual Library*.

Otra posibilidad es encontrar personas según la organización a la cual pertenecen. Para realizar este tipo de búsqueda, se puede utilizar entre otros:

✸ *Open Market's Commercial Sites Index* : Ofrece una lista de servicios comerciales y productos. Un usuario puede buscar por palabra clave o por orden alfabético. Es un servicio no pago ofrecido por Open Market Inc.

✸ *The Internet Business Directory* : Ofrece una lista de negocios e incluye la posibilidad de realizar una búsqueda por palabra clave dentro de esta lista.

ESTRUCTURA DE LAS APLICACIONES EN LA WWW

Presentación de los componentes involucrados en aplicaciones WWW

HTML (Hypertext Mark-Up Language)

El lenguaje HTML se utiliza para crear hipertextos en la Web. Fue desarrollado como un lenguaje de marcas para indicar la estructura lógica de un documento. Permite identificar las partes estructurales de un documento, tales como párrafos, listas, encabezados, etc. Según estas identificaciones de las partes del documento, los programas que traducen documentos HTML (*browsers* Web) los muestran en una forma adecuada para leerlo.

Esta organización permite separar la especificación estructural de un código HTML de su apariencia mostrada a través de un *browser* Web.

HTML fue definido utilizando el *Standard Generalized Mark-Up Languaje (SGML)*, un estándar internacional (ISO 8879:1986, Information Processing - Text and Office Systems-) para el procesamiento de información de textos.

SGML es un meta-lenguaje (un lenguaje para definir lenguajes). El objetivo es ayudar a presentar la información *on-line* para una eficiente distribución electrónica, búsqueda y recuperación en una forma independiente de los detalles de apariencia de un documento. Cuando el programa que hace la presentación mezcla el documento SGML con la información del estilo, se puede observar la apariencia del documento. La *Figura 4.6* refleja la idea básica de este proceso .

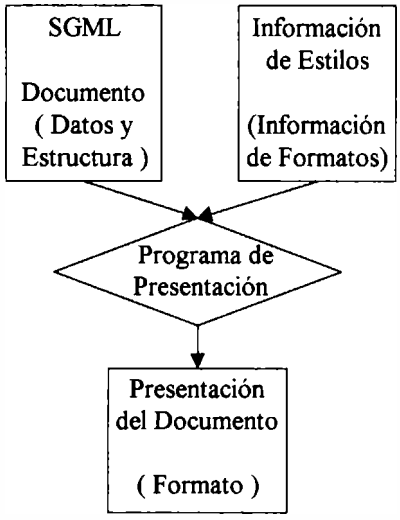


Figura 4.6

Organización del Procesamiento de Documentos SGML

Los *datos* de un documento consisten en los contenidos del mismo, ya sea texto o multimedia. Los *tags* (marcas) en un documento SGML identifican la *estructura*, encabezados, títulos, párrafos, etc. Finalmente el *formato* de un documento es su apariencia final después de mezclar los datos, la estructura y la especificación de cómo presentar estos datos. Estas partes son independientes, los datos del documento pueden crearse sin tener en cuenta la estructura que van a tener. La estructura puede agregarse sin preocuparse acerca de cómo van a ser presentados. Las especificaciones del formato pueden crearse siguiendo un estilo en particular.

Es conveniente utilizar SGML para codificar los datos debido a que es un estándar internacional. Los *tags* en un documento SGML favorecen la reusabilidad (los segmentos de los documentos puede re-utilizarse en otros documentos) y la búsqueda, debido a que ciertas marcas permiten identificar el contenido del documento y ayudan en la búsqueda electrónica .

Filosofía del Lenguaje HTML

HTML sigue la misma filosofía de independencia de datos, estructura y formato que SGML.

Aunque HTML no es un lenguaje complicado como algunos lenguajes de programación, para utilizarlo el autor debe seguir ciertas reglas específicas para *marcar* las partes del documento. La idea de marcar un texto para expresar su estructura es diferente del método de procesamiento de texto WYSIWYG (*what you see is what you get).* En este caso los autores concentran los datos, la estructura y el formato en una vez .En caso de trabajos individuales esto puede ser útil, pero para grandes sistemas de documentos e información, los lenguajes de marcas son más eficientes.

Cuando se utiliza HTML, el programador define la estructura del documento de manera tal que cualquier *browser* pueda entenderla y presentar el documento en la forma más adecuada. Esto permite desarrollar información HTML sin tener que crear versiones distintas para cada *browser* existente.

La ventaja de HTML es que se basa en un texto ASCII, con lo cual un archivo HTML es fácil de comprender y puede editarse en un editor de texto simple.

Niveles del Lenguaje HTML

El nivel 0 y 1 de HTML, abarca un conjunto de características básicas para expresar información, las cuales son reconocidas por casi todos los *browsers* actuales. La aparición de *browsers* gráficos como *Netscape* y *Mosaic*, provocó la necesidad de incorporar nuevas características que extienden la idea de hipertexto, permitiendo la interacción con el usuario. Una de estas incorporaciones es el ***Form*** especificado en un nivel 2 de HTML, y disponible para usuarios de *Netscape, Mosaic* y otros *browsers.*

Forms

Los *forms* se utilizan para obtener información de los usuarios, así como para implementar nuevos métodos de interacción con el usuario.

Presentan una interface que consiste en campos en blanco para que sean completados por el usuario, listas para ser chequeadas seleccionando alguna opción y otros métodos para obtener información por parte del usuario. Utiliza pares de variables (*nombre_del_campo*, *contenido_del_campo*). El usuario completa el o los campos que desee o selecciona datos de una lista. También se pueden definir valores por defecto.

Cada *form* tiene asociado un método, que especifica cómo se van a manejar los datos del *form* en el procesamiento, y una acción que identifica el programa ejecutable (*gateway program*)que procesará los datos del *form*.

El programa ejecutable puede acceder a BDs o a otro *software*. Según los resultados obtenidos, se puede presentar un documento HTML en el *browser* del usuario mostrando los resultados de la ejecución de este programa. El fin de la edición del *form* ocurre cuando el usuario pulsa un botón especial llamado botón de sumisión, es en este momento cuando el programa recibe el conjunto de datos del *form*. La *Figura 4.7* presenta las relaciones entre un *form*, un programa ejecutable y una BDs .

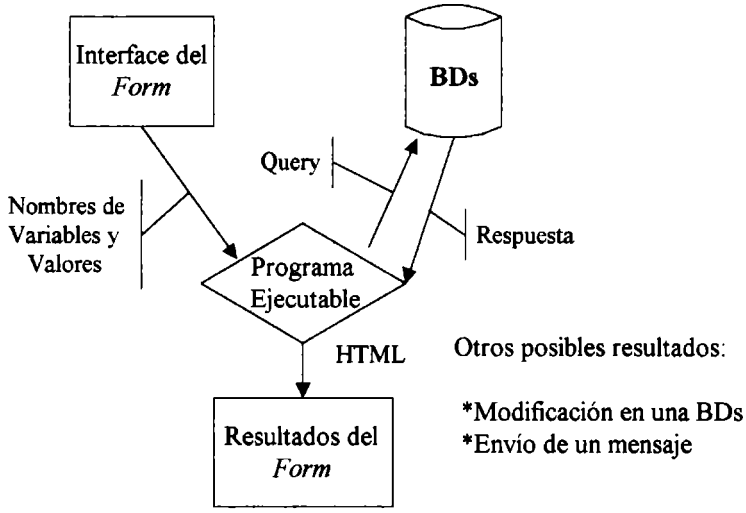


Figura 4.7

Relaciones y Funcionalidad de un Form

En algunos servidores Web, los programas ejecutables deben ubicarse en un directorio especial para este tipo de programas (usualmente es el directorio /cgi-bin u otro directorio designado en la configuración del servidor Web) .

HTTP (HyperText Transfer Protocol)

HTTP es el protocolo utilizado por los servidores Web para *negociar* la comunicación entre el servidor y los clientes.

Este protocolo define un conjunto de mensajes que corresponden a dos categorías : mensajes de pedido por parte de los clientes y mensajes de respuesta por parte del servidor.

Necesidad del Protocolo HTTP

Para realizar las operaciones básicas en Internet, no se necesita utilizar WWW. Se pueden transferir archivos ASCII o imágenes binarias de una máquina a otra utilizando FTP; nos podemos conectar a una máquina remota utilizando Telnet ; la mayoría de las máquinas soportan el correo electrónico, etc.

La cuestión a tener en cuenta es que ciertos servicios básicos como telnet y FTP, establecen conexiones con el usuario de larga duración. Además en los tiempos antes que surgiera el protocolo HTTP, no existía una manera de publicar un recurso de hipermedia para toda la comunidad de Internet.El único recurso era escribir acerca del sitio determinado en los *newsgroups* y entonces permitir a los usuarios acceder a esta información a través de operaciones FTP anónimas. Con lo cual la información hipermedia podía ser accedida sólo por un grupo de usuarios.

Ninguno de estos servicios básicos solos o combinados permitían que máquinas diseminadas por el mundo colaboren en un ambiente hipermedia.

En 1991, *Tim Berners-Lee* implementa el protocolo HTTP, el cual permite publicar información hipermedia que pueda ser accedido en forma global y además transferirla a otros sitios. Este protocolo es muy poderoso y constituye la esencia de la World Wide Web.

Características de HTTP

Entre las principales características acerca de este protocolo podemos mencionar :

◆Es un protocolo *sin estado*, esto significa que toda comunicación entre el servidor y el cliente es única y diferente; comienza con el pedido del cliente y termina con la respuesta correspondiente del servidor. No hay un registro como en el caso de FTP (que mantiene la sesión hasta que el usuario se desconecte o sea por *timeout* del servidor). En definitiva sólo hay un pedido y una respuesta.

◆Es rápido. El cliente hace un pedido, el servidor responde y punto.

◆Cuando el servidor HTTP transmite información al cliente, incluye un encabezado MIME (*Multipart Internet Mail Extension*) para "decirle" al cliente que tipo de datos forman la respuesta. La interpretación de estos datos (imagen, video, etc) corre por parte del cliente, el cual debe contar con el utilitario apropiado para dicha interpretación. El formato MIME se encuentra detallado más adelante (Pág 82).

CGI (Common Gateway Interface)

A partir que surge el protocolo HTTP se logra la colaboración remota, los clientes podían pedir información hipermedia a los servidores remotos y verla localmente. Si consideramos una sesión del cliente antes que surgiera CGI, el cliente sólo puede navegar de un *link* de hipertexto a otro, cada uno de los cuales puede contener texto, video, sonido,etc. Las acciones asociadas a este tipo de sesión se ven reflejadas en la *Figura 4.8* .

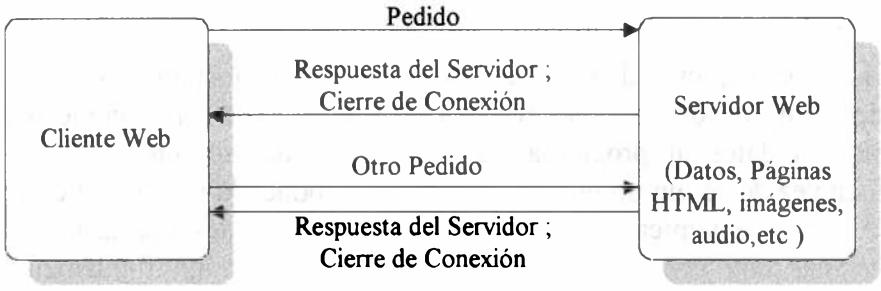


Figura 4.8

Sesión de comunicación sin utilizar CGI

En este caso se navega de un link a otro,lo que produce recuperar datos ya existentes en algún servidor. No existe una forma de pedir datos al usuario y no es posible construir documentos HTML en forma dinámica en el momento de un pedido.

CGI es una interface entre el servidor HTTP y otros recursos de una computadora. Es un conjunto de variables y convenciones para pasar información desde el servidor al cliente y a la inversa. Podemos decir que CGI es el *gateway* o puerta común que utiliza el servidor para comunicarse con aplicaciones diferentes al *browser*. Un programa CGI actúa como un enlace entre cualquier aplicación que lo necesite y el servidor, siendo este último el responsable de recibir información del *browser* , así como de enviar de vuelta los datos.

Su nombre se debe a lo siguiente:

Common : La idea es que cada servidor y cliente, sin tener en cuenta su plataforma, utiliza el mismo mecanismo estándar para el flujo de datos entre el cliente, el servidor y el programa de ejecución (programa *gateway*). Esto permite un alto grado de portabilidad entre la vasta variedad de máquinas y sistemas operativos.

Gateway : Un programa CGI en general actúa como un mediador entre el servidor HTTP y cualquier otro programa que puede aceptar en ejecución cierta entrada como línea de comando (por ejemplo, por la entrada estándar (stdin) o variables de ambiente). Esto significa que si tenemos por ejemplo, un programa SQL que no tiene mecanismo para comunicarse con un servidor HTTP, puede ser accedido por un programa CGI (*gateway)*. Los programas CGI se pueden desarrollar utilizando diferentes lenguajes.

Interface : El mecanismo estándar provee un ambiente completo para los programadores. No se necesita conocer los detalles de los servidores HTTP, una vez que se aprende la interface, se puede desarrollar un programa CGI ; todo lo que se debe saber acerca del protocolo HTTP, es cómo fluyen los datos como entrada y salida.

El servidor no sólo se encarga de enviar información ya existente según el pedido del usuario, sino que también puede presentar diferentes documentos dependiendo del pedido. La programación CGI permite implementar extensiones del lado de los servidores Web. Hasta el momento los servidores HTTP son los únicos que soportan CGI.

La especificación de CGI permite crear nuevos documentos en forma dinámica, es decir en el momento que el usuario hace el pedido. El cliente puede pasar argumentos o datos al programa CGI a través del servidor HTTP. Un programa CGI, en vez de ser un programa estático que produce la misma salida cada vez, actúa en forma dinámica respondiendo a las necesidades del usuario. La creación de páginas con *contenido dinámico* ofrece una ventaja fundamental : personalizar la entrega de información de interés específico al cliente.

Flujo de Datos

Podemos ver en la *Figura 4.9* como es el flujo de datos utilizando CGI.

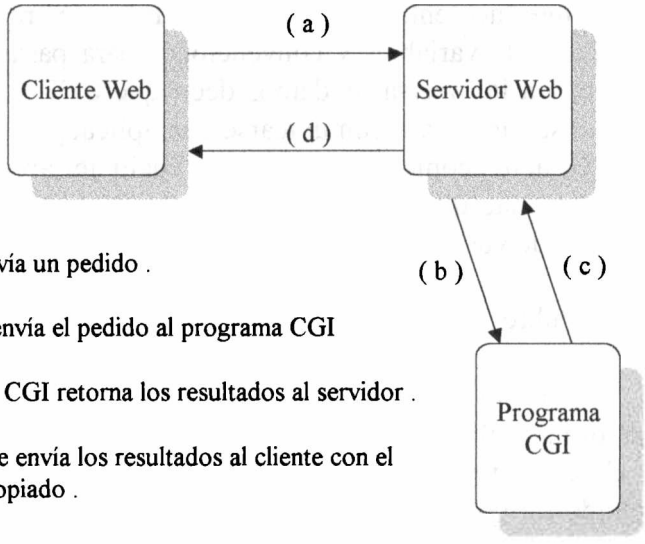


Figura 4.9

Interacción Cliente/Servidor

Flujo de Datos utilizando CGI

En primer lugar, se transmiten los datos del cliente al servidor(a). El servidor entrega el pedido al programa CGI para su ejecución (b). Si existe una salida como respuesta, es enviada al servidor (c). En este caso el servidor le envía la salida al cliente (d). Se termina la conexión inicial entre el cliente y el servidor(e).

En forma más detallada, la transacción se realiza de la siguiente manera :

(a) El cliente envía un pedido (un determinado URL) al servidor. Este pedido debe incluir el tipo de servicio deseado (por ejemplo : HTTP, FTP, telnet, etc) y la ubicación del recurso (por ejemplo: //nombre_de_máquina ó IP/ nombre_de_archivo).

(b) El servidor HTTP analiza el pedido y decide qué hacer a continuación. En caso en que no sea un pedido HTTP, se incluye el servicio apropiado. Por ejemplo para un pedido FTP, se recuperará el archivo apropiado y se lo envía al *browser* del cliente.

Si es un pedido HTTP, el servidor ubica el archivo solicitado para retornarlo al cliente. Se envía un archivo HTML y en la mayoría de los casos el servidor no analiza o interpreta el archivo, el cliente se encarga de analizar las marcas (*tags*) para presentarlo al usuario.

Si el servidor reconoce que el archivo pedido es un archivo ejecutable o un programa CGI, se va a ejecutar incorporándole lo siguiente :

* A través de las *variables de ambiente*, se envía al programa CGI el encabezado recibido del cliente, si lo hubiera, y su propio encabezado.

* En caso que se utilice parámetros, se envía estos parámetros ingresados por el usuario al programa CGI. Estos datos pueden ser enviados a través de variables de ambiente o como una entrada al programa por línea de comando, dependiendo del método utilizado en el *form* de pedido de datos (GET o POS).

(c) El programa CGI analiza la entrada recibida a través del servidor y genera una respuesta y/o salida para enviar de vuelta al servidor, considerando lo siguiente :

* Si no existen datos para retornar, de cualquier manera el programa CGI debe enviarle al servidor una indicación que se ha ejecutado. Hay que recordar que en este punto la conexión HTTP *todavía está abierta*, con lo cual es muy importante que el programa CGI retorne una respuesta (aún cuando no involucre datos) para poder terminar la conexión.

* Si existen datos para retornar al cliente, el programa CGI debe incluir antes de los datos, un encabezado que el servidor pueda entender, el cual debe representar un formato acorde con las convenciones de formatos MIME. Por lo tanto los datos pertenecerán a un tipo indicado en el encabezado de la respuesta

(d) El servidor lee la salida del programa CGI y nuevamente decide qué hacer, según el encabezado recibido. Si el encabezado es de tipo "*Location*", el servidor recupera el archivo o indica al cliente que recupere el archivo. Si el encabezado es de tipo "*Content-Type*", el servidor le envía los datos de vuelta al cliente, con lo cual el cliente es el responsable de manejar los datos y presentarlos al usuario en forma apropiada.

(e) Una vez que el cliente ha recibido los datos, la conexión HTTP se cierra.

Salida del Programa CGI

La salida de un programa CGI debe tener un formato especial para que tanto el servidor como el cliente la puedan entender. Se compone de dos partes separadas por una línea en blanco: el encabezado, que incluye datos que utilizará el servidor para construir la cabecera HTTP de su respuesta al cliente, y el cuerpo, que se compone de datos que constituyen el documento a enviar al cliente.

➲ *Encabezado*

Existen (actualmente) tres encabezados que el servidor puede reconocer :

❖ *Content-type*: Una de las principales funciones del encabezado es indicar el tipo de datos que generará el programa CGI, debido a que puede generar no sólo documentos HTML, sino también sonido o imágenes, entre otros. Este tipo se indica por medio de la línea *Content-type* . La sintaxis utilizada es la sintaxis MIME (*Multipart Internet Mail Extensions*) :

Content-type: type/subtype <line feed> <line feed>

El formato MIME apareció en Internet como una extensión del protocolo SMTP (*Simple Mail Transfer Protocol*). Este protocolo es el protocolo estándar para el intercambio de correo en Internet, previsto para transferir sólo archivos de texto. Con la aparición de la multimedia, surgió la necesidad de transferir , además de archivos de texto, imágenes, sonidos, etc. Con el formato MIME es posible intercambiar archivos multimedia entre dos máquinas. El diálogo SMTP se ha enriquecido con nuevos mandatos que permiten al emisor, entre otras cosas, indicar el tipo y la longitud del documento que envía.

El mandato *Content-type* permite indicar la naturaleza del archivo que se va a transmitir. La lista de los diferentes tipos posibles se ha normalizado. Cada tipo se define por la asociación de un tipo general (imagen, sonido, video, texto,etc) y un subtipo que indica el formato exacto del archivo.

La máquina que recibe esta información necesita realizar una asociación entre cada tipo de datos recibido y una aplicación capaz de gestionar estos datos.

Existe un archivo especial que asocia cada tipo/subtipo con una o más extensiones de archivos. Antes de enviar el archivo, el emisor examina la extensión del mismo y recupera el tipo/subtipo asociado para crear el mandato *Content-type*.

Los tipos más habituales en la programación CGI comprenden los siguientes :

Tipo/Subtipo	Descripción
text/html	Es el tipo más utilizado, indica al cliente que los datos deben interpretarse como mandatos HTML .
text/plain	Indica al cliente que los datos son de texto plano, que no deben interpretarse en ningún sentido .
image/gif *image/jpeg* *image/x-xbitmap*	Son los tres tipos de imágenes soportados en general por los clientesWWW sin llamar a un *viewer* externo.
audio/basic	Es un tipo de formato que engloba todos los formatos de sonido *.au y *snd. En general, el cliente llama a un programa externo para interpretar estos sonidos.
application/postscript	Indica al cliente que los datos están en formato postscript. En general, el cliente llama a un intérprete externo para mostrar el documento postscript .

❖ *Location* : En este caso, el servidor ignora todo dato que continúa al encabezado y *redirige* la salida , es decir le indica al usuario que recupere el archivo especificado por el URL. Es útil cuando como respuesta al programa CGI se desea ir a un URL existente y no generar un documento nuevo. La sintaxis es :

Location : URL

❖ *Status*: En este caso el servidor altera el número y texto de un mensaje por defecto que normalmente le retornaría al cliente. Cuando no se especifica esta línea, el código de retorno es '*200 OK*". La sintaxis es :

Status : código mensaje

➲ *Cuerpo*

La segunda parte de la salida estándar de un programa CGI es el cuerpo, el cual contiene los datos del documento. Debe corresponder al tipo MIME anunciado en la línea *Content-type* en el encabezado. Caso contrario el cliente será incapaz de mostrar el documento.

Desarrollo

Sistema IR de documentos CFPs

CAPITULO 5

Este capítulo incluye :

INTRODUCCIÓN AL SISTEMA IR DE DOCUMENTOS CFPS

Definición general de los objetivos y ventajas del sistema de Recuperación de Información de Documentos CFPs

Como hemos mencionado en el *Capítulo 2 -Nuestro Enfoque-* (Pág 13), el objetivo de este sistema es organizar la información referente a los pedidos de papers para conferencias (CFPs), recuperando la información relevante de dichos documentos y almacenándolos en una BDs para su posterior consulta. Además de almacenar esta información relevante sobre el documento CFP también se conserva dicho documento en su formato original, de manera tal que *puede ser consultado* por los usuarios que deseen obtener más información acerca de dicho CFP.

A manera de introducción presentamos las funcionalidades básicas que comprenden las fases que conforman el sistema.

El proceso de recuperación de información, puede dividirse en tres fases :

- Fase de recolección de documentos .
- Fase de recuperación de información de documentos.
- Fase de incorporación de datos recuperados a una BDs.

✸ Fase de Recolección de Documentos

La recepción de los documentos CFPs a analizar se realiza a través del correo electrónico. Es decir, el autor de un CFP debe enviar este documento a una cuenta de *e-mail*. La forma de difundir los CFPs no varía ya que actualmente se utiliza el correo electrónico para ésto. Sin embargo, entre las ventajas que brinda utilizar el sistema se encuentran las siguientes :

➲ El CFP puede alcanzar una mayor difusión, debido a que el Sistema de Consulta de Información de Documentos CFPs que actúa en forma conjunta con este sistema (detallado en el *capítulo 6* (Pág 133)), es de dominio público por lo tanto dicho CFP puede ser consultado por cualquier usuario de Internet sin necesidad que pertenezca a algún grupo de interés en particular. En consecuencia la difusión de CFPs no tiene un ámbito de recepción limitado.

➲ El organismo encargado de difundir la conferencia puede optar por enviar el CFP a esta cuenta de *e-mail* sin la necesidad de enviarlo a todos los usuarios y listas de interés que considera involucrados con el tema. Evitando tener que contar con listas actualizadas de usuarios y de personal disponible para realizar dicha tarea.

➲ Se podría evitar en ciertos casos la recepción múltiple de CFPs. Si un usuario por ejemplo pertenece a más de una lista de interés y el encargado de difundir la conferencia lo envía a varias listas, puede llegar a recibir el mismo CFP por varias vías distintas.

✸ Fase de Recuperación de Información de Documentos

Esta fase extrae los documentos recibidos en la fase anterior. A cada uno de los CFPs extraídos se le aplica un *proceso de análisis automático de texto* que, combinando reglas heurísticas y técnicas IR, recupera la información considerada relevante de dicho documento. Esta fase es el núcleo del sistema.

✸ Fase de Incorporación de Datos Recuperados a una BDs

Si la fase anterior recuperó acerca de un documento la suficiente información de manera tal que sea representativa del mismo, estos datos son incorporados a una BDs para su posterior consulta. Si dicho documento ya se encuentra almacenado en la BDs se modifica con los nuevos datos recuperados, dado que para la difusión de una conferencia se suele publicar más de una versión de CFP, con lo cual en la BDs se mantendrán los datos de la última versión publicada.

Estas fases se presentan con mayor profundidad en la siguiente sección.

DEFINICIÓN CONCEPTUAL DEL SISTEMA IR DE DOCUMENTOS CFPS

Definición y presentación funcional del Sistema de Recuperación de Información de Documentos CFPs

En primer lugar presentamos un esquema general del Sistema de Recuperación de Información de Documentos CFPs, el cual fue presentado en el esquema de nivel 1 en el *capítulo 2 - Nuestro Enfoque* - (Pág 13).

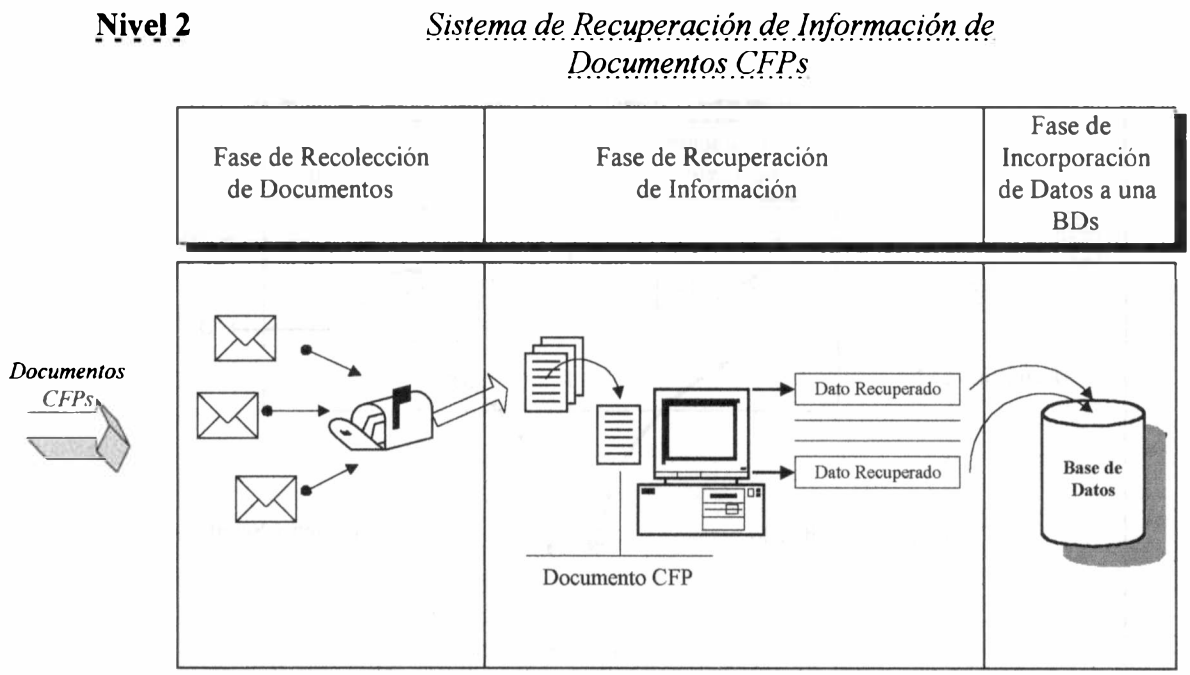


Figura 5.1 *- Visión Esquemática del Sistema de Recuperación de Información de Documentos CFPs-*

Este esquema representa las tres fases presentadas en la sección anterior.Cada una de ellas será detallada a continuación .

1.- FASE DE RECOLECCIÓN DE DOCUMENTOS

La incorporación de documentos CFPs al sistema, tiene su punto de partida cuando el comité organizador de una conferencia envía su CFP a la cuenta de *e-mail* destinada para el uso del sistema de Recuperación de Información.

El usuario que extrae los *e-mails* de esta cuenta es el propio sistema sin la intervención de personas a cargo. Esta recolección se hace periódicamente en forma automática.

En esta fase, el estudio se centra en el servicio prestado por el correo electrónico por medio del cual se van recibiendo los documentos CFPs, acumulándose conformando la muestra a ser procesada luego por la *fase de recuperación de información* .

2.- FASE DE RECUPERACIÓN DE INFORMACIÓN

La fase de recuperación de información es el núcleo de este sistema. Presentamos esta fase mediante un esquema general que representa su funcionalidad

Nivel 3 *Fase de Recuperación de Información*

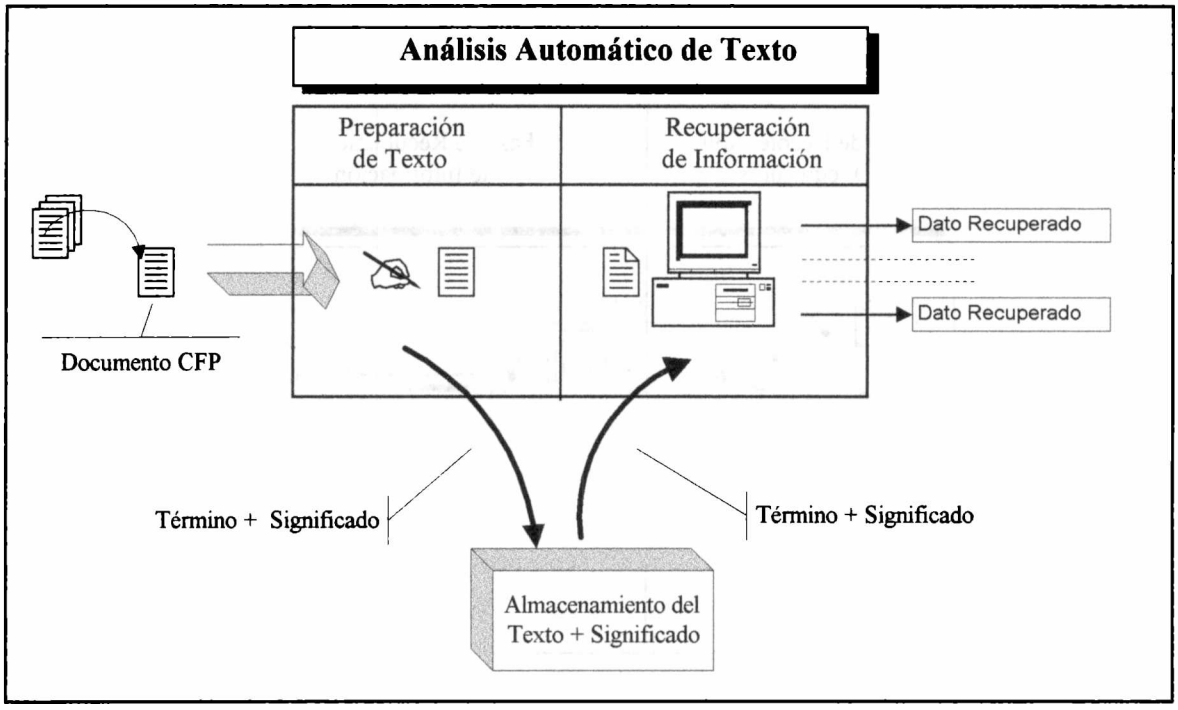


***Figura 5.2** - Visión Esquemática de la Fase de Recuperación de Información -*

En este esquema se pueden reconocer los siguientes pasos :

① Cuando se inicia esta fase, en primer lugar se verifica en la cuenta de *e-mail* la existencia de al menos un documento. En caso negativo se termina el proceso. En caso afirmativo se continúa al paso siguiente .

② Se determina la cantidad de *e-mails* que arribaron. Se extrae del mailbox cada uno de estos documentos transformándolos en archivos de texto.

③ A cada uno de estos documentos se le aplica un **análisis automático de texto**. Como resultado se obtienen, en caso de éxito, los datos considerados relevantes. Estas nociones serán explicadas mas adelante.

Se repite el paso ③ hasta que se terminen de analizar todos los documentos que arribaron a la cuenta de *e-mail* .

2.1.- Análisis Automático de Texto

A partir de un documento de texto, el objetivo de este análisis es recuperar los datos relevantes del documento.

Antes de continuar profundizando sobre este análisis debemos determinar qué significa que un dato sea relevante para un documento. Los *datos relevantes* serán aquellos que sean representativos de un documento. Para ampliar este concepto analizamos la noción de representación de un documento en un sistema de recuperación de información.

Representación de Documentos

Como se ha visto en el *Capítulo 3* referente a *Information Retrieval (IR)* (Pág 29), en el general de los casos los sistemas de recuperación de información almacenan sólo una representación de los documentos en vez del texto completo, ya que no es necesario almacenar todo el texto. Por lo tanto una cuestión asociada a la definición de un sistema de recuperación de información es justamente obtener la representación que se considere adecuada. Dicha representación podría ser por ejemplo, una lista de palabras extraídas del documento consideradas representativas o significativas de dicho documento.

Con el fin de obtener dicha representación en el caso de nuestro sistema analizamos diferentes cuestiones. La primera a tener en cuenta fue el hecho que este sistema tiene como objetivo analizar documentos de tipo CFP. Como adelantamos en el *capítulo 2 -Nuestro Enfoque-* (Pág 13) cuando presentamos la idea general del sistema, luego de analizar un número significativo de CFPs observamos que estos documentos cuentan con una estructura general donde se pueden encontrar ciertas categorías de términos, tales como : sigla, título, fecha, ciudad y país .

Una *categoría de término* hace referencia a un conjunto de caracteres que está asociado a un significado. Estas categorías de términos se conservan a pesar de la variedad de los CFPs aunque puede variar el orden, la existencia o el formato de las mismas. El contar con esta información acerca de la estructura de los documentos a analizar nos permite definir una representación de documentos basada en éstas categorías de términos.

Como **representación de un documento CFP**, consideramos a las siguientes categorías de términos :

- Sigla .
- Título .
- Tema .
- Fecha .
- Ciudad .
- País .
- URLs asociadas .
- Conferencia asociada .

Todas en conjunto conforman una representación de un documento CFP, sin embargo algunas de ellas son más representativas que otras. No se cumple que en todos los casos se encuentren todas estas categorías en el documento, o que puedan ser todas recuperadas por el sistema. En ciertos casos sólo se recuperará algunas de ellas.

Además, puede suceder que un CFP no haga referencia solo a una conferencia sino que también haga referencia a una conferencia asociada, si sucede esto se intenta recuperar el título y/o la sigla de dicha conferencia. Con lo cual contamos no sólo con los datos de la conferencia de difusión del CFP, sino también con el título y/o sigla de la conferencia a la cual está asociada, si es que la hubiera.

Consideramos como **condición necesaria** para obtener la representación de un documento que se haya recuperado la sigla ó el título de la conferencia, debido a que cualquiera de estas dos categorías permiten reconocer una determinada conferencia. La existencia de al menos una de estas categorías implica que la representación del documento es válida y por lo tanto es considerada, esto último sería la ***condición de éxito*** del análisis automático del texto.

En caso de no contar con al menos una de estas categorías, consideramos que el sistema no obtuvo la información suficiente para clasificar dicho documento, con lo cual la información referente a él no es almacenada. Hacemos esta consideración debido a que el resto de las categorías que pudieran haber sido recuperadas tales como fecha, ciudad o país, no permiten identificar una determinada conferencia. Por lo tanto el usuario a partir de los datos presentados no podría llegar a decidir si dicha conferencia es de su interés o no.

Por lo tanto, el análisis automático de texto tendrá como objetivo reconocer las categorías de términos que conforman la representación de los documentos recién definida. Estos datos son los que se consideran ***relevantes*** en el análisis.

Para alcanzar este objetivo, en este proceso de análisis se aplican ciertas operaciones tanto sobre el texto como sobre términos en particular. Básicamente podemos clasificar estas operaciones en dos clases:

- Las operaciones que tienen como objetivo "preparar " al texto de manera tal de facilitar la recuperación de información. Este tipo de operaciones permite por ejemplo, eliminar texto considerado irrelevante para la recuperación de información ó eliminar ciertas palabras del documento que se considera no resultan significativas para el proceso de la recuperación. De esta manera se puede reducir el tamaño del texto a analizar.
- Las operaciones que tienen como objetivo recuperar información a partir del texto.

Como se observa en el esquema de la *Figura 5.2*, el proceso de análisis automático de texto se encuentra dividido según estas dos categorías de operaciones. Básicamente se destaca un proceso que tiene como objetivo facilitar la recuperación de información preparando el texto, y otro que tiene como objetivo dicha recuperación.

El *proceso de preparación del texto* almacena en forma temporaria ciertos datos que serán luego utilizados por el *proceso de recuperación de información* propiamente dicho. Estos dos procesos no actúan en forma secuencial, sino que luego de contar con cierta información mínima comienza el proceso de la recuperación y a partir de ese momento, ambos procesos se superponen en su ejecución. Estas nociones son ampliadas a continuación en el detalle de ambos procesos.

2.1.1.- Preparación del Texto para la Recuperación

Este proceso tiene como objetivo preparar el texto de manera tal de facilitar y mejorar el proceso de recuperación. Se encuentra representado gráficamente en la *Figura 5.3*.

Este proceso recibe como entrada un documento CFP y produce como salida el texto en un formato preparado para la recuperación de información. Se aplica al texto y básicamente tiene como objetivo eliminar toda información considerada innecesaria para la etapa de recuperación. Además agrega otro tipo de información asociada a los términos del texto. Esta nueva información representa un significado asociado a dichos términos .

Los pasos que intervienen en este proceso son los siguientes :

- *Eliminar texto comprendido en el encabezado del e-mail*

En primer lugar dado el documento de texto, aplicamos una operación que tiene como objetivo eliminar del mismo el encabezado que corresponde al *e-mail*. Podríamos preguntarnos por qué nos interesa eliminar esta porción del texto. Luego de analizar los encabezados de *e-mail* de un número significativo de CFPs pudimos notar que la información dentro de ellos interfiere en la recuperación, debido a que este encabezado puede incluir términos pertenecientes a categorías buscadas tales como ciudad o país por ejemplo, sin que correspondan a los datos representativos del CFP. Esto provocaba una disminución en la precisión de la recuperación. Debido a esto y sabiendo que los datos incluidos en los encabezados de *e-mail* no son relevantes en la recuperación de la información, se aplica esta operación al documento.

También eliminamos posibles comentarios que suelen estar presentes antes del comienzo del contenido del CFP. Se reconocen comentarios con un formato específico. Estos pueden llegar a contener información que lleve a recuperar datos erróneos sobre las categorías de términos que representan a dicho CFP. Con lo cual es importante eliminar esta porción del texto.

Por lo tanto se elimina información innecesaria y se disminuye el volumen de texto a analizar.

Nivel 4 *Preparación de Texto*

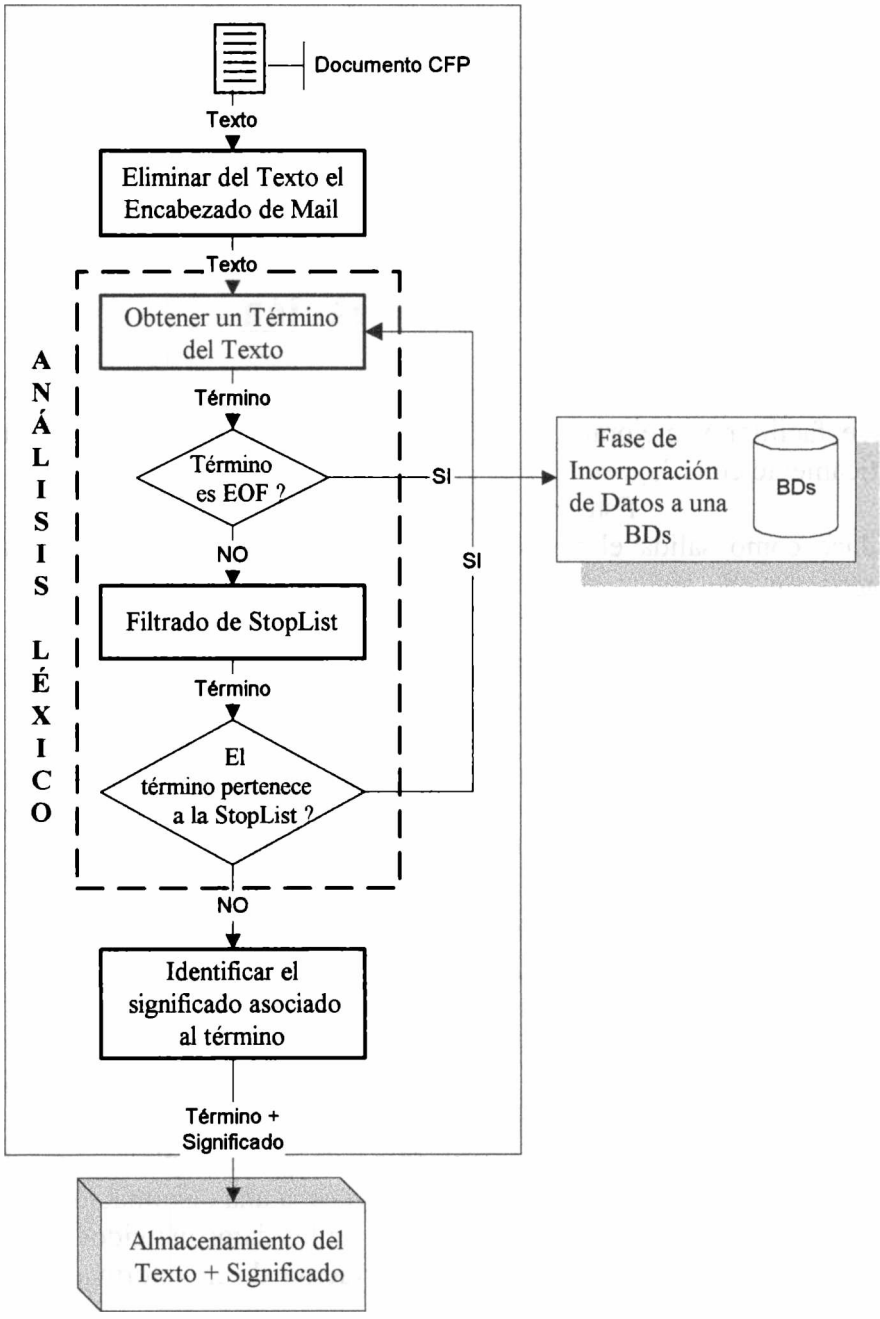


Figura 5.3 *- Visión Esquemática de la Preparación del Texto -*

❷ *Obtener un término a partir del Texto .*

Para identificar un término del texto utilizamos un *analizador léxico*, el cual divide el texto en términos.

Definimos un término como :

* Un conjunto de caracteres numéricos o alfanuméricos separados por un conjunto de caracteres especiales denominados delimitadores.

* Un subconjunto de delimitadores que incluye : **‘ , ( ) [ ] Enter** .

Por lo tanto el conjunto de caracteres delimitadores se divide en dos clases, aquellos que a su vez son considerados términos, con lo cual forman parte de la salida del analizador, y los que sólo actúan como delimitadores permitiendo identificar términos pero son ignorados por el analizador.

Sólo los términos definidos anteriormente son los considerados. Por lo tanto se elimina información innecesaria y se disminuye el volumen de texto a analizar.

Cuando el término obtenido es el fin del archivo entonces continúa la ***Fase de Incorporación de Datos a una BDs***. En caso contrario, continúa el paso siguiente .

❸ *Filtrar palabras no significativas.*

En este paso utilizamos una técnica de *Information Retrieval* (IR). Como se ha detallado en el *Capítulo 3* (Pág 32) referente a este tema, una de las técnicas de Recuperación Información es filtrar del documento las palabras que se encuentran en un diccionario negativo o *stoplist* [BaezaYates-Frakes92].

Estas listas de palabras se conforman de acuerdo al sistema de recuperación de información que se esté desarrollando. Contiene un conjunto de palabras consideradas no significativas para el proceso de la recuperación. Permiten mejorar la *performance* de los sistemas de recuperación de información.

En el caso de nuestro sistema, utilizamos para implementar esta técnica el método de remover los términos *stopwords*[BaezaYates-Frakes92] como parte del análisis léxico. Elegimos este método debido a que como de todos modos debemos realizar un análisis léxico para obtener los términos del texto, entonces cuando obtenemos un término determinamos si pertenece a la *stoplist*, en cuyo caso lo eliminamos del conjunto de términos a analizar.

La *stoplist* utilizada es una modificación de la presentada por Christopher Fox [BaezaYates-Frakes92], la cual a su vez deriva de la definida por Francis y Kucera a partir de un análisis de la literatura inglesa. Este es un conjunto de palabras que no tienen incidencia en las categorías de términos que se desean recuperar. Modificamos esta lista teniendo en cuenta que el sistema no sólo recupera términos, sino también frases completas como es el caso de las categorías de título y tema de la conferencia.

En el general de los casos, estos diccionarios negativos contienen las palabras con mayor frecuencia de ocurrencia en un texto, tal como las preposiciones por ejemplo. En los casos en que sólo se recuperan términos, esto es muy útil debido a que las preposiciones no son términos significativos o representativos de un documento entonces pueden ser ignoradas en la recuperación.

Sin embargo en nuestro caso, muchas preposiciones no están incluidas en este diccionario negativo ya que pueden formar parte de las categorías de términos que comprenden una frase. Por lo tanto este es un caso en el que se representa la dependencia de esta lista de palabras con respecto al sistema a desarrollar.

Existen ciertas listas generales de palabras consideradas *stopwords* (palabras no significativas), a partir de estas se puede adaptar de forma adecuada según el sistema a desarrollar.

Por lo tanto si el término que se está analizando está incluido en esta lista, se ignora y se vuelve al paso ❷ ,en caso contrario se continúa al paso siguiente. Es decir que eliminamos palabras que resultan innecesarias en la recuperación, en consecuencia se disminuye el volumen de texto a analizar.

Utilizamos como diccionario negativo a un conjunto de palabras en Inglés, ya que como se especificó en el *Capítulo 2 -Nuestro Enfoque-* (Pág. 13), se analizan documentos CFPs escritos en este idioma .

Diccionario Negativo o Stoplist utilizado en el Sistema de Recuperación de Información de Documentos CFPs

ABOVE	ACROSS	AFTER	AGAINST	ALL	ALMOST
ALONE	ALONG	ALREADY	ALSO	ALTHOUGH	ALWAYS
AMONG	ANY	ANYBODY	ANYONE	ANYTHING	ANYWHERE
AROUND	ASK	ASKED	ASKING	ASKS	AWAY
BACK	BACKED	BACKING	BACKS	BECAME	BECAUSE
BECOME	BECOMES	BEEN	BEFORE	BEGAN	BEHIND
BEING	BEINGS	BEST	BETTER	BETWEEN	BOTH
BUT	CAME	CAN	CANNOT	CASE	CASES
CERTAIN	CERTAINLY	CLEAR	CLEARLY	COME	COULD
DID	DIFFER	DIFFERENT	DIFFERENTLY	DO	DOES
DONE	DOWN	DOWN	DOWNED	DOWNING	DOWNS
DURING	EACH	EARLY	EITHER	END	ENDED
ENDING	ENDS	ENOUGH	EVEN	EVENLY	EVER
EVERY	EVERYBODY	EVERYONE	EVERYTHING	EVERYWHERE	FACE
FACES	FACT	FACTS	FAR	FELT	FEW
FIND	FINDS	FRIDAY	FULL	FULLY	FURTHER
FURTHERED	FURTHERING	FURTHERS	GAVE	GENERALLY	GET
GETS	GIVE	GIVEN	GIVES	GO	GOING
GOOD	GOODS	GOT	GREATER	GREATEST	GROUP
GROUPED	GROUPING	GROUPS	HAD	HAS	HAVE
HAVING	HE	HER	HERE	HERSELF	HIGHER
HIM	HIMSELF	HIS	HOW	HOWEVER	IF
ITSELF	JUST	KEEP	KEEPS	KIND	KNEW
KNOW	KNOWN	KNOWS	LARGE	LARGELY	LATER
LATEST	LEAST	LESS	LET	LETS	LIKE
LIKELY	LONG	LONGER	LONGEST	MADE	MAKE
MAKING	MAN	MANY	ME	MEMBER	MEMBERS
MEN	MIGHT	MONDAY	MORE	MOSTLY	MR
MRS	MUCH	MUST	MY	MYSELF	NECESSARY
NEEDED	NEEDING	NEEDS	NEVER		
NEWER	NEWEST	NEXT	NO	NOBODY	NON
NOONE	NOT	NOTHING	NOW	NOWHERE	OFF
OFTEN	OLD	OLDER	OLDEST	ONCE	ONE
ONLY	OPEN	OPENED	OPENING	OPENS	ORDER
ORDERED	ORDERING	ORDERS	OTHER	OTHERS	OUR
OUT	OVER	PARTED	PARTING	PER	PERHAPS
PLACES	POINT	POINTED	POINTING	POSSIBLE	PRESENT
PRESENTED	PRESENTING	PRESENTS	PUT	PUTS	QUITE
RATHER	REALLY	RIGHT	RIGHT	ROOM	ROOMS

SAID	SAME	SATURDAY	SAW	SAY	SAYS
SEE	SEEM	SEEMED	SEEMING	SEEMS	SEES
SEVERAL	SHALL	SHE	SHOULD	SHOW	SHOWED
SHOWING	SHOWS	SIDE	SIDES	SINCE	SMALL
SMALLER	SMALLEST	SO	SOME	SOMEBODY	SOMEONE
SOMETHING	SOMEWHERE	STILL	STILL	SUCH	SUNDAY
SURE	TAKE	TAKEN	THAN	THEIR	THEM
THERE	THEREFORE	THESE	THEY	THING	THINGS
THINK	THINKS	THIS	THOSE	THOUGH	THOUGHT
THOUGHTS	THROUGH	THURSDAY	THUS	TODAY	TOGETHER
TOO	TOOK	TOWARD	TUESDAY	TURN	TURNED
TURNING	TURNS	UNDER	UNTIL	UP	UPON
VERY	WANT	WANTED	WANTING	WANTS	WAS
WAY	WAYS	WE	WEDNESDAY	WELL	WELLS
WENT	WERE	WHAT	WHEN	WHERE	WHETHER
WHICH	WHILE	WHO	WHOLE	WHOSE	WHY
WITHOUT	WORKED	WOULD	YEAR	YEARS	YET
YOU	YOUNG	YOUNGER	YOUNGEST	YOUR	YOURS

❹ *Asociar un significado o token al término .*

Este paso tiene como objetivo asociarle un *significado* a un término dado. Este significado será referenciado como ***token*** a lo largo del desarrollo.

Para realizar el proceso de la recuperación de información, no sólo vamos a contar con el texto a analizar sino que además cada término tendrá asociado un token. Recordemos además que el sistema tiene como objetivo reconocer términos con un significado asociado (las *categorías de términos* antes mencionadas (Pág 93)).

Estos tokens asociados a los términos son fundamentales en el proceso de recuperación de información debido a que de éstos dependen la aplicación de las reglas heurísticas. Estas reglas están asociadas a determinados tokens y su aplicación puede modificar los mismos, esto sucede cuando una regla logra recuperar la categoría de término buscada, entonces los tokens que estaban asociados a los términos que forman dicha categoría son reemplazados por el token que la representa.

Definimos un conjunto de tokens a reconocer. Este conjunto lo elegimos luego de analizar un número significativo de CFPs y llegar a la conclusión que con dicho conjunto se podían recuperar las categorías de términos deseadas, debido a que los tokens fueron elegidos en función del comportamiento de las reglas. Este conjunto de tokens permite que las reglas puedan recuperar dichas categorías de términos.

Estos tokens como se detalla mas adelante (Pág. 104) pueden clasificarse según diferentes puntos de vista. Algunos son reconocidos por este paso, mientras que otros que serán presentados mas adelante, son reconocidos por el sistema de recuperación de información empleando las reglas heurísticas definidas.

Algunos de los tokens reconocidos por este paso pertenecen a las categorías de términos que representan a un documento CFP. Para reconocer algunos de ellos se utilizan vocabularios previamente definidos que actúan como diccionarios .

Los tokens reconocidos por este paso son los siguientes: ***día, año, mes, ordinal, país, ciudad, URL, estado, palabra clave, comilla, coma, paréntesis izquierdo , paréntesis derecho, fin de línea, continuación de título, comienzo de tema, comienzo de ciudad, final del titulo, conferencia asociada, comite de programa, CFP,especial*** y ***otro***.

A continuación se detallan algunos con mayor profundidad :

➲ **PAÍS** : Este token puede ser reconocido tanto en este paso como en el proceso de recuperación de información. En este paso se reconocen los países compuestos por sólo un término a partir de un vocabulario previamente definido que comprende un conjunto de nombres de países en Inglés, el cual actúa como un diccionario. Este tipo de término corresponde a una de las categorías que comprenden la representación del documento.

➲ **CIUDAD** : Este token puede ser reconocido tanto en este paso como en el proceso de recuperación de información. En este paso se reconocen las ciudades compuestas por sólo un término a partir de un vocabulario previamente definido que comprende un conjunto de nombres de ciudades en Inglés, el cual actúa como un diccionario. Este tipo de término corresponde a una de las categorías que comprenden la representación del documento .

➲ **ESTADO** : Este token corresponde al nombre de un estado de Norteamérica, el cual puede ser reconocido tanto en este paso como en el proceso de recuperación de información.. En este paso se reconocen estados compuestos por sólo un término a partir de un vocabulario previamente definido que comprende un conjunto de nombres de estados en Inglés, el cual actúa como un diccionario. Se utiliza este tipo para contemplar muchos casos, en los cuales los CFPs que corresponden a ciudades de Norteamérica, indican la ciudad y el estado donde se desarrollará la conferencia pero no indican el país. Por lo tanto al notar esto luego de analizar un gran número de CFPs, decidimos utilizar este token para poder reconocer el país en caso en que éste no se encuentre especificado en el CFP. Se utiliza sólo en el caso de Norteamérica ya que en el resto de los CFPs, procedentes de otros países, aún cuando se indique como lugar de desarrollo de la conferencia la ciudad y el estado también se indica el país.

➲ **MES** : Token que corresponde al nombre de un mes. Se consideran los nombres de los meses en Inglés. Se utiliza en el proceso de recuperación de la fecha.

➲ **ORDINAL** : Token que corresponde a un número ordinal. Se consideran los números ordinales en Inglés representados con números y palabras. Se utiliza en el proceso de recuperación de la fecha y del título de la conferencia.

➲ **URL** : Este tipo de término corresponde a una URL. Es un caso especial ya que está conformada por más de un término.

➲ **COMIENZO DE TEMA** : Token que se asocia con el término Inglés ON. Se utiliza en el proceso de recuperación del tema de la conferencia.

➲ **COMIENZO DE CIUDAD** : Token que se asocia con el término inglés IN . Se utiliza en el proceso de recuperación de la ciudad de la conferencia.

➲ **PALABRA CLAVE** : Token que corresponde a una palabra clave en la recuperación del título de la conferencia debido a que, en general, se encuentra incluida en el mismo. El conjunto de palabras claves se encuentra en un archivo externo al sistema, de manera tal que *puede ser modificado* agregando o eliminando palabras claves y así modificar la recuperación del título de la conferencia. Este conjunto incluye las siguientes palabras: *symposium, meeting, congress, conf, conference, international, workshop, sessions, colloquium, seminar, forum* y *annual.*

➲ **CONTINUACIÓN DE TÍTULO** : Token que corresponde a un término que indica posible continuación de título tal como : *and, or, of,on* y *for* . Se utiliza en el proceso de recuperación del título de la conferencia.

➲ **FINAL DE TÍTULO** : Token que corresponde a términos que resultan claves como indicadores del final del título. Estos términos comprenden : *university, sponsored y hotel* . Además corresponden a este token frases tales como : *will be held , to be held, was held* etc . Se utilizan en el proceso de recuperación del título de la conferencia.

➲ **CONFERENCIA ASOCIADA** : Token asociado a frases tales como: *in conjunction with , affiliated with, as part of, in association with, in connection to,* etc . Estas frases indican la posible existencia de una conferencia asociada.

➲ **COMITE DE PROGRAMA** : Token asociado a frases tales como : *program commitee, organization commitee,* etc. Estas frases indican en general, que a continuación se detalla una lista de las personas que conforman el comité de evaluación u organización. Distinguimos este tipo de token, debido a que como en esta lista suele incluirse junto con los nombres de las personas el país y la ciudad de residencia, esto puede provocar una recuperación errónea de estas categorías de términos, por lo tanto por medio de este token podemos distinguir donde comienza esta lista y así evitar considerar los nombres de países y ciudades que se incluyen en la misma. Cuando el sistema se encuentra en un estado en el cual está analizando este tipo de lista, no se consideran las categorías *ciudad* y *país* recuperadas.

➲ **CFP** : Token asociado a frases tales como : *Call for Papers, Call for Participation*, etc. Este token permitir delimitar un título y además permite cambiar el estado del sistema si este se encontrara en el estado de análisis de un comité de programa.

➲ **ESPECIAL** : Token que corresponde a caracteres delimitadores que se encuentran incluidos en un URL.

➲ **OTRO** : Token que se asocia con todo término que no se corresponde con ninguno de los tokens anteriores.

La función de estos tokens se verá reflejada en el proceso de recuperación.

❺ *Almacenar provisoriamente el término junto con su token asociado.*

Una vez que asociamos un token con un término esta información es almacenada provisoriamente, debido a que será utilizada por el proceso de recuperación de información.

En este punto final de la preparación del texto contamos ahora con la siguiente información : a partir del texto original hemos ido eliminando toda información considerada irrelevante para el proceso de la recuperación y además asociamos un token a los términos que van a intervenir en dicho proceso. De esta forma al ir almacenando cada término con su significado asociado, tenemos el texto mínimo a ser considerado más información adicional acerca de dichos términos.

Es necesario contar tanto con el término como con su significado, debido a que ambos cumplen una función en el proceso de la recuperación. El significado o token se utiliza en dicho proceso guiando la ejecución de las reglas heurísticas desarrolladas, y los términos comprenden el contenido de las categorías de palabras a buscar. Es decir que una vez reconocido el token que corresponde a una categoría buscada, el término será el dato que la represente.

Vamos almacenando cada término considerado útil con su significado en el mismo orden en que estos son extraídos del documento CFP.

A continuación se retorna al paso ❷ .

2.1.2.- Recuperación De Información

En un proceso de recuperación de información se puede utilizar *métodos estadísticos* o *pattern matching*.

Los sistemas IR basados en *métodos estadísticos*, no utilizan conocimiento acerca de la colección de documentos y necesitan una muestra suficientemente grande para aplicarlos. Se reconocen términos sin significado asociado. Los sistemas IR basados en *pattern matching*, reconocen términos por *matching* exacto es decir, reconocen si el término se encuentra o no en un documento dentro de un determinado contexto.

El sistema de recuperación de información desarrollado analiza documentos CFPs. Como ya hemos visto estos documentos tienen una estructura general donde se pueden encontrar ciertas categorías de términos. Por lo tanto el sistema puede reconocer términos con un significado asociado. Por otro lado el sistema trabaja con una muestra incremental de documentos, la cual va creciendo en el tiempo. Esta muestra a priori no es representativa de algún área particular de la informática y además, el objetivo del sistema no es enfocar un dominio en particular.

Por estas características ya mencionadas, se utiliza *reconocimiento basado en contexto* y *pattern matching* para reconocer si ciertas categorías de términos se encuentran en un documento CFP.

El proceso de recuperación de información desarrollado para este sistema, utiliza un conjunto de reglas heurísticas con el fin de recuperar las categorías de términos que conforman la representación de documentos.

En el paso de preparación del texto se reconoce un conjunto de tokens. En este paso se reconoce un nuevo conjunto que comprende :

- **SIGLA** : Token que representa a la sigla de la conferencia.

- **FECHA**: Token que representa a la fecha de la conferencia.

- **CIUDAD** : Token que representa a la ciudad donde se desarrolla la conferencia. Este token también puede haber sido recuperado en el paso de preparación del texto. En este paso se recuperan ciudades formadas por más de un término que se encuentran en el vocabulario predefinido y ciudades que no se encuentran contempladas en dicho vocabulario .

- **PAÍS** : Token que representa al país donde se desarrolla la conferencia. Este token también puede haber sido recuperado en el paso de preparación del texto. En este paso se recuperan países formados por más de un término que se encuentran en el vocabulario predefinido.

- **ESTADO** : Token que representa un estado de Norteamérica. Este token también puede haber sido recuperado en el paso de preparación del texto. En este paso se recuperan los estados formados por más de un término que se encuentran en el vocabulario predefinido.

Algunos de estos tokens representan las categorías de términos buscadas. Se recuperan a partir de las reglas heurísticas, y éstas también permiten recuperar el resto de las categorías de términos que representan al documento, tales como ***título*** y ***tema***.

Clasificación de Tokens

Ya hemos presentado el conjunto completo de tokens que pueden ser reconocidos por el sistema. Podemos clasificar dicho conjunto según diferentes puntos de vista :

☑ *Cantidad de términos a los cuales se asocia el token*

① Tokens que se asocian a un sólo término : *día, año, mes, ordinal, país, ciudad, estado, palabra clave, comilla, coma, paréntesis derecho, paréntesis izquierdo, fin de línea, continuación de título, comienzo de tema, comienzo de ciudad, final de título, especial, url, otro .*

② Tokens que se asocian a más de un término (frase) : *fecha, sigla, país, ciudad, fin de título,conferencia asociada, comite de programa,CFP .*

Los tokens que se encuentran en ambas clasificaciones son aquellos en que se pueden dar ambos casos, ser simples o estar compuestos por más de un término.

☑ *Finalidad de los tokens*

① Tokens que corresponden a las categorías de términos buscadas : *sigla,fecha, ciudad, país, URL.*

② Tokens que ayudan en la recuperación de las categorías de términos buscadas : *día, año, mes, ordinal, estado, palabra clave, comilla, coma, paréntesis derecho, paréntesis izquierdo, fin de línea, continuación de título,comienzo de tema, comienzo de ciudad, final de título, conferencia asociada, CFP,especial, otro.*

☑ *Asociación de tokens con reglas*

① Tokens que están asociados con reglas de recuperación, es decir que provocan la ejecución de un conjunto de reglas : *comilla, día, mes, año, ordinal, ciudad, país, estado,CFP, conferencia asociada, comite de programa, otro.* Este asociación de tokens con las reglas puede variar (detallado en la presentación del *conjunto de reglas*).

② Tokens que no están asociados a reglas de recuperación: *sigla, fecha, URL, coma, paréntesis derecho, paréntesis izquierdo, fin de línea, continuación de título, comienzo de tema, comienzo de ciudad, final de título, palabra clave, especial .*

Conjunto de Reglas

Para realizar este proceso de recuperación, definimos un conjunto de reglas que se pueden aplicar en la recuperación. En éste se detalla para cada token el conjunto de reglas asociadas a dicho token, es decir el conjunto de reglas que pueden ejecutarse cuando se está analizando un término asociado con este token.

Este conjunto tiene la característica de *no estar incluido* en el sistema de recuperación de información, sino que se encuentra en un archivo externo. De esta manera dicho conjunto puede ser modificado con el fin de alterar la recuperación que se obtiene a partir del mismo. Por ejemplo, si se desea analizar una estructura de documentos mas fija, es decir que se cuenta con más información sobre la estructura del documento, se puede reducir el conjunto de reglas utilizando sólo las necesarias para obtener una recuperación que se adecue a ésta estructura. Otro caso es aquel en que si se desea recuperar sólo algunas de las categorías de términos, se pueden deshabilitar del conjunto aquellas reglas que recuperan las categorías de términos no deseadas.

Como conclusión, al ser el conjunto de reglas modificables externamente y como la salida del sistema (las categorías de términos recuperadas) depende de la ejecución de estas reglas, entonces se puede modificar dicha salida manipulando este conjunto.

Si un CFP es analizado por este sistema, podemos obtener diferentes resultados dependiendo de la selección de las reglas habilitadas y su asociación con los tokens. Se puede seleccionar un subconjunto de reglas a aplicar y a su vez se puede cambiar el conjunto de reglas asociadas a un token determinado.

De esta forma el sistema permite determinar la combinación de reglas que sea más adecuada para alcanzar el objetivo de recuperación deseado.

Los elementos componentes y el formato de este conjunto se presentan más adelante.

Factores que determinan la ejecución de las reglas

Una regla tiene como objetivo recuperar una categoría de término.

Una regla está asociada a uno o más tokens. Esto quiere decir que está asociada a determinados significados.

La ejecución de la regla depende del significado (token) del término analizado. El token asociado al término permite tener una idea sobre el contexto que se está analizando. Definimos contexto al conjunto de tokens que se encuentra no muy distante del token analizado, es decir en un rango de pocas palabras. Por lo tanto, si en dicho contexto existe la posibilidad de la ocurrencia de alguna de las categorías de términos buscadas, se ejecuta la regla que intenta recuperar dicha categoría. Ciertas categorías se encuentran en los CFPs con diferentes formatos por lo tanto más de un token nos permite presuponer la existencia de dicha categoría. En este caso la regla que recupera esta categoría va a estar asociada a más de un token.

La ejecución de una regla también depende de un estado particular del sistema. Es decir que una regla se ejecuta sólo si el sistema se encuentra en un *estado adecuado* para dicha ejecución. En general los estados tienen que ver con la recuperación de las categorías de términos, es decir cuales categorías ya han sido recuperadas y cuales aún no.

Un estado se considera un *estado adecuado* para una regla si se han recuperado ciertas categorías de términos y aún no se ha recuperado la categoría buscada por dicha regla. Por ejemplo la regla que recupera la sigla de la conferencia se ejecuta sólo si aún no se ha recuperado dicha categoría (corresponde al estado **precondAbbrev** presentado a continuación). En otros casos la recuperación de una categoría depende de otras, por ejemplo si se ha recuperado el país de la conferencia y aún no se ha recuperado la ciudad (corresponde al estado **precondCountryCity** presentado a continuación), entonces se ejecuta la regla que recupera la ciudad debido a que es probable que la ciudad se encuentre junto al país.

En conclusión la ejecución de las reglas depende del estado actual del sistema y del significado(token) del término analizado.

Luego de haber presentado detalles acerca de las reglas y los factores que influyen en su ejecución podemos conocer el conjunto de estados del sistema y reglas contempladas en el conjunto asociado al sistema de recuperación de información de documentos CFPs :

Conjunto de Reglas

Tipo de Token	Conjunto de Reglas asociado	
	Estado Adecuado	**Regla**
Comilla	PrecondAbbrev	FindAbbrev
	PrecondAbbrevTitle	FindTitle
Día	PrecondDate	FindDate
	PrecondDateTitle	FindTitle
Mes	PrecondDate	FindDate
	PrecondDateTitle	FindTitle
Año	PrecondAbbrev	FindAbbrev
	PrecondAbbrevTitle	FindTitle
Ordinal	PrecondDate	FindDate
	PrecondDateTitle	FindTitle
	PrecondAllTitle	FindTitle
Ciudad	PrecondCity	RegistCity
	PrecondCityTitle	FindTitle
Estado	PrecondState	RegistState
	PrecondStateCity	FindCity
	PrecondStateTitle	FindTitle

País	PrecondCountry	RegistCountry
	PrecondCountryCity	FindCity
	PrecondCountryTitle	FindTitle
CFP	PrecondNot ProgComm	ProgComm
Conferencia Asociada	PrecondInConjW	InConjW
	PrecondTitle	FindTitle
URL	PrecondTitle	FindTitle
Comite de Programa	PrecondProgComm	progComm
Otro	PrecondCity	FindCity_dt
	PrecondCityTitle	FindTitle
	PrecondCountry	FindCountry_dt
	PrecondCountryCity	FindCity
	PrecondCountryTitle	FindTitle
	PrecondAbbrev	FindAbbrev
	PrecondAbbrevTitle	FindTitle
	PrecondState	FindState_dt
	PrecondStateCity	FindCity
	PrecondStateTitle	FindTitle

Estados del Sistema y Reglas que intervienen en el Conjunto de Reglas

* ESTADOS DEL SISTEMA :

♠ **PrecondCity :** El estado adecuado en este caso es que no se haya recuperado la ciudad, no se haya recuperado el país y no se esté analizando la lista del comité de programa. Este es el estado adecuado para intentar recuperar una ciudad. Se tiene en cuenta además que no se haya encontrado el país, debido a que si ya se recuperó el país, debería haber recuperado la ciudad junto al él. Es conveniente que este estado sea evaluado cuando se está analizando los tokens del tipo : *Ciudad* y *Otro* .

♠ **PrecondState** : El estado adecuado en este caso es que no se haya recuperado un estado Norteamericano, la ciudad, el país y no se esté analizando la lista del comité de programa. Este es el estado adecuado para intentar recuperar un estado Norteamericano (la recuperación del mismo facilita la recuperación de la ciudad y el país). Se tiene en cuenta que no se haya recuperado aún la ciudad y el país, debido a que si se recuperaron estas categorías no es necesario considerar este tipo de token, ya que sólo se utiliza como una ayuda en la recuperación de las dos categorías mencionadas. Es conveniente que este estado sea evaluado cuando se está analizando los tokens del tipo : *Estado* y *Otro* .

♠ **PrecondAbbrev :** El estado adecuado en este caso es que no se haya recuperado la sigla. Este es el estado adecuado para intentar recuperar la sigla de la conferencia. Es conveniente que este estado sea evaluado cuando se está analizando los tokens del tipo : *Comilla, Año* y *Otro*.

♠ **PrecondCountry** : El estado adecuado en este caso es que no se haya recuperado el país y no se esté analizando la lista del comité de programa. Este es el estado adecuado para intentar recuperar el país. Es conveniente que este estado sea evaluado cuando se está analizando los tokens del tipo : *País y Otro.*

♠ **PrecondDate** : El estado adecuado en este caso es que no se haya recuperado la fecha. Este es el estado adecuado para intentar recuperar la fecha de la conferencia. Es conveniente que este estado sea evaluado cuando se está analizando los tokens del tipo : *Día, Mes y Ordinal.*

♠ **PrecondTitle** : El estado adecuado en este caso es que no se haya recuperado el título. Este es el estado adecuado para intentar recuperar el título. Es conveniente que este estado sea evaluado cuando se está analizando el token : *Conferencia asociada.*

♠ **PrecondAbbrevTitle** : El estado adecuado en este caso es que se haya recuperado la sigla y no se haya recuperado el título. Este es el estado adecuado para intentar recuperar el título, debido a que en general, éste puede encontrarse junto a la sigla. Es conveniente que este estado sea evaluado cuando se está analizando los tokens que intentan recuperar la sigla : *Comilla, Año y Otro.*

♠ **PrecondDateTitle** : El estado adecuado en este caso es que se haya recuperado la fecha y no se haya recuperado el título. Este es el estado adecuado para intentar recuperar el título, debido a que en general, éste puede encontrarse junto a la fecha. Es conveniente que este estado sea evaluado cuando se está analizando los tokens que intentan recuperar la fecha : *Día, Mes y Ordinal.*

♠ **PrecondCityTitle** : El estado adecuado en este caso es que se haya recuperado la ciudad y no se haya recuperado el título. Este es el estado adecuado para intentar recuperar el título, debido a que en general, éste puede encontrarse junto a la ciudad. Es conveniente que este estado sea evaluado cuando se está analizando los tokens que intentan recuperar la ciudad : *Ciudad y Otro.*

♠ **PrecondCountryTitle** : El estado adecuado en este caso es que se haya recuperado el país y no se haya recuperado el título. Este es el estado adecuado para intentar recuperar el título, debido a que en general, éste puede encontrarse junto al país. Es conveniente que este estado sea evaluado cuando se está analizando los tokens que intentan recuperar el país : *País y Otro.*

♠ **PrecondCountryCity** : El estado adecuado en este caso es que se haya recuperado el país y no se haya recuperado la ciudad. Este es el estado adecuado para intentar recuperar la ciudad, debido a que en general, ésta puede encontrarse junto al país; será el caso en que la ciudad no pertenece al vocabulario predefinido. Es conveniente que este estado sea evaluado cuando se está analizando los tokens que intentan recuperar el país : *País y Otro.*

♠ **PrecondAllTitle** : El estado adecuado en este caso es que se hayan recuperado todas las categorías excepto el título. Se intenta recuperar el título a partir de ciertas palabras que indican un posible comienzo del mismo. Es conveniente que este estado sea evaluado cuando se está analizando el token : *Ordinal.*

♠ **PrecondInConjW** : El estado adecuado en estos casos es que se esté analizando un token que implica posible conferencia asociada. Este es el estado adecuado para intentar recuperar el título, debido a que en general, éste puede encontrarse junto a la conferencia asociada. Es conveniente que este estado sea evaluado cuando se está analizando el token : *Conferencia asociada* .

♠ **PrecondProgComm** : El estado adecuado en estos casos es que el sistema no se encuentre analizando una lista de comité de programa. Es conveniente que este estado sea evaluado cuando se está analizando el token : *Comite de Programa.*

♠ **PrecondNotProgComm** : El estado adecuado en estos casos es que el sistema se encuentre analizando una lista de comité de programa. Es conveniente que este estado sea evaluado cuando se está analizando el token : *CFP.*

♠ **PrecondStateCity** : El estado adecuado en este caso es que se haya recuperado un estado Norteamericano y no se haya recuperado la ciudad. Este es el estado adecuado para intentar recuperar la ciudad, debido a que en general, ésta puede encontrarse junto al estado. Es conveniente que este estado del sistema sea evaluado cuando se está analizando los tokens : *Estado y Otro.*

♠ **PrecondStateTitle** : El estado adecuado en este caso es que se haya recuperado un estado Norteamericano y no se haya recuperado el título de la conferencia. Este es el estado adecuado para intentar recuperar el título, debido a que en general, éste puede encontrarse junto al estado. Es conveniente que este estado de sistema sea evaluado cuando se está analizando los tokens : *Estado y Otro.*

* **REGLAS** :

♠ **FindSigla** : Intenta recuperar la sigla, considerando distintos formatos. En términos generales esta regla recupera una sigla si encuentra una comilla seguida por un año y precedida por una palabra o letras en mayúsculas. También se consideran los casos en que no exista la comilla o que la sigla junto con el año forma una sola palabra. Esta regla recupera tanto la sigla de la conferencia como la sigla de una conferencia asociada si es que existiera.

♠ **FindDate** : Intenta recuperar la fecha. En términos generales esta regla recupera una fecha simple o un rango de fechas considerando distintos formatos. Al recuperar la fecha controlamos que concuerde con el año asociado a la sigla debido a que podemos haber recuperado una sigla inválida como por ejemplo una sigla correspondiente a una conferencia anterior.

♠ **FindTitle** : Intenta recuperar el título. En caso de recuperarlo intenta identificar el tema a partir del mismo. En términos generales esta regla intenta recuperar el título de la conferencia en dos casos:

• El *primer caso* es una vez que se ha recuperado alguna otra categoría o si se ha encontrado alguna conferencia asociada. En este caso el título se busca como un texto que precede a la categoría recuperada hasta encontrar alguna otra categoría, el comienzo del CFP, algún token como *ordinal, conferencia asociada*,etc . Analizando un conjunto de CFPs observamos que el título de una conferencia suele encontrarse en un contexto delimitado por ciertas categorías tales como : *ciudad, país, estado, fecha, sigla* y *conferencia asociada,* por lo tanto si se recupera alguna de estas categorías, analizando el contexto intentamos determinar si el título se encuentra junto a alguna de ellas.

• El *segundo caso* es una vez que se han recuperado todas las categorías, donde el título se busca en el texto que se encuentra a continuación. Esta búsqueda se realiza a partir del token del tipo : *Ordinal*, debido a que este indica un posible comienzo de título. La idea es que en aquellos casos en que no se pudo recuperar el título antes de recuperar el resto de las categorías, se intenta recuperarlo del resto del texto, ya que observamos que el título suele aparecer nuevamente en el comienzo del desarrollo del CFP.

Para la recuperación de esta categoría consideramos un conjunto de palabras claves que en general se encuentran en un título y otras palabras claves que nos permiten identificar la continuación o el final del mismo. Existen ciertas palabras tales como : *on, in, the* etc que indican que el texto continua, por lo tanto si el texto que forma el título se encuentra dividido en frases, estas palabras actuan como nexos entre estas frases, obteniendo el título completo. Además existe un conjunto de frases tales como : *will be held, is to be held, sponsored by,* etc que nos permiten determinar el final del título.

Una vez recuperado el título determinamos si es el título de la conferencia o de una conferencia asociada analizando el contexto en el que se encuentra.

♠ **RegistCity** : Obtiene la ciudad a partir del término asociado al token de esta categoría que ha sido recuperada en el paso de identificación de términos. Se considera como ciudad sólo si comienza con mayúscula ya que en caso contrario puede ocurrir que ésta no corresponda a la conferencia sino que, por ejemplo forme parte de un URL o una dirección de *e-mail.* Además se verifica que este término no corresponda al nombre de una universidad (esto se conoce analizando el contexto), en cuyo caso tampoco se lo considera como ciudad. Si el término es recuperado como ciudad, se intenta recuperar el país correspondiente consultando en una BDs que contiene información de ciudades y países.

♠ **FindCity** : Intenta recuperar una ciudad que no está incluida en el vocabulario predefinido de ciudades con el que se cuenta. Esta regla se basa en que en la mayoría de los casos la ciudad se encuentra junto al país, con lo cual si se ha recuperado el país se intenta recuperar la ciudad junto a él, de esta forma podemos recuperar ciudades que no se encuentran en el diccionario de ciudades. Se verifica que el nombre de la posible ciudad no corresponda al nombre de una universidad (esto se conoce analizando el contexto), en cuyo caso no se lo considera como ciudad.

♠ **FindCity_Dt** : Intenta recuperar una ciudad compuesta por más de un término a partir del vocabulario predefinido de ciudades con el que se cuenta. Agrupa varios términos y los busca en el diccionario para ver si corresponden a una ciudad. Se verifica que el nombre de la posible ciudad no corresponda al nombre de una universidad (esto se conoce analizando el contexto), en cuyo caso no se lo considera como ciudad.

♠ **RegistState**:Obtiene un estado Norteamericano a partir del término asociado al token de esta categoría que ha sido recuperada en el paso de identificación de términos. Se verifica que dicho término no corresponda al nombre de una universidad (esto se conoce analizando el contexto), en cuyo caso no se lo considera como estado. Además (también analizando el contexto) se verifica si se encuentra incluido en el título de la conferencia, si es así tampoco se lo considera como estado ya que este token tiene por función facilitar la recuperación del país y la ciudad, con lo cual si forma parte del título no tiene sentido intentar recuperar estas categorías. En caso que el término corresponda a un estado correcto, se registra como país de la conferencia EE.UU.

♠ **FindState_Dt** : Intenta recuperar un estado Norteamericano que está compuesto por más de un término a partir del vocabulario predefinido de estados con el que se cuenta. Agrupa varios términos y los busca en el diccionario para ver si corresponden a un estado. Se verifica que dicho término no corresponda al nombre de una universidad (esto se conoce analizando el contexto), en cuyo caso no se lo considera como estado.Además (también analizando el contexto) se verifica si se encuentra incluido en el título de la conferencia, si es así tampoco se lo considera como estado ya que este token tiene por función facilitar la recuperación del país y la ciudad, con lo cual si forma parte del título no tiene sentido intentar recuperar estas categorías.

♠ **FindCountry_Dt** : Intenta recuperar un país que está compuesto por más de un término a partir del vocabulario predefinido de países con el que se cuenta. Agrupa varios términos y los busca en el diccionario para ver si corresponden a un país.

♠ **RegistCountry** : Obtiene el país a partir del término asociado al token de esta categoría que ha sido recuperada en el paso de identificación de términos.

♠ **InConjW** : Indica la existencia de una posible conferencia asociada.

♠ **ProgCom** : Dependiendo del tipo de token que la dispare, esta regla registra el estado en el cual el sistema se encuentra analizando una lista correspondiente al comité de programa o indica que este estado ha concluido.

Ejecución de las Reglas

El proceso de preparación de texto elimina la información irrelevante del texto, asocia un significado a cada término, y almacena tanto el término como su significado (token) en el mismo orden en el que se fueron recuperando a partir del texto.

Una vez que este proceso almacenó una cantidad mínima de términos y tokens necesaria para la ejecución de las reglas, comienza el proceso de ejecución de las mismas. Es necesario contar con una cantidad mínima de tokens y términos, debido a que las reglas analizan no sólo el significado de los términos sino también el contexto en el que se encuentran. Esta cantidad mínima representa el número de términos que necesita la regla de máximo análisis de contexto.

El proceso de recuperación de información cuenta con dos elementos básicos. Por un lado, la representación del texto a analizar formada por pares (término, token) y por otro lado con un conjunto de reglas a aplicar.

Este conjunto de reglas tiene el siguiente formato :

Token $_1$: (estado $_1$, regla $_1$) .....(estado$_k$, regla $_k$)
.....
.....
Token $_n$: (estado $_1$, regla $_1$) .....(estado $_p$, regla $_p$)

Los tokens$_{1..n}$, corresponden a la clasificación de tokens que provocan la ejecución de reglas por lo tanto tienen asociado un subconjunto de reglas.

Cada regla está compuesta por el par (estado $_i$, regla $_i$), donde estado $_i$ representa el estado del sistema adecuado para la ejecución de dicha regla. Estos subconjuntos de reglas no son disjuntos, es decir que una regla puede estar asociada a mas de un token.

El proceso de ejecución de las reglas va provocando cambios en el estado del sistema. Estos estados tienen que ver con el grado de recuperación obtenido hasta el momento *en el contexto del análisis de un documento CFP.*

El *estado inicial* corresponde al estado del sistema cuando va a comenzar el proceso de recuperación de información de un documento CFP y el *estado final* corresponde al estado cuando se terminó de analizar dicho documento.

Podemos comprender el proceso de ejecución de reglas a través de una *representación funcional.* Esta representación se compone de un conjunto de funciones donde se detalla el tipo y la definición de cada una de ellas. A medida que se produce la ejecución de dichas funciones el sistema va cambiando de estado, pudiendo en ciertos casos mantener el mismo estado, hasta llegar al estado final.

Representación Funcional del Proceso de Ejecución de Reglas

Funciones: ***Tipos***

RecuperarInformacion ::
([Token] , [(Token , [(EstadoAdecuado , Regla)])] , Estado) → Estado

Objetivo : Recuperar información a partir de un texto.
[Token] : Lista que representa los tokens asociados a los términos del texto analizar.
[(EstadoAdecuado , Regla)] : Lista que representa el conjunto de reglas a aplicar .
(EstadoAdecuado , Regla) : Representa una regla formada por un estado de sistema adecuado para su aplicación y la regla misma.

AsociarTokenConjuntoReglas ::
(Token , [(Token , [(EstadoAdecuado , Regla)])], Estado) → Estado

Objetivo : Dado un token , determinar el subconjunto de reglas que el mismo tiene asociado.

AplicarConjuntoReglas :: ([(EstadoAdecuado , Regla)] , Estado) → Estado

Objetivo : Dado un subconjunto de reglas, aplicar las mismas dependiendo del estado actual del sistema .

AplicarRegla :: (Regla , Estado) → Estado

Objetivo : Dada una regla y un estado , aplicar dicha regla .

Funciones: ***Definición***

RecuperarInformacion ([T] , L_S , E) = AsociarTokenConjuntoReglas (T, L_S , E)
RecuperarInformacion ((T : T_S) , L_S , E) =
RecuperarInformacion (T_S , L_S, AsociarTokenConjuntoReglas(T, L_S, E))

Donde : T representa un token
Ts representa una lista de tokens
[T] representa una lista formada por un solo token
(T : Ts) representa una lista formada por al menos un token
Ls representa el conjunto de reglas asociado al sistema
E representa un estado del sistema

AsociarTokenConjuntoReglas (T , [ ] , E) = E
AsociarTokenConjuntoReglas(T , ((T_i ,R_S) : L_S), E) =
AplicarConjuntoReglas (R_S, E) ⇔ T_i = T
AsociarTokenConjuntoReglas(T , ((T_i ,R_S) : L_S), E) =
AsociarTokenConjuntoReglas(T , L_S , E) ⇔ T_i <>T

Donde : [ ] representa una lista vacía
Rs representa el conjunto de reglas asociadas al token Ti

AplicarConjuntoReglas ([(E_A , R)] , E) = E $\Leftrightarrow E_A <> E$
AplicarConjuntoReglas ([(E_A , R)] , E) = AplicarRegla (R, E) $\Leftrightarrow E_A = E$
AplicarConjuntoReglas(((E_A ,R) : R_S), E) =AplicarConjuntoReglas (R_S , E) $\Leftrightarrow E_A <> E$
AplicarConjuntoReglas (((E_A ,R) : R_S), E) =
AplicarConjuntoReglas (R_S , AplicarRegla (R, E)) $\Leftrightarrow E_A = E$

Donde : R representa una regla
E_A representa el estado adecuado asociado a la regla R

AplicarRegla (R, E_1) = E_2 Pudiendo ser E_1=E_2 en el caso en que la regla no tuvo éxito en la recuperación de la categoría buscada .

Donde : E_1, E_2 representan estados del sistema

Esta representación funcional del motor de ejecución de reglas define el comportamiento del proceso de recuperación de información, cómo va analizando el texto y recuperando las categorías de términos deseadas.

En términos generales dada la lista de tokens que representa el texto de un documento CFP, se va analizando uno a uno todos los tokens que conforman el texto. Si el token tiene asociado un conjunto de reglas, se disparan aquellas reglas donde el estado del sistema coincida con el estado adecuado asociado a dicha regla. En este caso puede producirse un cambio en el estado del sistema o no, dependiendo del éxito obtenido en la recuperación.

Una vez finalizado el proceso de recuperación de información del documento CFP se continúa con la *Fase de Incorporación de Datos a una BDs.*

3.- FASE DE INCORPORACIÓN DE DATOS A UNA BDS

En esta fase evaluamos los resultados obtenidos del proceso de recuperación de información de un documento CFP.

Como ya hemos mencionado (Pág 94), consideramos como **condición de éxito** que dicho proceso haya recuperado al menos una de las dos categorías : *Título* o *Sigla* de la conferencia. Si se recupera alguna de estas categorías consideramos que se cuenta con información suficiente que permite reconocer una conferencia determinada.

En el caso en que se ha recuperado el *Título* realizamos un proceso adicional. Este proceso implica la utilización de otra técnica de recuperación de información (IR) llamada ***Stemming*** [BaezaYates-Frakes92]. Como hemos visto en el *Capítulo 3* (Pág 34), esta técnica permite obtener las raíces de las palabras con lo cual se pueden reconocer términos de una misma familia, es decir me permite relacionar términos similares morfológicamente.

El tipo de algoritmo de *stemming* que utilizamos es "Affix Removal", el cual remueve sufijos de los términos obteniendo así la raíz de las palabras. En particular utilizamos las reglas definidas por Porter [BaezaYates-Frakes92], las cuales fueron detalladas en el *Capítulo 3* (Pág 37). Este tipo de algoritmo de *stemming* es el más común aunque requiere participación humana debido a que se deben preparar las listas de afijos y las reglas para poder removerlos de una palabra. Al contar con las reglas definidas por Porter, nos pareció adecuado utilizar este método, obteniendo un buen resultado al aplicarlo. En el caso del método "Succesor Variety", la obtención del *stem* depende del texto en el cual se encuentra el término, mientras que en el caso del método "Affix Removal", el *stem* se obtiene en forma independiente. El método "N-Gram" no obtiene el *stem* sino que combina términos según el número de digramas o n-gramas que ellos comparten y se necesitan estructuras adicionales para contener esta información. El método "Lookup" tiene la desventaja que la tabla de los *stems* puede ser muy grande y además para el idioma Inglés no podemos asegurarnos de contar con todos los términos posibles y sus correspondientes *stems*.

La técnica de *stemming* no la aplicamos a las categorías de términos tales como ciudad, país, fecha o sigla, debido a que en estos casos la consulta es en forma exacta. Se aplica al título porque en general éste representa la temática de la conferencia, en cuyo caso se obtiene el tema de la misma. Es justamente en el caso de consultas sobre temas y/o títulos donde nos va a interesar abarcar la mayor variedad de conferencias relacionadas.

Aplicamos esta técnica de IR tanto en este proceso de indexación como en el procesamiento de *queries* (detallado en el *Capítulo 6* (Pág 139)), por lo tanto cuando se requiere información acerca de un tema ó título la consulta no se realiza mediante un *matching* exacto, sino mediante un *matching* de variantes morfológicas del término de la consulta. Por lo tanto se puede abarcar un mayor conjunto de conferencias al relacionar morfológicamente términos índices con términos de búsqueda. Aplicamos esta técnica a cada término componente del título e incorporamos a la BDs tanto el título completo como los *stems* de sus términos componentes.

El conjunto de categorías que pueden ser recuperadas a partir de un documento CFP comprenden :

- Título
- Sigla
- Tema
- Fecha de inicio de la conferencia
- Fecha de finalización de la conferencia
- Ciudad
- País
- Conjunto de URLs que se encuentran referenciadas en el documento
- Sigla y/o Título de una conferencia asociada

En el caso en que se ha recuperado el título y no se ha recuperado la sigla realizamos una operación especial. Esta consiste en obtener a partir del título una *sigla probable* de la conferencia. En el general de los casos la sigla de una conferencia se forma como una abreviatura del título de la misma junto con el año de realización. Por lo tanto se considera como sigla probable a la abreviatura obtenida a partir de la categoría título recuperada. Esta sigla se considera *probable* debido a que no es un dato recuperado a partir del documento, sino que es *generado por el sistema* a partir de un dato recuperado. Realizamos esta operación porque el identificador de la conferencia es la sigla de la misma, por lo tanto en caso en que no se la haya recuperado, la genera el sistema de manera tal que toda conferencia pueda ser identificada.

Una vez que hemos recuperado los datos, obtenemos un formato de los mismos necesario para su almacenamiento y posterior consulta, y se interactua con un manejador de BDs con el fin de incorporarlos a una BDs.

Pueden darse diferentes situaciones en el momento de incorporar los datos a la BDs :

1) Es la primera vez que se incorpora el CFP a la BDs : en este caso simplemente se incorporan los datos y se le asocia un dígito identificador a la sigla obtenida , de manera tal de distinguirla de otras conferencias con la misma sigla. En este caso le asociamos el dígito "0".

2) Ya se ha recibido una versión anterior del mismo CFP : en este caso se actualizan los datos almacenados en la BDs con los datos de la nueva versión. En el caso de los URLs sólo se incorporan aquellos que no se encontraban en la versión anterior.

3) Existe un CFP distinto que tiene la misma sigla : en este caso se agregan los datos correspondientes a la nueva conferencia con la sigla recuperada y un nuevo digito para diferenciarla del resto de las conferencias que tienen la misma sigla.

Una vez que se han incorporado los datos debemos actualizar la información referente a las conferencias asociadas, debido a que esta conferencia puede estar asociada con alguna que ya se encuentre almacenada en la BDs, en cuyo caso debemos actualizar los datos referentes a esta conferencia de manera tal que si es consultada podamos acceder también a la información de la conferencia asociada que será la que se está incorporando en este momento.

Podríamos haber obtenido la referencia a la conferencia asociada en el momento de la consulta, sin embargo esto provocaría una disminución en la *performance* aumentando el tiempo de respuesta, para evitar ésto, decidimos realizar esta actualización de información (aunque sea mas costosa) en el Sistema de Recuperación de Información de Documentos CFPs, debido a que en este caso no es tan crítico el tiempo de procesamiento.

Podemos representar esta fase con el siguiente esquema :

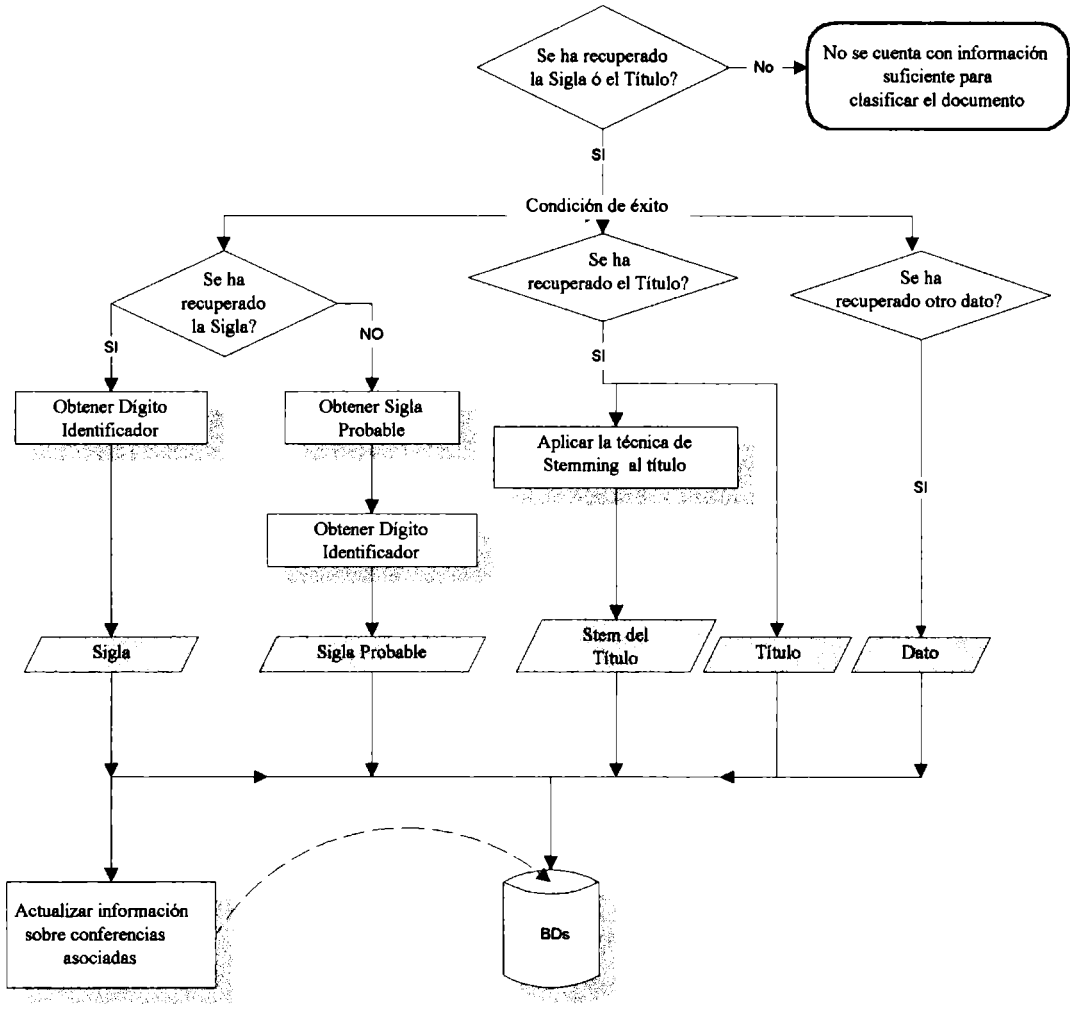


Figura 5.4 *- Evaluación de la Recuperación de Información e Incorporación de Datos a una BDs -*

ARQUITECTURA DEL SISTEMA IR DE DOCUMENTOS CFPS

Presentación de los módulos componentes del Sistema de Recuperación de Información de Documentos CFPs y su comunicación

El Sistema de Recuperación de Información de Documentos CFPs fue desarrollado en el lenguaje C para correr bajo el sistema operativo UNIX (Linux).

En esta sección presentamos una representación del sistema basada en un conjunto de módulos y de la comunicación entre ellos. Estos módulos pueden verse reflejados en la *Figura 5.5* .

● Módulo de Inicialización

Este módulo cumple las siguientes funciones :

① Carga el conjunto de reglas en una estructura con el formato presentado en la definición conceptual del sistema (pág. 112).

Debido a que este conjunto de reglas no es parte del sistema sino que es un componente externo y que además puede ser modificado, cada vez que el sistema analiza una colección de documentos CFP este conjunto de reglas debe ser incorporado al mismo, cargándolo a partir del archivo externo que contiene dicho conjunto. Este guiará el proceso de recuperación de información que se va a aplicar a la colección completa de documentos con la que se cuenta al comienzo de ejecución del sistema.

Para mantener el conjunto de reglas en el sistema utilizamos una estructura donde se van a cargar los tokens que ejecutan reglas, junto con sus subconjuntos de reglas asociadas, formadas cada una de ellas por nombres de funciones que representan el estado adecuado para la ejecución y la regla misma.

Este conjunto de reglas será utilizado por el *módulo ejecución de reglas.*

② Construye las estructuras que contienen los vocabularios predefinidos que serán utilizados como diccionarios en el proceso de recuperación de información. Estos vocabularios corresponden a :

.- Nombres de ciudades
.- Nombres de países
.- Nombres de estados
.- Palabras claves para la recuperación del título
.- Palabras *stopwords*
.- Palabras *stopwords* especiales para un título de conferencia

En estos casos, en los cuales el contexto del análisis corresponde a un vocabulario predefinido externo al documento analizado utilizamos en el proceso de reconocimiento, autómatas finitos determinista (DFAs) de mínimo estado.

Estos son generados en forma automática a partir de un vocabulario en este *módulo de inicialización del sistema*. Contamos con un DFA por cada vocabulario predefinido que se utiliza en el reconocimiento. Cada DFA se utiliza para reconocer términos que correspondan al vocabulario asociado, es decir nos permite reconocer términos *stopwords*, países, ciudades, estados y palabras claves.

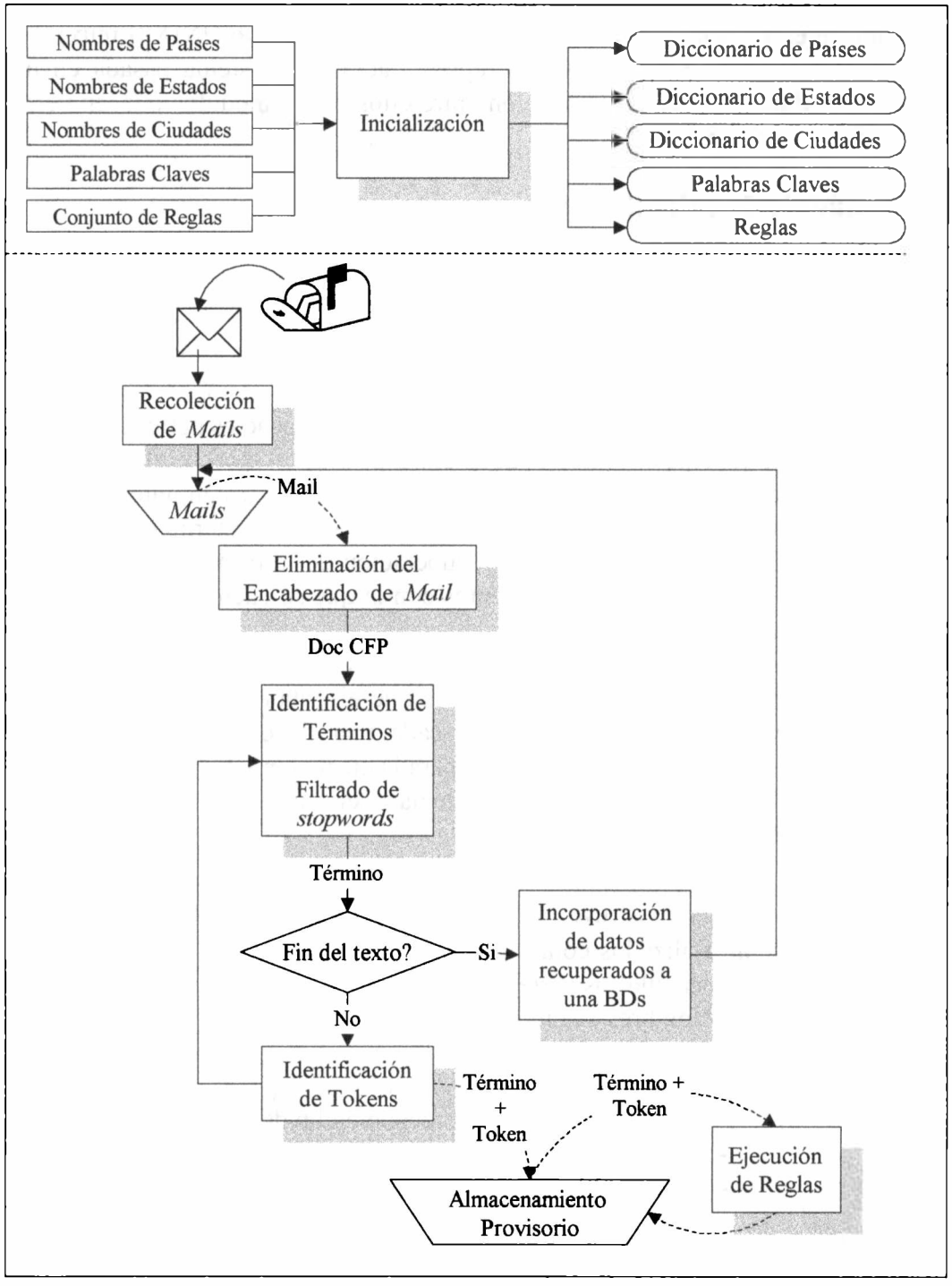


Figura 5.5 *- Módulos del Sistema IR de Documentos CFPs y la interacción entre ellos*

Autómata Finito - Definición -

Las máquinas reconocedoras o autómatas finitos, son dispositivos formales que deciden si una cadena determinada pertenece o no a un lenguaje establecido.

El autómata finito consta de :

- Una cinta de entrada, que solo se mueve en un sentido (unidireccional), dividida en celdas. En esta cinta se encuentra la tira de entrada ocupando una celda cada uno de sus símbolos. Estos símbolos pertenecen al alfabeto de entrada.
- Un control de estados, que determina el comportamiento del mecanismo de reconocimiento representado por un conjunto de estados.

Como norma general de funcionamiento, encontrándose el autómata en un estado y leído un símbolo de la cinta de entrada, la máquina transitará o pasará a otro estado, pudiendo incluso ser el mismo.

Cuando la transición se realiza de un estado a otro exclusivamente, se dice que el control es *determinista*. En caso contrario, cuando puede transitar a dos o más estados diferentes, se dice que el control es *no-determinista*.

Un autómata *acepta* o *reconoce* una cadena o tira de la cinta de entrada si, comenzando en un estado inicial, y después de transitar por estados intermedios, se llega a un estado final.

Un *autómata finito de mínimo estado*, es aquel que tiene la menor cantidad de estados posible.

Generación automática de DFA de mínimo estado

El DFA generado automáticamente para reconocer palabras de un vocabulario comprende un conjunto de estados asociados a un label formado por un conjunto de strings.

En el proceso de generación del autómata, el algoritmo le asigna como label a cada estado el conjunto de strings que la máquina aceptaría, si ese estado fuera el estado inicial. Examinando estos labels de estados, se puede determinar:

a) Las transiciones de salida de cada estado (cantidad de transiciones a partir de dicho estado).
b) El estado objetivo asociado a cada transición de salida.
c) Los estados finales.

Llamamos *label derivado L con transición a*, al label del estado que se alcanza por una transición sobre el símbolo *a*.

Un estado es un *estado final* sí y solo sí su label contiene el string nulo.

El algoritmo para generar un DFA de mínimo estado usando este mecanismo de labels comprende los siguientes pasos :

① Crea un estado inicial al cual se le asigna como label el conjunto de strings de entrada.

② A partir de este label se generan los *labels de estados derivados.* Por cada label de estado derivado con transición "*a*", si el estado derivado no existe, es creado y se crea un arco entre el estado anterior y este nuevo estado, si el estado ya existe, sólo se crea un arco entre ambos estados.Este mismo proceso se le aplica a todas las transiciones para dicho estado y a su vez, a todos los estados generados.

③ Una vez procesados todos los estados que comprende el DFA, todo aquel cuyo label contiene el string nulo (λ) se indica como *estado final.*

El proceso de generación puede verse reflejado en el siguiente ejemplo: Sea el conjunto de entrada el siguiente subconjunto de países representado por {CHILE, CHINA, INDIA, US, USA }.

El DFA generado a partir de esta lista de países, puede verse reflejado en el grafo presentado en la *Figura 5.6* .

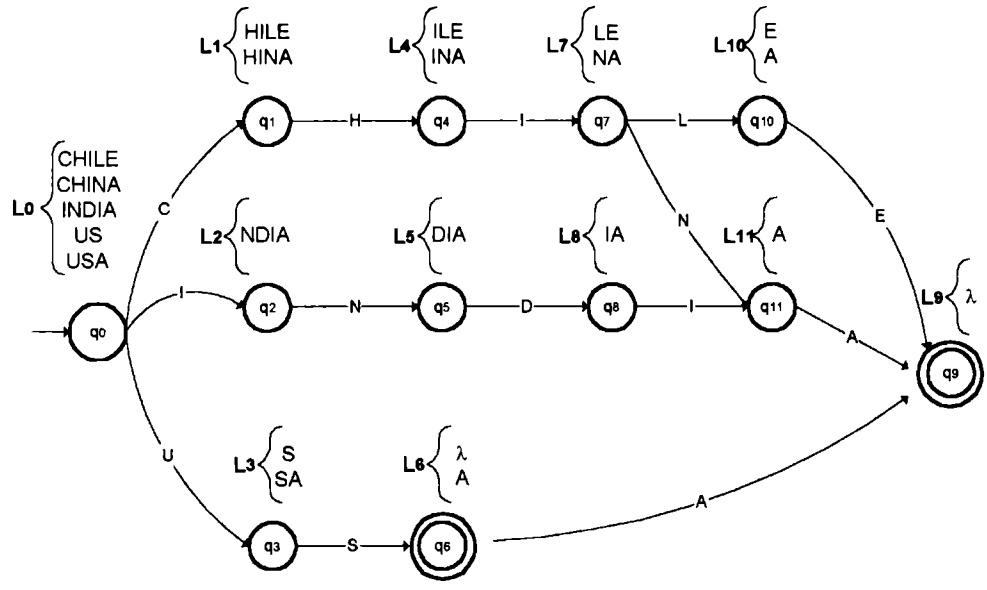


***Figura 5.6** - Ejemplo de un DFA generado a partir de una lista de países*

El proceso de generación de la *Figura 5.6* sigue la siguiente secuencia :

1) Se genera el estado q_0 con label L_0= {CHILE, CHINA, INDIA, US, USA}

2) A partir de L_0, se genera el conjunto de los *labels de estados derivados* , el cual comprende :

L_1 ={ HILE, HINA } Label de estado derivado con transición *C*
L_2 ={ NDIA } Label de estado derivado con transición *I*
L_3 ={ S, SA } Label de estado derivado con transición *U*

3) Como no existen estados con labels L_1, L_2 o L_3, entonces se crean los estados q_1 con label L_1, q_2 con label L_2 y q_3 con label L_3. Se genera un arco con transición *C* entre q_0 y q_1, un arco con transición *I* entre q_0 y q_2 y un arco con transición *U* entre q_0 y q_3.

4) A partir de L_1 se genera el conjunto de los *labels de estados derivados* , el cual comprende : L_4 ={ ILE, INA } Label de estado derivado con transición *H*.

5) Como no existe un estado q_i con label L_4 entonces se crea el estado q_4 con label L_4. Se genera un arco con transición *H* entre q_1 y q_4 .

6) A partir de L_2 se genera el conjunto los *labels de estados derivados*, el cual comprende : L_5 ={ DIA } Label de estado derivado con transición *N*.

7) Como no existe un estado q_i con label L_5 entonces se crea el estado q_5 con label L_5. Se genera un arco con transición *N* entre q_2 y q_5 .

8) A partir de L_3 se genera el conjunto de los *labels de estados derivados*, el cual comprende : L_6 ={ λ , A} Label de estado derivado con transición *S*

9) Como no existe un estado q_i con label L_6 entonces se crea el estado q_6 con label L_6. Se genera un arco con transición *S* entre q_3 y q_6 .

10) De esta misma forma , se van generando los estados q_7, q_8, q_9, q_{10} y q_{11}, asociados con sus labels correspondientes.

11) Se indica como estados finales los estados q_6 y q_9, debido a que ellos incluyen el string nulo {λ}.

Una vez que el DFA ha sido construido todo el espacio utilizado por los labels de estado es liberado, y sólo se conservan las tablas de estados y transiciones.

Reconocimiento de strings utilizando un DFA

Una entrada es *reconocida* por el DFA si partiendo del estado inicial, se llega a algún estado final.

Se dice que un DFA es *parcial*, si la función de transición no está definida para todos los posibles símbolos de los labels de cada estado. En este caso existe para cada transición no definida un *estado de error* (ó "estado muerto")implícito que pertenece al conjunto de estados finales. Cuando el DFA se bloquea en este estado se dice que *no ha sido reconocida* la entrada.

El proceso de reconocimiento puede verse reflejado con los siguientes ejemplos:

❶ Sea el string a reconocer : "**USA**", los pasos de reconocimiento son los siguientes :

➲ La máquina se encuentra en el estado inicial q_0 .

➲ Se toma el primer símbolo de la cadena de entrada a analizar, el símbolo "*U*".

➲ Como existe un estado derivado con transición "*U*" a partir del estado q_0, entonces se transita al nuevo estado q_3.

➲ Se toma el segundo símbolo de la cadena de entrada, símbolo "*S*".

➲ Como existe un estado derivado con transición "*S*" a partir del estado q_3, entonces se transita al nuevo estado q_6.

➲ Se toma el tercer símbolo de la cadena de entrada, símbolo "*A*".

➲ Como existe un estado derivado con transición "*A*" a partir del estado q_6, entonces se transita al nuevo estado q_9 .

➲ Se toma el cuarto símbolo de la cadena de entrada, símbolo que indica fin de la cadena.

➲ Como el estado actual q_9 es un estado final, entonces *se ha reconocido la cadena de entrada* "USA" (✓), ya que partiendo del estado inicial se ha llegado a un estado final.

❷ Sea el string a reconocer : "IRAN", los pasos de reconocimiento son los siguientes :

➲ La máquina se encuentra en el estado inicial q_0.

➲ Se toma el primer símbolo de la cadena de entrada a analizar, el símbolo "*I*".

➲ Como existe un estado derivado con transición "*I*" a partir del estado q_0, entonces se transita al nuevo estado q_2.

➲ Se toma el segundo símbolo de la cadena de entrada, símbolo "*R*".

➲ Como no existe un estado derivado con transición "*R*"a partir de q_2, se transita por defecto al *estado de error* especial, lo que indica que *no ha sido reconocida la cadena de entrada* "IRAN" (✗).En este caso la máquina se bloquea y termina el proceso.

Utilizar DFAs en este proceso de reconocimiento tiene la ventaja que pueden construirse rápidamente, además utilizan poco espacio y son de rápido acceso.

El *módulo de inicialización* recibe como **entrada** los siguiente archivos:

- Conjunto de reglas.
- Nombres de ciudades .

- Nombres de países .
- Nombres de estados Norteamericanos .
- Palabras claves para la recuperación del título .
- Palabras *Stopwords* para ser filtradas en el análisis léxico .
- Palabras *Stopwords* para ser filtradas en el procesamiento del título .

Produce como **salida** las siguientes estructuras, las cuales serán utilizadas por los módulos del sistema :

- Conjunto de reglas : utilizado en el *módulo de ejecución de reglas* .
- DFA representando el diccionario de ciudades : utilizado por el *módulo de identificación de tokens* y por el *módulo de ejecución de reglas.*
- DFA representando el diccionario de países : utilizado por el *módulo de identificación de tokens* y por el *módulo de ejecución de reglas.*
- DFA representando el diccionario de estados Norteamericanos : utilizado por el *módulo de identificación de tokens* y por el *módulo de ejecución de reglas.*
- DFA representando un conjunto de palabras claves : utilizado por el *módulo de ejecución de reglas* .
- DFA representando un conjunto de palabras *stopwords* : utilizado por el *módulo de filtrado de stopwords* (análisis léxico).
- DFA representando un conjunto de palabras *stopwords* especiales para el título : utilizado por el *módulo de incorporación de datos recuperados a la BDs* (procesamiento del título).

● Módulo de Recolección de Documentos CFPs

La recolección de los documentos CFPs de la cuenta de *e-mail* destinada para esto, se realiza mediante un script, cuyas funciones fueron presentadas en la definición conceptual del sistema (Pág. 92) .

Este módulo obtiene como **entrada** de la cuenta de *e-mail* destinada para la recolección de documentos CFPs, un conjunto de *e-mails.*

Produce como **salida** los *e-mails* recibidos, como archivos de texto. Estos a su vez, actuarán uno a uno como entrada al *módulo de eliminación del encabezado del e-mail.*

● Módulo de Eliminación del Encabezado del *e-Mail*

Como ya se especificó en el *Capítulo 4* (Pág. 60), el encabezado del *e-mail* se encuentra separado de su contenido por medio de una línea en blanco, por lo tanto en primer lugar se elimina el texto comprendido hasta esta línea. Pero luego de analizar un conjunto de CFPs, observamos que en la mayoría de los casos, el contenido del mensaje no comienza inmediatamente después de esta línea, sino que existe más información respecto al encabezado, manteniendo el mismo formato.

Por lo tanto, cuando se comienza a analizar el contenido del mensaje, se tiene en cuenta que esta información no corresponda a un encabezado, en cuyo caso también se la elimina y se continua este proceso hasta encontrar información que no corresponda a un posible encabezado.

Este módulo recibe como **entrada** un *e-mail* compuesto de un encabezado y un contenido .

Produce como **salida** el contenido del *e-mail*, el cual representa el documento CFP. Este a su vez, actuará como entrada al *módulo de identificación de términos*.

• Módulo de Identificación de Términos y Filtrado de *Stopwords*

La identificación de términos se realiza mediante un analizador léxico, el cual va leyendo el texto carácter a carácter, hasta encontrar un delimitador. Luego se verifica si el término obtenido es una palabra *stopword*, en cuyo caso se lo ignora y vuelve a obtener a un nuevo término. En caso contrario, este término será la **salida** de estos módulos.

El reconocimiento de palabras *stopwords* se realiza utilizando un DFA, tal como ha sido descripto en el *módulo de inicialización*, el cual fue generado a partir del conjunto de palabras *stopwords*. En caso en que se identifica el fin del texto, retorna el carácter nulo y continuará el *módulo de incorporación de datos recuperados a la BDs*.

Como **entrada** el *módulo de identificación de términos* recibe un documento CFP y produce como **salida** un término, el cual actúa como **entrada** para el módulo de filtrado de *stopwords* . La **salida** de este módulo será un término, quien a su vez actuará como entrada para el *módulo de identificación de tokens*.

• Módulo de Identificación de Tokens

Este módulo reconoce significados de términos (*tokens*) utilizando *pattern matching*. Este tipo de recuperación determina si un término se encuentra o no en un contexto dado; dependiendo del caso, este contexto podrá ser el documento CFP o un vocabulario predefinido.

En este módulo se utilizan los DFAs generados en el *módulo de inicialización* correspondiente a los diccionarios de ciudades, países, estados y palabras claves.

Recibe como **entrada** un término.

Produce como **salida** el término junto con un token (significado) asociado, los cuales son almacenados provisoriamente de manera tal que el *módulo de ejecución de reglas* pueda obtenerlos mas adelante y así ejecutar las reglas que sean convenientes.

● Módulo de Ejecución de Reglas

Este módulo tiene como objetivo ejecutar las reglas correspondientes al conjunto de reglas generado en el *módulo de inicialización*. La representación funcional de este módulo fue detallada en la definición conceptual del sistema (Pág. 113). Estas reglas intentan recuperar los datos relevantes del documentos CFP analizado, utilizando recuperación basada en contexto y *pattern matching* .

La *recuperación basada en contexto*, utiliza el contexto en el cual se encuentra el término que se está analizando, para recuperar categorías de términos. La recuperación no sólo depende del significado asociado al término, sino que también se analizan los términos que se encuentran en su contexto. El contexto a analizar será el documento CFP.

Recibe como **entrada** un término y un token asociado, los cuales fueron almacenados provisoriamente por el *módulo de identificación de tokens*.

En caso de tener éxito en la recuperación de alguna de las categorías de términos buscadas, produce como **salida** un cambio en el estado del sistema. Caso contrario el sistema mantiene el mismo estado.

● Módulo de Incorporación de los Datos Recuperados a una BDs

Este módulo tiene como objetivo preparar los datos que han sido recuperados de manera tal que sean adecuados para ser luego incorporados a la BDs de documentos CFPs. Las operaciones que realiza se encuentran detalladas en la definición conceptual del sistema (Pág. 114), las cuales incluyen la aplicación de la técnica de *stemming* al título(en caso de haber sido recuperado), la generación de las siglas probables (en caso de no haber recuperado la sigla) y el formato de los datos en una forma adecuada para su posterior consulta.

Luego estos datos recuperados son incorporados a la BDs. La interacción con la BDs se realiza utilizando el manejador de BDs MiniSQL (mSQL).

MiniSQL - Especificación -

MiniSQL fue desarrollado por David J. Hughes en la Universidad Bond, Australia.

MiniSQL o mSQL es un sistema manejador de BDs relacional diseñado para proveer un rápido acceso a datos almacenados con bajo requerimiento de memoria. Ha sido diseñado para ejecutar operaciones en forma rápida y con poco *overhead* de recursos.

Como su nombre lo indica, mSQL ofrece un subconjunto de operaciones SQL como interface de *query*. Aunque sólo soporta un subconjunto, todas las operaciones que sí soporta concuerdan con la especificación ANSI SQL. Permite que un programa o usuario manipule y recupere datos en estructuras de tablas. Aunque no soporta todas las operaciones relacionales definidas en la especificación ANSI, provee la capacidad de "unir" (*join*) múltiples tablas.

El *daemon* mSQL es una aplicación que espera conexiones sobre un socket TCP. Es un motor de proceso simple que acepta múltiples conexiones y serializa los *queries* recibidos.

El paquete mSQL incluye una API en C. Esta API y el motor de BDs han sido diseñados para trabajar en un ambiente cliente/servidor sobre una red TCP/IP. La API permite que cualquier programa escrito en C pueda comunicarse con el motor de BDs.

Objetivos de Diseño

El objetivo del diseño es que la BDs soporte operaciones cliente/servidor. Los datos no solo están disponibles vía una API desde máquinas remotas a través de la red, sino que también pueden ser transmitidos de forma tal que permitan que hosts de diferente diseño compartan información.

Soporte Cliente/Servidor

Su implementación está basada en una API que permite la comunicación con el servidor a través de sockets.

El servidor mSQL ofrece dos mecanismos por el cual puede ser contactado :

- Puerta TCP/IP
- Socket de dominio UNIX

Cuando una aplicación inicia una comunicación con el servidor mSQL la API cliente se conecta, ya sea con el socket local UNIX o con el socket TCP para negociar la sesión.

Se soportan ambos tipos de sockets por razones de *perfomance*. Acceder al servidor a través de un socket UNIX es más rápido que acceder a través de un socket TCP, debido a que no se invoca código del kernel de red. Si se utiliza interface TCP se requiere que los datos sean empaquetados y encapsulados como paquetes TCP/IP aún si el servidor está ubicado en la misma máquina. La interface TCP sólo debería ser usada por aplicaciones que corren sobre hosts de red remotos.

Para facilitar las operaciones cliente/servidor todos los datos enviados entre la aplicación del cliente y el servidor mSQL tienen un formato de texto ASCII. Esto concuerda con el paradigma SQL en el cual todos los datos son enviados al servidor en un texto basado en el string del *query*.

Soporte Multi - Usuario

mSQL soporta múltiples sesiones simultáneas de usuarios. Maneja múltiples sesiones sin la necesidad de múltiples procesos debido a que va encolando los requerimientos de los usuarios y los va procesando en orden.

Este diseño tiene la ventaja de no tener *overhead* de memoria extra según el número de clientes, y la desventaja es que si se reciben múltiples *queries* simultáneamente existe una demora debido a que los *queries* son encolados.

Los detalles del usuario en el momento de la conexión son utilizados en el control de acceso a las BDs. El servidor soporta control de acceso según la identificación del usuario (*username)*, el tipo de conexión (local o remota) y el *hostname* de la máquina remota.

Control de Acceso

El control de acceso es manejado por medio de un archivo (*msql.acl*) dentro del directorio de instalación. Este archivo está dividido en entradas por cada BDs que va a ser controlada.

En el caso del sistema que hemos desarrollado, el acceso a las BDs utilizadas se define de la siguiente manera:

- El acceso de escritura corresponde al Sistema de Recuperación de Información de Documentos CFPs.
- El acceso de lectura corresponde al programa CGI.

Utilizamos este manejador de BDs ya que sus características se adaptan a nuestro sistema, debido a que en la consulta se utilizan *queries* simples y como provee un rápido acceso, esto favorece la rapidez en la consulta de la información recuperada.

Este módulo obtiene como **entrada**, a partir del estado del sistema el conjunto de datos que han sido recuperados del documentos CFP.

Produce como **salida** la incorporación de estos datos a la BDs del sistema.

CAPITULO 6

SISTEMA DE CONSULTA DE DOCUMENTOS CFPs

Este capítulo incluye :

INTRODUCCIÓN AL SISTEMA DE CONSULTA DE DOCUMENTOS CFPS

Presentación y objetivo del Sistema de Consulta de Información de Documentos CFPs

El Sistema de Consulta tiene como objetivo permitir consultas sobre una colección de documentos CFPs. La información consultada es aquella que ha sido almacenada por el Sistema de Recuperación de Información de Documentos CFPs.

La consulta se realiza a través de un browser Web y una serie de *scripts* colaboran como interfase entre la consulta que recibe un servidor Web y la BDs.

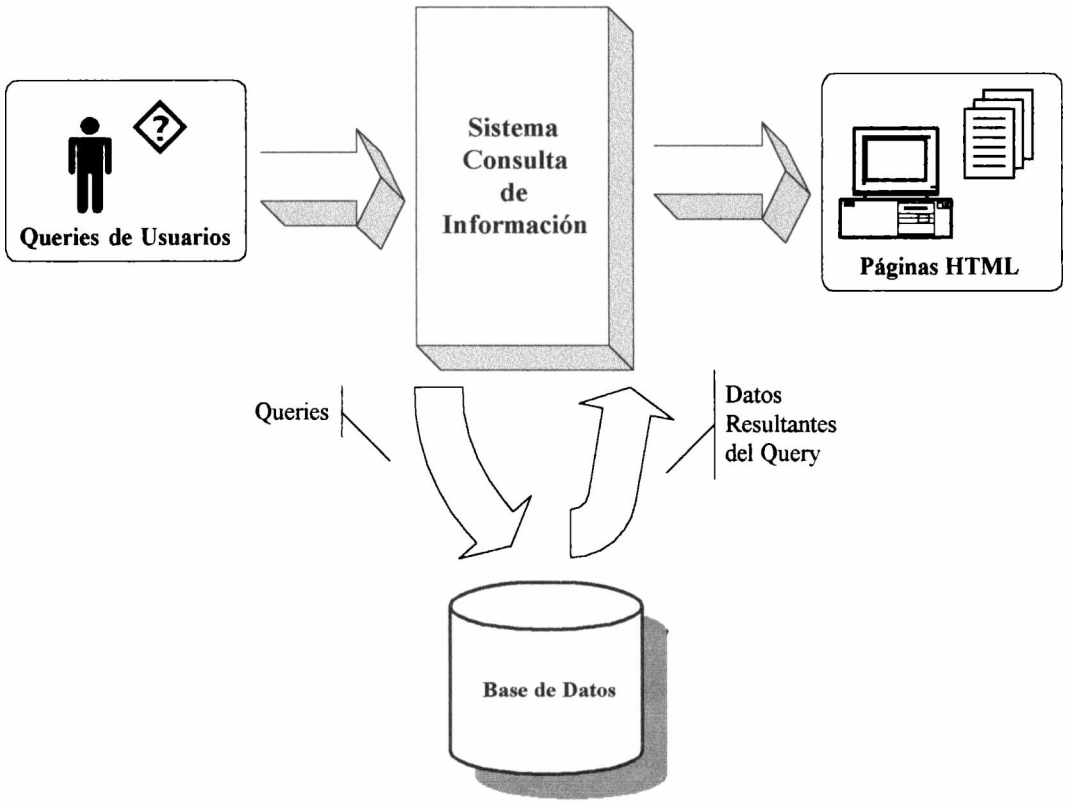


Figura 6.1 *Presentación del Sistema de Consulta de Información*

Como se puede observar en la *Figura 6.1*, este sistema recibe como entrada *queries* por parte de usuarios. La función del mismo será presentar a dichos usuarios información respecto a la consulta realizada, en caso de obtener esta información.

La consulta se realiza a través de una página HTML así como también la presentación de resultados.

Las páginas que presentan los datos como respuesta a una consulta, como ya se adelantado tienen la propiedad de ser generadas al momento por el servidor sin estar almacenadas en algún archivo y basándose en la consulta del usuario. Construir páginas HTML dinámicamente es un enfoque interesante para enfrentar la sobrecarga de información en ciertas áreas.

Esta consulta permitirá a los usuarios acceder según diferentes criterios, al conjunto de CFPs que consideren de su interés y a partir de él, identificar aquellos que considera se ajustan más a sus inquietudes.

DEFINICIÓN CONCEPTUAL DEL SISTEMA DE CONSULTA DE DOCUMENTOS CFPS

Presentación del Sistema de Consulta detallando los criterios de búsqueda y la presentación de resultados

El sistema permite al usuario realizar consultas sobre una colección de documentos CFPs.

Podemos destacar dos aspectos a detallar :

① Presentación de las opciones de consulta.

② Presentación de resultados de una consulta.

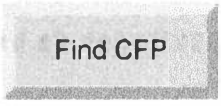
① Opciones de Consulta

La consulta se lleva a cabo mediante un página HTML creada para tal fin El diseño de esta página puede observase en la *Figura 6.2*.

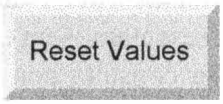
Mediante esta página HTML el sistema le presenta al usuario lo siguiente :

- Información de ayuda acerca de la presentación de los resultados de una búsqueda.
- Información acerca de la incorporación de un CFP al sistema de consulta.
- Criterios de búsqueda a seleccionar.
- Criterio de ordenamiento de los resultados de la búsqueda.

En esta página se presentan dos botones con la siguiente funcionalidad :



: Presionado este botón se realiza la consulta de documentos CFPs según los criterios seleccionados.



: Presionando este botón se inicializan los criterios de búsqueda anulando los datos ingresados hasta el momento .

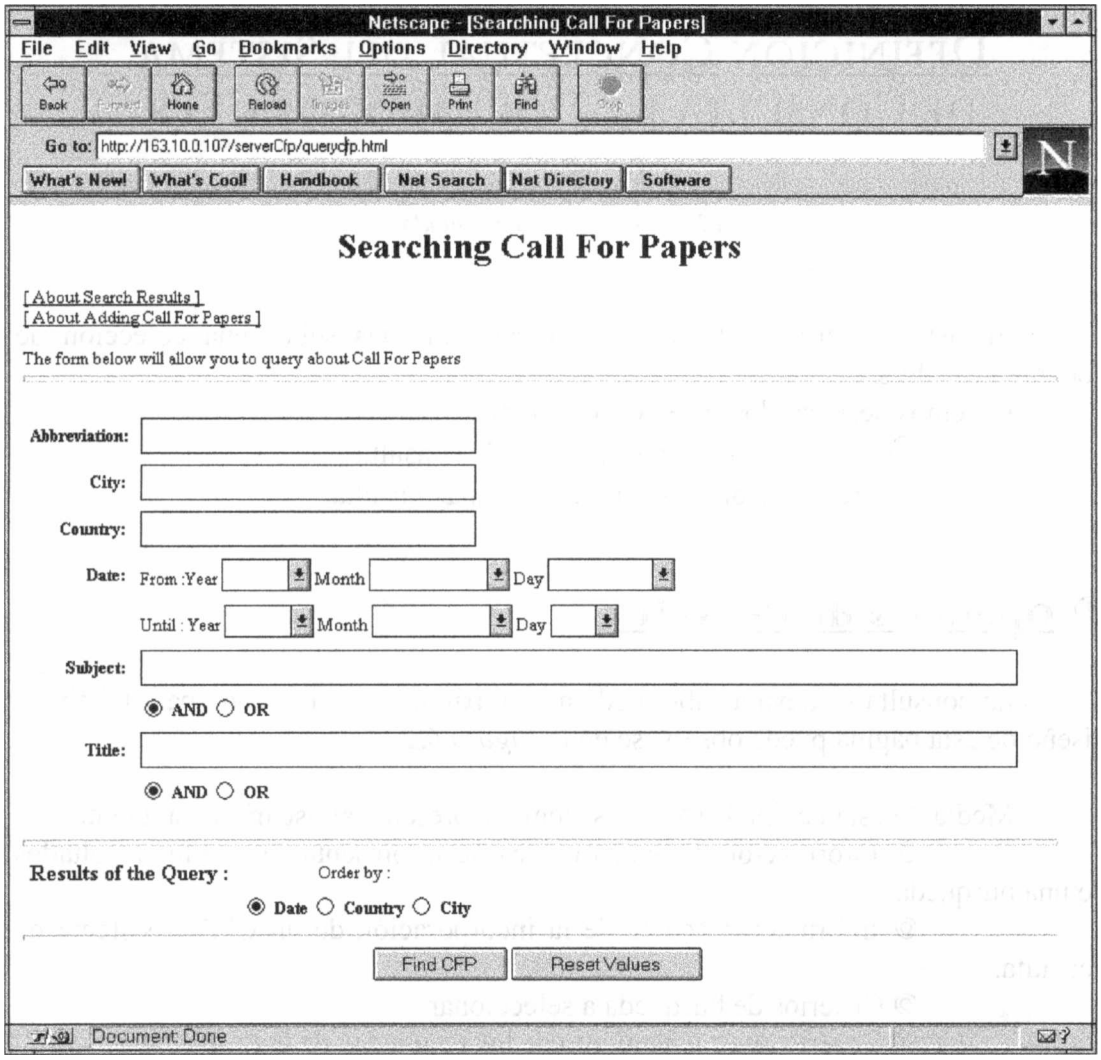


***Figura 6.2** - Página HTML de Consulta de Documentos Call For Papers*

Ayuda acerca de la presentación de los resultados de una Consulta

En la página de consulta existe un *link* (*[About Search Results]*) que permite acceder a otra página HTML donde brindamos información de ayuda acerca de la forma de presentación de los resultados. Presentamos un ejemplo de una página que podría haber sido generada a partir de una consulta con referencias de ayuda sobre la presentación de los resultados .

Por medio del *link [Searching Call For Papers]* se puede retornar desde esta página a la página de consulta de información.

El diseño de esta página puede observarse en la *Figura 6.3* .

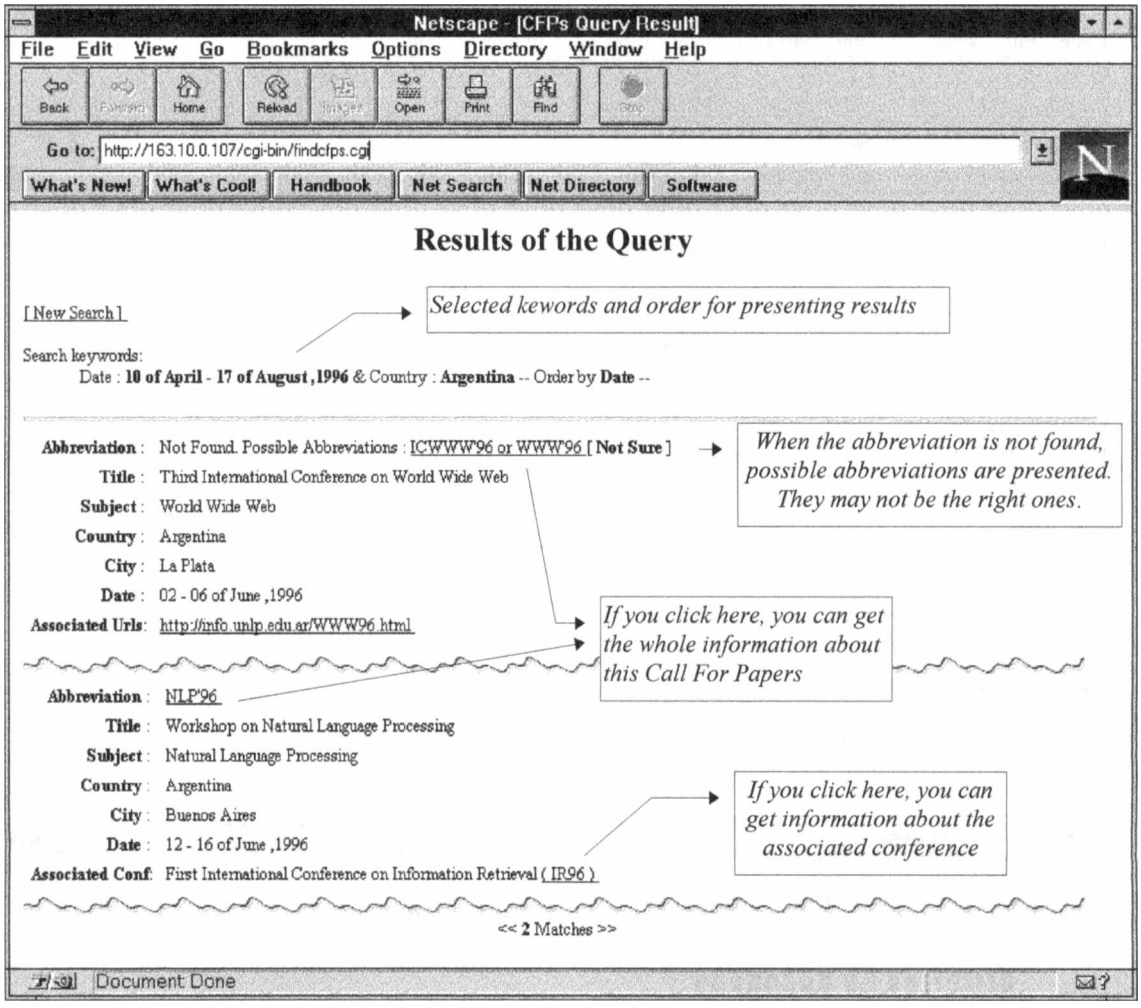


Figura 6.3 *- Página HTML de ayuda acerca de los resultados de una Consulta -*

Ayuda acerca de la Incorporación de CFPs al Sistema de Consulta

En la página de consulta existe un *link ([About Adding Call For Papers])* que permite acceder a otra página HTML donde brindamos información acerca de la manera de incorporar un CFP a este sistema de consulta.

La incorporación de información acerca de un documento CFP es justamente el objetivo del Sistema de Recuperación de Información de Documentos CFPs por lo tanto, como ya se ha explicado en la presentación de este sistema (Pág. 89), la forma de incorporar CFPs para poder ser analizados es enviándolos a través del correo electrónico a una cuenta de *e-mail* destinada a la recolección de CFPs .

En esta página se informa la dirección de *e-mail* a la cual se debe enviar el CFP para su análisis y posterior incorporación a la BDs de manera tal de poder ser consultado a través de este sistema.

Por medio del *link [Searching Call For Papers]* se puede retornar desde esta página a la página de consulta de información.

El diseño de esta página puede observarse en la *Figura 6.4*.

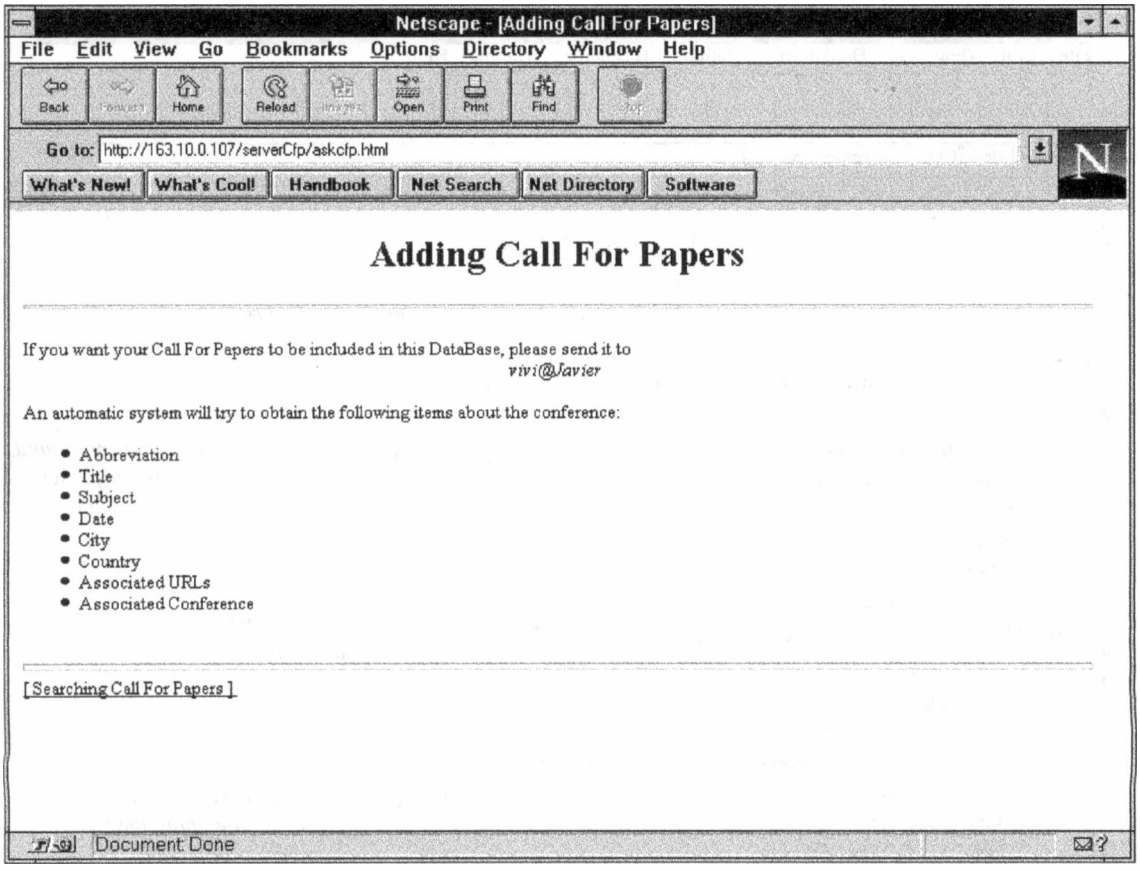


Figura 6.4 *- Página HTML acerca de la incorporación de CFPs al Sistema de Consulta*

Criterios de Búsqueda

El sistema permite realizar consultas sobre un solo criterio de búsqueda a combinar más de uno de manera tal de realizar una búsqueda más estringida.

Presentamos el conjunto de criterios de búsqueda mediante un *Form HTML* (el cual fue presentado con más detalle en el *Capítulo 4* (Pág. 77)) donde el usuario puede tipear los dato deseados.

Los criterios de búsqueda incluyen :

- **Abbreviation** : Corresponde a una sigla de conferencia. Al utilizar este criterio se buscan los CFPs que *incluyan* en el campo recuperado como sigla los datos ingresados por el usuario, es decir que no es una búsqueda exacta.

- **City** : Corresponde a una ciudad. Al utilizar este criterio se buscan las conferencias que se desarrollan en esta ciudad. No es una búsqueda exacta, sino que se buscan los CFPs que *incluyan* en el campo recuperado como ciudad los datos ingresados por el usuario.

➲ **Country** : Corresponde a un país. Al utilizar este criterio se buscan las conferencias que se desarrollan en este país. Es una búsqueda exacta.

➲ **Date** : Corresponde a una fecha de desarrollo de conferencias. En este caso se permiten diferentes opciones :

- Consultar una fecha exacta (Año, Mes,Día).
- Consultar un año completo (Año).
- Consultar un rango de años (Año, Año).
- Consultar un mes completo (Año, Mes).
- Consultar un rango de meses(Año, Mes, Mes)
- Consultar un rango de Fechas(Año, Mes,Día,Año, Mes, Día).

Al utilizar este criterio la búsqueda depende del caso que haya sido seleccionado según sea, un rango de fechas o una fecha exacta. Por lo tanto la búsqueda podrá ser exacta o no según el caso.

➲ **Subject** : Corresponde a un tema de conferencia.

➲ **Title** : Corresponde a un título de conferencia.

Al utilizar tanto el criterio de búsqueda de *título* como *tema* permitimos ingresar una o más palabras para buscar conferencias relacionadas. Además se permite al usuario seleccionar el conector lógico (AND, OR) para relacionar estas palabras de forma tal de poder ampliar o restringir la búsqueda deseada.

En estos casos realizamos un proceso especial a los datos ingresados por el usuario, el cual consiste en aplicar la técnica de Recuperación de Información (IR) conocida como ***Stemming*** [BaezaYates-Frakes92].

Esta técnica permite obtener las raíces de las palabras y así poder reconocer palabras de una misma familia. Al aplicar esta técnica a los datos ingresados como criterio de búsqueda para el título o el tema de una conferencia, se puede obtener una mayor cantidad de conferencias relacionadas con dicho tema/título debido a que no es una búsqueda exacta, sino que la consulta se realiza mediante un *matching* de variantes morfológicas de los términos. Es en estos casos donde interesa poder abarcar el mayor número de conferencias relacionadas con el criterio seleccionado.

Esta técnica se aplica en el proceso de indexación (a cargo del Sistema de Recuperación de Información de Documentos CFPs) en el caso de haber recuperado el título de la conferencia del CFP analizado, y también se aplica en el procesamiento de *queries* sobre los términos ingresados por el usuario.

Criterios de Ordenamiento

El sistema permite al usuario seleccionar un criterio de ordenamiento para la presentación de los resultados de la consulta.

Estos criterios comprenden :

- ***Date*** : Los resultados estarán ordenados por Fecha de Inicio y Fecha de Finalización de la conferencia.
- ***Country*** : Los resultados estarán ordenados por país.
- ***City*** : Los resultados estarán ordenados por ciudad.

Por defecto se considera como criterio de ordenamiento de los resultados de la consulta a la Fecha de realización de la conferencia (***Date***).

② Resultados de una Consulta

Como ya hemos adelantado, el sistema genera en forma dinámica una página HTML con los resultados de la consulta. En caso en que se produzca algún tipo de error generamos también dinámicamente una página especial de error que informa al usuario el error que se ha producido y le permite retornar a la página de consulta de información . Por lo tanto se tiene dos tipos de páginas resultado :

❶ Páginas con datos sobre CFPs como resultado de una consulta.
❷ Páginas con un mensaje de error asociado .

❶ Página Resultado de Consulta

Como resultado de una consulta generamos dinámicamente una página donde se informa lo siguiente :

- ❖ Los criterios de búsqueda que han sido seleccionados junto con los datos ingresados por el usuario de manera tal que el usuario tenga una visión de la búsqueda que ha realizado.
- ❖ El criterio de ordenamiento de presentación de los resultados que ha sido seleccionado o en caso en que no se haya seleccionado ninguno, por defecto se tiene el criterio de ordenar la información de los CFPs por la fecha de realización de las conferencias.
- ❖ La información recuperada de los CFPs que se encuentran según el o los criterios de búsqueda seleccionados.
- ❖ La cantidad de CFPs que se han encontrado a partir de la consulta.

Por medio del *link [New Search]* se puede retornar desde esta página a la página de consulta de información .

Presentación de la Información Recuperada sobre un CFP

La información acerca de un CFP consta de varios ítems. Estos dependen del grado de recuperación que se ha tenido al analizar dicho CFP. Se presentan aquellos ítems que han podido ser recuperados, el resto se informa que no son conocidos (*Not Found*).

Si dicho CFP no contenía o no se han podido recuperar direcciones URLs asociadas, este ítem no se presenta en el conjunto de ítems asociados al CFP.

En el caso que el CFP hiciera referencia a una conferencia asociada, se informa la sigla y/o título de la misma y si los datos de esta conferencia se encuentran en la BDs del sistema permitimos por medio de un *link*, que el usuario acceda también a la información asociada a esta conferencia. Si la conferencia de difusión del CFP no está asociada a otra conferencia, este ítem no se presenta en el conjunto de ítems asociados.

La presentación de estos ítems puede verse reflejada en la *Figura 6.3* .

Este conjunto de ítems comprende :

- **Abbreviation**

Este ítem corresponde a la sigla de la conferencia.

Como ya hemos visto en el desarrollo del Sistema de Recuperación de Información de Documentos CFPs (Pág. 116), en aquellos casos en que no se ha recuperado la sigla de la conferencia el sistema genera a partir del título dos siglas probables. Esto permite que un usuario pueda reconocer la conferencia aún cuando la sigla no ha podido ser recuperada. Generamos dos alternativas de siglas, donde una corresponde a la abreviatura del título, y otra corresponde a la abreviatura del *stem*. Como es el caso en que es una información generada por el sistema a partir de un dato recuperado y no un dato recuperado a partir del CFP, aclaramos en esta presentación que son *siglas probables* de manera tal de no dar lugar a posibles confusiones por parte del usuario. Para tomar conocimiento de la conferencia en caso en que la sigla no fuera conocida por el usuario, se cuenta con la información sobre el título de la misma.

Por lo tanto este ítem siempre estará presente y además actúa como un *link* que permite acceder al documento CFP en su formato original tal como ha sido recibido por el Sistema de Recuperación de Información. El usuario tiene la posibilidad en caso que le interese dicha conferencia, de acceder a la información completa sobre este CFP.

- **Title**

Este ítem corresponde al título de la conferencia en caso en que haya podido ser recuperado.

Subject

Este ítem corresponde al tema sobre el cual trata la conferencia en caso en que haya podido ser recuperado.

Country

Este ítem corresponde al país donde se desarrolla la conferencia en caso en que haya podido ser recuperado.

City

Este ítem corresponde a la ciudad donde se desarrolla la conferencia en caso en que haya podido ser recuperada.

Date

Este ítem corresponde a la fecha o el rango de fechas en el cual se desarrolla la conferencia en caso en que haya podido ser recuperada.

Associated URLs

Este ítem corresponde a un conjunto de URLs que han sido encontradas en el contenido del documento CFP. Estas actúan como *links* que permiten acceder a estas URLs asociadas. Por lo tanto el usuario además tiene la posibilidad de obtener al momento no sólo el documento CFP, sino también por medio de este ítem información que se encuentra relacionada con dicha conferencia.

Associated Conference

Este ítem corresponde a la sigla y/o título de una conferencia que está asociada a la conferencia difundida por el CFP. En caso en que exista la información de esta conferencia en el sistema, este ítem actuará como un *link* que permitirá al usuario acceder también a los datos de dicha conferencia.

❷ Página de Error

En caso en que se produzca algún tipo de error ya sea por parte del usuario o por parte del sistema, o en el caso en que la consulta realizada no obtenga resultados, generamos páginas especiales con el fin de informar al usuario lo sucedido.

Un ejemplo del diseño de estas páginas puede observarse en la *Figura 6.5* .

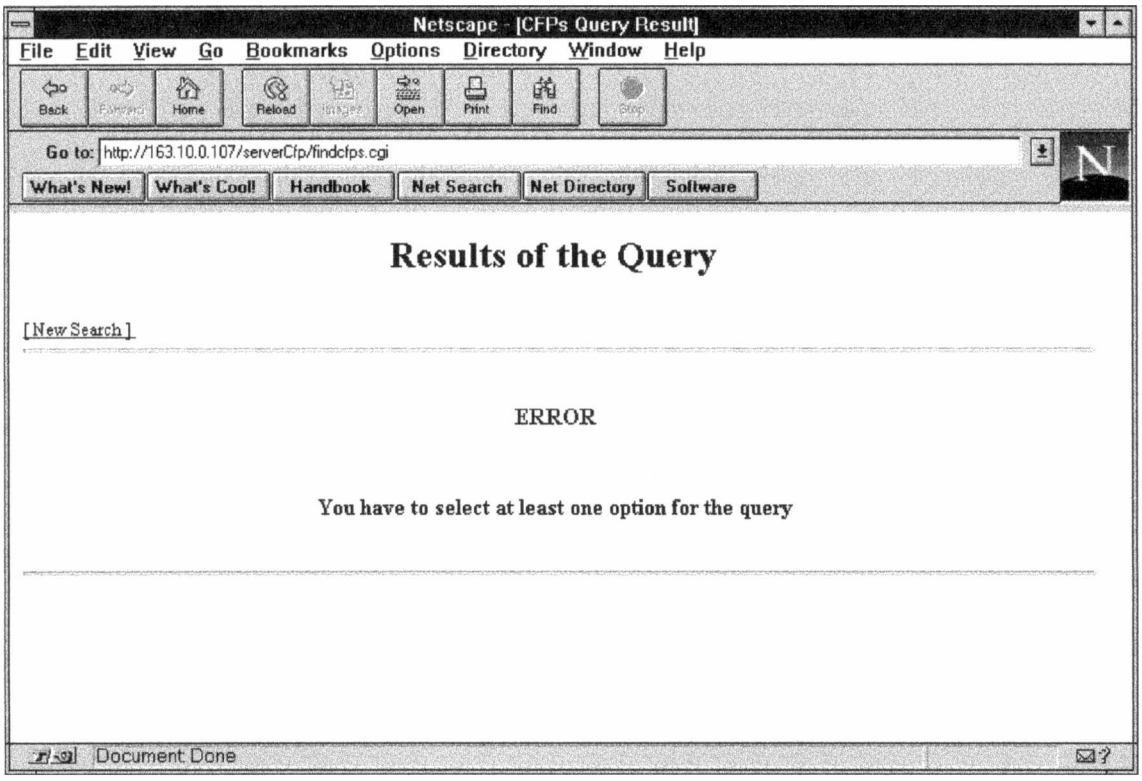


Figura 6.5 *Página HTML de error en el caso en que el usuario no ha seleccionado ninguna opción para realizar la consulta*

Se consideran los siguientes errores :

❖ En caso en que el usuario realiza una consulta pero no se obtiene ningún CFP que cumpla con los criterios seleccionados, el sistema le informa al usuario que no se ha encontrado información para dicha consulta y le muestra los criterios de búsqueda que ha seleccionado.

❖ En el caso en que el usuario realiza una consulta pero no ha seleccionado ningún criterio de búsqueda, el sistema le informa mediante una página de error, que debe seleccionar al menos un criterio de búsqueda para poder realizar una consulta (ejemplo presentado en la *Figura 6.5*).

❖ En caso en que el usuario realiza una consulta y existe alguna inconsistencia en la incorporación de datos a los ítems que representan la fecha o el rango de fechas, se lo informa mediante una página de error.

❖ En caso en que el usuario realiza una consulta y el server mSQL no se encuentra activo o existe algún problema en el acceso a las BDs ,se lo informa mediante una página de error.

En todos los casos por medio del *link [New Search]*se puede retornar desde estas páginas a la página de consulta de información.

ARQUITECTURA DEL SISTEMA DE CONSULTA DE DOCUMENTOS CFPS

Presentación de los módulos del sistema de Consulta de Información de Documentos CFPs

En esta sección presentamos detalles acerca del proceso de comunicación que permite realizar una consulta y obtener los datos resultantes de la misma.

El usuario realiza una consulta con el fin de obtener datos referentes a CFPs de su interés, accediendo a través de su *browser* Web a una página HTML creada para tal fin tal como se ve reflejada en la *Figura 6.2* en la definición conceptual del sistema (Pág. 136). Mediante esta consulta el usuario puede seleccionar uno o más criterios para recuperar información acerca de los CFPs y además seleccionar el criterio de ordenamiento de presentación de la información resultante.

Los datos acerca de una consulta son enviados al servidor Web quien a su vez los envía al programa CGI que realizará la consulta, el cual se encuentra referenciado en la página de consulta. Este proceso de comunicación se refleja en la *Figura 6.6* presentada a continuación .

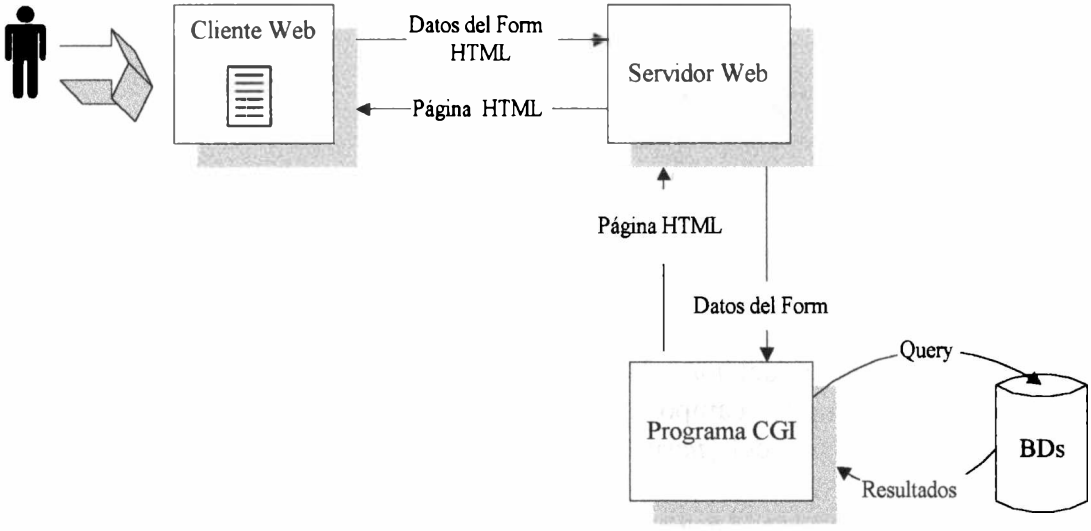


***Figura 6.6** - Proceso de comunicación en la realización de una consulta -*

PROGRAMA CGI

El *programa* CGI cumple las siguientes funciones :

- Recuperar los datos ingresados por el usuario.
- Construir el *query* y realizar la consulta.
- Generar la página HTML con los resultados de la consulta.
- Contemplar posibles errores generando la página de error correspondiente.

Estas funciones se reflejan en la *Figura 6.7* presentada a continuación

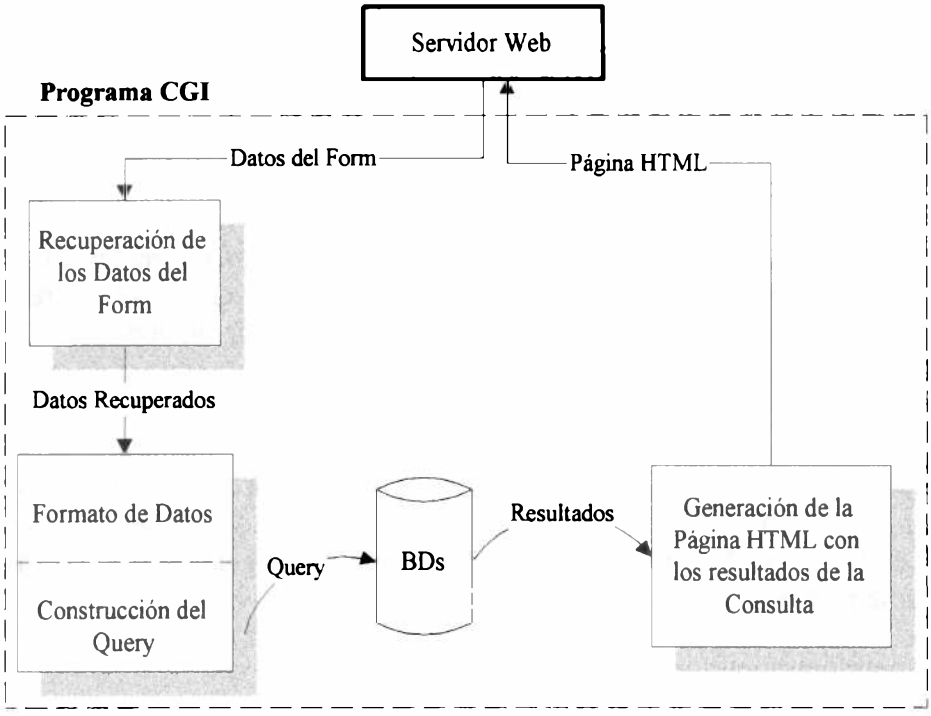


***Figura 6.7** - Funciones del programa CGI -*

A continuación detallamos cada función.

Recuperación de los Datos de la Consulta

En primer lugar recuperamos los datos que han sido ingresados por el usuario a través del *form* HTML presentado en la página de consulta. Se recuperan cada uno de los campos que contienen algún dato, y esta información se utiliza en la construcción del *query*.

Construcción del *Query* y Consulta

Para construir el *query* analizamos los datos ingresados por el usuario que han sido recuperados. Debe existir al menos un criterio de búsqueda para realizar la consulta, en caso contrario se informa al usuario generando una página de error.

En ciertos casos, según cual sea el dato ingresado por el usuario, aplicamos un proceso de transformación con el fin de obtener un formato acorde con los datos almacenados en la BDs para poder realizar la consulta. Los ítems a los cuales se les aplica algún tipo de transformación comprenden :

- **Título y Tema**

En el Sistema de Recuperación de Información de Documentos CFPs, cuando recuperamos estos ítems a partir del CFP les aplicamos dos técnicas IR antes de almacenarlos en la BDs.

En primer lugar realizamos un filtrado de palabras no significativas (técnica *stoplist* - presentada en el *Capítulo 3* - Information Retrieval - (Pág. 32)) y luego a cada una de las palabras no filtradas, les aplicamos la técnica de *stemming* (presentada en el *Capítulo 3* - Information Retrieval - (Pág. 34)) con el fin de obtener las raíces (*stems*) de dichas palabras para también almacenarlas en la BDs.

Para realizar la consulta es necesario aplicar las mismas técnicas a los datos ingresados por el usuario, por lo tanto a los ítems Título y Tema se les realiza primero un filtrado de palabras no significativas (utilizando la misma lista de *stopwords* que en el Sistema de Recuperación de Información) con el fin de filtrar a partir de estos datos las palabras no significativas ingresadas por el usuario, evitando considerar palabras innecesarias en la búsqueda. Aquellas palabras que no han sido filtradas, se les aplica la técnica de *stemming* obteniendo las raíces (*stems*) de las mismas. Estos términos *stems* son los que se incorporan en la construcción del *query* y por lo tanto los utilizados para realizar la consulta del título y/o tema y así obtener un *matching* con los *stems* almacenados en la BDs. De esta manera abarcamos un mayor número de conferencias debido a que no se busca un *matching* exacto, sino que se busca un *matching* de variantes morfológicas de los términos ingresados por el usuario.

❖ Fecha

Al analizar la fecha controlamos en primer lugar que sea una fecha válida, es decir que sea una fecha exacta o un rango de fechas válido (la fecha inicial sea menor que la fecha de finalización, etc). En caso de producirse algún error generamos una página HTML de error para informarle al usuario. Caso contrario transformamos las fechas al mismo formato en que están almacenadas en la BDs (año mes día) y las incorporamos al *query*.

❖ Otros ítems

En el caso de los ítems ciudad y país los transformamos a un formato tipo frase, es decir la primer letra en mayúscula y el resto en minúsculas antes de incorporarlos al *query* ya que en este formato están almacenados estos ítems en la BDs.

En el caso de la ciudad, la búsqueda no será exacta, sino que se buscarán aquellas ciudades que *incluyan* el dato ingresado por el usuario. De esta manera no se necesita ingresar una ciudad exacta para obtener información acerca de conferencias que se realicen en dicha ciudad.

En el caso de la sigla la transformamos a mayúscula y la incorporamos al *query* pero la búsqueda no será exacta, sino que indicamos en el *query* que se va a buscar toda sigla que *incluya* el dato incorporado al *query*. Realizamos esto debido a que las siglas almacenadas en la BDs están formadas por la abreviatura del título de la conferencia correspondiente junto con el *año* de realización de la misma, por lo tanto en caso en que el usuario ingrese sólo la sigla sin indicar el año, podemos recuperar el conjunto de conferencias asociadas a dicha sigla sin tener en cuenta el año de realización.

En estos dos últimos casos aplicamos a los términos correspondientes a estas categorías la operación *truncation*, la cual fue presentada en el *Capítulo 3 -Information Retrieval-* (Pág 26). Esta operación permite una fusión o combinación de términos usando caracteres comodines de tal forma que el término pueda hacer *matching* con múltiples palabras.

Luego de haber incorporado al *query* todos los criterios de búsqueda, analizamos el criterio de ordenamiento para la presentación de los resultados, seleccionado por el usuario o designado por defecto por el sistema, el cual también es incorporado al *query*.

Una vez concluida la construcción del *query* realizamos la conexión con el servidor mSQL, si no fuera posible dicha conexión generamos una página HTML de error informando al usuario que en ese momento no es posible realizar la consulta, debido a que el servidor no está activo. Además le enviamos un *e-mail* al administrador(root) informándole la existencia de este problema para que lo solucione. Caso contrario realizamos la consulta a la BDs.

Generación de la Página HTML con los Resultados de la Consulta

Luego de realizar la consulta analizamos los resultados retornados por el servidor mSQL. En caso en que no se ha podido obtener información acerca de CFPs con estos criterios de búsqueda seleccionados, generamos una página especial informando al usuario que no existe información respecto a su consulta. Caso contrario, generamos dinámicamente una página HTML a partir de los datos retornados por el servidor mSQL (no son páginas que se encuentran almacenadas en un archivo a las cuales se accede mediante un *link*).

Los datos de los CFPs se presentan ordenados según el criterio que ha sido elegido por el usuario (fecha, ciudad o país de cada conferencia) o por defecto estarán ordenados por la fecha de realización de las conferencias.

El servidor Web se encarga de retornar al *browser* Web dicha página. Para que el *browser* pueda presentar al usuario la página HTML con los resultados de la consulta, en primer lugar se le indica que el tipo de información a presentar corresponde a una página HTML, de manera tal que se puedan interpretar las marcas (*tags*) HTML. Para indicar esto se utiliza un encabezado del tipo :

Content-type : text/html

Este tipo de encabezados se encuentran detallados en el *Capítulo 4* - Internet y World Wide Web - (Pág.82) .

Un ejemplo del formato de la página generada se encuentra en la *Figura 6.3* en la definición conceptual del sistema (Pág. 137), presentando los criterios de búsqueda seleccionados, los datos relevantes recuperados de los CFPs así como también la posibilidad de acceder a la información completa referente a un CFP y de acceder a información relacionada mediante un conjunto de direcciones URLs asociadas . Además en caso de existir una conferencia asociada y si existe información referente a esta conferencia en la BDs se permite acceder también a este documento CFP.

Capítulo 7

Evaluación del Sistema IR de Documentos CFPs

EVALUACIÓN DEL SISTEMA IR DE DOCUMENTOS CFPS

Medidas *Recall* y *Precision*

Para evaluar el funcionamiento del sistema de Recuperación de Información de Documentos CFPs desarrollado, utilizamos las medidas de evaluación presentadas en el *Capitulo 3(IR)* (Pág. 56), *Recall* y *Precision.*

Siendo: | A | = Número de documentos relevantes para un *query*. Un documento se considera relevante para un *query* si corresponde al pedido representado por éste.

| B | = Número de documentos recuperados.

| A ∩ B | = Número de documentos relevantes recuperados .

La medida *recall* se define de la siguiente manera :

$$\text{RECALL} = \frac{|A \cap B|}{|A|}$$

Es decir, se define como el radio de los documentos relevantes recuperado para un *query* dado sobre el número de documentos relevantes para dicho *query* .

La medida *precision* se define de la siguiente manera :

$$\text{PRECISION} = \frac{|A \cap B|}{|B|}$$

Es decir, se define como el radio del número de documentos relevantes recuperados sobre el número total de documentos recuperados.

Realizamos la evaluación del sistema en base a una muestra de ***230 documentos CFPs*** de diferentes temáticas y nacionalidades. Definimos un conjunto de *queries*, seleccionando diferentes criterios de búsqueda, simples y combinados por más de un criterio.

Cada *query* tiene asociado lo siguiente :

- Conjunto de documentos CFPs considerados relevantes para dicho query. *Un documento CFP es relevante para un query si incluye el valor de la/s categoría/s de términos buscada/s.*
- Conjunto de documentos recuperados : Es el conjunto de documentos que se obtienen como respuesta al *query.*
- Conjunto de documentos relevantes recuperados : Dentro del conjunto de documentos recuperados, cuántos de ellos realmente son relevantes para el *query*, es decir cuántos de ellos realmente incluyen la/s categoría/s de términos buscadas. Para esto, analizamos cada documento CFP para comprobar la recuperación de las categorías de términos, determinando si fueron recuperadas en forma correcta o no.

Los resultados obtenidos se presentan en la siguiente tabla :

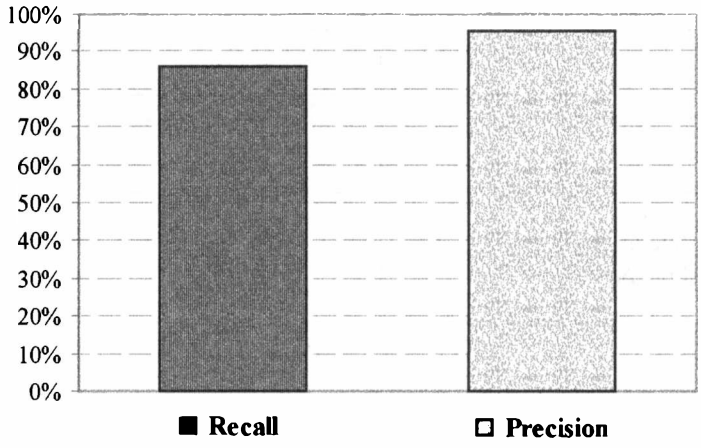
Query (Criterios de búsqueda)	$\|A\|$ = N^{ro} de documentos relevantes	$\|B\|$ = N^{ro} de documentos recuperados	$\|A \cap B\|$ = N^{ro} de documentos relevantes recuperados	**Recall** $\|A \cap B\| / \|A\|$	**Precision** $\|A \cap B\| / \|B\|$
Tema : *Logic*	52	39	35	**0,67**	**0,89**
Tema : *Logic* AND País: *Germany*	11	7	7	**0,63**	**1**
Tema: *Logic* AND Ciudad: *Leuven*	4	4	4	**1**	**1**
Tema: *Logic* AND Fecha Inicio: *June 1997* AND Fecha Fin : *October 1997*	14	13	13	**0,92**	**1**
Tema : *Logic* AND Fecha : *1996*	27	18	18	**0,66**	**1**
Sigla : *LICS*	2	2	2	**1**	**1**
Ciudad:*London* AND Fecha :*1996*	4	3	3	**0,75**	**1**
Ciudad : *New Brunswick*	7	6	6	**0,85**	**1**
Sigla : *PACT*	3	3	3	**1**	**1**
Título : *Functional*	13	10	9	**0,69**	**0,90**
Título : *Functional* AND País : *Argentina*	2	2	2	**1**	**1**
Tema : *Mathematics*	7	6	6	**0,85**	**1**
Sigla : *ASIAN*	3	2	2	**0,66**	**1**
País : *USA*	60	53	53	**0,88**	**1**
Tema : *Logic* AND *Functional*	6	3	3	**0,5**	**1**
Ciudad : *Passau*	2	1	1	**0,5**	**1**
Tema : *Compilation*	6	5	5	**0,83**	**1**
Fecha Inicio : *10 April 1996* AND Fecha Fin : *28 September 1996*	66	64	63	**0,95**	**0,98**
País : *Germany* AND Fecha : *September 1996*	13	10	10	**0,76**	**1**
Sigla : CP96	1	1	1	**1**	**1**
Título: *Programming Functional* OR Tema : *Autómata*	14	10	10	**0,71**	**1**

Query (Criterios de búsqueda)	|A| = N^{ro} de documentos relevantes	|B| = N^{ro} de documentos recuperados	|A∩B| = N^{ro} de documentos relevantes recuperados	**Recall** |A∩B| / |A|	**Precision** |A∩B| / |B|
País : Canada	2	2	2	**1**	**1**
Ciudad : *Paris*	6	6	6	**1**	**1**
País : Singapore	3	1	1	**0,33**	**1**
Ciudad : *Paris* AND Fecha : *January 1997*	4	4	4	**1**	**1**
Sigla : *ILPS*	1	2	1	**1**	**0,5**
Pais: *Scotland*	2	2	2	**1**	**1**
Pais: *UK*	8	6	6	**0,75**	**1**
Pais : *Francia*	15	16	15	**1**	**0,93**
Pais : *Russia*	1	1	1	**1**	**1**
Pais : *Francia* AND Fecha : *1997*	11	12	11	**1**	**0,91**
Pais : *Francia* AND Fecha Inicio : *August 1997* AND Fecha Fin : *September 1997*	3	4	3	**1**	**0,75**
Sigla : *POLP*	2	2	2	**1**	**1**
Tema : *Concurrent* OR *Parallel*	9	8	8	**0,88**	**1**
País : *Italy* AND Fecha Inicio : *March 1997* AND Fecha Fin : *November 1997* AND Título : *Languages* OR *Computation*	4	3	3	**0,75**	**1**
Fecha : *1998*	16	16	16	**1**	**1**
Sigla : ESSLLI	4	4	4	**1**	**1**
Fecha : *1997* AND Título : *Computation*	21	18	18	**0,85**	**1**
Sigla : ICLP97	1	3	1	**1**	**0,33**
Tema : *Algebra*	9	5	5	**0,55**	**1**
Sigla : *FOOL* AND País : *USA*	2	2	2	**1**	**1**

Query (Criterios de búsqueda)	|A| = N^{ro} de documentos relevantes	|B| = N^{ro} de documentos recuperados	|A∩B| = N^{ro} de documentos relevantes recuperados	**Recall** |A∩B| / |A|	**Precision** |A∩B| / |B|
País: *Italy* AND Fecha Inicio: *July 1997* AND Fecha Fin : *October 1997*	6	6	5	**0,83**	**0,83**
Fecha : *June 1997* AND Ciudad : *Amsterdam*	4	4	4	**1**	**1**
Título : *Foundations the objects oriented languages*	3	3	3	**1**	**1**

Los valores promedio de *recall* y *precision* son los siguientes :

Recall	Precision
85,80 %	95,50 %



Podemos observar que tenemos un mayor porcentaje de *precision*. Le dimos una mayor prioridad a la precisión de la información recuperada, dado que en el sistema desarrollado nos pareció conveniente proveer información confiable en lugar de ofrecer mayor cantidad de información pero con menor precisión.

Grado de Exactitud en la Recuperación

Otra forma de evaluar el sistema se basa en el grado de exactitud que se obtiene al recuperar cada categoría de término. Utilizando la misma muestra que en el caso anterior, evaluamos cada una de las categorías de términos, obteniendo el porcentaje de documentos en los cuales el sistema recuperó dicha categoría en forma correcta. Además discriminamos el número de documentos CFPs en los cuales se recuperó la categoría en forma incorrecta y los casos en que no ha podido ser recuperada. Los resultados fueron los siguientes :

Muestra : 230 documentos Call For Papers

Categoría	CFPs que incluyen la categoría	CFPs con categoría recuperada		CFPs con categoría no recuperada	% Recuperación Correcta
		Recuperación correcta	*Recuperación incorrecta*		
Sigla	141	118	13	10	**83,69 %**
País	226	219	3	4	**96,90 %**
Ciudad	227	212	5	10	**93,39 %**
Fecha	230	217	6	7	**94,35 %**
Título	230	177	7	46	**76,96 %**
Tema	230	150	7	73	**65,22 %**
Conf. Asociada	59	40	2	17	**67,80 %**

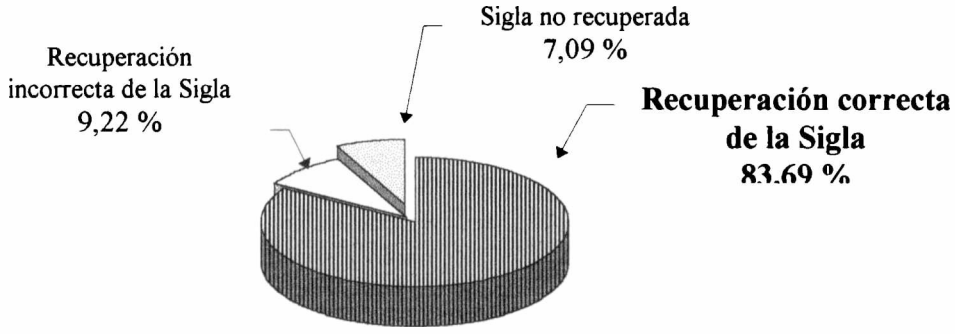
Estos valores de recuperación de las categorías de términos buscadas, se encuentran representados en los gráficos siguientes, en estos se incluye :

▲ Porcentaje de documentos CFPs donde la recuperación de la categoría ha sido correcta.

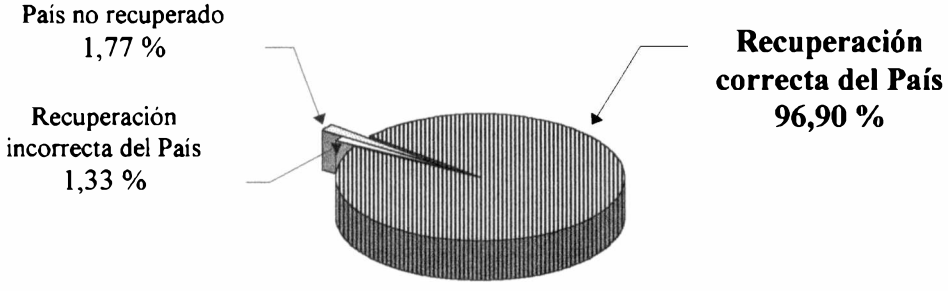
▲ Porcentaje de documentos CFPs donde la recuperación de la categoría ha sido incorrecta.

▲ Porcentaje de documentos CFPs donde la categoría no ha sido recuperada.

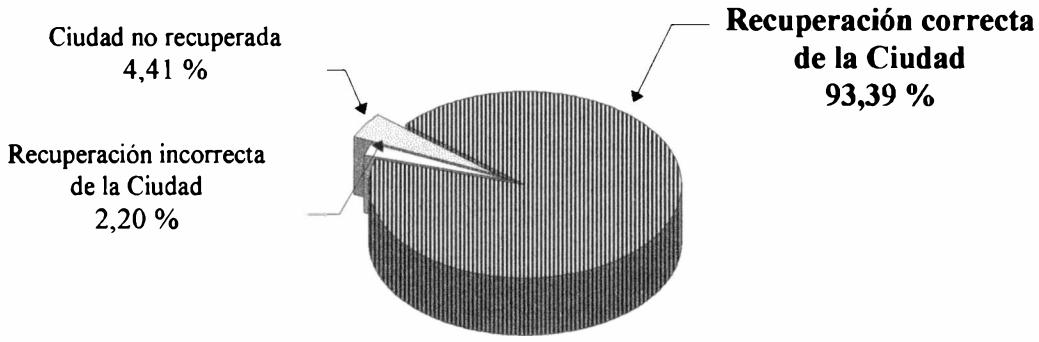
Representación gráfica de la recuperación de la SIGLA DE LA CONFERENCIA



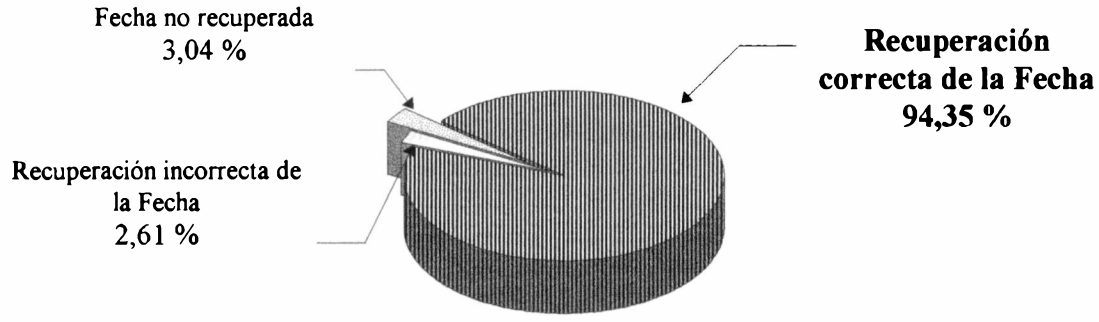
Representación gráfica de la recuperación del PAIS DE LA CONFERENCIA



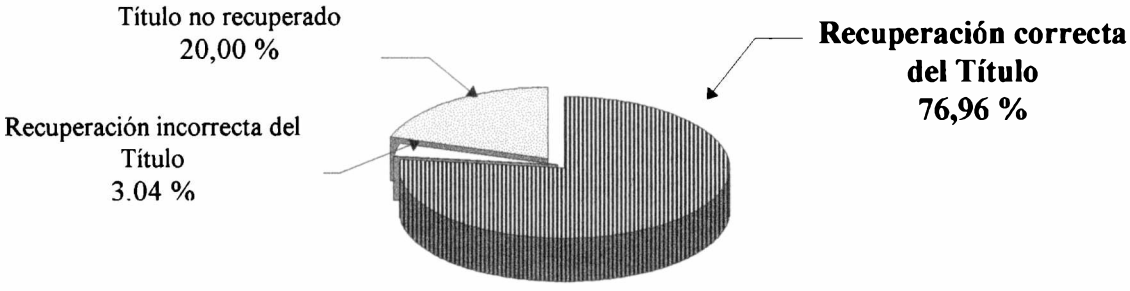
Representación gráfica de la recuperación de la CIUDAD DE LA CONFERENCIA



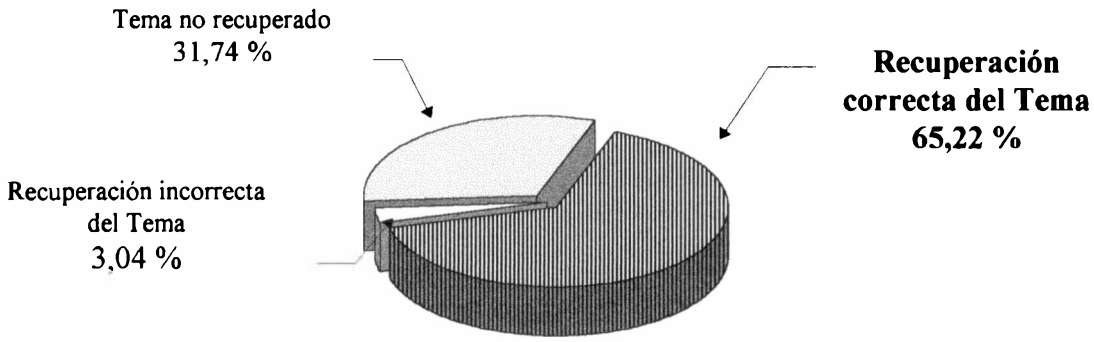
Representación gráfica de la recuperación de la FECHA DE LA CONFERENCIA



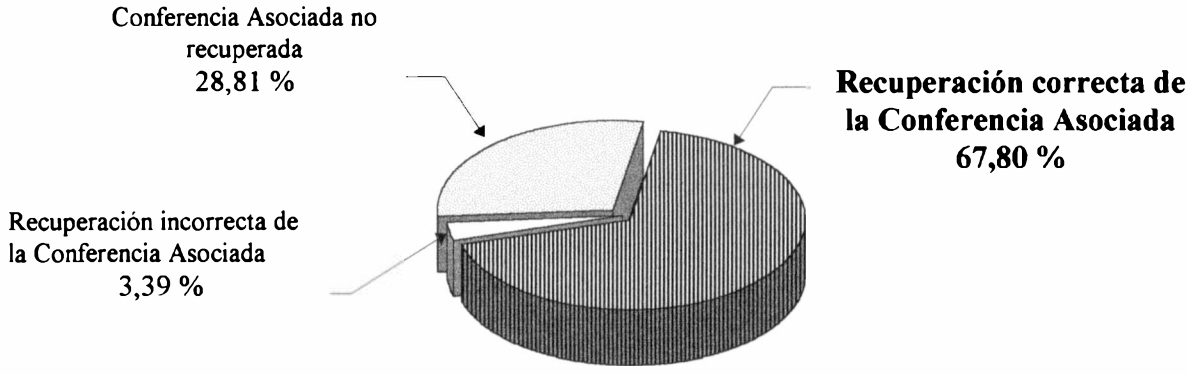
Representación gráfica de la recuperación del TITULO DE LA CONFERENCIA



Representación gráfica de la recuperación del ***TEMA DE LA CONFERENCIA***



Representación gráfica de la recuperación de ***CONFERENCIAS ASOCIADAS***



CONCLUSIONES

CAPITULO 8

CONCLUSIONES

Presentación de los inconvenientes surgidos en el desarrollo, las conclusiones obtenidas y las posibles ampliaciones futuras

❖ INCONVENIENTES

En el desarrollo del sistema, surgieron ciertos inconvenientes especialmente en la fase de recuperación de información. Entre ellos podemos mencionar los siguientes :

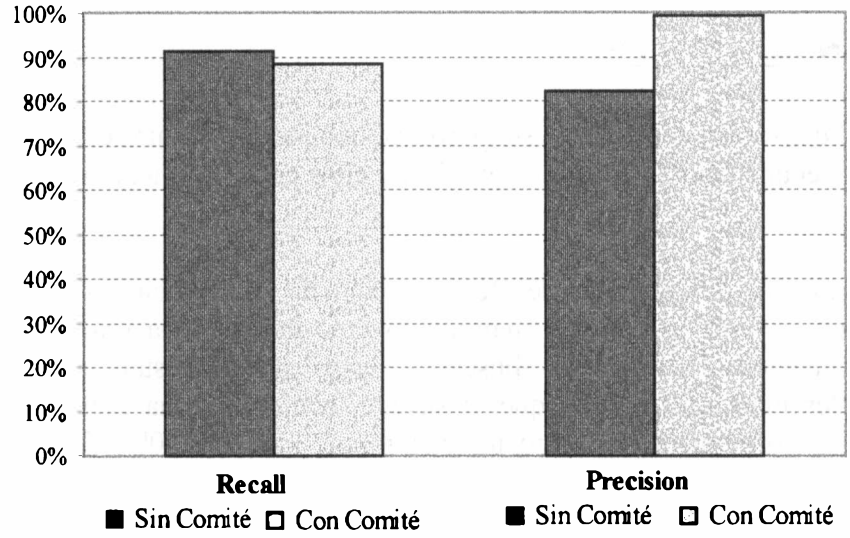
- El desarrollo del Sistema de Recuperación de Información se basa en el análisis de texto en lenguaje natural, lo cual implica una gran variedad de representación de los datos buscados, estos pueden estar expresados de múltiples maneras. A pesar que el sistema contempla una gran variedad de posibles representaciones, la recuperación no podrá ser 100% efectiva, ya que las reglas no son exactas.

- La mayoría de los CFPs hacen referencia a una lista de personas que forman el comité de programa, la cual en general incluye los nombres de los integrantes junto con la ciudad, estado y/o país de residencia. En ciertos casos en los que el sistema no recuperaba la ciudad y/o el país de desarrollo de la conferencia antes de comenzar a analizar esta lista, recuperaba como lugar de desarrollo algunos de estos datos mencionados, lo cual disminuía la precisión en la recuperación. Para solucionar esto existe un estado especial del sistema que indica que se está analizando una lista de este tipo, en cuyo caso no se intenta recuperar las categorías ya mencionadas, ya que lo mas probable es que sea un dato correspondiente a las personas que conforman el comité y no un dato correspondiente al CFP.

  Podemos ver cómo varían las medidas de *recall* y *precision* en ambos casos, identificando un estado especial de análisis de Comité de Programa y en caso contrario. Para esto analizamos un conjunto de *queries* acerca de documentos CFP, determinando los valores de evaluación correspondientes.

	Sin identificar Comité de programa		**Identificando Comité de programa**	
Query	**Recall**	**Precision**	**Recall**	**Precision**
Pais: *Scotland*	1	0,66	1	1
Pais: *Russia*	1	0,5	1	1
Tema : *Algebra*	0,66	1	0,55	1
País : *French*	1	0,93	1	0,93
Tema: *Concurrent* OR *Parallel*	1	1	0,88	1
País : *UK*	0,75	0,85	0,75	1
País : *USA*	0,88	0,98	0,88	1
País : *Canada*	1	0,66	1	1
Promedios :	**91,12%**	**82,25 %**	**88,25 %**	**99,12%**

Podemos observar a partir de esta tabla que disminuye un poco la recuperación si se contempla o no un estado especial de análisis de la lista de comité de programa, sin embargo la precisión aumenta en gran medida.



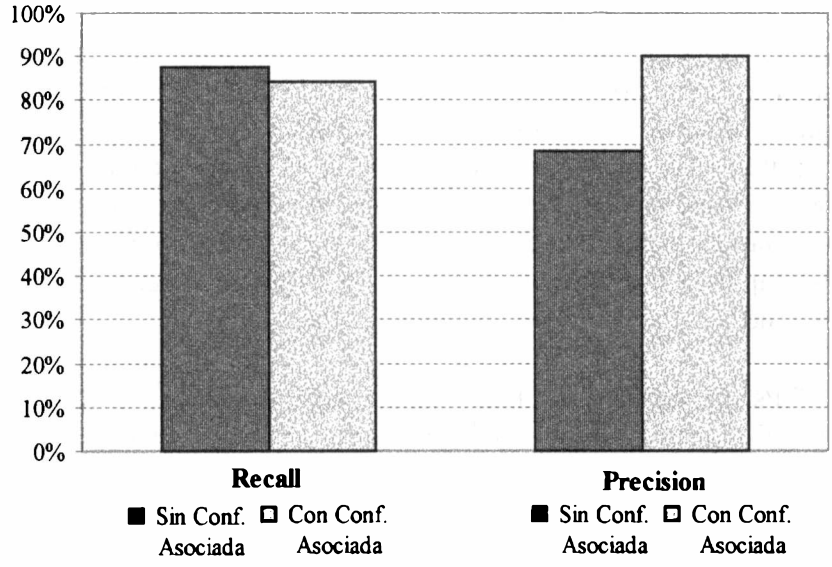
- Algunos CFPs no hacen referencia sólo a la conferencia que difunden, sino que en algunos casos también hacen referencia a una conferencia asociada, por lo tanto, para aumentar la precisión en la recuperación el sistema intenta diferenciar ambas conferencias. Analizando los CFPs observamos que cierto conjunto de palabras en general introduce una conferencia asociada. Utilizando este conjunto como guía podemos diferenciar el título y la sigla de la conferencia de los correspondientes a la conferencia asociada.

  Podemos representar cómo varían las medidas de *recall* y *precision* en ambos casos. Para esto analizamos un conjunto de queries acerca de documentos CFP que incluyen conferencias asociadas, determinando los valores de evaluación correspondientes .

	Sin identificar Conferencia Asociada		**Identificando Conferencia Asociada**	
Query	**Recall**	**Precision**	**Recall**	**Precision**
Sigla: *ASIAN*	0,66	0,5	0,66	1
Sigla : *ILPS*	1	0,25	1	0,5
Tema : *Logic*	0,67	0,87	0,67	0,89
Tema: *Logic* País : *Germany*	0,63	0,87	0,63	1
Tema : *Logic* Fecha Inicio: *June 1997* Fecha Fin : *October 1997*	0,92	0,92	0,92	1
Sigla : *POLP*	1	0,37	1	1
Tema : *Computer* Fecha :*1997*	0,85	0,90	0,85	1
Sigla : *CP96*	1	0,5	1	1
País : *Singapore*	0,66	1	0,33	1

Sigla: *ESSLLI*	1	0,8	1	1
País : *Germany* AND Fecha: *September 1996*	1	1	0,90	1
Sigla : ICLP97	1	0,25	1	0,33
Sigla : *LICS*	1	0,66	1	1
Promedios :	**87,61 %**	**68,38 %**	**84,30 %**	**90,15 %**

Podemos observar a partir de esta tabla que la recuperación no varía en gran medida si se contempla o no la conferencia asociada, sin embargo la precisión aumenta .



- Al aplicar el proceso de recuperación de información a un conjunto de CFPs, observamos que muchas de las categorías recuperadas eran erróneas y las mismas correspondían al encabezado de *e-mail* del documento CFP. Notamos que la información dentro de los encabezados de *e-mail* interfería en la recuperación, debido a que ciertas sentencias contienen datos que pueden pertenecer a las categorías buscadas, sin que se correspondan con los datos del CFP. Esto provocaba una disminución en la precisión de la recuperación. Debido a esto, y sabiendo que los datos incluidos en los encabezados de *e-mail* no son relevantes en la recuperación de información, decidimos eliminar esta porción del texto del documento CFP a analizar .

- Un gran porcentaje de las conferencias se realiza en EE. UU., y en la mayoría de los CFPs cuando hacen referencia al lugar de realización indican el nombre del estado pero no el país, entonces hubo que considerar los estados Norteamericanos como ayuda para poder recuperar el lugar de realización de tales conferencias.

Al analizar un conjunto de CFPs, encontramos casos de ciudades con el mismo nombre que sin embargo pertenecen a diferentes países. Para resolver esta ambigüedad, el sistema intenta recuperar la ciudad a partir de la existencia del estado o país donde se realiza la conferencia, debido a que en otro caso no sería posible determinar a qué país corresponde la ciudad recuperada.

- La categoría *título* de la conferencia es la categoría que presenta una mayor variedad en su representación, por lo tanto ocasiona una mayor dificultad en su recuperación. El sistema no enfoca un área temática en particular, con lo cual no se puede recuperar el título de una conferencia en base a los temas tratados. Al analizar un conjunto de CFPs, observamos la existencia de un conjunto de palabras que en general se encontraban incluidas en los títulos de las conferencias, este conjunto de palabras nos permite identificar un *posible* título. A partir de éste, analizamos el contexto en el cual se encuentra para determinar si realmente puede llegar a ser un título de una conferencia y luego determinamos el comienzo y fin del mismo. Este caso de recuperación es el que implica un mayor análisis, debido a que es el más inexacto. A su vez el sistema desarrollado provee la facilidad que el conjunto de palabras claves utilizadas en la identificación del título *puede ser modificado externamente*, ya que dicho conjunto no es parte del sistema, sino que se obtiene a partir de un archivo externo.

- Muchos CFPs indican como lugar de desarrollo de la conferencia una universidad. El inconveniente asociado a esto es que algunas universidades tienen nombre de ciudades o estados e incluso hasta se han dado casos que tienen nombre de ciudades que pertenecen a un país diferente al de dicha universidad, de manera tal que tuvimos que distinguir los nombres de las universidades de los nombres de ciudades y estados. Esto se realiza analizando el contexto en el que se encuentran tales categorías al intentar su recuperación.

- En algunos CFPs observamos que el título incluía el nombre de un estado Norteamericano. Como el estado se utiliza como una ayuda para la recuperación de la ciudad y el país, si se da este caso el estado no sería de utilidad para esta recuperación, debido a que la ciudad y el país no se encontrarían junto a él, con lo cual no consideramos este término como un estado.

- Debido a que la sigla de una conferencia está formada en general, por iniciales correspondientes al título de la misma, puede darse el caso en que una sigla corresponda a conferencias diferentes. Esto nos provocaba un problema debido que utilizamos la sigla como clave en la representación de los datos y el manejador de BDs que utilizamos sólo permite claves formadas por un solo campo. Para poder identificar las conferencias entre sí, le agregamos a la clave un dígito que permite esta identificación.

Cuando se intenta incorporar datos a la BDs y la clave ya existe, debemos verificar si los datos a incorporar corresponden a la/s conferencias almacenadas, en cuyo caso se actualizan los datos, o si es una conferencia nueva, donde se incorpora con la misma sigla y un nuevo dígito de identificación. Esto provoca una mayor cantidad de accesos a la BDs .

- Al analizar el conjunto de CFPs observamos que en ciertos casos, la sigla incluía letras minúsculas. Para obtener una mayor recuperación de esta categoría de término, hicimos pruebas para también contemplar este formato. Los resultados obtenidos al incluir esto nos dio un mayor grado de recuperación a expensas de una disminución en el grado de precisión, debido a que recuperaba datos que no correspondían a siglas. Como en un sistema de recuperación de información se intenta mantener un balance entre el grado de recuperación y el grado de precisión, decidimos no incluir este formato debido a que pesaba mas la disminución de la precisión que el aumento de la recuperación.

- Muchos CFPs incluyen la fecha de emisión del mismo y en ciertos casos el sistema recuperaba esta fecha considerándola como fecha de realización de la conferencia. Para distinguir la fecha de emisión del CFP de la fecha de realización de la conferencia, consideramos que esta última debía ser posterior a la fecha actual.

- Con ciertas categorías de términos tales como : *ciudad, estado* y *país* tuvimos el inconveniente que en ciertos casos formaban parte de una dirección de *e-mail* o de un URL, con lo cual esto provocaba confusión con los datos representantes de estas categorías. Para solucionar este inconveniente consideramos como válidos los términos del tipo *ciudad, estado* y *país* que comienzan con mayúscula.

❖ CONCLUSIONES

Al desarrollar el Sistema de Recuperación de Información de Documentos CFPs y el Sistema de Consulta de Información, pudimos alcanzar los objetivos planteados, es decir explorar la utilización de técnicas IR, reflejando su utilidad, tanto en la indexación como en la recuperación de información textual, y que el sistema tenga algún tipo de fin práctico, en este caso la organización de documentos CFPs.

El proceso de recuperación de información desarrollado se basa en un conjunto de reglas que tienen como objetivo recuperar de un documento cierto datos considerados relevantes. Estas reglas no son exactas y están complementadas con técnicas IR, las cuales facilitan tanto la recuperación de información (indexación) como la consulta de datos.

Utilizar la técnica de filtrado de *Stoplist* en la recuperación de información nos permite eliminar del texto los términos no significativos, con lo cual se disminuye el volumen de texto a analizar. De esta manera evitamos analizar términos que no corresponden a las categorías buscadas o que no ayudan en la recuperación de las mismas.Utilizar esta técnica en la consulta nos permite eliminar del pedido del usuario aquellos términos considerados no significativos en la búsqueda de información. Los términos no significativos tienen la característica de ser términos con una alta frecuencia de ocurrencia en los textos, con lo cual no son de utilidad para discriminar información.

Utilizar la técnica de *Stemming* tanto en la recuperación de información como en el procesamiento de *queries,* nos permite relacionar palabras morfológicamente con el objetivo de mejorar la efectividad de la recuperación y en una consulta, podemos abarcar un mayor número de conferencias relacionadas. Utilizando esta técnica, en general se incrementa la recuperación al costo de decrementar la precisión.

Para presentar los resultados obtenidos de la recuperación de información, desarrollamos un sistema de consulta, el cual utiliza una interfase WWW. A través de un *browser* Web se pueden realizar consultas según diferentes criterios acerca de documentos CFPs.

La WWW permite proveer información en forma tal que pueda ser accedida por diferentes usuarios trabajando en distintas plataformas y en lugares distantes. Además permite a los usuarios acceder a información almacenada sin requerir conocimiento acerca de los mecanismos subyacentes de implementación de tales accesos.

Al realizar este sistema de consulta analizamos temas relacionados con la World Wide Web tales como :

➲ El lenguaje HTML para la creación de páginas de hipertexto, basado en la filosofía de independencia de datos, estructura y formato de SGML .Con este lenguaje se define la estructura de los documentos de manera tal que cualquier *browser* pueda entenderla .

➲ La extensión de la capacidad del servidor a través de la programación CGI. CGI es el *gateway* o puerta común que utiliza el servidor para comunicarse con aplicaciones diferentes al *browser* . Un programa CGI actúa como un enlace entre cualquier aplicación que lo necesite y el servidor, siendo este último el responsable de recibir información del *browser* así como enviar de vuelta los datos. La especificación de CGI permite crear nuevos documentos en forma dinámica, lo que nos permite generar información personalizada para el usuario, construyendo en forma dinámica páginas HTML con los resultados de la consulta realizada.

❖ AMPLIACIONES FUTURAS

En el sistema desarrollado, utilizamos el manejador de BDs mSQL versión 1.1, el cual tiene ciertas limitaciones, como por ejemplo que la clave de búsqueda debe ser un campo simple; no se puede tener una clave compuesta por más de un campo.

La clave utilizada en la tabla que contiene la representación de los documentos CFPs es la sigla de la conferencia, sin embargo sería conveniente que la clave estuviera formada por más de un campo.

Otro limitación de esta versión es que no permite utilizar paréntesis en las operaciones booleanas del *query*, lo cual impide alterar la precedencia de los operadores, restringiendo las consultas y en algunos casos ampliando la complejidad de los *queries* .

Actualmente se está desarrollando la versión 2 del mSQL (ya existe la versión beta), la cual presenta muchas mejoras sobre la versión 1.1, incluyendo la posibilidad de tener claves compuestas por más de un campo y la incorporación de paréntesis. Por lo tanto sería beneficioso para el sistema utilizar esta nueva versión del mSQL.

El Sistema de Recuperación de Información de Documentos CFPs desarrollado recupera las categorías básicas de identificación de un documento CFP. Se podría ampliar el sistema con el fin de recuperar nuevas categorías de términos que complementarían la información brindando mayores detalles acerca del documento, teniendo en cuenta de no producir una sobrecarga de información ya que de cualquier manera el usuario siempre puede acceder al documento CFP original y obtener así la información que considera relevante y no se encuentra en el conjunto de datos brindado por el sistema .

El sistema desarrollado tiene como objetivo recuperar información de un tipo de documento especial conocido como CFP, sin embargo la estuctura general del sistema podría aplicarse para el reconocimiento de información sobre otro tipo de documento modificando el conjunto de reglas y el reconocimiento de los significados de los términos, sin que se necesite alterar el proceso de recuperación de información.

Anexo

Glosario

CAPITULO 9

GLOSARIO

Anchor : Area de un documento hipertexto que es el origen de un *link* de hipertexto. Típicamente se identifican por estar subrayados o intensificados.

ANSI : (*American National Standards Institute*) Agrupación que define los estándares para la industria de procesamiento de la información.

Archie : Sistema para indexar contenidos de los servidores FTP. Permite una búsqueda según los nombres de los archivos y su ubicación.

ASCII : (*American Standard Code for Information Interchange*) Código de caracteres de 7 bits que pueden representar 128 caracteres, algunos de los cuales se utilizan como caracteres de control para el control de las comunicaciones .

Browser : Programa cliente de la Web que recibe los datos del servidor Web, los interpreta y los presenta en pantalla.

CERN : (*Centre Europeen pour la Recherche Nucleaire*) Laboratorio Europeo donde se originó la Web en 1989.

CFP : (*Call For Papers*) Invitación a participar en conferencias. Tiene como objetivo difundir el desarrollo de las mismas e invitar a participar en ella mediante la presentación de *papers*.

CGI : (*Common Gateway Interface*) Gateway o puerta común que utiliza el servidor para hacer interfaz, es decir comunicarse, con aplicaciones diferentes al browser. Permite escribir un programa que recibe información de clientes, y en base a esto produce ciertos resultados como una página Web determinada.

DFA : (*Deterministic Finite Automata*) Dispositivo formal que decide si una cadena determinada pertenece o no a un lenguaje establecido. El control es determinista cuando una transición se realiza de un estado a otro exclusivamente.

Documento : Pieza de información que un usuario quiere recuperar. Puede ser un archivo de texto, una página WWW, una imagen o una sentencia de un libro.

E-Mail : (Correo electrónico) Uno de los primeros servicio desarrollados para Internet. Permite el envío de mensajes de una computadora a otra, facilitando la comunicación en forma rápida y a través de grandes distancias .

Feedback : Procedimiento que permite al usuario de un sistema IR, alterar su pedido de información durante una sesión de búsqueda, según la información que se ha recuperado y, de esta manera mejorar la recuperación .

FTP : (*File Transfer Protocol*) Permite la transferencia de archivos bidireccional .

Form : Elemento HTML que permite a los usuarios ingresar información para su procesamiento.

Gateway : Puerta. Dispositivo o conjunto de dispositivos que interconectan dos o mas redes, permitiendo la transferencia de datos entre ellas.

Gopher : Sistema de BDs distribuidas al que se accede mediante un menú.

Hipermedia : Hipertexto que puede incluir multimedia : texto, gráficos, video, audio, etc.

Hipertexto : Texto que no está restringido a una secuencia simple de observación, contiene vínculos (*links*) a otros textos.

Home Page : Página de entrada para acceder a un servidor local o una página que define una persona como su página principal, a menudo conteniendo información personal o profesional.

Host : Computadora conectada a una red. Pueden dividirse en dos clases: servidores que proporcionan recursos, y usuarios que acceden a ellos.

HTML : (*HyperText Mark-Up Language*) Código que permite definir páginas con información estructurada en forma de hipertexto. Es el mecanismo utilizado para crear páginas Web.

HTTP : (*HyperText Transfer Protocol*) Protocolo con que se realizan las comunicaciones entre servidor y cliente. Se utiliza para transferir documentos hipertextos.

Indexar : Proceso de convertir una colección de datos de una manera adecuada para facilitar la búsqueda y recuperación .

Internet : Colección de redes de computadoras distribuidas globalmente que intercambian información a través del protocolo TCP/IP.

IP : (*Internet Protocol*) Protocolo estándar de Internet que define un *datagram* Internet como la unidad de información pasada a través de Internet y provee las bases del servicio de conexión. El protocolo Internet a menudo se lo conoce como TCP/IP, debido a que IP es uno de los protocolos fundamentales.

IP *Datagram* : Unidad básica de información en el transporte a través de Internet. Contiene la dirección de origen y el destino junto con los datos.

IR : (*Information Retrieval*) Estudio de los sistemas para la indexación, búsqueda y recuperación de datos, particularmente de texto u otra forma no estructurada.

IRC : (*Internet Relay Chat*) Servicio que permite que muchas personas "charlen simultáneamente". Permite mantener conferencias en tiempo real .

Link : Conexión entre un documento hipertexto y otro.

Matching : Coincidencia; semejanza. La coincidencia puede ser exacta, es decir dos cadenas de caracteres coincidentes carácter por carácter, o aplicar otra regla de comparación utilizando un *símbolo comodín* que coincida con todo.

MIME : (*Multipurpose Internet Mail Extensions*) Especificación para formatos de documentos multimedia.

Mosaic : Programa cliente (*browser*) gráfico de la WWW, originalmente desarrollado por NCSA (*National Center for Supercomputing Applications*).

mSQL : (*MiniSQL*) Sistema manejador de BDs relacional diseñado para proveer un rápido acceso a datos almacenados con bajo requerimiento de memoria. Ofrece un subconjunto de operaciones SQL como interface de *query*, las cuales concuerdan con la especificación ANSI SQL.

Newsgroup : Grupo de discusión o foro.

Netscape : Programa cliente (*browser*) gráfico de la WWW. Se basó en el programa Mosaic desarrollado por NCSA.

NNTP : (*Network News Transport Protocol*) Protocolo para la distribución de noticias y artículos Usenet.

Página : Archivo simple HTML.

Parsing : Análisis sintáctico de una cadena de caracteres.

Puerto : Los protocolos de transporte utilizan el puerto para distinguir entre múltiples destinos. Los protocolos Internet utilizan números enteros positivos para identificar los puertos. Algunos números de puertos están reservados para servicios estándares (por ejemplo para el correo electrónico).

Precision : Medida standard de la *performance* de un sistema IR. Se define como el número de documentos relevantes recuperados dividido por el número total de documentos recuperados. En una situación ideal, la precision es del 100 % ; esto se alcanzaría recuperando sólo un documento. Un sistema intenta maximizar las medidas *recall* y *precision* simultáneamente.

Protocolo : Conjunto de reglas y mensajes que rigen el diálogo entre dos procesos informáticos .

Query : Sentencia formal que indica un requerimiento de información.

Recall : Medida standard de la *performance* de un sistema IR. Se define como el número de documentos relevantes recuperados dividido por el número total de documentos relevantes en la colección. En una situación ideal, la recuperación es del 100 %, esto se alcanzaría recuperando todos los documentos. Un sistema intenta maximizar las medidas *recall* y *precision* simultáneamente.

Relevancia : Medida abstracta de cómo un documento satisface la necesidad de información del usuario. Idealmente, el sistema debería recuperar todos los documentos relevantes, desafortunadamente esta es una noción subjetiva y difícil de cuantificar.

RFC : (*Request For Comments*) Serie de documentos que describen estándares o proponen nuevos estándares para protocolos y tecnologías de Internet.

Robot : (*Spider*) Programa que atraviesa la Web buscando URLs. Comienza en una página particular de la Web, y accede a todos los *links* que contiene. De esta forma atraviesa el grafo que forma la WWW. Va recordando la información de los servidores para crear índices .

SGML : (*Standard Generalized Mark-up Language*) Estándar para definir lenguajes de marcas; HTML es una instancia de SGML.

Servidor : Aplicación de software que provee información o servicios según los pedidos de programas clientes.

Site : Sección de archivos de una computadora donde residen los documentos Web (u otros documentos servidos por otros protocolos); por ejemplo Web site, Gopher site, FTP site.

SMTP : (*Simple Mail Transfer Protocol*) Protocolo estándar de Internet para transferir mensajes de correo electrónico de una máquina a otra. Especifica como interactuan dos sistemas de *e-mail* y el formato de los mensajes de control que intercambian para transferir los mensajes.

Socket : Mecanismo de comunicación entre procesos que permiten a los mismos "hablar" unos con otros, aun si se encuentran en diferentes máquinas. El proceso de comunicación vía *sockets* se basa en el modelo cliente/servidor. El proceso servidor crea un *socket* cuyo nombre es conocido por los procesos clientes. Estos procesos clientes pueden "hablar" con los procesos servidores vía una conexión a través del *socket*. Las conexiones por medio de *sockets* son bidireccionales.

Stoplist : Lista de palabras que se considera no tienen un valor de índice, es decir que no tienen peso para discriminar información. Se utiliza para eliminar de un texto, términos que no serán relevantes para la representación de un documento.

Stemming : Proceso de remover prefijos y sufijos de palabras en un documento o query. Tiene como objetivo agrupar palabras que tienen el mismo significado conceptual, con lo cual el usuario no necesita ser tan específico en un query.

Stopword : Una palabra tal como preposiciones o artículos que tienen poco contenido semántico. También se refiere a palabras que tienen alta frecuencia de ocurrencia en una colección. Como las *stopwords* aparecen en muchos documentos y por lo tanto no ayudan a la recuperación, entonces usualmente se remueven del documento o del query. Algunos sistemas utilizan una lista predeterminada de *stopwords*. Sin embargo, las *stopwords* dependen del contexto.

SQL : (*Structured Query Language*) Lenguaje estándar para construir queries sobre datos representados por tablas. Las filas son los registros de la tabla y las columnas, los campos.

Tag : Código de formato utilizado para marcar una parte de un elemento HTML .

TCP : (*Transmision Control Protocol*) Protocolo de nivel de transporte estándar de Internet. Provee un servicio de transmisión bidireccional y confiable. Permite que un proceso en una máquina envíe una conjunto de datos a un proceso en otra máquina. Es orientado a la conexión en el sentido que antes de transmitir datos, ambos participantes deben establecer una conexión. El software que implementa TCP usualmente reside en el sistema operativo y utiliza el protocolo IP para transmitir información a través de Internet. El protocolo utilizado en Internet, a menudo se lo conoce como TCP/IP debido a que TCP es uno de los dos protocolos fundamentales.

Telnet : Protocolo que permite compartir información a través de la red utilizando la técnica de emulación de terminal. Permite el "login" remoto en otras máquinas de Internet.

Término : Palabra simple o concepto que ocurre en un documento o un query. También puede hacer referencia a palabras en el texto original.

Thesaurus : Estructura que comprende una lista de términos, donde cada término puede ser una palabra simple o una frase, junto con las relaciones entre los términos. Su objetivo es proveer un vocabulario común, preciso y controlado, el cual asiste en la coordinación de la indexación y recuperación de información. Se diseña para áreas temáticas específicas, en consecuencia es dependiente del dominio.

URL : (*Uniform Resource Locator*) Esquema de direccionamiento en la Web; identifica un recurso de Internet.

Usenet : Sistema de diseminación de discusiones de texto. El espacio de discusión Usenet se divide en *newsgroups*, cada uno de los cuales trata un determinado tema.

Verónica : Servicio que permite localizar archivos. Se pueden incluir textos descriptivos para cada archivo que se publica. Se puede buscar por nombre de archivos, así como también por tema .

WAIS : (*Wide Area Information Servers*) Sistema de búsqueda en BDs que permite encontrar ocurrencias de textos en archivos de Internet.

WWW : (*World Wide Web*) Sistema de comunicación e información de hipertextos, popularmente utilizado en Internet, donde la comunicación de datos se basa en el modelo cliente/servidor. Los clientes Web (*browsers*) pueden acceder a múltiples protocolos y a información hipermedia utilizando un esquema de direccionamiento.

Referencias Bibliográficas

Capítulo 10

REFERENCIAS BIBLIOGRÁFICAS

❒ [BaezaYates-Frakes92] *Ricardo Baeza-Yates ,William B. Frakes.* **Information Retrieval - Data Structures & Algorithms** - 1992 -

❒ [BernersLee95] *Tim Berners Lee.* **Style Guide for online hypertext** - 1995 -

❒ [Blair97] *David C. Blair.* **An evaluation of retrieval effectiveness for a full-text document retrieval system** - 1997 -

❒ [Chakravarthy95] *Anil S. Chakravarthy and Kenneth B. Haase.* **NetSerf : Using Semantic Knowledge to Find Internet Information Archives** - MIT Media Laboratory - SIGIR 1995 -

❒ [Chaléat-Charnay96] *Philippe Chaléat, Daniel Charnay.* **HTML y la Programación de servidores Web** - 1996 -

❒ [Church95] *Kenneth Ward Church .* **One Term or Two ?** - AT & T Bell Laboratories - Murray Hill, USA - SIGIR 1995 -

❒ [Comer88] *Douglas E. Comer.* **Internetworking with TCP/IP** -Principles, protocols and architecture -

❒ [Crocker82] *David H. Crocker.* **Standard for the format of Arpa Internet Text Message** - 1982 -

❒ [Davison95] *Andrew Davison.* **Coding With HTML Forms** - Junio 1995 -

❒ [Davison96] *Andrew Davison.* **Interactive Forms in HTML** - Junio 1996 -

❒ [December-Ginsburg95] *John December-Mark Ginsburg.* **HTML & CGI** - Unleashed - 1995

❒ [Dutt96] *G. Dinesh Dutt.* **CGI and the World Wide Web** - Febrero 1996 -

❒ [Hersh95] *William R. Hersh,Diane L. Elliot, David H. Hickam, Stephanie L. Wolf, Anna Molnar, Christine Leichtenstien.* **Towards New Measures of Information Retrieval Evaluation** - Oregon Health Sciences University, Portland, USA and University of Ulm, Ulm, Germany - SIGIR 1995 -

❒ [Hoch94]*Rainer Hoch.***Using IR Techniques for Text Classification in Document Analysis** - German Research Center for Artificial Intelligence (DFKI) - SIGIR 1994 -.

❒ [Hughes95] *David J. Hughes .* **MiniSQL A Lightweight Database Server Versión 1.1** - 1995 -

❒ [Hughes95] *David J. Hughes.* **MiniSQL A Lightweight Database Engine or Designing a Lean, Mean Data Machine** - Fiddich Technologies - 20 Abril 1995

❒ [Kraaij96] *Wessel Kraaij, Renée Pohlmann.* **Viewing Stemming as Recall Enhancement** - Institute of Applied Physics, Netherlands Organization for Applied Scientific Research (TNO), Deft - The Netherlands - and Research Institute for Language and Speech (OTS/SST), Utrecht University, Utrecht - The Netherlands - SIGIR 1996 -

❒ [Mulhem96] *P. Mulhem and L. Nigay.* **Interactive Information Retrieval System: From User Centered Interface Design to Software Design** - Laboratoire CLIPS - IMAG - Grenoble, France - SIGIR 1996 -

❒ [Nielsen93] *Jakob Nielsen.* **Hypertext and Hypermedia** - 1993 -

❒ [Paice94] *Chris D. Paice.* **An Evaluation Method for Stemming Algorithms** - Department of Computing, Lancaster University , Bailrigg, Lancaster,UK - SIGIR 1994 -

❒ [Samuel96] **The mSQL FAQ** - Mantenido por Peter Samuel - Agosto 1996 -

❒ [Saracenic95] *Tefko Saracenic.* **Evaluation of Evaluation in Information Retrieval** - School of Communication, Information and Library Studies , Rutgers University, New Brunswick , USA - SIGIR 1995 -

❒ [Swoboda96] *Nick Swoboda.* **CGI-BIN Building HTML - Base interfaces** - Marzo 1996 .

❒ [VanRijsbergen79] *C. J. Van Rijsbergen.* **Information Retrieval** -1979 -

❒ [Weinman96] *William E. Weinman.* **The CGI Book** The Complete World Wide Web programming reference -1996 -

❒ [Wilkinson94] *Ross Wilkinson.* **Effective Retrieval of Structured Documents** - Dept. of Computer Science, RMIT, Melbourne , Australia - SIGIR 1994 -