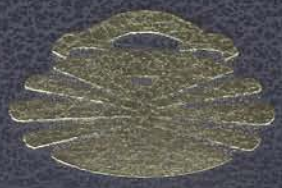

UNIVERSIDADE DA CORUÑA

Departamento de Tecnoloxías da Información
e as Comunicacións

TESIS DOCTORAL

ARQUITECTURA DE DATOS AVANZADA DE UN DIRECTORIO WEB, CON OPTIMIZACIÓN DE CONSULTAS RESTRINGIDAS A UNA ZONA DEL GRAFO DE CATEGORÍAS

FIDEL CACHEDA SEIJO

DIRECTOR: DR. ANGEL VIÑA CASTIÑEIRAS

A Coruña, Julio 2002

UNIVERSIDADE DA CORUÑA

Departamento de Tecnoloxías da Información e as Comunicacións

TESIS DOCTORAL

ARQUITECTURA DE DATOS AVANZADA DE UN DIRECTORIO WEB, CON OPTIMIZACIÓN DE CONSULTAS RESTRINGIDAS A UNA ZONA DEL GRAFO DE CATEGORÍAS

FIDEL CACHEDA SEIJO

DIRECTOR: DR. ANGEL VIÑA CASTIÑEIRAS

A Coruña, Julio 2002

UNIVERSIDADE DA CORUÑA

DEPARTAMENTO DE TECNOLOXÍAS
DA INFORMACIÓN E AS COMUNICACIÓNS

Facultade de Informática
Campus de Elviña, s/n.
15071 A Coruña
Telf. 981 167 000
Fax 981 167 160
E-mail: tic@udc.es

D. Angel Viña Castiñeiras, catedrático de Ingeniería Telemática del Dpto. de Tecnoloxías da Información e as Comunicacións en la Universidade de A Coruña,

CERTIFICA que la memoria titulada "*Arquitectura de Datos Avanzada de un Directorio Web, con Optimización de Consultas Restringidas a una Zona del Grafo de Categorías*", ha sido realizada por D. Fidel Cacheda Seijo bajo mi dirección en el Departamento de Tecnoloxías da Información e as Comunicacións de la Universidade de A Coruña y concluye la tesis que presenta para optar al grado de Doctor en Informática.

A Coruña, 19 de Noviembre de 2001

Universidade da Coruña · DEPARTAMENTO DE TECNOLOXÍAS DA INFORMACIÓN E AS COMUNICACIÓNS

V°B°: D. Juan Ares Casal
Director del Dpto. de Tecnoloxías da Información e as Comunicacións

Fdo.: Dr. Ángel Viña Castiñeiras
Director de la Tesis Doctoral

AGRADECIMIENTOS

Esta tesis no podría haber sido escrita sin el apoyo de otra gente. Quisiera expresar mi agradecimiento por esta ayuda, en particular a las siguientes personas.

En primer lugar, agradecer a mi director de tesis, D. Angel Viña, la oportunidad que me ha brindado al introducirme en el innovador mundo de la recuperación de información en el Web. Su experiencia y comentarios acerca del enfoque al afrontar una disertación de este tipo me han guiado enormemente.

Quisiera también agradecer al personal docente e investigador del Departamento de Tecnoloxías da Información e as Comunicacións de la Universidade de A Coruña las facilidades que siempre me han dispensado, y en mayor medida, al grupo del Área de Ingeniería Telemática que dirige el profesor Viña, por el acogedor y estimulante ambiente de trabajo que han creado. A Victor Carneiro y Carmen Guerrero por haber soportado estoicamente mis comentarios y por siempre tener a punto alguna sugerencia.

También agradecer por su gran ayuda para la recopilación de información, al servicio de bibliotecas de la Universidade de A Coruña en general, y especialmente al servicio de préstamos interbibliotecario por resolver siempre de manera ágil y rápida todas y cada una de mis peticiones.

Al Departamento de Matemáticas de la Universidade de A Coruña por tener siempre un momento para atender mis consultas y por sus oportunos comentarios.

Finalmente quisiera agradecer a mi familia, amigos y especialmente a Nahir, sin cuyo continuo apoyo y paciencia no habría sido posible complementar este trabajo.

RESUMEN

Arquitectura de Datos Avanzada de un Directorio Web,

con Optimización de Consultas Restringidas

a una Zona del Grafo de Categorías

Fidel Cacheda Seijo

Desde su origen, el World Wide Web ha sufrido un crecimiento exponencial que ha generado un gran volumen de información heterogénea accesible para cualquier usuario. Esto ha llevado a la utilización de herramientas eficientes para gestionar, recuperar y filtrar dicha información. En concreto, los directorios Web son taxonomías que clasifican documentos Web, sobre los que posteriormente se realizarán consultas. Este tipo de sistemas de recuperación de información presenta un tipo específico de búsquedas, en donde la colección de documentos está restringida a una zona del grafo de categorías. Esta disertación presenta una arquitectura de datos específica para directorios Web que permite mejorar el rendimiento ante búsquedas restringidas. Dicha arquitectura se basa en una estructura de datos híbrida, constituida por un fichero invertido conteniendo embebido múltiples ficheros de firmas. En base al modelo propuesto se definen dos variantes: la arquitectura híbrida con información total y la arquitectura híbrida con información parcial. La validez de esta arquitectura ha sido analizada mediante el desarrollo de ambas variantes para su comparación con un modelo básico, demostrando una clara mejoría en el rendimiento de las consultas restringidas, destacando especialmente el modelo híbrido con información parcial al responder adecuadamente bajo cualquier carga del sistema de búsqueda. A nivel general, la arquitectura propuesta se caracteriza por su facilidad de implementación, derivada de las estructuras de datos empleadas, su flexibilidad respecto al crecimiento del sistema y especialmente, por el buen rendimiento ofrecido ante búsquedas restringidas.

ABSTRACT

Web Directory Advanced Data Architecture,

with Optimisation of Restricted Searches

to an Area of the Category Graph

Fidel Cacheda Seijo

The World Wide Web has undergone an exponential increase since it was created which has created a great volume of heterogeneous information accessible to every user. This has lead to the use of efficient tools in order to manage, retrieve and filter that information. Specifically, Web directories are taxonomies for the classification of Web documents to be later consulted. This kind of Information Retrieval systems present a specific type of search where the document collection is restricted to one area of the category graph. This thesis introduces a specific data architecture for Web directories which improves the performance of restricted searches. That architecture is based on a hybrid data structure composed of an inverted file with multiple embedded signature files. Two variants based on the proposed model are presented: hybrid architecture with total information and hybrid architecture with partial information. The validity of this architecture has been analysed by means of developing both variants to be compared with a basic model. The performance of the restricted queries was clearly improved, specially the hybrid model with partial information, which yielded a positive response under any load of the search system. In a general meaning, the proposed architecture proves an easy implementation, due to the data structures used, is particularly flexible with regard to the growth of the system, and specially, shows a high performance for the restricted searches.

CONTENIDOS

PREFACIO

Desde su origen, a principios de los años 90, el World Wide Web ha sufrido un crecimiento exponencial en número de servidores Web y de usuarios, que nadie podría haber imaginado. En este medio se combinan una gran cantidad de información textual, junto con otro tipo de medios como imágenes, audio, vídeo, que convierten al Web en una gran y ubicua base de datos carente de estructura. Esto implica la necesidad de herramientas eficientes que permitan gestionar, recuperar y filtrar la información de esta base de datos.

Por este motivo, en los años posteriores a la explosión inicial del Web fueron surgiendo diversos sistemas de recuperación de información especialmente adaptados para este nuevo entorno. En general, se distinguen tres sistemas de búsqueda en el Web. El primero de ellos son los robots o motores de búsqueda que indexan una porción importante de los documentos disponibles en el Web como una base de datos de texto. El segundo se trata de los directorios Web, que clasifican aquellos documentos Web más relevantes en base a su temática. Y por último se encuentran los metabuscadores, que envían una consulta a otros servicios de búsqueda para realizar un procesamiento posterior de los resultados.

Entre estos sistemas, los directorios Web son taxonomías que clasifican documentos Web. Se caracterizan porque disponen de un grafo dirigido acíclico de categorías en base a la cual se catalogan y asocian documentos. Obviamente, el número de documentos catalogados es reducido (respecto a la totalidad de documentos disponibles en el Web), ya que prima la calidad de los documentos indexados frente a la cantidad, no como el caso de los motores de búsqueda.

Al igual que en el resto de sistemas de búsqueda en el Web, los directorios se basan en la realización de búsquedas entre los documentos indexados, a partir de una serie de palabras clave indicadas por el usuario. Para tales efectos, la indexación basada en la técnica de

ficheros invertidos es la que muestra un mejor rendimiento, en cuanto a tiempos de respuesta y espacio de almacenamiento. De hecho, hoy en día, la técnica de ficheros invertidos constituye el método de indexación empleado en la gran mayoría de los sistemas comerciales de búsqueda, tanto en entornos Web como en sistemas tradicionales.

Sin embargo, los directorios Web se caracterizan por disponer de un tipo especial de búsquedas, en donde el conjunto de documentos buscados está restringido a aquellos asociados a una zona concreta del grafo de categorías, marcada por su nodo raíz. Este tipo de búsquedas constituye un valor añadido respecto al resto de sistemas de búsqueda, al permitir restringir la consulta a un conjunto de documentos relacionados con una temática definida y de calidad probada. Ahora bien, la técnica de ficheros invertidos no permite una resolución eficiente de este tipo de consultas, ya que requiere la realización de un proceso de filtrado posterior al proceso de búsqueda estándar. Este filtrado hace que el rendimiento no sea adecuado, especialmente en los casos en que el número de elementos incluidos en las listas invertidas es elevado.

En esta disertación se introduce una arquitectura de datos específica para directorios Web, que permite mejorar el rendimiento ofrecido por las búsquedas restringidas a una zona del grafo de categorías. Dicha arquitectura se basa en la indexación por medio de la técnica de ficheros de firmas de las categorías asociadas con cada documento. De esta manera se define una estructura de datos híbrida, constituida por un fichero invertido en donde cada una de las listas invertidas contiene embebido su propio fichero de firmas. Para ello se ha definido un identificador de documento compuesto (que contiene las firmas de todas las categorías asociadas al documento), de tal manera que por medio del propio proceso de búsqueda se obtiene dinámicamente el fichero de firmas asociado a cada consulta.

A través de este fichero de firmas se realiza un primer filtrado inexacto que permite eliminar la mayoría de los documentos que no se clasificarán (esto es, que no pertenecen a la zona del grafo sobre la que se ha restringido la búsqueda), para después aplicar de manera eficiente el filtrado exacto.

Para una correcta implantación de la técnica de ficheros de firmas, es necesario el estudio del método de generación de firmas mediante códigos de superposición para representar las categorías a las que se encuentra asociado un documento, teniendo en cuenta la compleja estructura del grafo de un directorio Web.

En base a la arquitectura propuesta se definen dos variantes, el modelo híbrido con información total y el modelo híbrido con información parcial. En la primera de las variantes, todas y cada una de las categorías que constituyen el grafo de categorías disponen de una firma que las identifica, y por lo tanto, son indexadas según la técnica de los ficheros de firmas para ser asociadas con los documentos. En cambio, la variante con información parcial únicamente asigna firmas a aquellas categorías incluidas en los primeros niveles del grafo, lo que implica una reducción en el espacio de almacenamiento requerido y que las categorías de los niveles inferiores no serán identificables en el proceso de filtrado.

Ambas variantes de la arquitectura híbrida propuesta han sido implementadas, junto con un modelo básico de directorio Web, con el objetivo de validar el rendimiento ofrecido en cada caso ante las búsquedas restringidas a una zona del grafo de categorías. Para la correcta evaluación del rendimiento, se ha desarrollado una herramienta de simulación de

usuarios de un sistema de búsqueda en el Web, denominada USim, capaz de reproducir diversas situaciones de carga y que garantiza una evaluación objetiva y completa de estos sistemas. Esta herramienta está basada en las conclusiones derivadas de un estudio de los accesos realizados por usuarios reales sobre un directorio Web, que ha permitido ajustar a modelos matemáticos el comportamiento del conjunto de usuarios frente a un sistema de búsqueda en el Web.

Las implementaciones de las variantes de la arquitectura híbrida propuesta demuestran una clara mejoría en el rendimiento de las consultas restringidas a una zona del grafo de categorías. Sin embargo, destaca el modelo híbrido con información parcial al mantener dicha mejora bajo cualquier carga del sistema, mientras que el modelo con información total presenta una caída del rendimiento ante situaciones de carga elevada. Naturalmente, los modelos propuestos en ningún momento implican una penalización en el resto de procesos recuperación de información asociados con un directorio Web, típicamente procesos de búsqueda o de navegación a través del grafo de categorías.

Esta memoria está estructurada de la siguiente forma:

Capítulo 1: Realiza una descripción pormenorizada de las técnicas asociadas con la indexación de texto, con especial énfasis en las técnicas de fichero invertido y fichero de firmas, y las variantes y optimizaciones asociadas. Asimismo, se exponen los principales sistemas de búsqueda en el Web, describiendo las técnicas de indexación empleadas en cada caso, con una atención especial en el caso de los directorios Web.

Capítulo 2: En este capítulo se realiza un estudio estadístico de los accesos realizados por usuarios reales sobre un directorio Web. Este análisis constituye la base para el diseño e implementación de una herramienta de simulación del comportamiento de los usuarios ante este tipo de sistemas de búsqueda, al mismo tiempo que muestra aspectos fundamentales relacionados con el tipo de búsquedas tratado.

Capítulo 3: Presenta una breve introducción a la evaluación de sistemas de recuperación de información, para a continuación tratar el diseño, implementación y operación de USim, una herramienta de simulación de usuarios, que permite evaluar de manera controlada múltiples sistemas de búsqueda simulando un entorno de operación real.

Capítulo 4: En este capítulo se expone la arquitectura híbrida propuesta, tanto en la variante con información total como la variante con información parcial. Se describe en detalle la estructura de datos híbrida, constituida por un fichero invertido que contiene embebidos múltiples ficheros de firmas. Igualmente, se analiza la técnica de los códigos de superposición, asociada a los ficheros de firmas, que demuestra ser lo suficientemente flexible como para ser adaptada al entorno de un grafo dirigido acíclico de categorías de manera eficiente.

Capítulo 5: Describe los detalles de implementación de la arquitectura propuesta, en sus dos variantes, frente al modelo básico. Por una parte se demuestra la flexibilidad de los modelos híbridos ante el crecimiento del sistema, tanto en número de documentos como de categorías. Y por otra parte, se destaca la mejora del rendimiento obtenido según los modelos propuestos, siendo analizado bajo diferentes situaciones de carga, empleando la herramienta de simulación previamente expuesta.

Capítulo 6: Se presenta un sumario de conclusiones y posibles trabajos futuros sobre la base de la arquitectura de datos desarrollada.

También se incorporan diversos apéndices con información detallada de los análisis estadísticos realizados, así como una guía de usuario para aquellas personas que deseen utilizar la herramienta de simulación desarrollada, próximamente de libre distribución para la comunidad científica internacional.

Contribuciones

Esta tesis presenta una arquitectura de datos híbrida de fichero invertido y ficheros de firmas, con las variantes de información total e información parcial, con el objetivo de mejorar el rendimiento ofrecido por los directorios Web ante consultas restringidas a una zona del grafo de categorías. Concretamente, esta tesis incluye las siguientes contribuciones:

- Análisis de las principales técnicas de indexación y su aplicación directa en los sistemas de búsqueda en el Web. En este estudio se definen las principales estructuras de datos y bloques que constituyen el modelo de un directorio Web, localizándose los problemas para la resolución eficiente del tipo de consultas planteado.
- Análisis de los accesos realizados a un directorio Web, a través del cual se examina el comportamiento de los usuarios de los sistemas de recuperación de información en el Web, frente al caso de los usuarios de sistemas tradicionales. De esta manera, se obtienen las características diferenciadoras (respecto a las búsquedas y demás peticiones realizadas), claves para la definición y optimización de la arquitectura propuesta. Además, por otra parte, se demuestra que dicho comportamiento se ajusta a un modelo matemático, lo que constituye la base para la definición de USim, una herramienta para la simulación de usuarios en este entorno. Esta herramienta constituye la base para una correcta evaluación del rendimiento de sistemas de búsqueda en el Web, y en concreto, para la evaluación de los diferentes modelos propuestos.
- Definición de los ficheros de firmas como una técnica válida para representar la información de un grafo dirigido acíclico de categorías y los documentos asociados. La técnica de los códigos de superposición demuestra gran flexibilidad a la hora de soportar la compleja estructura de un grafo categorías, al garantizar un rendimiento independiente del número de categorías y documentos del sistema, mientras que los parámetros influyentes se caracterizan por ser valores estables.
- Diseño de una arquitectura de datos híbrida constituida por la técnica de fichero invertido, conteniendo embebido cada lista invertida un fichero de firmas que representa el conjunto de categorías asociadas a cada documento. Sobre dicha arquitectura se definen la variante con información total y la variante con información parcial. Esta última constituye una optimización derivada del comportamiento de los usuarios y que mejora el rendimiento del sistema, tanto en espacio de almacenamiento como en tiempos de respuesta.
- Implementación de la arquitectura propuesta, en sus dos variantes, lo que permite determinar la flexibilidad ofrecida por la arquitectura en general, respecto al número de documentos y categorías aceptados por el sistema. Asimismo, el modelo con

información parcial se distingue por requerir un menor aumento de espacio de almacenamiento, frente al modelo con información total.

- Evaluación del rendimiento de los modelos híbridos propuestos, frente a un modelo básico, ante consultas restringidas a una zona del grafo de categorías. El modelo híbrido con información parcial destaca por la significativa mejora aportada sobre el modelo básico en todas las situaciones de carga del sistema de búsqueda, mientras que la variante con información total presenta una pérdida de eficiencia en situaciones de carga elevada.

Asimismo, la presente tesis doctoral constituye la base del proyecto de la Comisión Interministerial de Ciencia y Tecnología (CICYT) titulado "*Arquitecturas Distribuidas para Técnicas de Búsqueda en Internet*", con referencia TIC2001-0547. Este proyecto se centra en las tecnologías de recuperación de información en Internet. En una primera fase, se analizan los directorios Web y sus posibles mejoras en el rendimiento, para a continuación centrarse en los robots o motores de búsqueda, con especial énfasis en diferentes modelos de arquitecturas distribuidas.

1. SISTEMAS DE RECUPERACIÓN DE INFORMACIÓN

La recuperación de información (IR) estudia la representación, almacenamiento, organización y acceso a los elementos de información. La representación y organización de los elementos de información debe proporcionar un fácil acceso al usuario a la información en la que está interesado [Baeza-Yates, 99d].

El usuario, por su parte, debe traducir su necesidad de información en una consulta que pueda ser procesada por el motor de búsqueda o el sistema de información. Comúnmente, esta transformación produce una serie de palabras clave que resumen la descripción de la necesidad de información del usuario.

En base a la consulta del usuario, el objetivo primordial de un sistema de IR es recuperar información que puede ser útil o relevante para el usuario. Es importante la diferencia existente entre recuperar información y recuperar datos.

La recuperación de datos, en el contexto de un sistema de IR, consiste básicamente en determinar que documentos de una colección contienen las palabras clave insertadas por el usuario, lo cual no es suficiente para satisfacer la necesidad de información del usuario. De hecho, un usuario de un sistema de IR está interesado en obtener información sobre un tema determinado.

En un sistema de recuperación de información se obtienen objetos que pueden ser poco precisos o con pequeños errores, que no serán detectados por los usuarios. La principal razón para esto es el hecho de que la recuperación de información debe tratar texto en lenguaje natural que no siempre está bien estructurado y puede ser semánticamente ambiguo. Por otra parte, un sistema de recuperación de datos (como por ejemplo, una base de datos relacional) gestiona datos que presentan una estructura y una semántica bien definidas.

Un sistema de recuperación de información debe, de alguna manera, interpretar el contenido de los elementos de información o documentos de la colección y ordenarlos de acuerdo con el grado de relevancia para la consulta del usuario. La dificultad no está sólo en conocer como extraer esta información sino también en conocer como utilizarla para decidir la relevancia de cada documento. En consecuencia, la noción de relevancia es un concepto clave en la recuperación de información. De hecho, el objetivo primordial de un sistema de IR es recuperar todos los documentos que son relevantes para una consulta, mientras que se recuperan tan pocos documentos no relevantes como sea posible.

1.1. Evolución histórica

Durante aproximadamente 4.000 años el ser humano ha organizado información para más tarde recuperarla y utilizarla. Un ejemplo típico es el índice de contenidos de un libro. En el momento que el volumen de información crece más allá de unos cuantos libros, se hace necesario construir una estructura especializada para asegurar un acceso más rápido a la información almacenada. Una antigua y popular estructura para este propósito es una colección de palabras o conceptos seleccionados con punteros asociados a los documentos (o información) relacionada: el *índice*. De una forma u otra, los índices son el núcleo de los sistemas de recuperación de información modernos.

El proceso de indexado, conocido originalmente como catalogación, es la técnica más antigua para la identificación del contenido de elementos para facilitar su recuperación. En la época de los egipcios, en Babilonia, las bibliotecas ordenaban por tema las tablas cuneiformes [Hyman, 89]. Prácticamente hasta el siglo XIX no hubo cambios importantes en la catalogación, únicamente cambios en los métodos empleados para representar la información básica [Norris, 69]. A finales del sigo XIX los temas de indexación se convirtieron en jerárquicos. En 1963 la Librería del Congreso estadounidense inició un estudio para la automatización mediante ordenadores del archivo bibliográfico. Desde 1966 hasta 1968 la Librería del Congreso trabajó con su proyecto piloto MARC 1. MARC (Machine Readable Cataloging) estandariza la estructura, contenidos y codificación de los registros bibliográficos. El sistema se hizo operativo en 1969 [Avram, 75].

El primer sistema comercial de catalogación es DIALOG, desarrollado por Lockheed Corporation en 1965 para la NASA. Se comercializó en 1978 con tres archivos gubernamentales de índices de publicaciones técnicas. En 1988, cuando fue vendido a Knight-Ridder, DIALOG contenía unos 320 índices usados por más de 91.000 suscriptores de 86 países [Harper, 80].

La introducción de los computadores para asistir en la catalogación no cambió el modo de operación básico de los catalogadores o indexadores humanos encargados de determinar los términos adecuados para cada elemento a indexar. La estandarización de las estructuras de datos, como por ejemplo las empleadas en MARC, permitió compartir índices entre distintas entidades y redujo la sobrecarga manual para el mantenimiento del catálogo. Sin embargo, el proceso todavía requería que el catalogador insertara términos para su localización posterior. El usuario, pasó de realizar una búsqueda física entre las tarjetas del catálogo a disponer de una búsqueda basada en computadores y que mostraba el resultado en una pantalla por medios equivalentes a las tarjetas originales.

En los años 80, la significativa reducción del coste de capacidad de procesamiento y la memoria en los ordenadores permitió el acceso al texto completo de un elemento, frente a los términos manualmente especificados [Kowalski, 00]. Este hecho permitió cambiar radicalmente la forma de indexar y buscar: no es necesario insertar manualmente términos para ser indexados, y al usuario se le muestran aquellos elementos que mayor relevancia presentan con la búsqueda realizada.

Respecto al proceso de recuperación de información, es importante distinguir las dos vistas del problema de la recuperación de información: la perspectiva humana y la perspectiva del computador. Desde el punto de vista centrado en el computador, el problema de la recuperación de información consiste en construir índices eficientes, procesar las consultas de los usuarios con un rendimiento adecuado y diseñar algoritmos de ordenación que mejoren la calidad del conjunto de respuestas. Desde la perspectiva humana, el problema de la recuperación de información consiste en estudiar el comportamiento de los usuarios, comprender sus necesidades principales y determinar los efectos de esas necesidades sobre la organización y operación de los sistemas de recuperación.

En una historia más reciente, las bibliotecas han sido las primeras instituciones que han adoptado sistemas de IR para recuperar información. En la primera generación, estos sistemas consistían básicamente en una automatización de tecnologías previas (por ejemplo, catálogos de tarjetas), permitiendo únicamente búsquedas sobre autores y títulos de las obras. En la segunda generación se aumentaron las funcionalidades de búsqueda, permitiendo consultas sobre temas, palabras clave y otras características más complejas. En la actualidad se está desarrollando la tercera generación centrándose en aspectos de interfaz gráfica, características de hipertexto y en arquitecturas de sistemas abiertas.

En cambio, a principios de los años 90 un hecho cambió la percepción de la recuperación de información: la introducción del World Wide Web. El Web se está convirtiendo en un repositorio universal de conocimiento humano y cultural que ha permitido compartir ideas e información en una escala nunca antes conocida. Su éxito está basado en la concepción de una interfaz de usuario estándar que siempre opera de la misma forma, sin importar el entorno computacional utilizado. Como consecuencia, al usuario se ocultan detalles como los protocolos de comunicación, la ubicación de las máquinas y los sistemas operativos. Es más, cualquier usuario puede crear sus propios documentos Web y enlazarlos con otros documentos Web sin ningún tipo de restricciones. Esto constituye un aspecto clave ya que convierte la Red en nuevo medio de publicación accesible a y para todo el mundo [Agosti, 01].

Respecto a la recuperación de información, se puede considerar que ha habido tres cambios fundamentales debido a las mejoras tecnológicas y el boom del Web. En primer lugar, se ha reducido drásticamente el coste por tener acceso a varias fuentes de información, lo que permite llegar a un volumen de audiencia no posible anteriormente. Segundo, los avances en todos los tipos de comunicación digital han proporcionado un mayor acceso a las redes, lo que implica que la fuente de información puede estar disponible incluso si está físicamente ubicada a grandes distancias, y con tiempos de respuesta reducidos. Y en tercer lugar, la libertad para publicar cualquier tipo de información que cualquiera juzgue interesante ha contribuido a la popularidad del Web. Por primera vez en la historia, muchas personas tienen acceso gratuito a un inmenso medio de publicación.

Fundamentalmente, el bajo coste, gran capacidad de acceso y libertad de publicación ha permitido y animado a la gente a utilizar el Web como un medio altamente interactivo para compartir información.

A pesar de las características y el éxito del Web, también la Red ha introducido nuevos problemas: encontrar información útil en el WWW es normalmente una tarea tediosa y difícil. Por ejemplo, para satisfacer una necesidad de información el usuario debe navegar a través del espacio de hiperenlaces buscando la información de interés. Sin embargo, debido a que el espacio formado por los hiperenlaces es vasto y poco conocido, esta navegación es comúnmente ineficiente. En realidad, el principal obstáculo parte de la ausencia de un modelo de datos bien definido para el Web, lo que implica que la definición y estructura de la información es de baja calidad.

Estas dificultades han atraído y renovado el interés en la recuperación de información y sus técnicas como soluciones prometedoras, por lo que la IR se ha convertido en un aspecto tecnológico clave para el futuro desarrollo y evolución del World Wide Web.

1.2. Conceptos básicos

La adecuada recuperación de información está directamente asociada con la tarea del usuario y la vista lógica de los documentos del sistema de recuperación.

Por una parte, el usuario de un sistema de recuperación de información debe trasladar su necesidad de información en una consulta en el lenguaje adecuado para el sistema. En el caso de un sistema de recuperación de información esto normalmente consiste en especificar un conjunto de palabras que representan la semántica de la necesidad de información. En este caso se considera que el usuario está realizando una tarea de *recuperación*. En cambio, si se considera el ejemplo de un usuario que tiene interés en un tema poco definido (por ejemplo, baloncesto) sobre el que realiza una búsqueda para a continuación simplemente ojear los documentos obtenidos, puede consultar un documento sobre "Los Ángeles Lakers" y de ahí seguir a otros documentos sobre "Los Ángeles" y de ahí a documentos sobre "Turismo en California". En esta situación se considera que el usuario está *navegando* sobre los documentos de la colección, no buscando. El proceso de navegación también es considerado un proceso de recuperación de información, en el cual los objetivos no están claramente definidos en el inicio y cuyo propósito final puede sufrir cambios durante la interacción con el sistema [Baeza-Yates, 99d].

Los sistemas de recuperación de información clásicos permiten únicamente realizar la tarea de recuperación, mientras que los sistemas basados en hipertexto están especialmente preparados para una rápida y ágil navegación sobre los documentos. De hecho, las bibliotecas digitales modernas, y especialmente los sistemas de búsqueda en el Web están especialmente orientadas hacia la combinación de ambas tareas con el objetivo de mejorar las capacidades de recuperación.

Tanto las tareas de recuperación como de navegación se consideran, en la jerga del World Wide Web, acciones "*pull*" ya que el usuario solicita esa información de manera interactiva. En contraste, se encuentra las acciones "*push*", que realizan la recuperación de

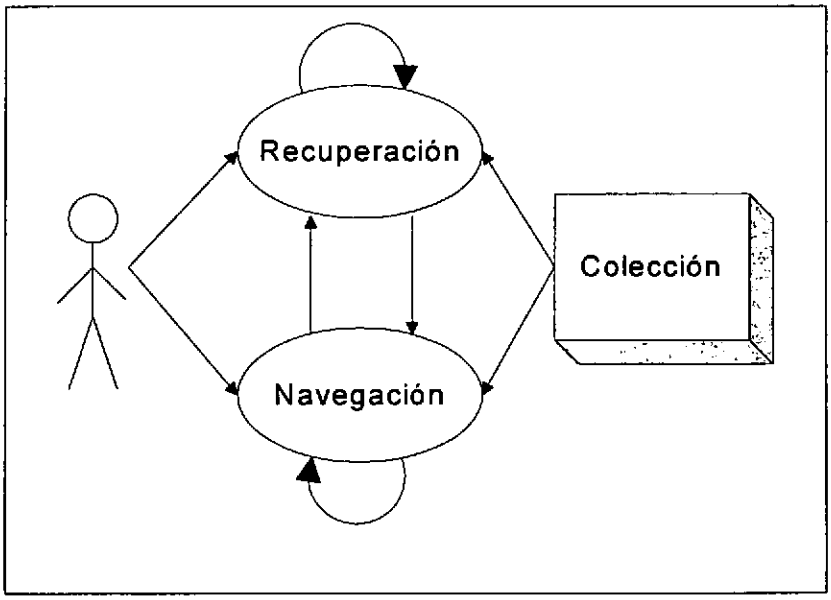

Figura 1-1: Interacción del usuario con el sistema de recuperación

información de manera automática utilizando agentes software que envían la información directamente hacia el usuario.

Respecto a la vista lógica de los documentos de una colección, frecuentemente es representada por medio del conjunto de términos indexados, también denominadas palabras clave. Estas palabras clave pueden ser extraídas directamente del texto de los documentos o bien, pueden haber sido especificadas por expertos en la materia. En ambos casos, sin importar su origen, estas palabras clave representan la vista lógica de los documentos.

La tecnología actual hace posible que los documentos puedan ser representados con todo el conjunto de palabras que los conforman. En este caso, se considera que el sistema de recuperación adopta una vista lógica de texto completo. Sin embargo, sobre esta base puede ser necesario reducir el número palabras clave consideradas, principalmente debido al elevado número de documentos disponibles (como por ejemplo, en el World Wide Web). En general se utilizan técnicas de eliminación de palabras comunes, de obtención de raíces gramaticales o de identificación de grupos de nombres. Estas operaciones reciben el nombre de operaciones de texto, provocando la reducción de la complejidad de la representación de los documentos y convirtiendo la vista lógica de texto completo en un conjunto de términos indexados.

1.3. El proceso de recuperación de información

En esta sección se describe el proceso de recuperación de información que se realiza en un sistema de búsqueda genérico. En la Figura 1-2 se describe la arquitectura global de un sistema de este tipo en base a lo expuesto en [Baeza-Yates, 99d].

A nivel general, el proceso de recuperación se inicia con la definición de la base de datos de texto. Para ello es necesario especificar:

a) Los documentos que formarán parte de la colección.

b) Las operaciones que se realizarán sobre el texto de los documentos. Estas operaciones serán las que transformarán los documentos originales y generarán la vista lógica de los mismos.

c) El modelo de texto, esto es, la estructura del texto y los elementos a recuperar.

Una vez que se ha definido la vista lógica de los documentos, se pasa al proceso de indexado que se encargará de construir el índice asociado al texto. El índice es una estructura de datos crítica ya que permitirá el acceso eficiente a grandes volúmenes de datos. Es posible la utilización de múltiples estructuras de datos, sin embargo, la más frecuentemente empleada hoy en día es el fichero invertido. Los recursos empleados por el índice (típicamente medidos en tiempo y espacio de almacenamiento) durante la generación y construcción del índice son recuperados a través del proceso de búsqueda.

A partir de la indexación de la base de datos de documentos, el proceso de recuperación se inicia. Para ello, el usuario especificará una necesidad de información que será analizada y transformada siguiendo las mismas operaciones realizadas sobre el texto de los documentos. A continuación, las operaciones de consulta (operaciones lógicas o de conjuntos) pueden ser aplicadas antes de obtener la consulta real, que representa la necesidad de información original del usuario. Seguidamente, la consulta es procesada para obtener los documentos recuperados de la colección. Este procesamiento debe ser realizado con unos tiempos de respuesta mínimos, al utilizar como base la estructura del índice construida previamente. Antes de ser enviados los documentos recuperados al usuario, son

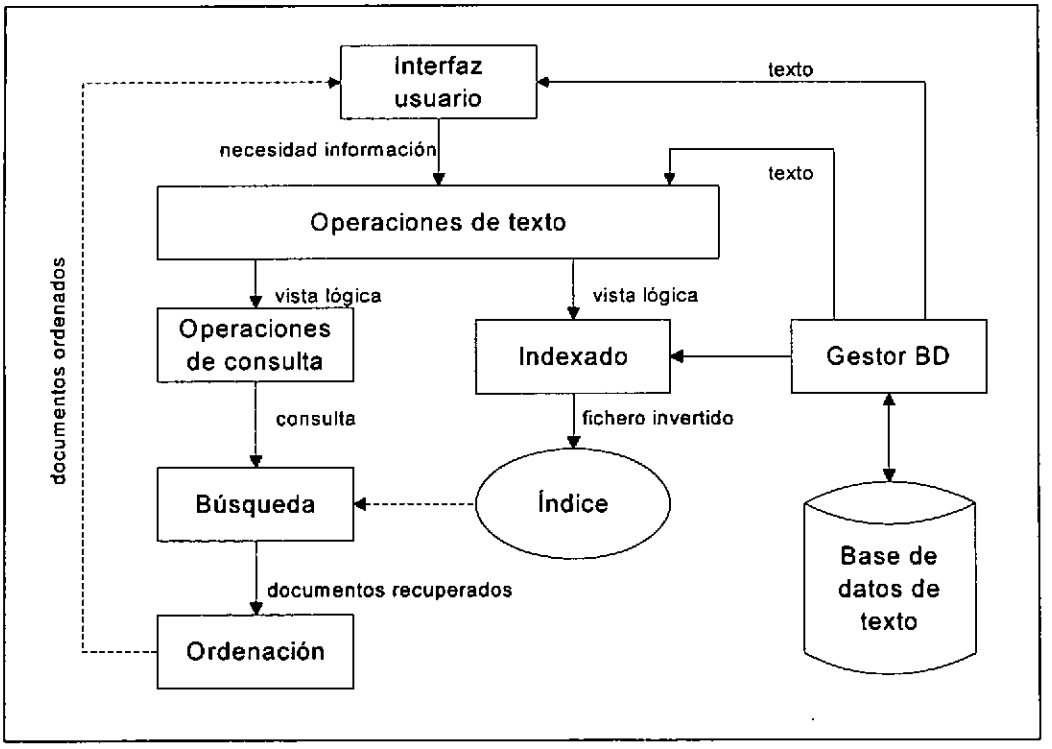

Figura 1-2: Arquitectura de un sistema de recuperación de información

ordenados de acuerdo a un criterio de relevancia basado en la consulta original del usuario.

Por si solo, cada uno de los componentes de un sistema de recuperación, constituyen una parte fundamental en el conjunto del sistema cuyo diseño e implementación pueden afectar drásticamente al rendimiento ofrecido por el sistema, tanto desde un punto de vista de la calidad de las búsquedas como desde la perspectiva de los tiempos de respuesta. Sin embargo, el núcleo de todos los sistemas de recuperación de información está en el índice, ya que es esta estructura de datos la que representa el conjunto de documentos contenidos en la colección y permite la realización de búsquedas sobre los mismos de manera eficiente.

En consecuencia, en la siguiente sección se examinan en detalle las técnicas de indexación, ya que constituyen la base del trabajo de tesis realizado.

1.4. Técnicas de indexación

A continuación se muestran las principales técnicas de indexación tradicionales. Esta sección es especialmente importante ya que el conocimiento de estas técnicas es clave para la realización y compresión de nuevos desarrollos, ya que la mayor parte se basan en variaciones y/o extensiones de las técnicas aquí presentadas.

En concreto, se describen principalmente las técnicas de ficheros invertidos y ficheros de firmas. Con todo, se considera relevante describir como primer método la inspección de texto completo, que aunque no se trate propiamente de un método de indexación, ha sido utilizado para la recuperación de información directamente a partir de documentos, por lo que se puede considerar como el método más básico y sencillo de indexación.

1.4.1. Inspección de texto completo

La forma más directa y simple de localizar los documentos que satisfacen una determinada consulta, materializada en una serie de términos de búsqueda, es la búsqueda directa en todos los documentos de esos términos. En caso de que la consulta englobe varios términos o cadenas de caracteres unidos mediante operadores de lógica booleana, se necesita un paso adicional para determinar si los documentos encontrados satisfacen la expresión booleana [Faloutsos, 95].

El estado del arte respecto a este método de localización de información se centra básicamente en la localización de cadenas de caracteres en documentos. Un algoritmo elemental para tal efecto es el siguiente:

- Comparar los caracteres de la cadena de búsqueda con los correspondientes caracteres del documento.
- Si ocurre un error, desplazar la cadena de búsqueda una posición a la derecha dentro del documento y continuar hasta que se localiza la cadena o se alcanza el final del documento.

La simplicidad de este algoritmo ofrece un pobre rendimiento. Considerando *m* el número de caracteres de la cadena de búsqueda y *n* el número de caracteres del documento, la complejidad del algoritmo tiende a *O(m*n)*.

Knuth et al. exponen en [Knuth, 77] un algoritmo que requiere *O(m+n)* comparaciones. La idea básica de este algoritmo consiste en desplazar la cadena de búsqueda en más de un carácter cuando se predice un error. Se basa en un preprocesamiento de la cadena de búsqueda, cuyo tiempo de ejecución tiende a *O(m)*. Sin embargo, el algoritmo que proporciona un mejor rendimiento es el propuesto por Boyer y Moore en [Boyer, 77]. El concepto sobre el que se asienta este algoritmo consiste en realizar las comparaciones de izquierda a derecha, de tal forma que si ocurre un error, la cadena de búsqueda puede ser desplazada *m* posiciones a la derecha. El número de comparaciones en el peor de los casos es *n+m*, y normalmente es sensiblemente menor. En [Sunday, 90] se pueden encontrar ciertas mejoras aplicables a este algoritmo.

Por otra parte, existen otras soluciones que se alejan de los enfoques más tradicionales. Por ejemplo, en [Aho, 75] se propone un método basado en la teoría de autómatas finitos que permite la localización de múltiples cadenas simultáneamente. Y en [Wu, 92] se plantea un algoritmo tolerante a los errores de deletreo.

A nivel general, las principales ventajas del método de inspección de texto completo son la ausencia de sobrecarga de espacio y el mínimo esfuerzo necesario para inserciones de nuevos documentos y/o actualizaciones de los ya existentes. Obviamente, la principal desventaja son los tiempos de respuesta que proporciona, que son especialmente elevados si la colección de documentos es de grandes dimensiones.

Como se ha comentado previamente, este método no se puede considerar estrictamente un método de indexado, básicamente porque no utiliza un índice para agilizar el proceso de localización de información. Sin embargo, es relevante ya que puede tener aplicación en cooperación con algún otro método de indexación que realice un filtrado de los documentos o incluso a través de hardware de propósito específico a tales efectos [Hollaar, 83].

1.4.2. Ficheros invertidos

La técnica de fichero invertido, o de índice invertido, es un mecanismo orientado a palabras para la indexación de una colección de documentos de texto para agilizar las tareas de búsqueda. La estructura de un fichero invertido se compone de dos elementos: el vocabulario y las ocurrencias [Navarro, 99]. El vocabulario es el conjunto formado por todas las ocurrencias de distintas palabras que aparecen en el texto. Asociado a cada palabra se encuentra una lista de los documentos en donde se puede encontrar esa palabra. El conjunto de todas esas listas es lo que constituye las ocurrencias (ver Figura 1-3).

Algunos autores diferencian el concepto de fichero invertido y lista invertida. Un fichero invertido es el modelo descrito, mientras que en una lista invertida cada elemento de la lista apunta a una posición de un documento. Esto simplemente es un problema de la granularidad del direccionamiento que abarca desde las posiciones del texto hasta bloques lógicos [Navarro, 99].

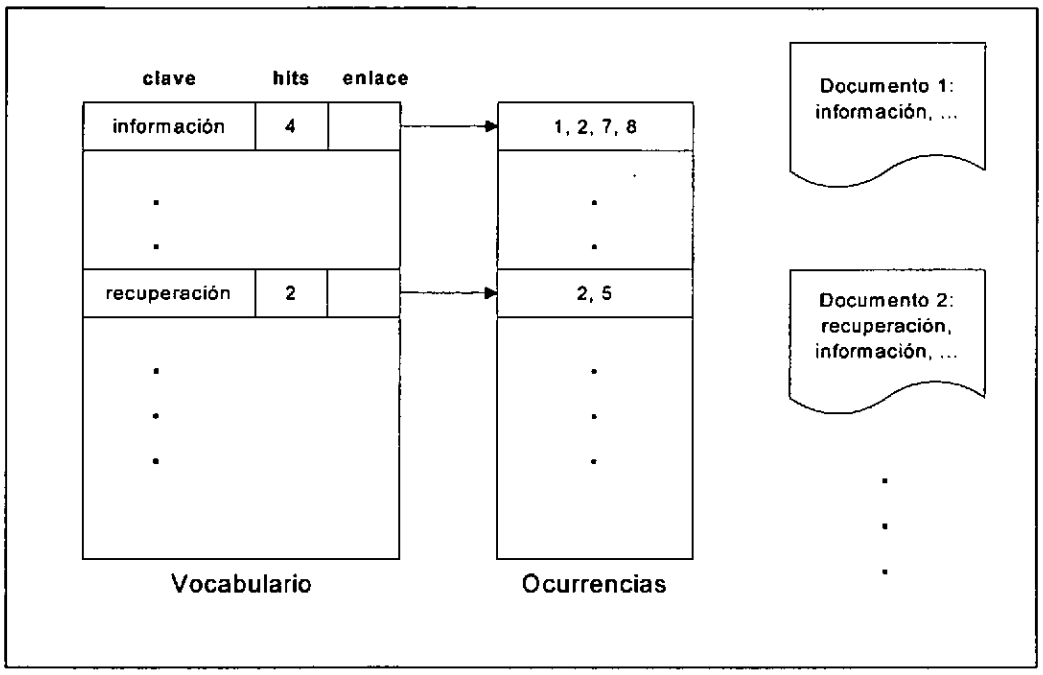

Figura 1-3: Indexación mediante fichero invertido

La utilización de un fichero invertido aumenta la eficiencia en las búsquedas varios órdenes de magnitud, por lo que es una estructura básica para grandes volúmenes de información. Por otra parte, la principal desventaja es la cantidad de espacio necesario para almacenar la estructura del índice que puede variar entre un 10% y 100% del texto original, o incluso más. Además, es necesario tener en cuenta las penalizaciones para las inserciones de nuevos documentos y las actualizaciones en los documentos ya existentes, que requieren accesos al índice para su modificación.

Normalmente, se imponen una serie de restricciones a la hora de construir los índices, que consecuentemente afectarán directamente a las búsquedas realizadas a posteriori [Harman, 92]. Algunos ejemplos de restricciones son los siguientes:

- En algunos casos, puede ser útil utilizar un vocabulario controlado, que constituirá el vocabulario que será indexado. Aquellas palabras del texto no pertenecientes al vocabulario no serán indexadas, y por lo tanto, posteriormente no podrán ser buscadas.
- Normalmente se dispone de una lista de palabras comunes (artículos, preposiciones, etc.) que por razones de volumen de información no se incluyen en el índice, y por lo tanto no serán localizables en la búsqueda.
- Existe un conjunto de reglas que se encargan de definir el inicio y el final de una palabra para ser indexada. Estas reglas se encargan del tratamiento de espacios en blanco, signos de puntuación, etc. y pueden tener un impacto significativo en los términos indexados.

Es importante destacar que estas restricciones encargadas de determinar lo que será indexado son críticas para la efectividad de las búsquedas, y por lo tanto son un parámetro clave a la hora de la construcción del índice.

El espacio utilizado por el vocabulario se puede considerar reducido. Según la ley de Heaps el vocabulario crece según $O(n^{\beta})$, en donde β es una constante entre 0 y 1 (dependiente del texto) [Heaps, 78], que en la práctica toma valores en 0,4 y 0,6. Por ejemplo, considerando 1 gigabyte de texto el vocabulario aproximadamente será de sólo 5 megabytes. Además, este espacio puede ser reducido considerando la técnica de stemming u otras técnicas de normalización del texto.

En cambio, las ocurrencias requieren un espacio mucho mayor. En general, el espacio extra es de $O(n)$, ya que cada palabra que aparece en los documentos debe ser referenciada una vez. Es importante tener en cuenta que las palabras comunes no serán almacenadas en el índice, y aún en esos casos el tamaño del índice puede ser importante. En la literatura se ofrecen diversas cifras al respecto y aunque a nivel general es muy difícil concretar en único valor, la tendencia actual es a la reducción del tamaño de los índices por debajo del 100% del texto original.

El proceso de búsqueda en un fichero invertido requiere tres pasos, aunque alguno de ellos puede ser obviado ante determinadas consultas:

- Búsqueda en el vocabulario: cada una de las palabras o patrones presente en la consulta es aislada y buscada en el vocabulario. Es importante destacar que las búsquedas de frases o por proximidad se dividen en palabras individuales.
- Recuperación de las ocurrencias: para cada palabra del vocabulario se recupera su lista de ocurrencias asociada.
- Manipulación de las ocurrencias: las ocurrencias son procesadas para resolver operaciones de lógica booleana, frases o proximidad.

Las consultas formadas por una única palabra pueden ser buscadas utilizando diversas estructuras de datos con el objetivo de mejorar el rendimiento: hashing, árboles B, tries. En los siguientes apartados se describirán en detalle el modo de operación y el rendimiento ofrecido para las principales técnicas. Sin embargo, a nivel general, simplemente almacenando por orden alfabético las palabras del vocabulario se consigue un rendimiento muy competitivo, ya que una palabra puede ser localizada realizando una búsqueda binaria con un coste de $O(\log n)$.

Si la consulta está formada por una única palabra, el proceso finaliza con el envío de la lista de ocurrencias. Puntualizar que en el caso de que el patrón coincida con múltiples palabras se hace necesaria la unión de múltiples listas de ocurrencias.

Las consultas que involucren a varias palabras son ligeramente más complejas de resolver en el caso de los ficheros invertidos. Cada elemento o palabra debe ser buscado de manera separada, se deben recuperar cada una de las listas y realizar las operaciones correspondientes en caso de que se hayan utilizado operadores lógicos en la búsqueda. Esto implica la realización de operaciones de unión, intersección o resta de conjuntos.

Las consultas de proximidad o de frases incorporan un mayor nivel de complejidad, ya que para cada palabra se debe obtener una lista (preferiblemente ordenada) de los documentos. Las listas deben ser combinadas, para lo cual se recorren intentando localizar documentos en donde las palabras aparezcan en secuencia (en el caso de una frase) o lo suficientemente cerca (para el caso de las búsquedas de proximidad). En el caso de que alguna de las listas

sea lo suficientemente reducida se puede realizar una búsqueda binaria en el resto a partir de los valores de la menor de las listas. Además, para que el índice pueda soportar este tipo de búsquedas es necesario almacenar, para cada documento, las posiciones en donde aparecen cada una de las ocurrencias del término, lo que aumenta sensiblemente el tamaño del índice.

A nivel general las principales ventajes de la técnica de fichero invertido se centran en la velocidad de localización de palabras que es logarítmica. Por otra parte las principales desventajas se centran en la sobrecarga de almacenamiento necesaria para el índice, que aunque anteriormente podría sufrir una penalización grave (entre el 50% y el 300% según Haskin en [Haskin, 81]), en la actualidad gracias a las técnicas de compresión (ver sección 1.4.2.4) la sobrecarga se ha reducido a valores menores del 100%. Por otra parte, dependiendo de la técnica utilizada el coste de actualización y reorganización del índice puede ser demasiado elevado, por lo que su utilización no es aconsejable en entornos altamente dinámicos. Y finalmente, uno de los problemas más relevantes se presenta por la necesidad de realizar operaciones sobre las listas de ocurrencias obtenidas, que pueden ser especialmente perjudiciales si el número de elementos y/o de listas es muy elevado.

En los siguientes apartados se describen las principales alternativas de implementación existentes para los ficheros invertidos que mejoran determinados aspectos de la técnica elemental.

1.4.2.1. Arrays ordenados

Un fichero invertido implementado usando una estructura de un array ordenado almacena la lista de palabras clave (el vocabulario) en un array ordenado, incluyendo el número de documentos asociados a esa clave y un enlace a lista de documentos (ver Figura 1-3). Para localizar un término en el array se realiza una búsqueda binaria, aunque en el caso de dispositivos de almacenamiento secundarios deben ser adaptados la estructura de almacenamiento y el algoritmo de búsqueda [Harman, 92].

La utilización de esta estructura para el almacenamiento del índice presenta problemas a la hora de gestionar las actualizaciones del índice (principalmente, inserciones de nuevos términos) ya que el coste es elevado. Por otra parte, las principales ventajas se centran en la facilidad de implementación y el alto rendimiento ofrecido.

La construcción de un fichero invertido utilizando una estructura de un array ordenado se divide en varios pasos. Inicialmente, el texto es analizado léxicamente para la obtención de las distintas palabras junto con los documentos en donde pueden ser localizadas. Esta primera parte del proceso constituye la principal carga de tiempo y almacenamiento durante la elaboración del índice. En segundo lugar la lista de términos debe ser ordenada alfabéticamente, creando la lista de los documentos en donde se localiza cada término [Harman, 92]. Y opcionalmente, se puede realizar un postprocesamiento del fichero invertido para calcular los pesos de los términos, o realizar reorganizaciones o compresiones del archivo.

La primera parte del proceso, consistente en la obtención de la lista inicial de palabras, se descompone en diferentes operaciones que van desde la eliminación de las palabras

comunes hasta la obtención de la raíz de la palabra (proceso denominado comúnmente "*stemming*").

La segunda parte del proceso consiste en la inversión de la lista de términos. Esta operación se suele realizar con una ordenación de los términos manteniendo los elementos duplicados Este proceso de ordenación puede ser bastante costoso en el caso de grandes vocabularios, principal problema que presenta esta técnica y que ha impulsado el desarrollo de otras variantes descritas en los siguientes apartados. Seguidamente se realizará un proceso de eliminación-mezcla de los términos duplicados que generará la lista de documentos asociados a cada término.

Normalmente, en la técnica de fichero invertido se almacenan los dos componentes (vocabulario y ocurrencias) de manera separada. Como se ha comentado anteriormente el vocabulario posee unas dimensiones medianamente reducidas frente al gran tamaño que puede llegar a ocupar la lista de ocurrencias. El proceso de búsqueda, en consecuencia, se centra en la localización de los términos utilizando una búsqueda binaria en el vocabulario para después acceder directamente a la lista de ocurrencias asociada.

Sobre la técnica básica de indexación usando arrays ordenados han surgido diversas mejoras. Entre ellas cabe destacar la propuesta por Harman et al. en [Harman, 90] en donde se presenta una técnica que mejora la construcción de los ficheros invertidos para grandes conjuntos de datos. Básicamente, en esta técnica se elimina el paso intermedio de ordenación de la lista utilizando un tipo especial de árboles binarios. Por otra parte, Fox et al. diseñaron FAST-INV, una técnica para la generación de un fichero invertido basada en la gran cantidad de memoria disponible hoy en día en los ordenadores y el orden inherente de los datos de entrada [Fox, 91].

1.4.2.2. Árboles B

Otra estructura de implementación utilizada para la técnica de ficheros invertidos se basa en los árboles B.

Los árboles B son un caso específico de los árboles de búsqueda, cuyo componente más conocido son los árboles binarios. En un árbol binario cada nodo interno contiene una clave de tal forma que el subárbol izquierdo contiene claves menores que la del padre y el subárbol derecho contiene claves mayores. Este tipo de árboles se adapta de manera adecuada para ser utilizados directamente en memoria principal.

En cambio, cuando se necesita acceder a memoria secundaria, los árboles de búsqueda m-arios presentan un mejor rendimiento ya que los nodos internos son mayores. En concreto, los árboles B son tipo concreto de estos árboles. Un árbol B de orden m se define como [Baeza-Yates, 92]:

- La raíz y todos los nodos internos del árbol tienen entre m y $2m$ claves.
- Si k_i es la clave de la posición i-ésima, entonces todas las claves del hijo (i-1)-ésimo son menores, mientras que todas las claves del i-ésimo hijo son mayores.
- Todos los nodos hoja tienen la misma profundidad.

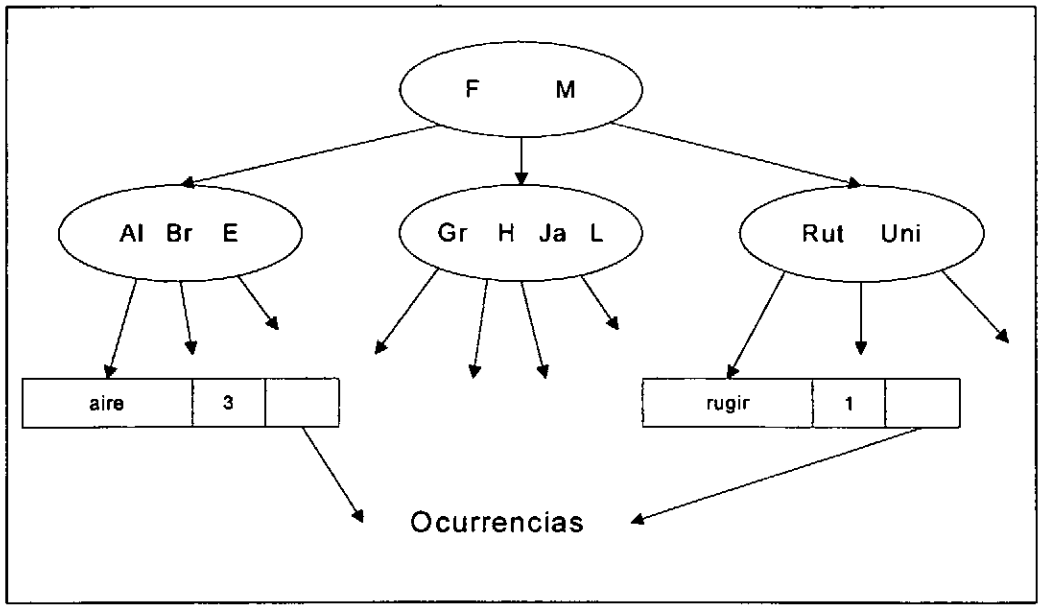

Figura 1-4: Ejemplo de árbol B prefijo

En [Cutting, 90] se presentan los árboles B como una estructura eficiente para los ficheros invertidos en situaciones de datos dinámicos, en donde el número de actualizaciones es preponderante sobre las búsquedas.

Sin embargo, son Bayer y Unterauer ([Bayer, 77]) quienes proponen la utilización de un tipo especial de árboles B, los árboles B prefijos, que utilizan prefijos de las palabras como claves primarias en el árbol. Este tipo de árbol demuestra ser especialmente útil en el caso del almacenamiento de índices de texto, como es el caso de los ficheros invertidos.

Cada clave es la palabra más corta que permite distinguir las claves almacenadas en el siguiente nivel, y la clave no tiene porque ser, necesariamente, un prefijo de un término real del índice. Los nodos hoja almacenan las palabras clave junto con los datos asociados a los términos del vocabulario, típicamente la lista de ocurrencias (ver Figura 1-4).

La principal ventaja de la utilización de árboles B se centra en la facilidad para realizar inserciones en el índice frente al caso de los arrays ordenados. Para insertar un nuevo registro simplemente se localiza el punto de inserción, si no hay suficiente espacio en ese nodo, se divide y se promociona una clave al nivel anterior. Este proceso se repite recursivamente hasta llegar al nodo raíz, que en caso de no disponer de más espacio aumenta la profundidad del árbol en uno [Baeza-Yates, 92].

Asimismo, la búsqueda es normalmente más rápida que el caso de un array ordenado, ya que el número de comparaciones necesarias es igual a la profundidad del árbol. Por otra parte, las principales desventajas se centran en la cantidad de almacenamiento empleado (debido a la presencia de nodos intermedios, que pueden ser de gran tamaño) y la complejidad de implementación.

1.4.2.3. Tries

Un fichero invertido puede ser también implementado utilizando una estructura de datos denominada "*trie*". Los tries son una estructura de árbol recursiva especialmente indicada para la localización de prefijos. Se basan en la descomposición de las cadenas de caracteres para representar las distintos prefijos y para facilitar la búsqueda. Los tries fueron inventados por Briandais en [Briandais, 59] y el nombre fue sugerido por Fredkin en [Fredkin, 60], derivado de "information re*trie*val".

Partiendo de un alfabeto ordenado, se construye un árbol ordenado lexicográficamente. La raíz del trie utiliza el primer carácter, el nodo hijo utiliza el segundo carácter y así sucesivamente. Si el restante subtrie contiene únicamente una cadena, el identificador de esta cadena (o sea, la posición en la que se encuentra) se almacena en un nodo externo [Baeza-Yates, 92]. En la Figura 1-5 se muestra el trie binario para la cadena "01100100010111..." analizada hasta la posición 8; los nodos externos indican la posición del prefijo en el texto.

Un tipo de estructura trie especialmente útil en el campo de la recuperación de información es el árbol Patricia. Un árbol Patricia ("Practical Algorithm To Retrieve Coded In Alphanumerical") es un trie con la restricción adicional de que los nodos con un único descendiente son eliminados [Morrison, 68]. De esta forma se mantiene un contador en los nodos internos encargados de indicar el siguiente bit a inspeccionar. En la Figura 1-6 se muestra el árbol Patricia correspondiente al trie binario de la Figura 1-5.

Considerando una cadena de n caracteres, el índice constará de n nodos externos

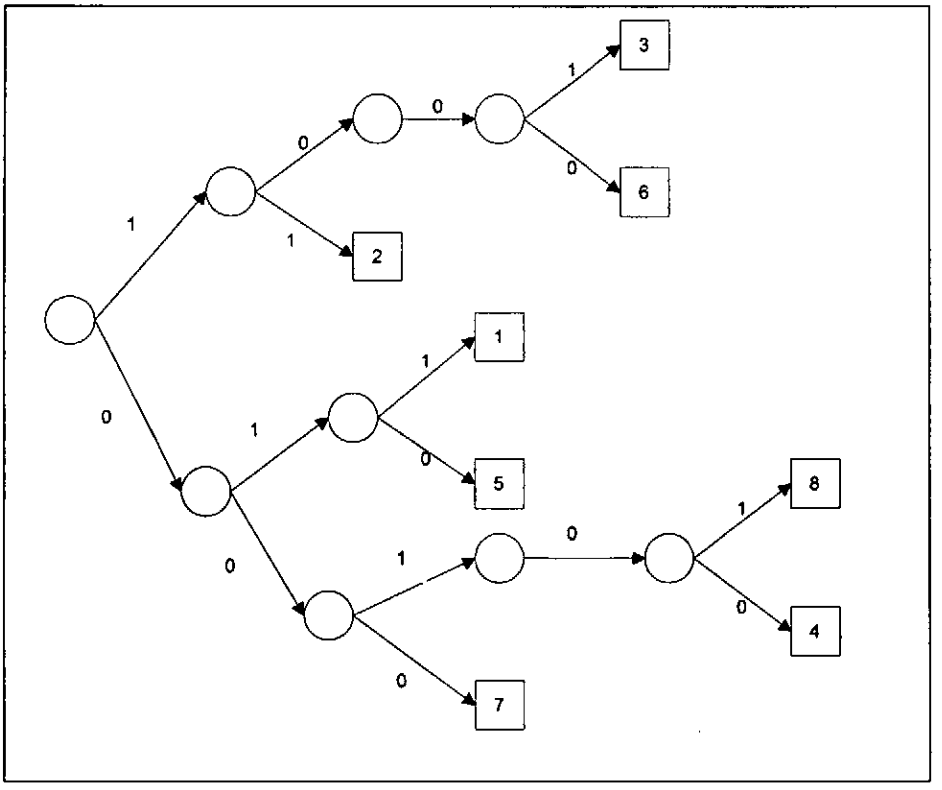

Figura 1-5: Ejemplo de trie binario

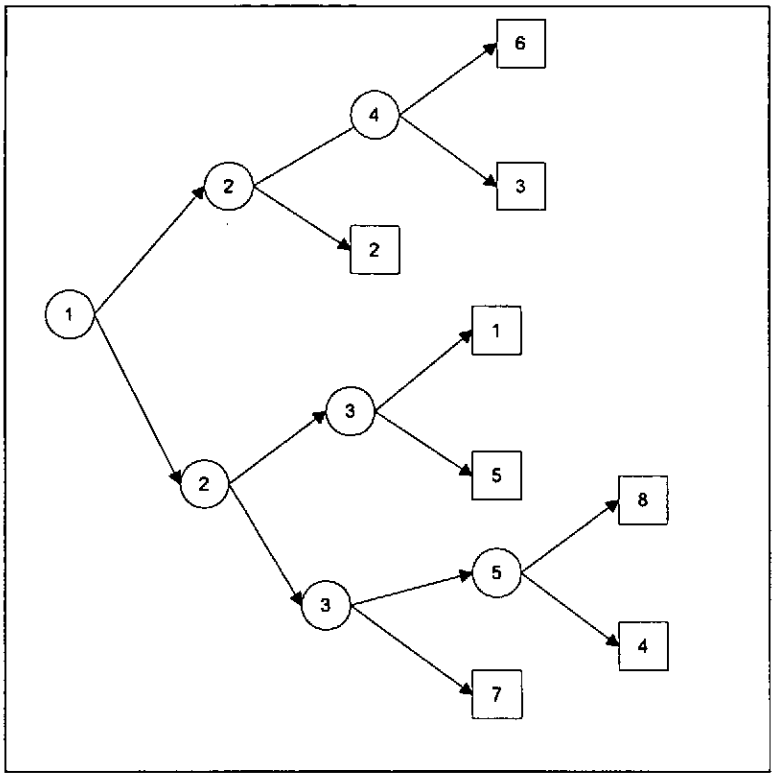

Figura 1-6: Ejemplo de árbol Patricia

correspondientes a las *n* posiciones de la cadena, y *n-1* nodos internos.

Asociado a los árboles Patricia se encuentran las estructuras PAT, originalmente propuestas por Gonnet en [Gonnet, 83]. Esta estructura se caracteriza por romper con el modelo tradicional de texto utilizado en los sistemas de recuperación de información. Según el modelo tradicional se parte de una colección de documentos, a los cuales se les asignan un conjunto de palabras clave, pudiendo opcionalmente, asignar pesos de relevancia a cada término. Esta orientación ha sido utilizada (y todavía es utilizada) ampliamente, pero presenta ciertos problemas: (i) la estructura subyacente asumida puede ser válida para determinadas aplicaciones (p.e. archivos bibliográficos), en cambio puede no serlo para otras; (ii) es necesario realizar una extracción previa de las palabras clave del texto, lo cual puede inducir a ciertos errores al realizar este proceso y por lo tanto perder algunos términos; (iii) las consultas deben ser realizadas utilizando palabras clave [Gonnet, 92].

Utilizando las estructuras PAT el texto es considerado como una gran cadena de texto. Cada posición en el texto se corresponde con una cadena de texto semi-infinita (denominada *sistring*). Las principales ventajes de este modelo son: (i) no se necesita ninguna estructura del texto (aunque si la hay, puede ser utilizada); (ii) no se utilizan palabras clave, las consultas se basan en prefijos de sistrings, o lo que es lo mismo, en cualquier sub-cadena del texto.

La principal ventaja de este tipo de estructura en la generación de los índices es el uso potencial en cualquier tipo de búsqueda (búsqueda por literal, por proximidad, etc.), que utilizando una aproximación directa de fichero invertido puede ser ineficiente o demasiado compleja. Esto es el caso de las búsquedas de frases (especialmente aquellas que están

compuestas por palabras muy frecuentes), la búsqueda de expresiones regulares o la búsqueda por proximidad. Por otra parte, esta estructura obtiene un rendimiento respecto al tamaño y tiempo de búsqueda equiparable a las otras técnicas ya vistas [Gonnet, 92].

1.4.2.4. Compresión de ficheros invertidos

Las técnicas de compresión aplicadas a ficheros invertidos operan principalmente sobre la lista de ocurrencias o lista invertida, si bien, también se pueden aplicar técnicas de compresión sobre el conjunto de palabras que conforman el vocabulario.

La compresión de las palabras clave incluidas en el vocabulario no permite obtener unas reducciones significativas en el espacio de almacenamiento requerido, al estar optimizadas para la compresión de documentos completos. Simplemente mencionar la técnica de los códigos de Huffman canónicos [Hirschberg, 90], en donde cada palabra es sustituida por un código de este tipo, cuya longitud depende de la frecuencia de la palabra.

La compresión en las listas invertidas se consigue mediante la numeración de los documentos secuencialmente desde uno, ordenando las entradas de la lista invertida, representando la secuencia de identificadores como una secuencia de saltos y empleando técnicas de representación compacta de enteros pequeños [Moffat, 94]. Por ejemplo, la siguiente lista invertida:

$$1, 4, 17, 91, 113, ...$$

Podría ser representada como la siguiente secuencia de diferencias (que puede ser comprimida):

$$1, 3, 13, 74, 22, ...$$

Las técnicas de compresión aplicadas a las secuencias de este tipo se dividen en dos clases [Moffat, 92]. Los métodos globales que usan el mismo tipo de codificación para todas las entradas, y por lo tanto presentan la ventaja de ser generales, pero insensibles a la frecuencia de cada término. Los métodos locales se caracterizan por calcular cada código teniendo en cuenta uno o más parámetros de la distribución de los valores (normalmente, la frecuencia de cada término).

Uno de los métodos globales más simples es el código gamma propuesto en [Elias, 75], en donde cada entero positivo x es representado mediante la codificación de $\lfloor \log_2 x \rfloor$ en unario, seguido del valor de $x - 2^{\lfloor \log_2 x \rfloor}$ en binario. Esta codificación genera códigos de longitud variable, pero en donde cada conjunto de bits puede ser unívocamente decodificado en un entero positivo. En la Tabla 1-1 se muestran algunos valores de códigos gamma.

La utilización de códigos de longitud variable permite representar de manera más sucinta los valores menores (y más frecuentes) que los valores mayores (y menos frecuentes), lo que supone una mejora sobre la codificación binaria estándar.

Por otra parte, los códigos locales obtienen una ventaja adicional al considerar la variabilidad de las frecuencias de cada palabra. Por ejemplo, la palabra "*el*" tendrá

x	gamma(x)
1	0
2	10 0
3	10 1
4	110 00
7	110 11
15	1110 111
63	111110 11111

Tabla 1-1: Codificación gamma para listas invertidas

asociada una secuencia de diferencias formada por valores pequeños, comúnmente uno. En cambio, palabras más raras tendrán listas invertidas formadas por diferencias mucho mayores. Por lo tanto, los métodos locales, que adaptan su codificación en función de la frecuencia de cada palabra, pueden obtener unas tasas de compresión mayores, aún sin ser generales.

Las técnicas de compresión de listas invertidas aplicadas a grandes volúmenes de documentos (como es el caso de los sistemas de búsqueda) permiten realizar compresiones del índice por un factor de seis [Bell, 93]. Esta aproximación presenta la desventaja de que las listas invertidas deben ser decodificadas al ser recuperadas, pero esta descompresión es rápida. Además, existen técnicas de compresión que insertando una pequeña cantidad de información adicional de indexado en cada lista permiten evitar una gran parte del proceso de decodificación [Zobel, 98].

Una característica importante de las listas invertidas comprimidas es que los mejores porcentajes de compresión se consiguen en las listas de mayor tamaño, esto es, los términos más frecuentes. Por lo tanto, no es necesario eliminar las palabras comunes del índice en el momento de la creación del fichero invertido, sino que la decisión de la utilización o no de las palabras comunes queda relegada al momento de la consulta.

En consecuencia, la compresión de las listas invertidas permite obtener una mejora en el espacio de almacenamiento utilizado por el índice, y algo que posiblemente pueda ser considerado de mayor relevancia, es la posibilidad de insertar las palabras comunes en el índice sin aumentar drásticamente el tamaño final de la estructura.

1.4.3. Ficheros de firmas

La técnica de indexación denominada ficheros de firmas (del inglés "*signature files*") se basa en la idea de generar un filtro inexacto: los ficheros de firmas proporcionan un test rápido que permite descartar múltiples elementos que no se clasificarían. Esta técnica garantiza que todos los elementos correctos pasarán el test, mientras que habrá algunos elementos adicionales (denominados, "*false hits*" o "*false drops*" o "falsos aciertos") que pueden pasar el test accidentalmente [Faloutsos, 92a]. Esta técnica representa básicamente una abstracción de los documentos de texto en un sistema de información [Lee, 89]. El fichero de firmas es utilizado en una primera fase de la consulta para eliminar tantos documentos como sea posible, teniendo en cuenta el impacto de los falsos aciertos en la precisión del sistema de búsqueda.

La investigación alrededor del diseño y el rendimiento de los métodos de codificación superpuestos aplicada a la recuperación de información fue iniciada por C. N. Mooers en 1949. Él inventó un ingenioso dispositivo mecánico, basado en tarjetas con muescas y agujas, que era capaz de gestionar de manera eficiente consultas conjuntivas sobre una base de datos de entradas bibliográficas [Mooers, 49]. Este método atrajo gran interés, y en 1960 Stiassny sugirió la utilización de pares de letras para la creación de la firma de cada palabra, y demostró que, para un tamaño de firma concreto, la probabilidad de falsos aciertos se minimizaba si el número de "1"s era igual al número de "0"s en las firmas de los documentos [Stiassny, 60]. Por otra parte, Kautz y Singleton en [Kautz, 64] discuten el problema de diseñar un sistema de firmas que no genere falsos aciertos, atacando el problema desde el punto de vista de la codificación y la teoría de la información. Aunque este método es interesante desde un punto de vista teórico, a nivel práctico presenta problemas ya que no es capaz de gestionar fácilmente un vocabulario creciente y presenta una gran sobrecarga para el diseño del conjunto de firmas.

Files y Huskey en [Files, 69] describen la aplicación de la técnica de ficheros de firmas a una base de datos de entradas bibliográficas, eliminando las palabras comunes y utilizando una función hash para la generación de las firmas. Por otra parte, Pfaltz et al. en [Pfaltz, 80] sugiere la utilización de varios niveles de ficheros de firmas. Una firma del nivel i-ésimo se crea superponiendo un número constante de firmas del nivel (i-1)-ésimo. Uno de los trabajos más relevantes es el de Roberts en [Roberts, 79] en donde utiliza un fichero de firmas para una aplicación de directorio telefónico. En este trabajo se presentan varios aspectos novedosos e interesantes, como un almacenamiento en disco por columnas (ver sección 1.4.3.3) y la asignación de firmas en función de la repetición de cada término, concepto investigado y mejorado por Faloutsos en [Faloutsos, 85b].

A nivel general, las principales características de los métodos basados en firmas respecto a los otros métodos de indexación son las siguientes: los métodos basados en firmas son más rápidos que la técnica de inspección de texto completo (1 ó 2 órdenes de magnitud por encima). Por otra parte, comparado con la técnica de fichero invertido, los ficheros de firmas presentan una baja sobre carga para la creación del índice (entre un 10% y 15% según [Christodoulakis, 84]) frente a los valores entre un 40% y 100% del fichero invertido (dependiendo de la técnica utilizada y los métodos de compresión). Asimismo, la técnica de ficheros de firmas puede gestionar las inserciones de una manera más simple que la técnica de ficheros invertidos, ya que únicamente se necesita una operación de inserción al final del índice, sin necesidad de reorganizar ni rescribir ninguna porción del índice. Este tipo de sistemas son especialmente adecuados para la creación de índices en dispositivos WORM (una escritura, múltiples lecturas) y aumentan la concurrencia en la lectura ya que los lectores pueden seguir operando con la estructura mientras se está realizando la inserción.

Por otra parte, el principal inconveniente de los ficheros de firmas es el tiempo de respuesta ante consultas, que es lineal con el número de elementos en la base de datos, frente al tiempo logarítmico del fichero invertido. Esto hace que esta técnica se haya usado extensivamente en bases de datos de tamaños medios, pero hoy en día se hace impracticable utilizar directamente esta tecnología como método de indexación de grandes volúmenes de datos. En resumen, la técnica de ficheros de firmas ofrece un rendimiento a medio camino entre la inspección de texto completo y los ficheros invertidos.

La técnica de ficheros de firmas se basa en la generación de firmas para cada uno de los términos indexados, que a su vez pasarán a formar parte de la firma final del documento. En la siguiente sección se describen los principales métodos de generación de firmas, para a continuación describir diferentes optimizaciones sobre la técnica básica.

1.4.3.1. Métodos de generación de firmas

Básicamente existen dos métodos de generación de firmas. El primero de ellos se denomina Firmas de Palabras (del inglés Word Signatures, también identificado por las siglas WS). Este método propone que a cada palabra del documento le sea asignado un patrón de longitud *f* [Larson, 83][Tsichritzis, 83]. Estos patrones constituyen las firmas de cada palabra que son concatenados para formar la firma del documento (ver Figura 1-7) [Faloutsos, 87]. Las firmas del conjunto de documentos pertenecientes a la colección se almacenan conjuntamente, constituyendo el fichero de firmas.

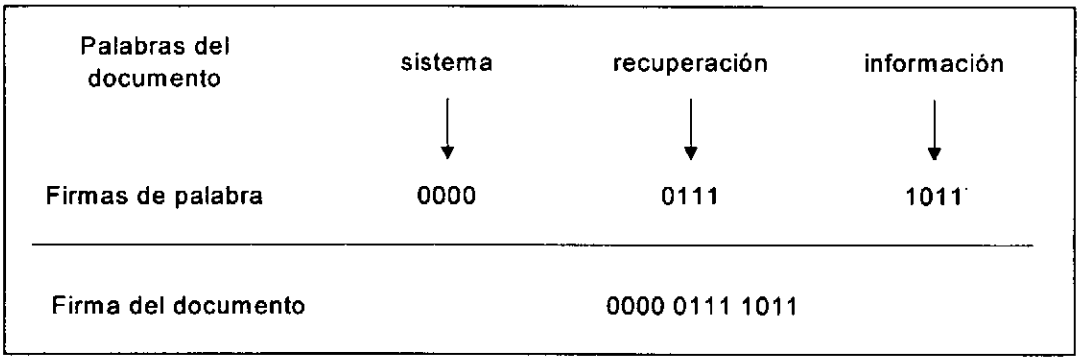

Figura 1-7: Método de firma de palabra (WS)

Durante la realización de una consulta, en primer lugar se obtiene la firma de la consulta utilizando la misma técnica que en el caso de los documentos. El fichero de firmas es buscado secuencialmente, intentado localizar documentos que coincidan con la firma de la consulta. La utilización de una búsqueda binaria para las comparaciones de las firmas produce un mejor rendimiento frente a las comparaciones de texto en la inspección de texto completo [Eastman, 89]. Igualmente, en este caso se puede aplicar la técnica de eliminación de palabras comunes mejorando el rendimiento total del sistema [Faloutsos, 85a].

La principal característica de este método es que se conserva la información de la posición relativa de las palabras dentro del documento. Esto representa una ventaja, ya que para aquellas consultas que hacen referencia a la posición de las palabras en el documento pueden ser resueltas de manera ágil. Y a su vez, esto representa la principal desventaja de este método ya que se debe examinar cada entrada del fichero de firmas de manera secuencial para localizar el término buscado.

Eastman en [Eastman, 89] sugiere que esta técnica es más adecuada para documentos o bases de datos muy estructuradas, en cambio, los sistemas de recuperación de información requieren técnicas que ofrezcan un rendimiento más eficiente, como la generación de firmas mediante códigos de superposición.

Los ficheros de firmas han estado normalmente basados en el método de generación de códigos conocida como códigos de superposición, definidos por Mooers en [Mooers, 49]. En los siguientes párrafos se realizará una descripción detallada de la teoría de los códigos de superposición, tanto por ser el tipo de codificación empleada en los ficheros de firmas, como porque constituyen uno de los ejes del presente trabajo de tesis doctoral, al ser aplicados sobre estructuras de datos no tradicionales.

La utilización de códigos de superposición en los ficheros de firmas fue introducida por Faloutsos y Christodoulakis [Faloutsos, 84], basándose en la técnica de códigos superpuestos inicialmente desarrollada por Mooers, y ampliada por Harrison [Harrison, 71] y Roberts [Roberts, 79]. En esta técnica, cada documento se divide en bloques lógicos (bloques de texto que deben contener un número constante de palabras distintas y no comunes). Cada palabra del bloque tiene asociada una firma (mediante una técnica de hashing o directamente generando las firmas y almacenándolas en una tabla). Para obtener la firma de cada bloque se realiza el OR lógico bit a bit de cada una de las firmas de las palabras, tal y como se ilustra en la Figura 1-8.

Palabras	Firmas de palabra
sistema	001 000 110 010
recuperación	000 010 101 001
Firma de bloque	001 010 111 011

Figura 1-8: Método de códigos de superposición (*D=2*, *b=12* y *w=4*)

Por motivos de rendimiento, cada documento se divide en bloques lógicos que contienen un número constante de palabras distintas, denominado *D*. A su vez, a cada palabra se le asocia una firma, constituida por un patrón de bits de tamaño *b*, de los cuales *w* bits se establecen a "1", mientras que el resto toman el valor "0". Los valores de *D*, *b* y *w* son parámetros de diseño de este tipo de sistemas.

Esta técnica permite un ahorro sustancial en el tamaño del índice generado, ya que cada bloque de texto (que en gran parte de los trabajos de investigación se asocia con un documento) se representa por una firma de tamaño constante, frente a la técnica anterior en donde las firmas generadas para cada documento son de tamaño variable.

Al igual que en el caso anterior, las consultas siguen la misma pauta. En primer lugar se convierte la consulta a una firma formada por las firmas de cada una de las palabras. Entonces se realiza una búsqueda secuencial a través del fichero de firmas localizando los documentos o bloques que coinciden con la consulta realizada, simplemente por medio de una operación de AND lógico entre la firma de la consulta y la firma de cada documento. Como se ha comentado anteriormente, esta técnica se utiliza en conjunción con una lista de palabras comunes que son eliminadas del texto del documento para conseguir un mejor rendimiento.

Un concepto importante en los ficheros de firmas es la probabilidad de falsos aciertos, representada como F_d. Intuitivamente, es la probabilidad de que el test sobre la firma falle generando un falso acierto. De todas formas, es importante destacar que esta técnica produce falsos aciertos, sin embargo, nunca se producirán falsos fallos. En otras palabras, el método del fichero de firmas garantiza que, ante una consulta, se recuperarán todos los documentos que satisfagan dicha consulta, y a mayores, pueden ser recuperados otros documentos que no se ajusten a la consulta, garantizando que nunca serán obviados documentos que satisfagan la consulta.

Se define la probabilidad de falsos aciertos, F_d, como la probabilidad de que la firma de un bloque semeje correcta, a pesar de que el bloque no lo sea (asumiendo que un bloque es correcto si contiene la palabra buscada) [Faloutsos, 92a]:

$$F_d = Prob\{firma\ correcta \mid bloque\ no\ es\ correcto\}$$

O lo que es lo mismo, la probabilidad de falsos aciertos se mide como el número de falsos aciertos dividido por el número total de documentos menos los aciertos reales:

$$F_d = \frac{número\ falsos\ aciertos}{N - número\ aciertos\ reales}$$

El fichero de firmas se puede considerar como una matriz binaria $b \times N$. Stiassny probó en su trabajo que para un valor concreto de b, el valor de w que minimiza la probabilidad de un acierto falso es aquel en el que cada fila de matriz contiene "1"s con una probabilidad del 50 por ciento. O lo que es lo mismo, se minimiza la probabilidad de falsos aciertos en aquellos casos en los que la firma de cada bloque contiene igual número de "0"s que de "1"s [Stiassny, 60]. Para un valor concreto de b y D, el valor óptimo de w se calcula como:

$$w = \frac{b \ln 2}{D}$$

Esta es la razón por la que los documentos deben ser divididos en bloques lógicos. Si los documentos no estuviesen divididos en bloques lógicos, aquellos documentos más largos generarían firmas saturadas de "1"s, lo que provocaría siempre un falso acierto. A partir de la expresión anterior, se obtiene el valor óptimo de la probabilidad de aciertos falsos:

$$F_d = 2^{-w}$$

En el capítulo 4 se realiza una exposición detallada del cálculo de la probabilidad de falsos aciertos, así como su ajuste al modelo propuesto.

La forma básica de almacenamiento de un fichero o matriz de firmas consiste en almacenar las filas de manera secuencial. A este método se le denomina SSF (Sequential Signature File). En la Figura 1-9 se muestra el almacenamiento de la estructura, la cual suele ir asociada con un archivo de punteros a los bloques lógicos o alternativamente a los documentos.

Aunque el método SSF ha sido utilizado directamente, ofrece un bajo rendimiento cuando el tamaño del fichero es considerable. En la literatura se pueden encontrar múltiples

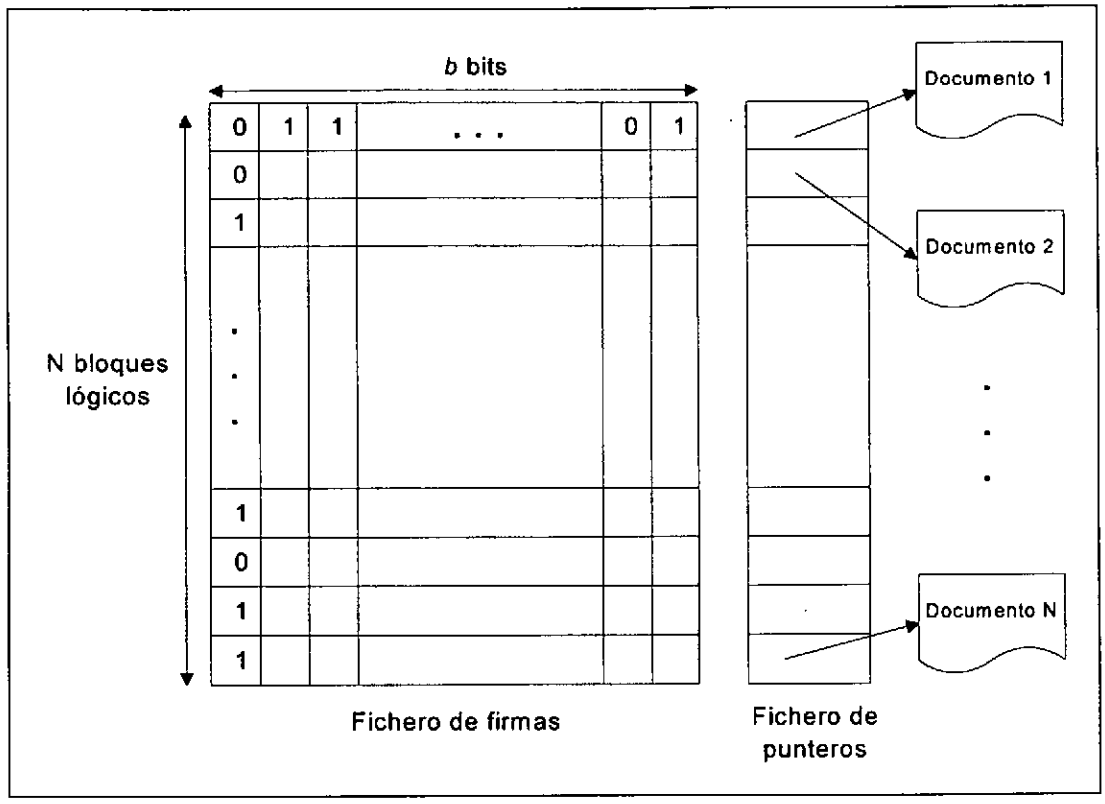

Figura 1-9: Estructura del método SSF

métodos que intentan mejorar el tiempo de respuesta del SSF, reducir el espacio de almacenamiento o agilizar las inserciones. Sin embargo todos los métodos propuestos se engloban dentro de alguno de las siguientes ideas:

- Compresión: en el caso de que la matriz de firmas sea deliberadamente difusa, ésta puede ser comprimida.
- División vertical: el almacenamiento de la matriz de firmas por columnas mejora los tiempos de respuesta en el caso de las búsquedas, mientras que se penalizan las inserciones.
- División horizontal: el agrupamiento de firmas similares puede resultar en tiempos de respuesta mejor que la aproximación lineal básica.

La principal ventaja de la técnica de ficheros de firmas en general se centra en su simplicidad: únicamente se necesita una comparación a nivel de bit para determinar si un término se encuentra en un bloque de texto. Por otra parte, las inserciones de nuevos documentos se realizan simplemente añadiendo nuevas firmas de bloque al final del fichero de firmas, sin ninguna reordenación ni reubicación de las estructuras previamente almacenadas. Sin embargo, el hecho de tener que realizar una búsqueda secuencial es una desventaja cuando se aplica a grandes volúmenes de datos [Croft, 88]. En los siguientes apartados se describen diferentes variantes de compresión y técnicas de división aplicadas directamente a ficheros de firmas que permiten mejorar este aspecto, aunque el rendimiento alcanzado nunca será equiparable a la técnica de los ficheros invertidos. Una segunda desventaja de este método se deriva del hecho de que la técnica de los códigos de superposición no representa la posición relativa de las palabras en el texto, por lo que

aquellas búsquedas que hacen referencia a las posiciones de las palabras en el texto requieren un acceso directo a los documentos.

1.4.3.2. Compresión

En esta sección se examinan el conjunto de métodos de compresión de ficheros de firmas propuestos por Faloutsos y Christodoulakis en [Faloutsos, 87]. Estos métodos generan firmas de documentos difusas (con muy pocos "1"s), que son comprimidas antes de ser almacenadas secuencialmente. En este trabajo demuestran que utilizando las técnicas de compresión se obtiene una mejor probabilidad de falsos aciertos que el método SSF, para el mismo tamaño de índice.

La idea en la que se basan estos métodos consiste en un usar un vector de bits de tamaño b, con un valor de b grande, y cada palabra del documento generará un número reducido de posiciones a "1" (el valor óptimo es $w = 1$). De esta manera, la firma del bloque o documento será difusa y podrá ser comprimida (ver Figura 1-10).

La técnica de compresión comúnmente empleada en estos casos es la denominada codificación run-length [McIlroy, 82], en donde simplemente se almacena el número de ceros consecutivos existentes en la firma (ver Figura 1-11). Inicialmente, McIlroy propuso esta técnica como un medio para comprimir un diccionario de aproximadamente 30.000 palabras. Este método proporciona una buena medida de compresión, aunque las búsquedas se ven penalizadas, ya que para determinar si un bit de la firma original es "1" es necesario decodificar y sumar los valores de los intervalos anteriores en el formato comprimido. Por lo tanto, suponiendo que los bits de las consultas están uniformemente distribuidos, de media sería necesario decodificar la mitad de los intervalos.

Con el objetivo de mejorar los tiempos de respuesta de las búsquedas utilizando compresión, a costa de aumentar ligeramente el espacio de almacenamiento, se define el método BC (bit-Block Compresión). El vector de bits se divide en grupos de bits consecutivos (denominados bit-blocks). Cada uno de estos grupos se codifica individualmente. El tamaño de estos bloques se debe escoger de acuerdo con lo expuesto en [Faloutsos, 87] para obtener una compresión óptima. Para cada bloque se crea una firma de tamaño variable y consistente de, como máximo, tres partes:

sistema	0000 0000 0000 0010 0000
recuperación	0000 0001 0000 0000 0000
información	0000 1000 0000 0000 0000
gestión	0000 0000 0000 0000 1000
Firma de bloque	0000 1001 0000 0010 1000

Figura 1-10: Método de compresión de ficheros de firmas ($b=20$, $w=1$)

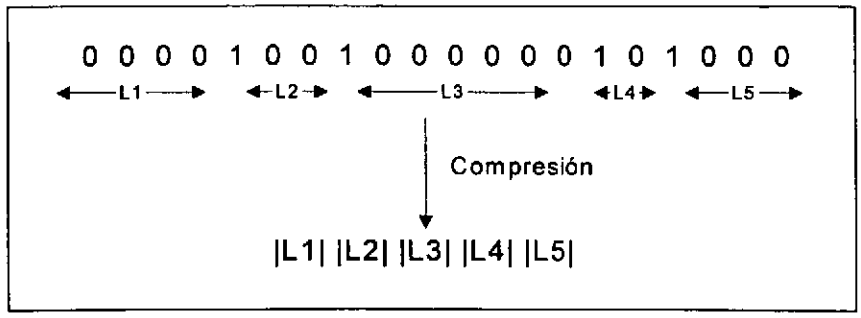

Figura 1-11: Compresión utilizando la codificación run-length

- Parte I: es un campo de un bit de longitud que indica si existe algún "1" en el bloque de bits, en cuyo caso se almacena un 1. Si el bloque está vacío, lo que se representa almacenando un 0, la firma del bloque finaliza aquí.
- Parte II: indica el número de "1"s en el bloque. Si en el bloque existen *s* bits a "1", se almacenan *s-1* "1"s consecutivos y se termina con un bit a "0".
- Parte III: contiene el desplazamiento de los bits "1" desde el principio del bloque de bits.

En la Figura 1-12 se ilustra como se realiza la compresión de la firma del documento de la Figura 1-10. Como se observa en la figura, el almacenamiento puede ser realizado de dos maneras: en la primera alternativa se almacena de manera consecutivas la información de cada bloque de bits, mientras que en el segundo caso el almacenamiento de la firma comprimida se realiza de tal forma que las primeras partes se almacenan de manera consecutiva, a continuación las segundas y por último las terceras partes. La segunda alternativa permite unos mejores tiempos de respuesta ante las búsquedas; por ejemplo, suponiendo que se intentan localizar firmas con el bit en la posición once a "1", examinando la parte I, el tercer bit, se concluye inmediatamente que el bloque de bits completo está vacío, por lo que la búsqueda en esa firma finalizaría.

Sobre este método Faloutsos y Christodoulakis proponen una ligera modificación para hacerlo insensible a modificaciones en el número de palabras por bloque, *D*. Esto supone una mejora ya que se elimina la necesidad de dividir los documentos en bloques lógicos, facilitando la resolución de consultas complejas. Este método fue denominado VBC (Variable bit-Block Compresión).

La idea consiste en utilizar un tamaño de bloque de bits (denominado b_{opt}) diferente para cada documento acorde con el número de bits a "1" en la firma (denominado *w*). El tamaño de las firmas (denominado *b*) se mantiene constante para todos los documentos. De esta manera lo que se consigue es que aquellos documentos cuya firma tenga poco peso, o lo que es lo mismo, un número reducido de "1"s, empleen un tamaño de bloque mayor, lo que implica un menor número de bloques de bits, una menor parte I, una menor parte II y menos desplazamientos en la parte III pero de mayor tamaño. Por otra parte, si la firma consta de muchos "1"s, el tamaño de bloque será menor, habrá mayor número de bloques y por consiguiente, la parte I y parte II serán mayores, mientras que la parte III tendrá más desplazamientos pero de menor tamaño [Faloutsos, 87].

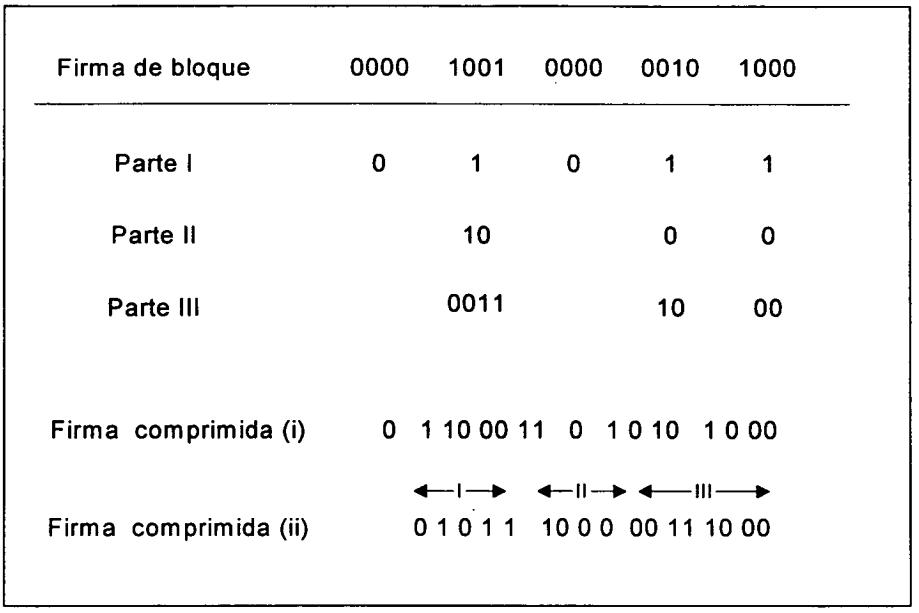

Figura 1-12: Método de compresión BC, con un tamaño de bloque de 4 bits

Respecto a la sobrecarga de espacio de los métodos BC y VBC, frente al SSF, en ambos casos el espacio requerido es menor para un valor constante de la probabilidad de falso acierto. Los tiempos de respuesta obtenidos son ligeramente inferiores al SSF debido a que es necesario un menor número de operaciones de entrada/salida. Por otra parte, las operaciones en memoria principal son ligeramente más complejas debidas a la compresión y descompresión, pero no constituyen el cuello de botella. Respecto a las inserciones, ambos métodos requieren un procesamiento previo para la realización de la compresión, pero el proceso de inserción en la estructura de datos es tan simple como en el método SSF [Faloutsos, 92a].

1.4.3.3. División vertical

La idea en la que se basan las técnicas de división vertical es evitar trasladar a memoria porciones del fichero de firmas que no serán útiles. Esto es posible almacenando la matriz de firmas invertida, considerando columnas de 1 bit [Roberts, 79][Faloutsos, 88], en lugar del método SSF que realiza un almacenamiento orientado a documentos; o según el modelo propuesto en [Lin, 88], que almacena la matriz invertida a nivel de conjuntos de bits, denominados "*frames*".

El método básico se denomina BSSF (Bit-Sliced Signature Files) y su estructura se muestra en la Figura 1-13. Para facilitar las inserciones se propone la utilización de *b* archivos diferentes, uno por cada bit de las firmas, denominados "*ficheros-bit*".

La búsqueda de una palabra requiere la recuperación de *w* vectores de bits (en lugar de los *b* vectores de bits del modelo SSF) sobre los que se realizará un AND lógico. El vector de resultado tendrá N bits (uno por cada bloque lógico) y aquellas posiciones con un "1" representarán a los bloques que se han clasificado. La inserción de un nuevo bloque lógico o documento requiere únicamente *b* accesos a disco, uno por cada fichero-bit.

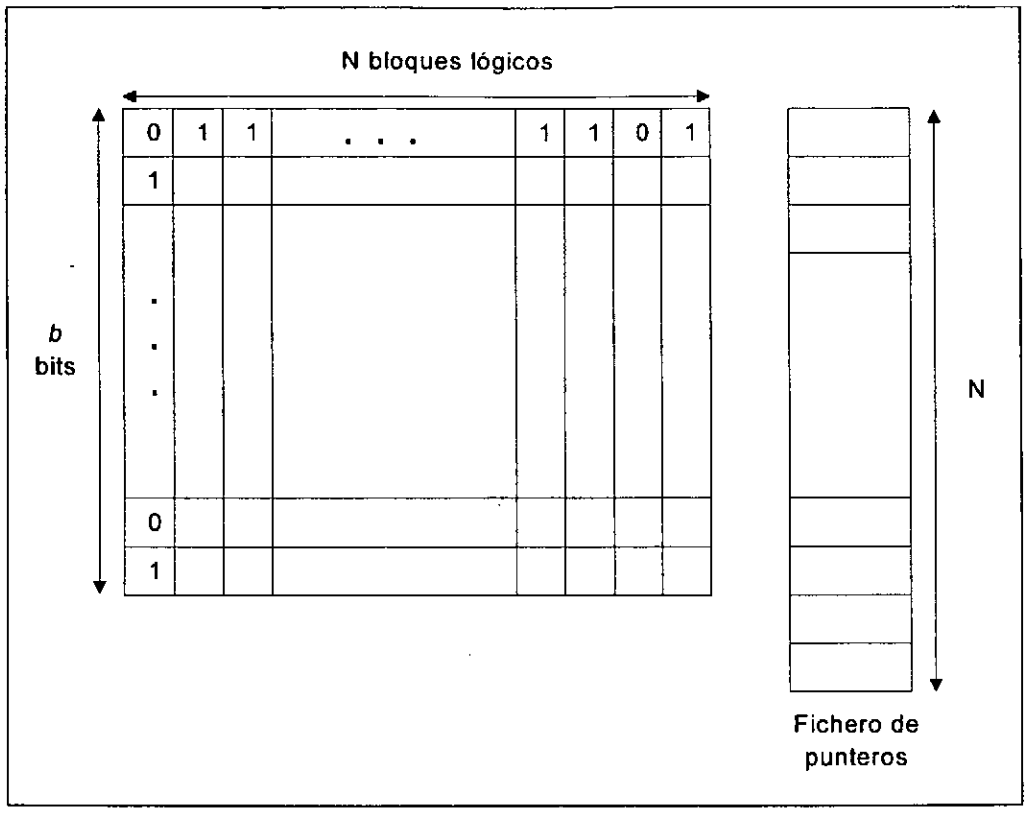

Figura 1-13: Estructura del método BSSF

El diseño tradicional de la técnica de ficheros invertidos sugiere la elección de un valor de w_{opt} con el objetivo de que las firmas de los documentos contengan aproximadamente un 50% de "1"s. Un valor típico para w en un entorno de un sistema de búsqueda es de 10 [Christodoulakis, 84], lo que implica en la técnica de BSSF se requieren 10 accesos a disco para una consulta de una palabra. En [Lin, 88], Lin y Faloutsos, sugieren utilizar un valor de w menor que el óptimo, de esta manera el número de accesos a disco se verá reducido. El principal problema, consiste en que para mantener la misma probabilidad de falsos aciertos es necesario aumentar el tamaño de las firmas de los documentos.

Por otra parte, una alternativa al método BSSF dentro de la división vertical es la conocida como FSSF (Frame-Sliced Signature File). La idea de este método consiste en que cada palabra se asocie con una firma que tenga los bits a "1" próximos entre sí dentro de la firma del documento [Lin, 88]. De esta manera, estos bits se almacenan de manera conjunta y pueden ser recuperados con muy pocos accesos aleatorios a disco.

La principal motivación de esta estructura es que los accesos aleatorios a disco son más costosos que los accesos secuenciales, ya que implican un movimiento de la cabeza lectora del disco.

Más detalladamente, el funcionamiento del método es como sigue: la firma de un documento, de b bits de longitud, se divide en k porciones de s bits consecutivos cada una de ellas. A estas porciones se las denomina "*frames*", de donde se deriva el nombre de este método. Para cada palabra del documento, una función hash selecciona uno de los k frames de la firma y a través de otra función hash la palabra establece w bits a "1" (no

necesariamente distintos) en ese frame. Los valores asignados a *b*, *k*, *s* y *w* son parámetros de diseño.

En la Figura 1-14 se muestra un ejemplo de aplicación de esta técnica de almacenamiento. En este caso, la firma está formada por 12 bits, que se dividen en 2 frames de 6 bits cada uno de ellos. A la palabra "*sistema*" se le asigna el segundo frame en donde se establecen 3 bits, mientras que la palabra "*recuperación*" se le asigna el primer frame en donde se establecen 3 bits.

Palabras	Firmas de palabra
sistema	000000 110010
recuperación	010110 000000
Firma de documento	010110 110010

Figura 1-14: Método de FSSF (*b=12*, *s=6*, *k=2* y *w=3*)

La matriz se almacena de manera vertical, pero almacenando de manera consecutiva no las columnas, sino los frames de la firma. De esta manera, ante una consulta de una palabra sólo es necesario recuperar un frame, por lo que sólo es necesario un acceso aleatorio a disco (y múltiples secuenciales). En caso de una consulta de *n* palabras, como máximo, será necesario acceder a *n* frames. Las inserciones serán mucho más rápidas que el método BSSF ya que sólo se requiere el acceso a *k* frames, frente a los *b* accesos del método anterior.

A nivel teórico es importante destacar el modelo GFSSF (Generalized Frame-Sliced Signature File), el cual partiendo del modelo FSSF, ofrece una aproximación más general al considerar que cada palabra selecciona *n* frames en los que establecerá *w* bits en cada uno de ellos (no necesariamente distintos). La firma del documento se obtendrá de la manera normal por medio de la realización del OR de las firmas de todas las palabras del documento [Faloutsos, 92a].

Es importante destacar que los métodos SSF, BSSF y FSSF son casos especiales del modelo GFSSF:

- Considerando *k=b* y *n=w* se obtiene el modelo BSSF, ya que se están considerando tantos frames como bits, en los cuales se establecerán w bits a "1".
- Considerando *n=1* se obtiene el modelo FSSF, ya que únicamente se selecciona un frame al generar la firma de la palabra.
- Considerando *k=1*, *n=1* se obtiene el método SSF, ya que la firma del documento se genera sobre un único frame.

En [Faloutsos, 92a] se exponen los resultados de la comparación del rendimiento ofrecido por los métodos de división vertical en base a un estudio teórico, en donde se comprueba que el modelo genérico GFSSF obtiene un mejor rendimiento que sus casos particulares BSSF y FSSF. De todas formas, es importante destacar que es necesario realizar una cuidadosa elección de los parámetros del GFSSF.

1.4.3.4. División vertical y compresión

Existen una serie de métodos que se basan en la división vertical de la matriz de firmas, pero aprovechando las técnicas de compresión. La idea básica consiste en la generación de una matriz de firmas difusa, con el objetivo de almacenar la matriz según un esquema de división vertical y comprimir cada columna mediante el almacenamiento de las posiciones de los "1"s [Faloutsos, 88]. Los métodos de esta clase están directamente relacionados con las técnicas de inversión utilizando una tabla hash.

El primero de estos métodos se denomina CBS (Compressed Bit Slices), y se basa en el método BSSF sobre el que se realiza una compresión de las columnas almacenadas. Aunque el método BSSF es más rápido que el método básico SSF en la realización de las búsquedas, se podría considerar una optimización, ya que es necesario recuperar w ficheros-bit para la búsqueda de una palabra, por lo tanto forzando el valor de $w=1$, únicamente sería necesario acceder a un fichero-bit. Y por otra parte, las inserciones son bastante costosas ya que son necesarios b accesos a disco, uno por cada fichero-bit.

Si se fuerza el valor de $w=1$ esto provoca que el valor de b deba aumentar para mantener constante la probabilidad de falsos aciertos. Esto genera una matriz de firmas y los ficheros-bit dispersos, con un número de "1"s muy reducido, por lo que se pueden aplicar técnicas de compresión. La forma más sencilla de compresión para cada fichero-bit o columna de la matriz de firmas consiste en almacenar las posiciones de los "1"s. Sin embargo, el tamaño de cada fichero-bit es impredecible, por lo que se almacenan en particiones de tamaño B_p, que se convierte en un parámetro de diseño. Según crece el tamaño de la columna serán necesarios más particiones, que se irán creando bajo demanda. Todas las particiones correspondientes a una columna se encuentran enlazadas por medio de punteros. Asimismo, es necesario un directorio, construido mediante una tabla hash, con b punteros hacia las particiones, uno por cada columna de la matriz de firmas.

Además, este método permite evitar la utilización del archivo de punteros, ya que en lugar de almacenar la posición de cada "1" es posible almacenar directamente un puntero al documento que contiene la palabra asociada a la firma. Asimismo, no es necesario realizar la división en bloques lógicos de los documentos, ya que el número de palabras por documento puede ser variable sin prejuicio en el rendimiento del método.

Por lo tanto, los ficheros-bit comprimidos contendrán punteros a los documentos apropiados. Al conjunto de todos los ficheros-bit o columnas comprimidas se les denomina nivel 1 u ocurrencias, por la similitud existente con la técnica de ficheros invertidos (ver sección 1.4.2). La lista de ocurrencias está constituida por particiones que contienen punteros a los documentos asociados, así como un puntero extra a la siguiente partición, en caso de ser necesario (ver Figura 1-15).

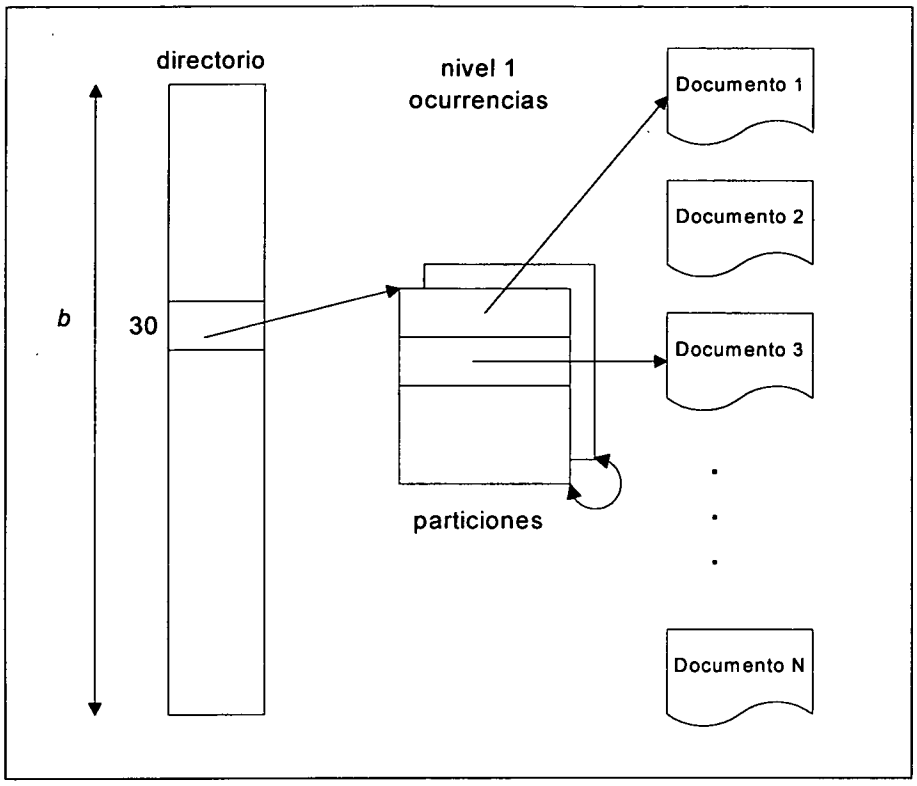

Figura 1-15: Método de CBS

El proceso de búsqueda se realiza mediante la aplicación de una función hash a la palabra buscada para obtener el bit activado de la firma. A continuación se accede al directorio o tabla hash para acceder a la partición con los punteros a los documentos relevantes.

Una variante del método CBS es el método DCBS (Doubly Compressed Bit Slices). En este método la estructura se compone de una tabla hash o directorio, un fichero intermedio, una lista de ocurrencias y el conjunto de documentos (ver Figura 1-16). El método es similar al CBS. En primer lugar se utiliza una función hash $h_1()$ que devuelve un valor en el rango *[0, b-1]* y determina una posición del directorio. A continuación, se utiliza otra función hash $h_2()$ para distinguir entre sinónimos (palabras que se correspondan con la misma posición de la tabla hash); esta función devuelve una cadena de bits de h bits de longitud. Estos códigos hash se almacenan en un fichero intermedio, formado por particiones de B_i bytes de longitud. Cada partición contiene registros con pares código-puntero, en donde el puntero se asocia con la cabeza de una lista enlazada de particiones con las ocurrencias.

Como se puede observar en la Figura 1-16, la búsqueda se realiza en tres pasos. En primer lugar, se aplica la primera función hash a la palabra para obtener la posición del directorio, que contendrá un puntero hacia las particiones del fichero intermedio. A continuación se aplica la segunda función hash que permitirá localizar la entrada correspondiente a esta palabra entre las particiones intermedias (en el ejemplo $h=3$, y se obtiene la cadena 011). Y finalmente, siguiendo el puntero de la partición intermedia correspondiente a las particiones de la lista de ocurrencias se pueden obtener los documentos que se clasifican, incluyendo tanto a aquellos documentos correctos como los falsos aciertos.

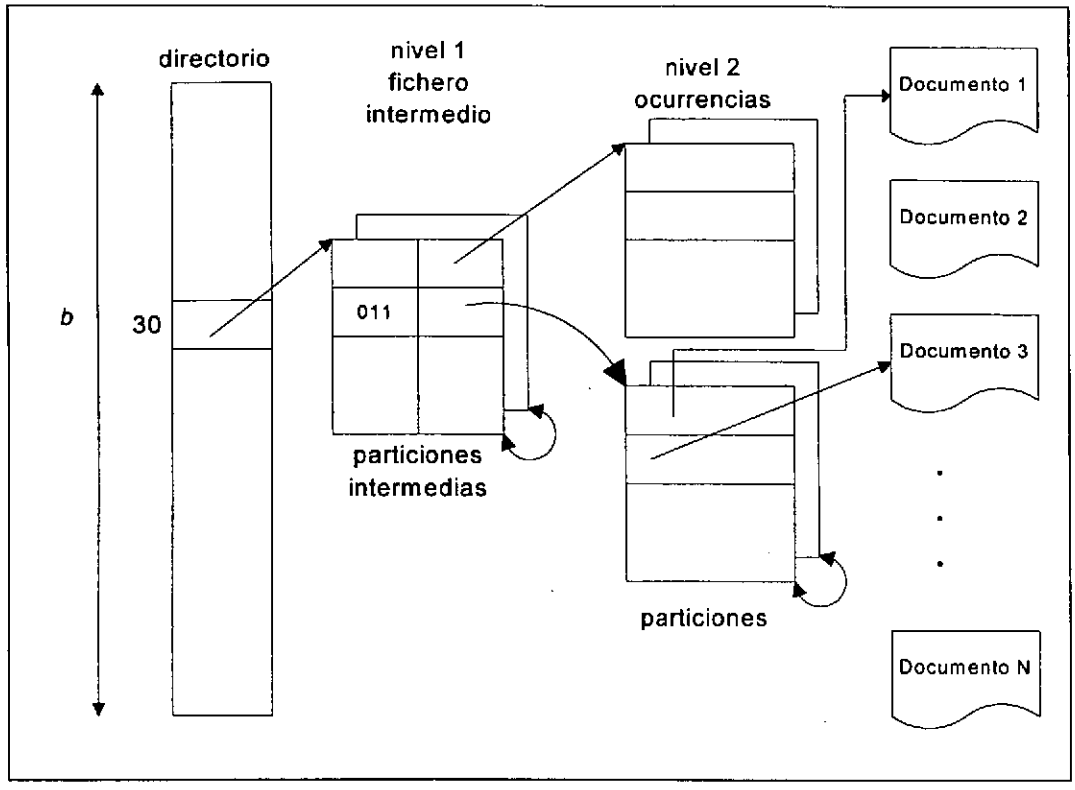

Figura 1-16: Método de DCBS

Una última variante de los métodos de división vertical y compresión es el NFD (No False Drop). Este método parte del DCBS y modifica el fichero intermedio para almacenar un puntero a la palabra en el texto, por lo que cada registro del fichero intermedio pasa a estar compuesto por el código hash de la palabra, un puntero a las particiones con las ocurrencias y un puntero a la palabra en el texto. De esta manera, se garantiza que cada palabra puede ser totalmente diferenciada de sus sinónimos y por lo tanto la probabilidad de falsos aciertos es nula.

A nivel general, el rendimiento ofrecido por estos tipos de métodos es bueno. Los tiempos de respuesta ante las búsquedas son más rápidos que los métodos de división vertical sin comprimir ya que se requiere un menor número de accesos a disco, introducen una sobrecarga para las estructuras de datos de entre el 20 y 25%. Las inserciones siguen sin necesitar una modificación de la estructura de datos creada, basta con una operación de inserción al final del archivo de datos [Faloutsos, 88].

1.4.3.5. División horizontal

Los métodos que realizan una división horizontal del fichero de firmas tratan de evitar la búsqueda secuencial del fichero de firmas completo, con el objetivo de intentar mejorar los tiempos de respuesta. Así pues, se agrupan las firmas en conjuntos, dividiendo la matriz de firmas de manera horizontal.

El criterio bajo el que se agrupan las firmas establece dos tipos de técnicas: aquellas independientes de los datos y las dependientes de los datos. En el caso en que el criterio de agrupación ha sido decido a priori (por ejemplo, utilizando una función hash) constituye

los métodos independientes de los datos, mientras que los grupos pueden ser definidos al mismo tiempo que la estructura de datos se va construyendo lo que constituye los métodos dependientes de los datos.

La primera aproximación a este tipo de métodos fue la propuesta por Gustafson en [Gustafson, 71] y se basa en la aplicación de la combinatoria a la técnica de los ficheros de firmas (ver Figura 1-17). Por ejemplo, considerando que los documentos están compuestos por palabras clave que los describen, se puede diseñar una función hash que asocie una palabra clave con un número en el rango desde 0 hasta 15. Por lo tanto, la firma de la palabra es una cadena de 16 bits, de los cuales está a "1" únicamente aquel marcado por la función hash. La firma del documento consiste en la superposición de las firmas de sus palabras clave. Si cada documento tiene asociadas 6 palabras clave (o en caso de tener un número menor, se generan los restantes bits a "1" al azar), esto produce un total de $C(16, 6)=8.008$ combinaciones posibles[†].

El aspecto más interesante de este método consiste en que el número de documentos recuperados en una consulta decrece rápidamente (casi exponencialmente) con el número de palabras indicadas en una consulta conjuntiva. Por ejemplo, si la búsqueda está formada por una palabra serán necesarios $C(15, 5)=3.003$ registros de la tabla hash, mientras que si está formada por dos palabras serán necesarios $C(14, 4)=1.001$ registros, y así sucesivamente. De todas formas, el método de Gustafson presenta ciertos problemas: las consultas que no son conjuntivas son difícilmente gestionadas; el rendimiento del sistema se deteriora según aumenta el tamaño del fichero de firmas; si el número de palabras clave por documento es elevado, incluso para búsqueda de varias palabras clave el resultado podría ser demasiado amplio.

Por otra parte, Lee y Leng en [Lee, 89] y [Lee, 90] proponen realizar una división horizontal de los ficheros de firmas, tanto para entornos secuenciales como paralelos. Para ello sugieren emplear una parte de la firma de cada documento como clave para la división del fichero de firmas, de esta manera aquellas firmas que tengan la misma clave se almacenan conjuntamente en una misma partición (ver Figura 1-17). Para procesar una consulta, en primer lugar es necesario determinar la clave de la firma de la palabra buscada y examinar únicamente las particiones asociadas.

Para la selección de la clave proponen tres métodos: división por prefijo fijo, división por prefijo extendido y división por clave flotante. El primero de los métodos, división por prefijo fijo, es el más sencillo de todos. Los primeros k bits de la firma constituyen la clave. Aquellas firmas que tenga el mismo prefijo se agrupan en la misma partición (ver Figura 1-18(a)). En el caso de una búsqueda se toman los primeros k bits para determinar la clave, y la búsqueda se restringe a las particiones que contienen la clave de la búsqueda.

El método de división por prefijo extendido intenta conseguir una distribución de la carga uniforme, evitando que haya particiones que se activen en todas las búsquedas. Para asegurar esto, basta con que cada clave contenga al menos un cero. En este método, la clave de una firma es el menor prefijo que contiene el número de ceros predefinido, denominado z (ver Figura 1-18(b)).

[†] $C(m, n)$ representa las combinaciones de m elementos tomados de n en n.

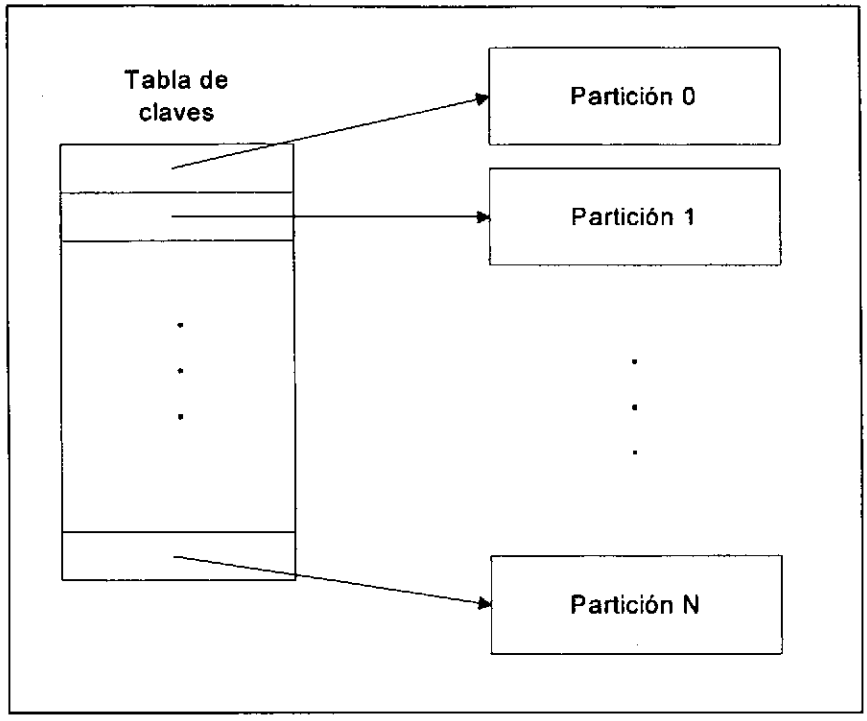

Figura 1-17: Método de división horizontal de fichero de firmas en particiones

En el método de claves flotantes se consideran claves de tamaño fijo, *k* bits, pero no necesariamente prefijos, sino que por el contrario, las claves se seleccionan como aquella parte de la firma de *k* bits con el menor número de unos. Concretamente, se examinan las partes de la firma de izquierda a derecha, consecutivas, de *k* bits de longitud y no superpuestas, y aquella cadena más a la izquierda con el menor peso constituye la clave. La clave en este método está constituida por el valor de la clave y la posición de la clave. Este método intenta maximizar el número de ceros en las claves, ya que de esta manera se minimiza el tamaño del conjunto de particiones utilizadas en una búsqueda (ver Figura 1-18(c)).

En el trabajo de Lee y Leng se analiza el rendimiento de los tres métodos, el primero de ellos es el más simple y ofrece un buen rendimiento ya que únicamente una partición es buscada por palabra, aunque la probabilidad de activación de cada partición no es uniforme. El método de prefijo extendido presenta ciertos problemas derivados del relativamente gran número de particiones generadas y el tamaño de las mismas no está uniformemente distribuido, aunque la probabilidad de utilización de cada partición está correctamente distribuida. Por último el método de claves flotantes es el más complejo, si bien el tamaño de las particiones generadas no es muy variable y la probabilidad de activación de cada partición es mejor que en el caso de prefijos fijos.

Respecto a los métodos de división horizontal dependientes de los datos, el más interesante es el propuesto en [Sacks-Davis, 83] y [Sacks-Davis, 87]. En estos trabajo se sugiere la utilización de dos niveles de firmas y que para la asignación de las firmas se tenga en cuenta la frecuencia de repetición de cada palabra. En un segundo nivel, las firmas de los documentos se almacenan de manera secuencial siguiendo el modelo SSF. En el primer nivel se generan firmas de bloque, que son creadas superponiendo las firmas de todas las palabras en ese bloque, sin tener en cuenta los límites de cada registro. En el primer nivel

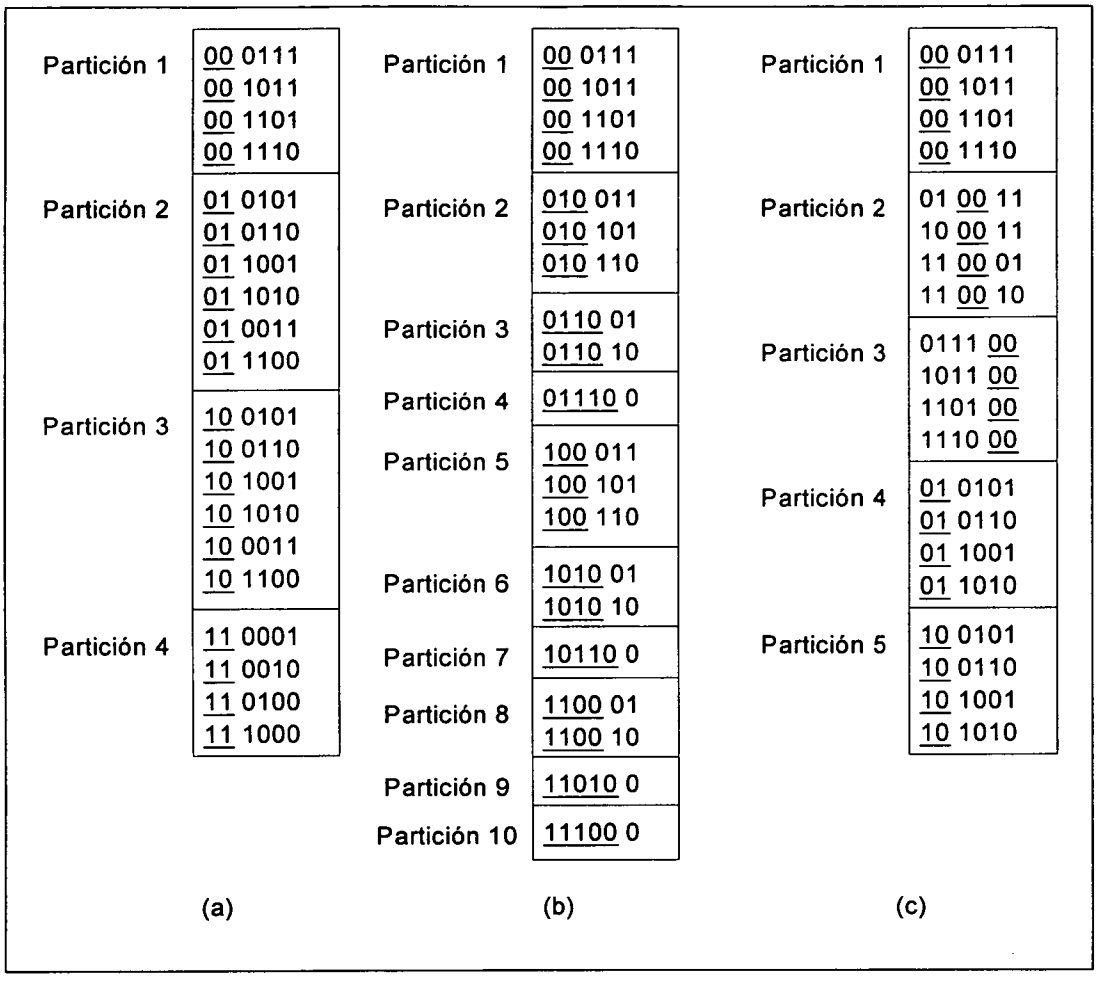

Figura 1-18: Distribución de firmas basado en (a) prefijo fijo, (b) prefijo extendido y (c) clave flotante

se almacena la matriz de firmas de manera invertida, como en el modelo de división vertical.

Para el procesamiento de una búsqueda es necesario la generación de dos claves: la clave correspondiente a los bloques y la clave correspondiente al segundo nivel de la estructura. La búsqueda se realiza en primer lugar examinando las firmas de bloques, y a continuación concentrándose únicamente en aquellas porciones del segundo nivel de firmas que se han clasificado. En el análisis de los tiempos de respuesta realizados se obtienen unas mejoras sustanciales respecto al modelo BSSF, sin embargo, en el caso de consultas conjuntivas compuestas por múltiples palabras, los bloques pueden generar falsos aciertos por contener las palabras buscadas, pero no en el mismo registro. Los autores proponen determinadas formas de intentar minimizar la probabilidad de falsos aciertos en los bloques de firmas.

Asimismo en [Lee, 95] se propone una generalización del modelo anterior denominada método de codificación superpuesta basado en múltiples niveles. Este método se centra en la generalización a múltiples niveles, no únicamente a dos como el caso anterior.

Asimismo, Deppisch en [Deppisch, 86] propone un nuevo método de división horizontal dependiente de los datos basado en una estructura de datos similar a un árbol B para proporcionar un acceso más eficiente al fichero de firmas, denominados árboles S. Los

nodos hoja de estos árboles están compuestos por un número constante de firmas similares. La firma de cada nodo se realiza con el OR de las firmas de sus hijos, recursivamente hasta alcanzar la raíz del árbol. Además este tipo de árboles se mantiene balanceado de la misma manera que los árboles B.

Este método presenta una baja sobrecarga de almacenamiento, mientras que los tiempos de respuesta son difíciles de estimar analíticamente. Sin embargo, a través de este método se pierde la propiedad de la sencillez de las inserciones característica de los ficheros de firmas, ya que puede ser necesario realizar una re-estructuración del árbol S.

1.4.4. Clustering

La principal idea de la técnica de clustering consiste en agrupar a los documentos similares. Esta técnica se basa en la hipótesis de grupo: documentos fuertemente relacionados tienden a ser relevantes a las mismas consultas. Por lo tanto, la agrupación de documentos similares acelerará las búsquedas.

La técnica de clustering ha sido estudiada en el entorno de la recuperación de información y del entorno bibliotecario [Salton, 69][Rijsbergen, 79]. Es importante destacar que esta técnica puede ser aplicada tanto a documentos como a términos. En el segundo caso se permite la creación de clases de términos que son normalmente relevantes entre sí o sinónimos. En cualquier caso, la descripción de las técnicas de agrupamiento se enfocará hacia su aplicación en el almacenamiento y posterior recuperación de documentos.

La técnica de clustering involucra dos procesos, por un lado la generación de un conjunto de clusters y por otro la búsqueda en los clusters. A continuación se realiza un análisis detallado de las técnicas asociadas a ambos tipos de métodos.

1.4.4.1. Métodos de generación de clusters

Los métodos de generación de clusters se basan en la representación de un documento como un vector, el cual es procesado y se le asignan determinadas palabras clave. Esto constituye el proceso de indexación, que puede ser realizado manual o automáticamente.

El proceso de indexación es similar al descrito en la sección 1.4.2, empleando un diccionario de palabras comunes, una lista de sufijos y prefijos para reducir cada palabra a su raíz y un diccionario de sinónimos para facilitar la asignación de cada término con su concepto asociado [Salton, 71].

Cada documento es representado con un vector t-dimensional, en donde t representa el número de términos indexados o conceptos máximo. La ausencia de un término se indica con 0 en el vector, mientras que la presencia del término se indica con un 1 (constituyendo un vector de documento binario) o mediante un entero positivo que representa el peso del término, que refleja la importancia del término en el documento. En la literatura es posible encontrar múltiples funciones de peso, de las cuales las más relevantes son las siguientes:

- Frecuencia del término k en el documento i. Es una medida fácil de obtener y que presenta un rendimiento más efectivo que un peso binario.

- Frecuencia de documento inverso. Se define como la frecuencia del término *k* en el documento *i*, dividido por el número de documentos en donde aparece el término *k*. Representa una medida más efectiva que la anterior [Salton, 83].

Estos procedimientos permiten la representación de los documentos como puntos en un espacio *t*-dimensional. El siguiente paso en la generación del cluster consiste en la división de los puntos en grupos. Esta división debe ser válida, correcta y al mismo tiempo eficiente. Los criterios para una división correcta se establecen en [Rijsbergen, 79] y son los siguientes: (i) el método debe ser estable aún con el crecimiento del número de documentos, las particiones no deben cambiar drásticamente con la inserción de nuevos documentos; (ii) pequeños errores en la descripción de los documentos deben producir pequeños cambios en las divisiones; (iii) el método debe ser independiente de la ordenación de los documentos. El principal criterio que mide la eficiencia de un método de generación de clusters se basa en el tiempo invertido, siendo despreciable el espacio de almacenamiento necesario para la implementación del método.

Existen múltiples métodos de generación de clusters, sin embargo ninguno combina adecuadamente las características de corrección y eficiencia [Faloutsos, 85a]. En concreto se definen dos familias de métodos: aquellos basados en una matriz de similitud de documentos (en donde prima el criterio de corrección) y los métodos iterativos en donde prima la eficiencia.

Los métodos basados en una matriz de similitud de documentos aplican técnicas de teoría de grafos y necesitan un tiempo cuadrático al número total de documentos. Es necesaria una función de similitud entre documentos que permite determinar cuan cerca se encuentran dos documentos relacionados. Existen múltiples funciones de similitud en la literatura [Salton, 83], si bien todas ellas proporcionan un rendimiento similar.

A partir de la matriz de similitud, se establece un umbral de tal forma que si dos documentos tienen una medida de similitud superior al umbral se conectan por medio de un eje. Los componentes que estén conectados entre sí serán los grupos obtenidos como resultado. La posterior recuperación de información puede ser mejorada si se crean jerarquías de clusters realizando grupos de clusters de manera recursiva. Un método posible para esto consiste en la utilización de umbrales decrecientes, aunque Van Rijsbergen propone en [Rijsbergen, 79] un método basado en la proximidad de documentos vecinos.

El principal problema de este tipo de métodos es la necesidad de utilizar un umbral determinado empíricamente para la construcción de los grupos. Además, el valor del umbral afecta directamente a la división final imponiendo una estructura a los datos, en lugar de detectar la estructura subyacente en los datos. A este respecto en [Zahn, 71] se propone un método de generación de clusters en donde la influencia del umbral de similitud parece minimizada.

Por otra parte, los métodos iterativos se caracterizan por que ofrecen tiempos proporcionalmente menores al cuadrado de los documentos, típicamente $O(n \log n)$ o $O(n^2 / \log n)$. Este tipo de métodos se basa directamente en las descripciones de los documentos y no requieren el cálculo previo de la matriz de similitud. En cambio, en este tipo de métodos la clasificación final depende del orden en el que son procesados los documentos. Además, requieren la determinación empírica de ciertos parámetros como son: el número

de clusters que se van a generar, el tamaño máximo y mínimo de cada cluster, un umbral de la similitud entre documentos y clusters, un control de superposición entre clusters (en caso de estar permitida) y una función objetivo que será optimizada.

El procedimiento a nivel general de los métodos iterativos consiste en un primer paso que determinará una partición inicial y a continuación, y de manera iterativa, se irán reasignando documentos a clusters hasta que se considere que los documentos se encuentran en los grupos adecuados. El método más rápido y simple es el expuesto en [Salton, 78] en donde cada documento es procesado una única vez y es asignado a alguno de los clusters (o más si está permitido) existentes o crea un nuevo cluster. Asimismo, este tipo de métodos se suelen emplear en colaboración con los métodos de matriz de similitud, utilizando los métodos iterativos para una primera partición de documentos y a continuación aplicar técnicas de teoría de grafos para subdividir cada uno de los clusters creados [Salton, 83].

Los métodos expuestos se basan en criterios estadísticos para determinar la similitud entre documentos, sin embargo existe una familia de técnicas que se basan en un "espacio de documentos dinámicos" para la generación de los clusters [Friedman, 71]. En este método los vectores de documentos relevantes son modificados y aproximados al vector de consulta, mientras que los vectores de documentos no relevantes son alejados del vector de consulta.

Las principales ventajas de este método son una mejora de la efectividad para consultas homogéneas y similares y la facilidad para gestionar la inserción y borrado de documentos. Por otra parte, se requiere un procesamiento adicional por cada consulta que es realizada y el resultado obtenido por una misma consulta puede variar en el tiempo, lo cual puede no ser bien aceptado por los usuarios.

1.4.4.2. Búsqueda de clusters

El proceso de búsqueda en una agrupación de clusters se basa en la consulta que es representada como un vector t-dimensional, el cual es comparado con cada uno de los clusters generados. Para la realización de la comparación es necesario definir una función de similitud entre la consulta y un cluster, para lo cual existen varias formulaciones propuestas.

La función de similitud más sencilla es la función coseno del ángulo entre dos vectores. Sea d_j el vector del documento j y q el vector representando la consulta, entonces la similitud entre d_j y q se define como [Baeza-Yates, 99c]:

$$sim(d_j, q) = \frac{\vec{d_j} \bullet \vec{q}}{\left|\vec{d_j}\right| \times \left|\vec{q}\right|} = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,q}^2}}$$

Sobre esta función de similitud básica se proponen modificaciones, como la expuesta en [Yu, 77] basada en la estimación previa del número de documentos relevantes en cada cluster, para continuar la búsqueda únicamente en aquellos clusters con suficientes

documentos. También en [Croft, 80] se plantea la utilización de técnicas de reconocimiento de patrones para la búsqueda en clusters.

De igual forma, la búsqueda en clusters potencia la utilización de técnicas de retroalimentación, en donde el usuario una vez que una primera consulta ha sido realizada selecciona aquellos documentos que considera relevantes, y el sistema, en base a esta nueva información, reformula el vector de consulta para realizar una nueva búsqueda. La reformulación de la consulta se realiza normalmente sumando al vector de consulta los vectores de los documentos relevantes y restando aquellos documentos no relevantes [Rocchio, 71]. En este tipo de técnicas son necesarias dos o tres iteraciones para obtener resultados notables.

1.5. Recuperación de información en el Web

El World Wide Web se originó a finales de los años 80 [Berners-Lee, 94] y ninguno de sus creadores pudo imaginar el impacto que tendría. La cantidad de información textual disponible estaba estimada del orden de 6 terabytes a mediados de 1999, a través de más de 800 millones de páginas distribuidas en más de 3 millones de servidores Web [Agosti, 01]. Además hay que tener en cuenta que otros medios, como imágenes, audio, vídeo, están también disponibles, lo que convierte al World Wide Web en una gran y ubicua base de datos sin estructura, sobre la que los usuarios desean localizar información, para lo que son necesarias herramientas que permitan gestionar, recuperar y filtrar la documentación disponible en esa gran base de datos.

De hecho, nadie sería capaz de localizar determinada información simplemente explorando la inmensidad del Web, en un tiempo razonable. Por el contrario son necesarias herramientas que ayuden a los usuarios a localizar aquellas páginas Web más relevantes para sus necesidades de información. En concreto, pocos años después del inicio y extensión del World Wide Web surgieron los primeros sistemas de recuperación de información en el WWW.

Básicamente se definen tres formas diferentes de sistemas de búsqueda en el Web. Las dos primeras se caracterizan por disponer de sus propias estructuras de datos sobre las que realizan el proceso de búsqueda, mientras que el tercer tipo carece de dichas estructuras. El primer tipo son los denominados robots o motores de búsqueda que intentan indexar la totalidad del Web como una base de datos de texto. El segundo se corresponde con los directorios Web, que clasifican los documentos Web en una ontología. El tercer tipo está representado por los metabuscadores o multibuscadores, que acceden a otros buscadores para la obtención de los resultados de la búsqueda, realizando una combinación del total de resultados obtenidos.

En el siguiente apartado se describen en detalle las características que convierten en únicos a los sistemas de recuperación de información en el Web, frente a los sistemas de recuperación de información tradicionales. A continuación, se examinan los tres tipos de sistemas de recuperación.

1.5.1. Características

Los sistemas de recuperación de información en el Web presentan unas características completamente diferentes a los sistemas de recuperación de información tradicionales, principalmente debido a las propias características de su entorno de trabajo: el World Wide Web. Asimismo, los propios usuarios de estos sistemas de recuperación presentan unas características totalmente diferentes a sus homólogos de sistemas tradicionales, aspecto que será discutido en detalle en el capítulo 2.

En concreto, las principales características que convierten a los sistemas de recuperación de información en el Web en únicos son las siguientes [Huang, 00][Baeza-Yates, 99b]:

- **Volumen de datos**. Por una parte, el volumen de información disponible hoy en día en Internet hace que estos sistemas de información deban tratar cantidades ingentes de datos que se escapan del contenido tratado por los sistemas tradicionales. Por otra parte, el crecimiento exponencial del World Wide Web presenta para los sistemas de recuperación unos aspectos de escalabilidad complejos de resolver. El número de servidores Web está estimado en 2,4 millones según NetSizer [NetSizer, 01], el número de páginas Web disponibles estaba estimado en 350 millones de páginas en Julio de 1998 [Bharat, 98], mientras que estimaciones más recientes cifran el número de páginas en 800 millones de páginas, con 6 terabytes de texto asociados a mediados de 1999 [Agosti, 01], y en un valor cercano a los 1.700 millones de páginas hoy en día.
- **Dinamismo**. El contenido del World Wide Web está cambiando día a día, con apariciones de nuevas páginas, modificaciones de páginas existentes y eliminaciones de páginas obsoletas. De hecho, se estima que el 40% del Web cambia cada mes [Kahle, 97]. Los sistemas tradicionales están diseñados para el tratamiento de bases de datos textuales con una gran componente estática.
- **Heterogeneidad**. En el Web se encuentran disponibles gran variedad de tipos de documentos: páginas Web, documentos textuales puros, imágenes, ficheros de audio, vídeos, etc. Por otra parte, existen documentos escritos en multitud de lenguajes, de hecho, se estima que en Internet se emplean más de 100 lenguajes diferentes.
- **Distribución**. Por las características intrínsecas del World Wide Web, los datos se encuentran dispersos sobre millones de ordenadores y plataformas. Estos ordenadores están interconectados sin ninguna topología predefinida, el ancho de banda y la fiabilidad de cada red de interconexión sufre grandes variaciones. El Web está basado en los hiperenlaces y esto es lo que constituye y conforma la red de documentos. De hecho, de media, cada página Web presenta más de 8 enlaces a otras páginas Web.
- **Redundancia y falta de estructura**. La mayoría de los sistemas tradicionales de recuperación de información parten de una base de datos o al menos documentos estructurados. En cambio, en el Web cada documento HTML ha sido elaborado de manera independiente, por lo que no existe una estructura de definición de documentos. Además, es necesario tener en cuenta los datos duplicados, ya que se estima que aproximadamente el 30% de las páginas Web son duplicadas [Broder, 97][Clarke, 95], y esto sin tener en cuenta detalles de redundancia semántica.

- **Calidad**. El Web se considera como un nuevo medio de publicación, sin embargo, no existe ningún proceso editorial previo, lo que provoca que el contenido informativo no siempre sea de una calidad adecuada. En consecuencia, los datos disponibles en el Web pueden ser falsos, inválidos, obsoletos, con errores de escritura (léxicos, gramaticales, etc.). Se estima que los errores tipográficos para palabras comunes se producen en 1 de cada 200, y en el caso de palabras más complejas (por ejemplo, apellidos extranjeros) en 1 de cada 3 [Navarro, 98].
- **Usuarios**. Los usuarios de un sistema de recuperación de información en el Web presentan un comportamiento totalmente diferente a los usuarios tradicionales en muchos aspectos. A nivel general, los usuarios buscarán una gran variedad de conceptos sin ningún tipo de nexo común, las consultas se realizarán de forma vaga y difusa, especificando muy pocas palabras clave y comprobando muy pocos resultados.

Estos aspectos constituyen las principales diferencias entre los sistemas de recuperación en el Web y los tradicionales, lo que también marcará ciertas diferencias en las estructuras de datos empleadas para la búsqueda. De hecho, inicialmente los sistemas de búsqueda en el Web se basaron en la tecnología desarrollada para el entorno de recuperación de información en medios estáticos y de poco volumen de datos. Sin embargo, cada día se hace más patente la necesidad de aplicar nuevas técnicas de indexación y representación de los datos para mejorar el rendimiento de este tipo de sistemas, aspecto en donde se centra el presente trabajo de tesis doctoral.

1.5.2. Tipos de motores de búsqueda

En el entorno de los sistemas de recuperación de información para el Web se consideran típicamente tres tipos básicos: robots, directorios y metabuscadores [Baeza-Yates, 99b].

Cada uno de estos sistemas presenta unas características particulares y permite resolver un tipo determinado de consultas. A nivel general, los robots gestionan un gran volumen de información, aunque la calidad de su contenido no siempre está garantizado, mientras que los directorios Web realizan una clasificación de páginas Web por lo que la calidad de su contenido es muy elevada, mientras que la cantidad es muy reducida. El caso de los metabuscadores es atípico y especial, ya que no disponen de ninguna estructura de datos propia, sino que acceden a otros sistemas de búsqueda para realizar las consultas de los usuarios.

El principal problema que presentan este tipo de sistemas en conjunto es el ocultismo que los rodea sobre las técnicas y tecnologías empleadas para su adecuado funcionamiento. Aunque existen casos puntuales de investigadores que publican trabajos relacionados con sistemas comerciales, la tónica general es la falta de información, principalmente por motivos comerciales a fin de evitar la copia o imitación, por parte de competidores, de los mecanismos empleados. A pesar de esto, en los siguientes apartados se describen las estructuras de datos empleadas por estos tipos de sistemas, basándose en diferentes investigaciones publicadas al respecto.

1.5.2.1. Robots

Estos sistemas se caracterizan por modelar el Web como una base de datos de texto. La principal diferencia con los sistemas de recuperación de información tradicionales consiste en que las consultas se deben realizar sin tener acceso al texto, utilizando únicamente los índices disponibles, ya que, de otra manera, sería demasiado costoso archivar una copia de los documentos del Web o demasiado lento el acceso a las páginas Web.

En este tipo de sistemas de búsqueda prima la cantidad de información disponible para sus usuarios. El objetivo ideal consiste en la indexación de la totalidad del Web, debiendo para ello disponer de estructuras de datos adecuadas para soportar este volumen de información. Estos sistemas son apropiados para búsquedas específicas o concretas, ya que en otro caso la cantidad de resultados puede desbordar al usuario y la calidad de las respuestas puede no ser siempre la óptima.

La mayor parte de los robots disponibles en el Web utilizan una arquitectura centralizada, en donde los robots recorren el Web enviando páginas a un servidor central en donde son indexadas. Estos robots, también denominados crawlers, spiders, wanderes o walkers, no son más que agentes software que envían peticiones a los servidores Web. La mayoría utilizan una estructura de fichero invertido para la indexación del texto recibido de los robots, índice que es almacenado de manera centralizada para responder a las consultas de los usuarios.

También existen robots que presentan una solución distribuida para la resolución del problema, como el caso de Harvest [Bowman, 94]. Este sistema se basa en una arquitectura distribuida para reunir y distribuir los datos, aunque presenta el inconveniente de que requiere la coordinación de múltiples servidores Web para su puesta práctica, algo que hoy en día es impensable.

La principal potencia de este tipo de sistemas de búsqueda se centra en el volumen de información que examinan, lo que los convierte en ideales para búsquedas específicas. En cambio, su principal inconveniente está basado en la poca calidad de alguna de la información indexada, lo que puede provocar la aparición de documentos de baja calidad entre los resultados.

El primer paso para cualquier robot de búsqueda consiste en la obtención del primer conjunto de URLs y recorrerlas de manera adecuada, aspecto que puede mejorar las páginas indexadas [Cho, 98]. Además es necesario tener en cuenta el estándar de exclusión de robots [Koster, 94] que permite a un administrador restringir el acceso a su sitio Web a este tipo de agentes.

Sin embargo, el aspecto principal de cualquier sistema de búsqueda se encuentra en la estructura de datos empleada, y es el aspecto central de esta disertación. En concreto, los robots se caracterizan por emplear variantes de la técnica de fichero invertido (ver sección 1.4.2) [Baeza-Yates, 99b]. Actualmente, el tamaño de un fichero invertido se sitúa en aproximadamente el 30% del tamaño del texto, lo que implica que para un conjunto 100 millones de páginas Web se genera un índice de aproximadamente 150 gigabytes, aunque este tamaño puede ser reducido utilizando técnicas de compresión [Witten, 94].

El proceso de búsqueda se realiza en base a una búsqueda binaria en el vocabulario de la estructura de fichero invertido, y en caso de que la consulta esté constituida por múltiples vocablos, los resultados individuales deben ser combinados para obtener la respuesta final. Esta etapa de combinación de resultados puede ser poco eficiente si los resultados individuales son muy numerosos [Baeza-Yates, 99b]. Otros aspectos más específicos de las búsquedas, como las búsquedas por proximidad, no son implementados por algunos robots ya que su inclusión en el índice es demasiado costoso a nivel de espacio de almacenamiento, aunque sí existen motores de búsqueda que proporcionan esta utilidad, sin embargo los detalles de implementación no han sido publicados. Por último, antes de mostrar los resultados se realiza una ordenación en base al ajuste de cada documento a la consulta, con el objetivo de mostrar al usuario únicamente aquellos más relevantes.

Estos pasos constituyen, a nivel general, las tareas de un motor de búsqueda genérico para el Web. A continuación, y de manera más detallada, en base a la información publicada en [Brin, 98] y [Page, 98] se describe uno de los paradigmas de estos sistemas de búsqueda en el Web hoy en día, Google [Google, 01].

El motor de búsqueda se basa en un sistema constituido por múltiples PCs, estimados en aproximadamente 6.000 máquinas. En la Figura 1-19 se muestra la arquitectura del sistema de Google. La descarga de las páginas Web se realiza por medio de varios robots distribuidos, coordinados por medio de un servidor de URLs. Las páginas obtenidas se envían al "*storeserver*" que las comprimirá y almacenará en un repositorio. Cada página se identifica por medio de un identificador de documento (denominado docID). El indexador se encarga de leer del repositorio un documento, descomprimirlo y analizarlo sintácticamente. Cada documento se convierte en un conjunto de ocurrencias de palabras denominadas "*hits*". Un hit representa a la palabra, su posición en el documento y una aproximación del tamaño de la fuente y otras características. El indexador distribuye estos hits en un conjunto de barriles, creando un primer índice parcialmente ordenado. Además el indexador realiza otra importante función: durante el análisis sintáctico obtiene los enlaces entre páginas y almacena esa información (enlace más texto del enlace) en un fichero de enlaces.

El "*URLresolver*" a partir del fichero de enlaces convierte las URLs relativas en URLs absolutas y a su vez en identificadores de documentos, inserta el texto del enlace en el índice y genera una base de datos de pares de identificadores de documentos que representan a los enlaces, lo que será utilizado para el cálculo del PageRank. El "*sorter*" accede al contenido de los barriles y los reordena por identificador de palabra para generar el índice invertido, a partir del cual se creará el nuevo vocabulario.

El buscador utiliza directamente el vocabulario, el índice invertido y el PageRank para responder a las consultas de los usuarios.

El sistema de búsqueda empleado por Google se basa en diversas estructuras de datos, algunas de las cuales tienen un papel más preponderante en las tareas de indexación off-line, mientras que otras se emplean activamente en la búsqueda. Las estructuras de datos definidas en la arquitectura de alto nivel son las siguientes:

- **El repositorio**: contiene el texto HTML completo de todas las páginas Web. Cada página se almacena comprimida, archivando además el identificador de documento, la longitud y su URL.

- **El índice de documentos**: almacena información sobre los documentos, incluyendo estado del documento, un puntero al repositorio, un checksum y varias estadísticas.
- **Vocabulario**: contiene la totalidad de las palabras indexadas. Se mantiene en memoria principal para agilizar las búsquedas.
- **Lista de hits**: se corresponde con una lista de ocurrencias de una palabra concreta en un documento concreto, incluyendo información sobre posición, fuente y características del texto (mayúsculas, minúsculas, etc.). Esta lista se almacena comprimida utilizando un algoritmo de compresión especialmente diseñado para este caso.
- **Índice avanzado**: se denomina al índice almacenado en los barriles (constituido por 64 elementos). Cada barril almacena un rango de identificadores de palabras. Si un documento contiene palabras que se corresponden con un barril, se almacena su identificador de documento, junto con la lista de palabras con sus listas de hits correspondientes.
- **Índice invertido**: es la estructura de datos empleada directamente por el buscador.

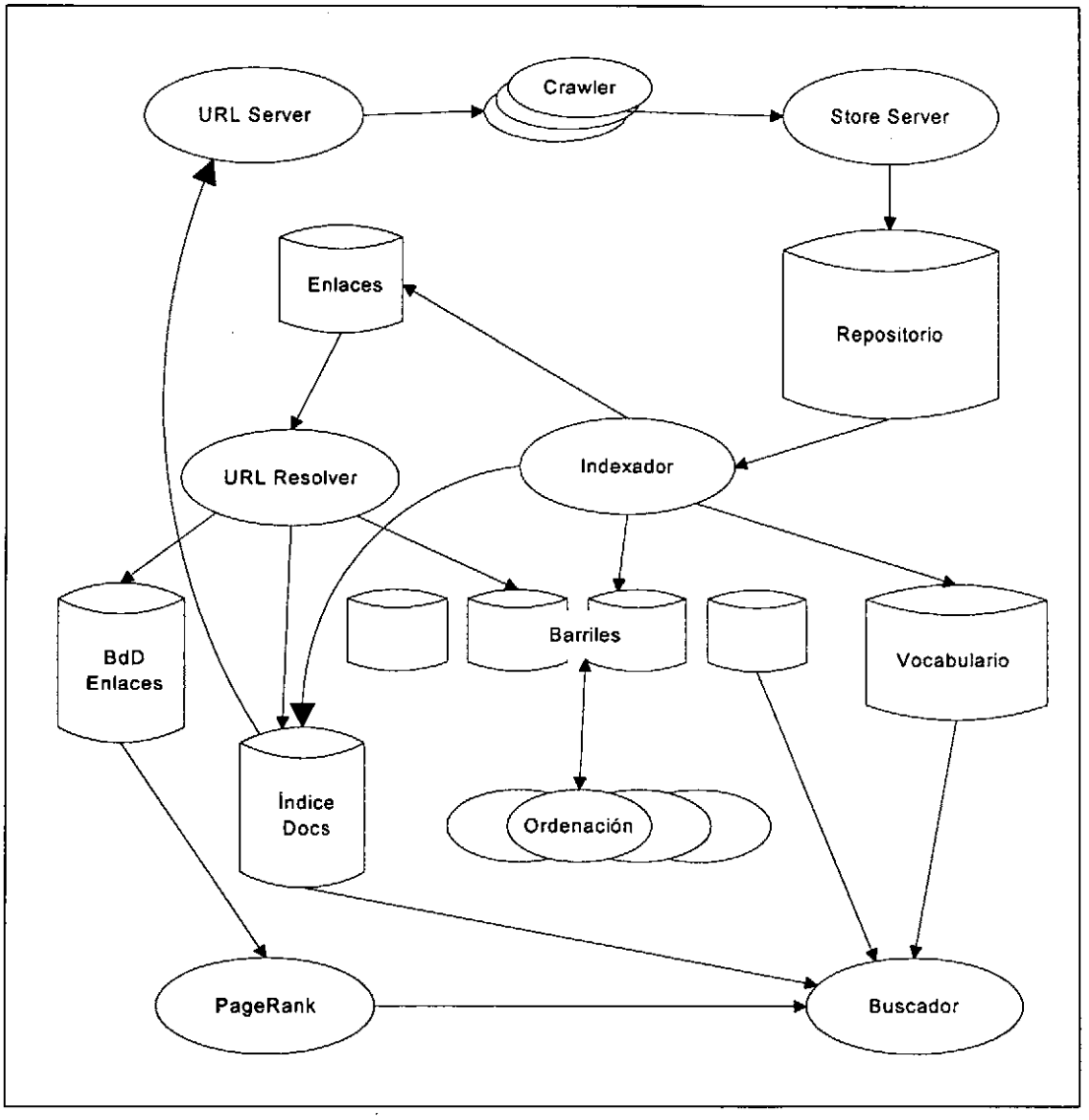

Figura 1-19: Arquitectura de alto nivel de Google

Está formado por un conjunto de barriles similares a los anteriores, excepto que han sido previamente ordenados por identificador de palabra. Para cada identificador de palabra, el vocabulario contiene un puntero al barril en el que se encuentra el identificador. Asociado a cada identificador de palabra se encuentra una lista de documentos (con sus correspondientes listas de hits) que representa todas las ocurrencias de dicha palabra en todos los documentos.

La estructura más importante para una búsqueda eficiente es, obviamente, el índice invertido, y un aspecto fundamental es el orden en el que se disponen los identificadores de documentos en la lista. Básicamente se definen dos alternativas: realizar un almacenamiento ordenado por identificador, con lo cual las operaciones de combinación de listas se ven aceleradas. O bien, almacenar los identificadores ordenados según un criterio de importancia de los documentos, lo cual convierte en trivial las búsquedas por una única palabra (y en el caso de múltiples palabras probablemente los resultados se encuentren entre los primeros documentos), y penaliza sensiblemente las operaciones con varios vocablos. La solución adoptada por Google es un híbrido: mantiene una lista inicial ordenada por mayor relevancia y a continuación, el resto de documentos ordenados por identificador. De esta manera, asume que en la mayoría de las combinaciones con la primera lista será suficiente, y en caso necesario accederá a la segunda lista.

Por último simplemente destacar el algoritmo de ordenación de los resultados, denominado PageRank, y expuesto en [Page, 98] y [Cho, 98]. Sin entrar en detalles, comentar que es un algoritmo basado en votaciones entre páginas Web: un enlace desde la página A hacia la B se interpreta como un voto de A a B. Aquellas páginas con más votos se consideran mejores, lo cual a su vez le da mayor importancia a sus propios votos. La ordenación de los resultados tras la búsqueda se realiza en base a los términos buscados y a la importancia de cada página.

1.5.2.2. Directorios Web

Este tipo de sistemas de recuperación de información en el Web se caracteriza por combinar la búsqueda con la navegación. Los directorios constituyen las ontologías del Web [Huang, 00], al proporcionar una clasificación de páginas Web basada en una jerarquía de categorías. La principal característica de estos sistemas es el pequeño porcentaje de la totalidad de documentos disponibles en la Red gestionado por estos sistemas (se consideran que indexan menos de un 1% de todas las páginas), aunque se garantiza la calidad de los documentos obtenidos como respuesta a una consulta.

Se puede definir un directorio como una taxonomía jerárquica que clasifica el conocimiento humano [Baeza-Yates, 99b], por lo tanto, un directorio Web es una taxonomía jerárquica que clasifica la información disponible en el World Wide Web. Normalmente un directorio Web está constituido por un grafo dirigido acíclico de categorías a las que se asocian documentos Web. La construcción del grafo suele ser bastante flexible, de tal forma que un nodo poseerá un número variable de nodos hijo y nodos padre. Asimismo, los documentos pueden estar asociados con cualquier categoría, sin estar restringidos a únicamente aquellas categorías hojas (ver Figura 1-20).

El ejemplo por excelencia de directorio se considera Yahoo! [Yahoo!, 01], el cual aunque no fue el primer directorio en aparecer en Internet (este puesto queda reservado para

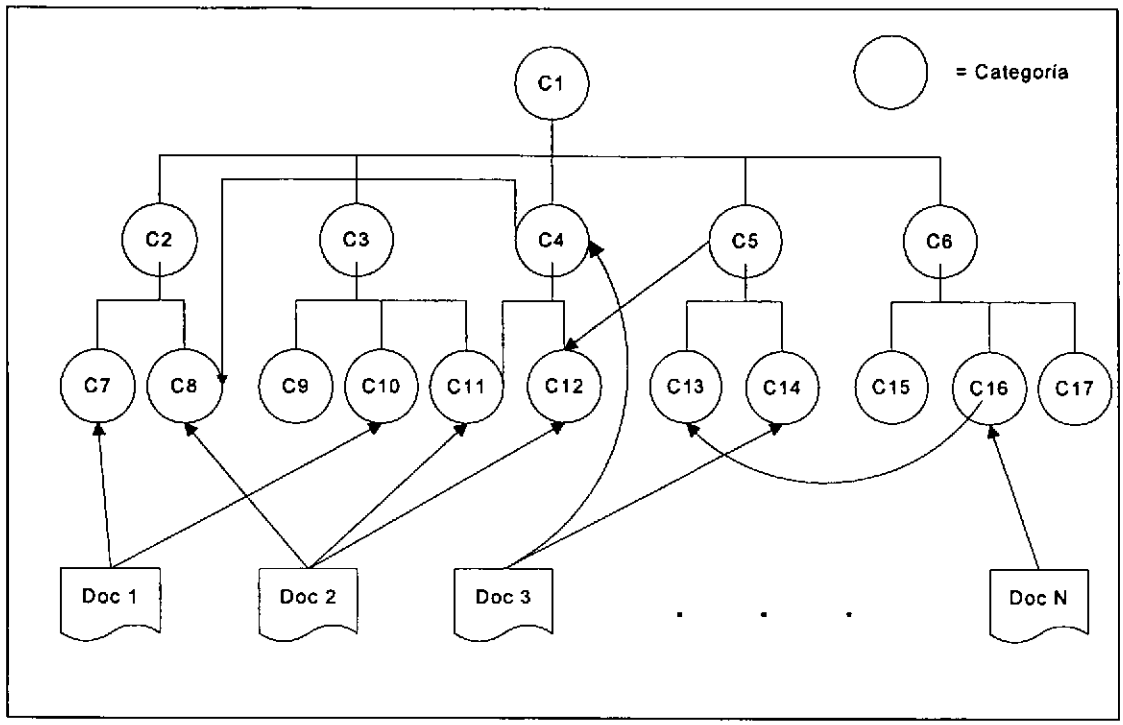

Figura 1-20: Grafo dirigido acíclico de un directorio Web

Galaxy [Galaxy, 01]), es considerado el directorio más completo al constar de alrededor de 150.000 categorías y aproximadamente un millón de páginas Web clasificadas [Labrou, 99].

La principal ventaja de estos sistemas se centra en la calidad de los resultados, ya que ante una consulta, en la mayoría de los casos, los resultados serán muy relevantes. Conjuntamente, una potente utilidad de estos sistemas de recuperación de información se centra en la posibilidad de realizar búsquedas limitadas a una zona del grafo de categorías, lo que limita ampliamente la temática de los resultados obtenidos. De hecho, esto constituye una de las características diferenciadoras de estos sistemas en donde se combina la navegación con la búsqueda, aspecto que es analizado en detalle en el presente trabajo de tesis doctoral.

Por el contrario, su principal problema se centra en la capacidad de categorización junto con el crecimiento acelerado del World Wide Web. Los esfuerzos entorno a la clasificación automática por medio de la utilización de técnicas de clustering y otras, han sido investigadas desde hace varias décadas y lo siguen siendo (ver sección 1.4.4), sin embargo, por el momento el procesamiento del lenguaje natural no es 100% efectivo en la extracción de los términos relevantes de un documento [Baeza-Yates, 99b]. Por lo tanto, en la mayoría de los casos la clasificación se realiza manualmente por un número limitado de personas. Además, aunque no es perfecta, comparada con la catalogación automática, la catalogación manual es la más precisa ya que los expertos se encargan de organizar los directorios e índices de tal manera que se facilite el proceso de búsqueda [Kobayashi, 00].

Este tipo de sistemas se suelen dividir en aquellos basados en una categorización automática (o al menos semi-automática) o manual. Dentro del primer grupo, merece especial mención el proyecto Taper [Agrawal, 97][Agrawal, 98][Indyk, 98] en donde se pretende que la respuesta a una consulta sea una lista de temas, frente a la tradicional lista

de documentos. Para ello, se basan en un análisis estadístico de los documentos, para, en base al mismo, clasificar los documentos en algún lugar de la jerarquía aportando también información de los enlaces existentes. Sin embargo, las tasas de error que deben soportar este tipo de sistemas son bastante elevadas, en concreto, en [Indyk, 98] la mejor tasa obtenida es del 21%.

Frente a los sistemas anteriores, los sistemas con clasificación manual garantizan unas tasas de errores mínimas,. El máximo exponente (como se ha comentado anteriormente) sigue siendo Yahoo!, que cuenta con un conjunto de personas encargadas de chequear y analizar documentos Web agrupados por áreas temáticas representativa de una porción de la jerarquía. Sin embargo, existen otros proyectos más interesantes a este respecto como el ODP o OpenGrid. El proyecto ODP (Open Directory Project, [ODP, 01]) se basa en la premisa de que ninguna compañía será capaz de disponer del número de empleados suficientes como para categorizar la totalidad del Web. En consecuencia, en este proyecto se permite a miles de voluntarios familiarizados con un determinado tema clasificar y gestionar una parte del directorio. Por otra parte, el proyecto OpenGrid [Lifantsev, 98] pretende emplear la opinión de miles de navegantes (no voluntarios como en el caso anterior) para la ordenación de los documentos en el Web. Para ello se basa en una ligera modificación al estándar HTML que permite incorporar información adicional a los enlaces. Sin embargo, este sistema sigue siendo una propuesta sin ningún modelo implementado, frente al caso de ODP con un sistema funcionando y utilizado por grandes compañías.

Sin embargo, ambos tipos de sistemas (automáticos y manuales) requieren una cierta estructura de datos adecuada para la información que deben gestionar, especialmente a la hora de realizar búsquedas. No obstante, la información disponible en la literatura sobre estos aspectos se puede considerar bastante reducida, si bien, como el resto de sistemas de recuperación de información actuales, la base sobre la que se asientan estos sistemas es una estructura de fichero invertido.

Un directorio Web está constituido por tres componentes básicos que representan la información almacenada en el mismo. Por una parte el vocabulario representa los vocablos indexados (tanto en los documentos como en las categorías del directorio), existe una estructura que representa la jerarquía de categorías existente en el directorio y se mantiene una base de datos de documentos con la información básica sobre cada uno (URL, título y descripción).

Y por otra parte también es necesario definir las estructuras que relacionan estos tres componentes. En primer lugar se requiere un índice para la relación entre las palabras y los documentos, igualmente para la relación entre palabras y categorías, y una tercera estructura que asocia cada documento con las categorías a las que pertenece (o lo que es lo mismo, cada categoría con los documentos que contiene). A continuación se listan las principales estructuras de estos sistemas:

- **Vocabulario**: consiste en la estructura que almacenará todas las diferentes palabras a las que hacen referencia los documentos y/o las categorías, constituyendo por lo tanto, un bloque común para varias estructuras de listas invertidas.
- **Base de datos de documentos**: es la estructura que contiene la información básica sobre cada documento. Normalmente está constituida por la URL de la página Web,

el título, una breve descripción y opcionalmente, alguna información estadística adicional.

- **Índice invertido documentos-palabras**: es la estructura que almacena los documentos que se encuentran asociados con cada palabra. Será el núcleo para la implementación de la búsqueda y la implementación dependerá de cada directorio Web, aunque siempre sobre la base de una estructura de lista invertida.
- **Índice invertido categorías-palabras**: comúnmente las propias categorías disponen de ciertas palabras clave asociadas que constituyen una representación de la temática asociada a dicha categoría. Normalmente este índice se almacenará de manera separada, aunque sería posible su inclusión conjunta con él de documentos, aunque el rendimiento ofrecido puede verse drásticamente afectado.
- **Jerarquía de categorías**: es fundamental para un eficiente funcionamiento del sistema una estructura de datos que represente el grafo dirigido acíclico existente entre las diferentes categorías. Esta estructura de datos será accedida continuamente durante la navegación del usuario a través de la ontología, por lo que primarán los accesos a los nodos hijo de una categoría, y en general a todos sus descendientes directos o indirectos. Típicamente, la estructura adecuada para tales efectos se suele ajustar a alguna variante de un árbol modificado para dar soporte a los múltiples padres que pueden poseer las categorías.
- **Índice invertido categorías-documentos**: en esta estructura se almacenarán los documentos asociados a cada una de las categorías, agrupados por categorías. De esta manera, se constituirá una lista de documentos para cada una de las categorías de la jerarquía. Se ha considerado más eficiente el almacenamiento en una estructura separada para su posterior utilización tanto en la operación de navegación como de búsqueda del usuario. Considerando únicamente la navegación a través de las categorías, sería más adecuado la incorporación a la estructura jerárquica la información de los documentos asociados a una categoría. No obstante, como las búsquedas constituyen otra operación básica y predominante en estos sistemas, se considera más conveniente su tratamiento en una estructura separada.

Sobre estas estructuras de datos varios procesos realizan operaciones tanto de consulta como de actualización. Por una parte, el proceso de categorización (con la indexación que conlleva inherentemente asociada) centra sus tareas en la inserción de nuevos documentos dentro de la ontología del directorio. Este proceso, comúnmente en un directorio Web se realiza de manera manual, aunque en caso de realizarse automáticamente no supone ninguna modificación sobre las estructuras de datos empleadas. Además, este proceso no es visible en ningún momento por el usuario.

El proceso de categorización realiza las incorporaciones de nuevos documentos al directorio, para lo cual procede a su incorporación a la base de datos de documentos, al mismo tiempo que le asocia una o varias categorías de la jerarquía en la que se situará dicho documento. Una vez que el documento ha sido insertado, se inicia un proceso de indexación que procesa las palabras clave asociadas al documento (por medio del título, descripción, etc.). La modificación de un documento simplemente consiste en el cambio de alguno de los parámetros de su inserción, mientras que el borrado consiste en la eliminación del documento de la base de datos y todas las relaciones asociadas; aunque comúnmente estos procesos son relativamente poco frecuentes.

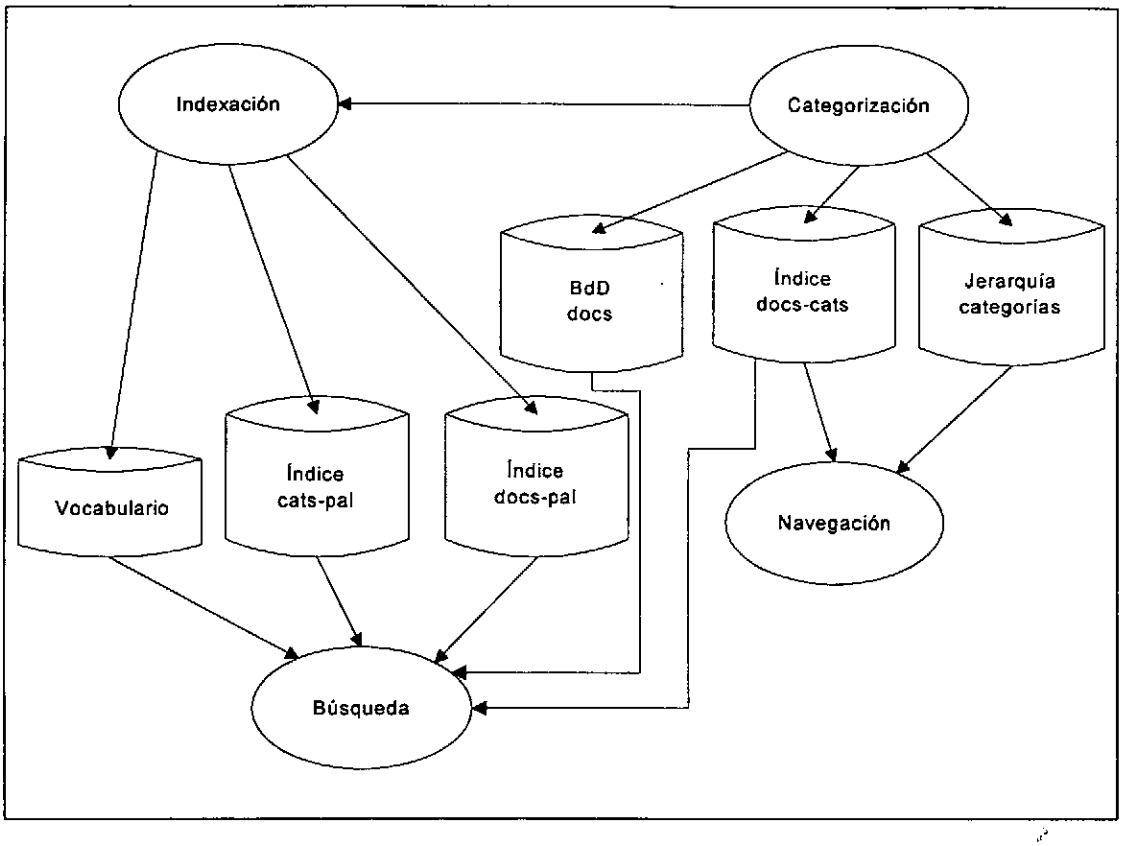

Figura 1-21: Arquitectura de alto nivel de directorio Web típico

Sin embargo, los procesos directamente accesibles para los usuarios son los de navegación y búsqueda, y por lo tanto, aquellos en donde el tiempo de respuesta es fundamental. El primero de ellos, la navegación, es característico de estos sistemas y se centra en un recorrido de la jerarquía a través de sus distintas categorías, mostrando en cada caso las diferentes categorías inferiores y los documentos asociados. Básicamente, este proceso requiere el acceso a la estructura de datos asociada con la jerarquía para la obtención de sus nodos inferiores, y también la utilización de la relación que asocia documentos y categorías para la localización de los documentos asociados con la categoría actual (en caso de existir documentos asociados).

El proceso de búsqueda es el más complejo de todos y el que requiere un rendimiento más optimizado. La búsqueda común que únicamente recupera los documentos relevantes para una serie de palabras clave, requiere el acceso al índice invertido de documentos y palabras clave a través del vocabulario, del mismo modo a como se realiza en el caso de los robots, aunque con un índice sensiblemente menor. Además, estos sistemas suelen comprobar si existe alguna categoría relacionada con la consulta por medio de una búsqueda en el índice de categorías y palabras, proceso análogo al anterior.

Sin embargo, uno de los valores añadidos de los directorios Web lo constituye las consultas restringidas a una zona de la jerarquía de categorías, lo que a su vez constituye un modelo de búsqueda más complejo. En este caso es necesario, una vez realizada la búsqueda común, contrastar los documentos obtenidos con los documentos pertenecientes a una zona de la ontología y realizar la intersección para la obtención del resultado final. Esto implica, por una parte un recorrido completo a través de una parte de la estructura jerárquica y un acceso a la estructura que asocia las categorías con sus documentos.

La definición de una estructura de datos adecuada para los directorios Web y especialmente para la resolución adecuada de las búsquedas restringidas a una zona de la jerarquía, constituye uno de los principales objetivos del presente trabajo, que será examinado en detalle tanto el diseño como la implementación en los capítulos 4 y 5.

1.5.2.3. Metabuscadores

Este tipo de sistemas de búsqueda se caracteriza porque envían una consulta insertada por el usuario a múltiples sistemas de búsqueda disponibles en Internet (tanto robots como directorios), reciben las respuestas, las procesan y unifican para mostrarlas al usuario.

La aparición de estos sistemas de búsquedas que carecen totalmente de estructuras de datos locales, se puede justificar debido a que los diferentes robots indexan partes diferentes del Web. De hecho, en [Bharat, 98] se estima que únicamente un 1% de las páginas indexadas por Altavista [Altavista, 01], HotBot [HotBot, 01], Excite [Excite, 01] e Infoseek [Infoseek, 01] se encuentran en todos ellos.

La principal aportación de este tipo de sistemas de búsqueda se centra en la eliminación de resultados erróneos (páginas no existentes o servidores no accesibles), junto con una correcta ordenación de los resultados de múltiples motores de búsqueda. Sin embargo, un posible problema consiste en los tiempos de respuesta proporcionados ya que en estas búsquedas se está añadiendo un paso intermedio. Aunque posiblemente el principal problema estriba en el desarrollo de un servicio público en base al servicio desarrollado por otros sistemas, lo que en cierta medida está limitando las posibilidades del resto de sistemas de búsqueda en el Web.

1.6. Conclusiones

A lo largo de este capítulo se han descrito las principales técnicas de indexación empleadas para la localización de información, destacando especialmente la técnica de ficheros invertidos y de ficheros de firmas.

Indudablemente, la técnica de fichero invertido se presenta hoy en día como la mejor de las alternativas disponibles, teniendo en cuenta los avances tecnológicos y las características de la información a indexar. De hecho, permite la realización de búsquedas en tiempos proporcionales al logaritmo del tamaño del vocabulario de todos los documentos indexados, con unos requerimientos de espacio bastante reducidos (variando entre un 30% y un 100%).

En cambio, la técnica de ficheros de firmas ofrece un rendimiento más pobre al realizar una búsqueda secuencial para todos los documentos empleando una capacidad de almacenamiento reducida (entre un 20% y un 30%), y con la salvedad de que se trata de una técnica de filtrado inexacto.

Por otra parte, se han analizado las principales diferencias que presentan los sistemas de recuperación de información en el Web frente a los sistemas de recuperación de

información tradicionales: volumen de datos, dinamismo, heterogeneidad, distribución, redundancia y falta de estructura, calidad y finalmente, los usuarios.

A su vez, los sistemas de búsqueda en el Web se dividen en tres categorías: robots o motores de búsqueda, directorios y metabuscadores. Los robots se caracterizan por indexar una gran cantidad de información (idealmente, la totalidad del World Wide Web) en base a la técnica de ficheros invertidos, frente a los directorios Web que se caracterizan por la catalogación de documentos Web en su jerarquía de categorías, lo que implica una gran calidad en sus contenidos.

El presente trabajo de tesis doctoral se centra en los directorios Web y las estructuras de datos empleadas, con especial énfasis en el rendimiento ofrecido por las búsquedas restringidas al conjunto de documentos asociados a una zona del grafo de categorías.

Tradicionalmente, los sistemas de búsqueda en el Web en general, y los directorios Web en particular, tienen una presencia bastante limitada en las publicaciones técnicas relacionadas con la recuperación de información, basándose en la mayoría de los casos en adaptaciones o migraciones de las técnicas de recuperación de información tradicional a la recuperación de información en el Web. En consecuencia, los directorios Web se caracterizan por estar basados en estructuras de fichero invertido que engloban diferentes aspectos: documentos, categorías y palabras clave, y las interrelaciones entre ellos, y que por lo tanto presentan una complejidad inherente mayor que el caso de los robots o motores de búsqueda.

Por lo tanto, cualquier mejora en el rendimiento de determinados aspectos del directorio Web (en este caso, ante búsquedas restringidas a una zona del grafo de categorías) debe garantizar que no se produzcan interdependencias con el resto de estructuras de datos que afecten a los procesos básicos del sistema de búsqueda: búsqueda y navegación a través de categorías.

2. ESTUDIO DE LOS ACCESOS A UN DIRECTORIO WEB

Desde la aparición de los primeros sistemas de búsqueda en el Web, su evolución ha sido imparable tanto en el volumen de la información tratada como en las diversas mejoras tecnológicas surgidas. Sin embargo, en la actualidad, el conocimiento existente sobre los usuarios de este tipo de servicios en Internet es reducido.

La mayoría de los estudios realizados hasta el momento se centran básicamente en un análisis de las consultas realizadas por los usuarios, con el objetivo de confirmar las diferencias existentes entre un usuario Web y un usuario tradicional de un sistema de recuperación de información. Por este motivo, en el presente análisis se estudia el comportamiento de los usuarios frente a un directorio Web, analizando tanto las búsquedas realizadas por los usuarios, como los documentos consultados y las categorías visitadas, con el fin de confirmar y contrastar las diferencias existentes y lo que es más importante, intentar obtener un modelo de comportamiento que se ajuste a un modelo matemático ([Cacheda, 01a], [Cacheda, 01c], [Cacheda, 01d]).

2.1. Trabajos relacionados

Recientemente han surgido varios estudios que examinan a los usuarios de un sistema de recuperación de información en Internet frente a los usuarios de un sistema tradicional.

El primero de estos trabajos fue presentado en 1.998 por Steve Kirsch, en donde se analizan las consultas realizadas sobre Infoseek [Kirsch, 98]. El estudio describe de forma sencilla algunas características básicas de las consultas de los usuarios, como las consultas más repetidas, la utilización de operadores lógicos y número medio de palabras por

consulta. La principal conclusión aportada se centra en el hecho de que las consultas realizadas son demasiado cortas lo que provoca un elevado número de resultados por consulta, imposibles de comprobar por el usuario.

A continuación Jansen et al. presentaron un trabajo en el SIGIR Forum de 1.998, en donde realizan un análisis detallado de las consultas realizadas a Excite [Jansen, 98]. En este análisis se examinan las consultas propiamente dichas, los términos de las consultas y se realiza un breve seguimiento de las consultas realizadas por los usuarios durante una sesión. Esta investigación aporta aspectos concretos del comportamiento de los usuarios Web frente a los usuarios tradicionales. A continuación se listan sus principales conclusiones:

- Las consultas de los usuarios Web son cortas, utilizando aproximadamente dos palabras de media (confirmando el estudio de Kirsch) y menos del 4% de las consultas emplean más de seis palabras.
- Los operadores lógicos se utilizan poco frecuentemente (uno de cada 18 usuarios) y el caso de los operadores simples, como el '+' y '-', es ligeramente mejor ya que uno de cada 12 usuarios tiende a utilizarlos.
- Un usuario consulta de media 2,21 páginas, y más de la mitad de los usuarios no visita más allá de la primera página de resultados.
- El número medio de consultas por usuario es inferior a 3, lo que indica que los usuarios no reformulan sus consultas. Esto se confirma por el hecho de que un 66% de los usuarios realizó una única consulta.
- Respecto a los términos o palabras usadas en las consultas es interesante destacar que existe un conjunto reducido de palabras que se repiten muchas veces y un gran conjunto de términos que se repiten una única vez.

Estas características hacen que el usuario de un servicio de búsqueda en el Web sea significativamente diferente de un usuario tradicional. Básicamente los usuarios Web no se encuentran cómodos utilizando los operadores lógicos y no consultan la totalidad de los resultados obtenidos, quedándose en la primera página de resultados.

Por último, en 1.999 Silverstein et al. presentan el trabajo que mayor número de consultas analiza (cerca de 1.000 millones de consultas), en este caso sobre Altavista [Silverstein, 99]. La primera parte de este estudio obtiene resultados similares al realizado por Jasen et al. En ambos casos se observa que el número de palabras por búsqueda es bastante reducido (alrededor de dos palabras por búsqueda). Asimismo se mantiene la distribución de palabras en las consultas, ya que un grupo reducido de consultas es repetido múltiples veces y al mismo tiempo existe un gran conjunto de consultas realizadas una única vez. Respecto a las consultas realizadas durante las sesiones de los usuarios, se observa que en general las sesiones son breves, aproximadamente dos consultas por sesión y el número de páginas de resultados visitadas no llega a dos (este valor es ligeramente inferior al estudio anterior). Como aspecto novedoso, la segunda parte del estudio se centra en el examen de las correlaciones existentes entre los términos utilizados en las consultas con el objetivo de examinar los pares de palabras más consultados.

2.2. El entorno de búsqueda

El presente análisis está basado en los accesos realizados a un directorio Web español, denominado BIWE (Buscador en Internet de Webs en Español, [Biwe, 01]). Durante un período que comprende desde el día 3 de mayo del 2.000 a las 15:00 hasta el día 18 de mayo del 2.000 a las 7:00 se han almacenado todos los accesos realizados por todos los usuarios al directorio. Estos accesos engloban desde consultas al motor de búsqueda, accesos a categorías hasta consultas de documentos.

En las siguientes secciones se describen el entorno de búsqueda y el archivo de transacciones en donde se han almacenado los accesos recibidos.

2.2.1. El directorio y el motor de búsqueda

En este apartado se definen las principales características del entorno utilizado por los usuarios, teniendo en cuenta que se trata de un directorio Web. Tal y como se ha descrito en la sección 1.5.2.2 los directorios permiten la localización de información mediante la navegación y las búsquedas.

En el proceso de navegación se recorre el grafo de categorías examinando los documentos asociados, y en el caso de una búsqueda, se mostrarán tanto documentos Web catalogados como categorías que coincidan con los conceptos buscados.

El motor de búsqueda está basado en el modelo vectorial, realizando una ordenación de los resultados según el modelo tradicional tf-idf (*term frequency-inverse document frequency*) modificado para el soporte de distintos pesos en función de la importancia de cada palabra. La búsqueda por defecto realizará un OR lógico de las palabras, aunque el algoritmo de ordenación garantiza que las primeras posiciones estarán ocupadas por aquellos documentos que se relacionen con todas las palabras (al igual que una operación de AND lógico).

Por otra parte el motor de búsqueda soporta distintos operadores. Los más sencillos son los siguientes:

- Operador '-': por ejemplo, la expresión *–información* indica al motor de búsqueda que recupere todos los documentos excepto los que contengan la palabra *información*.
- Operador '+': la expresión *+información* indica al motor de búsqueda que ignore aquellos documentos que no contenga la palabra *información*.
- Operador " ": la expresión *"recuperación de información"* indica al motor de búsqueda que recupere los documentos con el literal indicado, aunque en este caso el motor realizará un AND lógico entre las palabras. Es decir, no se garantiza que el orden y la proximidad de las palabras se mantengan.

Igualmente el motor de búsqueda soporta los operadores lógicos más comunes (*and*, *or*, *not*) y sus variantes en español (*y*, *o*, *no*). Los dos tipos de operadores pueden ser combinados utilizando los paréntesis.

Como se ha comentado anteriormente, una característica definidora de los directorios Web es el hecho de que las búsquedas pueden ser restringidas a los documentos contenidos o descendientes de una categoría determinada. En el caso del directorio analizado, por defecto la búsqueda se realizará sobre todos los documentos, aunque existe la posibilidad de restringir la búsqueda a una categoría y sus descendientes.

Al igual que la mayoría de los motores de búsqueda, también existe una búsqueda detallada en donde se permite modificar parámetros más específicos de las búsquedas como los campos sobre los que se va a realizar la búsqueda (título del documento, descripción del documento, URL del documento o palabras clave del documento), el número de resultados que se van a mostrar en cada página de resultados y la forma de agrupación utilizada al mostrar los resultados (por defecto no se agruparán, aunque se pueden mostrar los resultados sin agrupar pero indicando las categorías a las que pertenecen o agrupados por categorías).

De ahora en adelante se hará referencia a *cadena de búsqueda* como la cadena exacta que ha tecleado el usuario (incluyendo operadores, palabras comunes, etc.) y se denotará *término de búsqueda* a cada una de las palabras, una vez eliminados los operadores, palabras comunes, etc. Por lo tanto, una cadena de búsqueda estará constituida por uno o más términos de búsqueda. En caso de que una cadena de búsqueda carezca de términos de búsqueda (porque sean todos operadores o palabras comunes) se denomina *búsqueda vacía.*

El proceso de búsqueda se inicia cuando el usuario introduce la cadena de búsqueda, se procesan los términos de búsqueda y se muestra una página de resultados. Cada resultado incluye el título y la descripción del documento, junto con un enlace a una página intermedia (lo que permite establecer y controlar el número de accesos a cada página Web), que trasladará automáticamente al usuario a su localización en Internet. El usuario puede seguir cualquiera de estos enlaces para consultar un documento, o puede seguir a otras páginas de resultados a través de los botones de navegación, lo que producirá otra consulta al motor de búsqueda.

El directorio Web analizado está constituido por aproximadamente 60.000 documentos, con un grafo compuesto por aproximadamente 1.000 categorías. Como se ha comentado anteriormente, las categorías forman parte de un grafo, por lo que pueden disponer de varios padres. En concreto, cada categoría, de media, consta de 1,17 padres; de hecho la gran mayoría disponen de un único padre ya que únicamente un 16% de las categorías constan de más de un padre. Respecto a los documentos, en este caso el porcentaje de documentos situados en varias categorías es ligeramente más elevado con un 27% del total, y una media de 1,34 categorías a las que se asocia un documento.

2.2.2. El log de transacciones

El log de transacciones se encarga de almacenar todas las peticiones realizadas sobre el directorio Web analizado. En este caso concreto se ha almacenado información referente a los tres tipos de transacciones: búsquedas en el directorio, acceso a una categoría y consulta de un documento (obtenido a través de una búsqueda o navegando).

Para cada tipo de transacción se almacena una serie de información específica. Por ejemplo, para el caso de una búsqueda la información archivada es la siguiente:

- La cadena de búsqueda exactamente como fue introducida por el usuario.
- Los términos de búsqueda, una vez han sido eliminadas las palabras comunes, palabras reservadas y operadores.
- Fecha y hora del momento en el que se solicitó la búsqueda.
- El número del primer documento a mostrar en los resultados (p.e. 1, 11, 21, etc.).
- El número de documentos a mostrar en cada pantalla de resultados.
- Un campo booleano que indica si la búsqueda ha sido restringida a una categoría, en cuyo caso se almacenará también el identificador de dicha categoría.
- Los campos del documento sobre los que se ha realizado la búsqueda.
- El tipo de agrupación que se ha realizado sobre los resultados.

En caso de que la transacción sea un acceso a una categoría la información almacenada es la siguiente:

- El identificador de la categoría a la que se ha accedido (cada categoría está asociada con un identificador único).
- La fecha y la hora del momento en el que se realizó el acceso.

Y por último, si la transacción se corresponde con un acceso a un documento la información almacenada es la siguiente:

- El identificador del documento visitado (cada documento se identifica unívocamente a través de su identificador).
- La fecha y hora del acceso al documento.

Las transacciones fueron archivadas durante 16 días, desde el día 3 de mayo de 2.000 a las 15:00 hasta el día 18 de mayo a las 7:00. Durante ese tiempo se recibieron un total de 351.427 solicitudes, de las cuales 105.786 se corresponden a búsquedas, 87.974 a accesos a categorías y 157.667 a visitas a documentos.

2.3. Análisis de las búsquedas

En esta sección se describe el análisis realizado a las 105.786 búsquedas, con el objetivo de obtener un mejor conocimiento del comportamiento de los usuarios ante este tipo de sistemas de búsqueda.

En este sentido, una de las características más importantes es el número de pantallas de resultados que consultan los usuarios al realizar una búsqueda. En la Tabla 2-1 se muestran el número de pantallas visitadas junto con los porcentajes asociados. Se puede observar que aproximadamente el 68% de los usuarios visitaron únicamente la primera pantalla de resultados y más del 81% visitaron las dos primeras pantallas de resultados.

Pantallas visitadas	Porcentaje
1	67,881%
2	13,234%
3	5,966%
4	3,468%
5	2,272%
6	1,538%
7	1,119%
8	0,819%
9	0,598%
10	0,49%
Más	2,615%

Tabla 2-1: Pantallas de resultados vistas por los usuarios

De hecho, de media cada usuario únicamente consultará 2,18 pantallas, lo cual confirma el hecho de que los usuarios Web tienden a examinar muy pocos resultados. Esto implica que los algoritmos de ordenación utilizados por los buscadores cobran mayor importancia y presentan características significativamente diferentes a los de los sistemas de recuperación tradicionales, confirmando los resultados de [Jansen, 98] y [Silverstein, 99].

Por otra parte, los menores porcentajes se encuentran en las pantallas más elevadas (por encima de la pantalla número 20 los porcentajes son menores del 0,1%).

La siguiente parte del análisis se centra en las cadenas y términos de búsqueda y la utilización de los distintos operadores en las búsquedas. Para este propósito, Silverstein et al. en [Silverstein, 99] indican que la utilización de todas las búsquedas puede provocar resultados incorrectos, ya que aquellas búsquedas con más páginas de resultados examinadas pueden ser ponderadas en exceso. Por este motivo, la siguiente parte del análisis se centra en las 71.810 búsquedas correspondientes a las búsquedas de las primeras pantallas.

El primer dato relevante se refiere al número medio de palabras empleado por los usuarios para definir el concepto buscado, situado en únicamente 1,63 palabras. Este valor es ligeramente inferior a los obtenidos en [Jansen, 98] y [Silverstein, 99], y pone de manifiesto la dificultad a la hora de especificar correctamente una consulta, lo que se verá reflejado en el número de resultados obtenidos como respuesta.

De las 71.810 cadenas de búsqueda examinadas, se obtienen 26.654 cadenas de búsqueda diferentes. A su vez, estas cadenas de búsqueda están compuestas de 116.953 términos de búsquedas, de los cuales 18.966 son palabras diferentes. En la Tabla 2-2 se puede observar que cada cadena de búsqueda ha sido repetida 2,7 veces de media, mientras que cada término de búsqueda 6,2 veces. En principio esta información es poco relevante, aún teniendo en cuenta las desviaciones típicas, ya que no permite ver la forma en la que se distribuyen las cadenas y términos de búsqueda.

En la Tabla 2-3 se muestra información básica sobre las repeticiones de cadenas y términos de búsqueda. El dato más relevante es que un 23,4% de las 71.810 cadenas de búsqueda ha sido buscado una única vez, lo que implica que existe una larga cola de cadenas de búsqueda con frecuencia uno. Por otra parte, en el caso de los términos de búsqueda este porcentaje es sensiblemente menor (7,74%), debido a que el conjunto de búsquedas

realizadas una vez está compuesto de un conjunto de palabras o términos que se repiten varias veces, y un pequeño núcleo repetido una única vez, lo que provoca la reducción en el porcentaje.

Cadenas de búsqueda		Términos de búsqueda	
Total	71.810	Total	116.953
Cadenas de búsqueda únicas	26.654	Términos de búsqueda únicos	18966
Frecuencia máxima	1.452	Frecuencia máxima	1.429
Frecuencia media	2,69	Frecuencia media	6,17
Desviación típica frecuencia	12,27	Desviación típica frecuencia	25,77

Tabla 2-2: Estadísticas de cadenas y términos de búsqueda

Cadenas de búsqueda		Términos de búsqueda	
Búsquedas 1 vez	23,4 %	Términos 1 vez	7,74 %
Búsquedas 2 veces	12,13 %	Términos 2 veces	5,1 %
Búsquedas 3 veces	7,64 %	Términos 3 veces	3,99 %
Búsquedas + de 3 veces	56,83 %	Términos + de 3 veces	78,16 %

Tabla 2-3: Estadísticas de repeticiones de cadenas y términos de búsqueda

Hasta este punto se ha esbozado las características de la cola de la distribución, pero es necesario determinar como es la parte inicial. Para ello, podemos observar la Tabla 2-4 en donde se muestran las cadenas y términos de búsqueda más populares y la mediana de ambas distribuciones. Es importante destacar que la mediana de las cadenas de búsqueda se encuentra en la posición 2.490 (del total de 26.654 cadenas de búsqueda diferentes), por lo que aproximadamente un 9,3% de las cadenas de búsqueda genera la mitad de las 71.810 búsquedas analizadas. Esto denota que existe un pequeño porcentaje de búsquedas que está produciendo una gran parte del total de las búsquedas. Considerando los términos de búsqueda se hace más patente este hecho, ya que un 4,6% de los términos genera la mitad de los 116.953 términos analizados.

Respecto a la Tabla 2-4 merece una mención especial el hecho de que la cadena de búsqueda más repetida sea la cadena vacía con un 2% de las búsquedas. Kirsch en [Kirsch, 98] comenta la misma incidencia en su análisis, y aunque no se cita ninguna explicación en la bibliografía, la opción más lógica puede estar centrada en la complejidad que puedan tener las interfaces gráficas de los actuales buscadores para los usuarios de Internet noveles.

La visualización del gráfico de cadenas de búsqueda y términos frente a sus frecuencias no

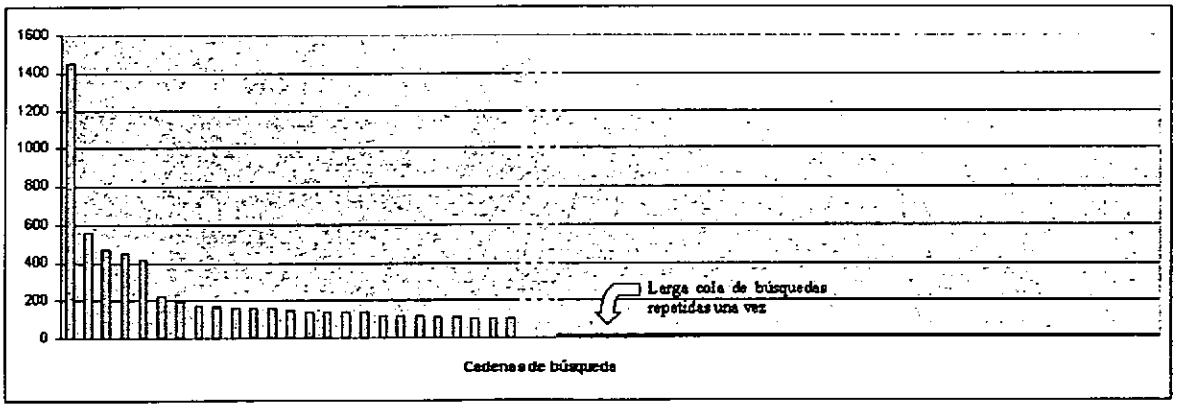

Figura 2-1: Distribución de las cadenas de búsqueda

Cadenas de búsqueda	Frecuencia	Porcentaje	Términos de búsqueda	Frecuencia	Porcentaje
	1.452	2,022%	sexo	1.429	1,222%
sexo	564	0,785%	gratis	1.289	1,102%
gran hermano	470	0,655%	fotos	908	0,776%
mp3	446	0,621%	gay	819	0,700%
gay	418	0,582%	mp3	802	0,686%
porno	222	0,309%	desnudas	695	0,594%
chat	195	0,272%	gran	583	0,498%
relatos eroticos	173	0,241%	porno	560	0,478%
hentai	166	0,231%	hermano	557	0,476%
famosas desnudas	162	0,226%	madrid	518	0,443%
moviles	158	0,220%	famosas	468	0,400%
famosas	156	0,217%	videos	432	0,369%
sexo gratis	142	0,198%	moviles	381	0,326%
amateur	138	0,192%	com	361	0,309%
....			....		
orihuela (mediana 2.490/26.654)	5	0,007%	champions (mediana 867/18.966)	24	0,021%

Tabla 2-4: Las 15 cadenas y términos de búsqueda más populares

es posible debido a la gran cantidad de elementos que lo constituyen. En cambio, en el siguiente gráfico (ver Figura 2-1) se muestra una versión reducida de la distribución de cadenas de búsquedas con el objetivo de visualizar los aspectos más relevantes de ambas distribuciones. Se puede observar claramente que se trata de una distribución no balanceada, en donde se produce un descenso muy abrupto inicialmente, mientras que en la parte final aparecen discontinuidades y una larga cola que representa los términos con frecuencia uno.

Los resultados obtenidos son totalmente equiparables a los obtenidos en [Jansen, 98] y que demuestran que la distribución de las palabras utilizadas en las búsquedas no se puede asimilar con la distribución de Zipf [Zipf, 49], a la que tradicionalmente se viene ajustando las palabras usadas en una lengua como el castellano o el inglés.

Otro aspecto importante de las búsquedas realizadas por los usuarios se refiere a las distintas opciones empleadas. En la Tabla 2-6 se muestran los porcentajes de utilización de los diferentes operadores disponibles en el motor de búsqueda.

La primera conclusión es obvia: en general los operadores (tanto lógicos como los más simples) son infrautilizados por los usuarios. El operador más empleado ha sido el operador lógico *Y*, con un 4% de las consultas, seguido por el operador " " para la construcción de literales. Otro operador bastante empleado es el '+', probablemente debido a su sencillez de uso. El porcentaje de utilización del resto de operadores cae por debajo del 1%, haciendo mención especial a la baja utilización de los operadores ingleses, siempre por debajo de la versión española, lo que pone de relieve la necesidad de disponer de operadores en distintas lenguas. El caso de los paréntesis merece una mención especial, ya que aunque su uso es elevado, la mayor parte de los casos van asociados a otros operadores lógicos o simples.

A nivel general se debe destacar el bajo porcentaje de utilización de operadores en las búsquedas, probablemente debido a la falta de conocimiento de los operadores booleanos

Operador *Y*	4,164%
Operador *AND*	0,362%
Operador *NO*	0,024%
Operador *NOT*	0,0006%
Operador *O*	0,033%
Operador *OR*	0,014%
paréntesis	3,513%
Operador +	2,055%
Operador -	0,826%
Operador " "	3,64%

Tabla 2-6: Utilización de operadores en las búsquedas

básicos o por desconocimiento de los operadores más simples, coincidiendo con los resultados obtenidos en [Jansen, 98].

Otra de las opciones importantes a la hora de realizar una búsqueda es la posibilidad de restringir una búsqueda a una categoría concreta, parámetro que ha sido utilizado únicamente en el 2,6% de las búsquedas. Es significativo destacar el hecho de que esta opción se encuentra fácilmente accesible para el usuario, al lado de la caja de búsqueda, frente al caso de las opciones de búsqueda detallada para las cuales es necesario acceder a otra página. Este hecho se ve reflejado en la Tabla 2-5, en donde destacan con valores cercanos al 100% las opciones utilizadas por defecto, lo que indica que el usuario no suele explotar las diferentes opciones disponibles aceptando en la mayor parte de los casos los valores por defecto.

A este respecto la excepción viene dada por el significativo porcentaje de uso de la opción que permite mostrar 30 resultados por pantalla frente al resto de valores. Probablemente esto sea debido a la existencia de algún metabuscador que utiliza esta opción del motor de búsqueda, aunque es imposible determinar a ciencia cierta el origen de este desfase.

La última parte de este análisis examina un aspecto que no ha sido previamente investigado en otros trabajos: el número de resultados que se han obtenido en cada una de las búsquedas. Para la obtención de esta información, se ha realizado un experimento adicional en donde se han repetido las 26.654 consultas formuladas por los usuarios y se ha comprobado el número de resultados obtenidos en cada caso.

Sin agrupamiento (defecto)	99,783%
Sin agrupamiento + categorías	0,174%
Agrupamiento por categorías	0,043%
Título + Descripción + Claves (defecto)	99,627%
Título + Descripción + Claves + URL	0,267%
Otras combinaciones	0,106%
10 documentos por pantalla (defecto)	81,892%
20 documentos por pantalla	0,06%
30 documentos por pantalla	17,785%
40 documentos por pantalla	0,006%
50 documentos por pantalla	0,085%

Tabla 2-5: Utilización de opciones de la búsqueda detallada

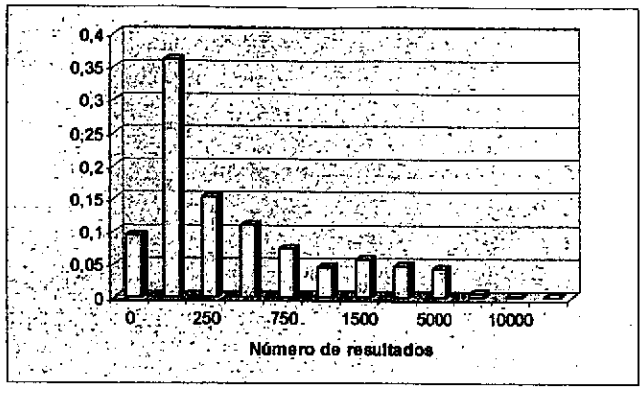

Figura 2-2: Histograma del número de resultados en las búsquedas

Considerando únicamente las 26.654 búsquedas, sin tener en cuenta sus frecuencias de repetición, cada búsqueda produjo de media unos 538 resultados (con una desviación típica de 1.113, bastante elevada que indica la gran fluctuación de los datos), con búsquedas que van desde los 0 resultados hasta consultas que obtuvieron 17.591 documentos. Considerando el total de las búsquedas consultadas en primeras páginas de resultados (es decir, considerando la frecuencia de cada una de las 26.654 consultas diferentes) el valor medio es prácticamente igual, situándose en 532 resultados por búsqueda (aunque en este caso la desviación típica es sensiblemente menor, 579). En la Figura 2-2 se muestran los porcentajes de consultas en función del número de resultados obtenidos. Como se puede observar las consultas predominantes son las que obtienen entre 100 y 250 resultados, y según van aumentando el número de resultados disminuye el número de consultas realizadas.

El hecho de que el número medio de resultados obtenidos por búsqueda sea tan elevado está directamente relacionado con el hecho de que los usuarios empleen pocas palabras en las búsquedas típicas, lo que indica que las consultas realizadas son demasiado genéricas, y en consecuencia el número de resultados recuperados será muy elevado.

2.4. Análisis de los accesos a categorías

Este análisis se centra en el estudio de los 87.954 accesos a categorías registrados y constituye el primer análisis orientado directamente hacia el concepto de directorio Web y su jerarquía de categorías. El directorio base consta de 911 categorías, de las cuales se ha accedido a 898. Cada categoría ha sido accedida una media de 96 veces con una desviación típica muy elevada (σ = 1.437), lo que indica una gran variabilidad en los accesos a las categorías. De hecho, la categoría principal, a través de la cual se accede al buscador, es la que más accesos presenta, con prácticamente la mitad del total (43.465). En consecuencia, para evitar desvirtuar los resultados obtenidos el resto del análisis se realizará sin tener en cuenta los accesos realizados a la categoría raíz.

En primer lugar se muestra un gráfico con las categorías accedidas junto con las frecuencias asociadas (ver Figura 2-3). Se puede comprobar que, al igual que sucedía en el caso de las búsquedas, existe un conjunto reducido de categorías que obtienen gran parte de los accesos (un 5,68% de las categorías genera la mitad de los accesos, sin tener en

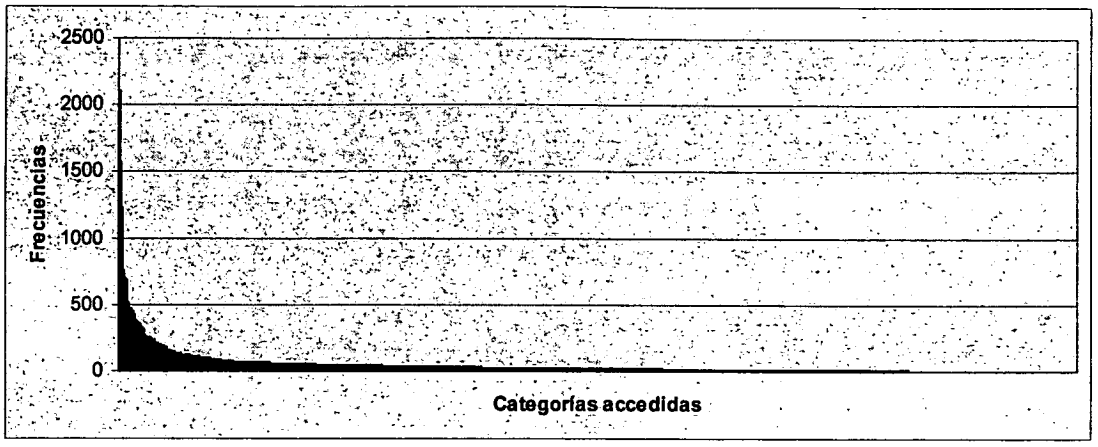

Figura 2-3: Frecuencias de las categorías accedidas

cuenta a la categoría raíz); sin embargo, en el caso de las categorías, la cola de categorías menos visitadas es de dimensiones reducidas, siendo aproximadamente un 1,78% del total de las categorías.

No obstante, es necesario un enfoque basado en la profundidad de las categorías para poder obtener información más práctica y relevante sobre el comportamiento de los usuarios. Para ello se ha analizado la profundidad a la que se encuentra cada categoría (teniendo en cuenta que aquellas categorías con varios caminos posibles se conservaba la menor profundidad) y se ha generado la Figura 2-4 en donde se muestran los accesos que acumulan las categorías de los diferentes niveles de profundidad (se considera que la categoría raíz es de profundidad 0).

En el gráfico se puede observar que las categorías de mayor profundidad tienen un porcentaje de accesos reducido, ya que su ubicación implica la necesidad de seguir 5 ó 6 enlaces. Por otra parte, el hecho de que las categorías de primer nivel posean un porcentaje de accesos inferior a las categorías de los siguientes niveles, indica que los usuarios localizan una categoría de primer nivel adecuada y a continuación se dedican a navegar por las diferentes subcategorías de ese nivel y los subsiguientes. También es importante tener en cuenta que a través de las búsquedas los usuarios pueden acceder directamente a categorías relacionadas con los conceptos buscados, sin necesidad de recorrer una parte del árbol, lo que podría aumentar el número de accesos a las categorías de los niveles

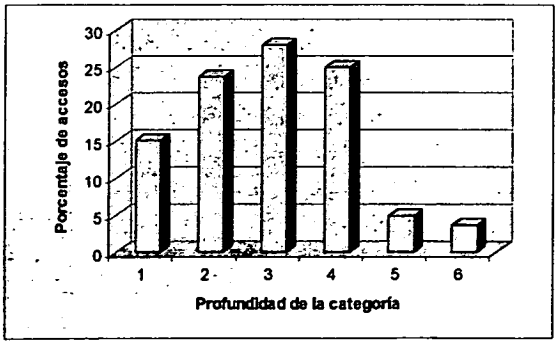

Figura 2-4: Porcentajes de acceso respecto a la profundidad de la categoría

intermedios.

2.4.1. Análisis de búsquedas restringidas a categorías

Durante el análisis de las búsquedas se destacó que un pequeño porcentaje de las búsquedas realizadas se encontraban restringidas a categorías, en concreto, 2.211 búsquedas han sido restringidas sobre alguna categoría y sus descendientes.

A nivel general, este tipo de búsquedas se comporta del mismo modo que el resto, en cuanto a operadores utilizados, a pantallas de resultados consultadas y a parámetros especiales disponibles en la búsqueda detallada. Sin embargo, es interesante investigar las categorías en las que han buscado los usuarios, especialmente en lo que se refiere a su profundidad en el grafo de categorías.

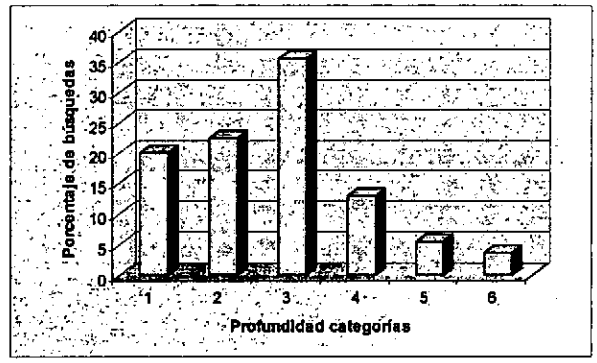

Figura 2-5: Porcentajes de búsquedas respecto a la profundidad de la categoría

En la Figura 2-5 se muestran los porcentajes de búsquedas realizadas en función de la profundidad de la categoría a la que estaba restringida. Se puede comprobar que aproximadamente un 77% de las búsquedas se han limitado a categorías de profundidad menor que 4. Esto coincide con el perfil genérico de un usuario Web, ya que éste procura visitar el número mínimo de páginas para encontrar la información buscada. En consecuencia, los usuarios raramente profundizarán en exceso en el grafo de categorías para la realización de una búsqueda restringida.

Otro aspecto interesante en este tipo de búsquedas se refiere a la capacidad de filtrado respecto a las consultas aplicadas a la totalidad del directorio. Para evaluar este aspecto, se han repetido estas consultas y examinado el número de documentos recuperados en cada caso, tanto restringiendo la consulta a la categoría especificada por el usuario, como si la consulta hubiese sido realizada sobre la totalidad del directorio. Por una parte, a nivel global estas consultas mantienen un comportamiento similar al resto, ya que de media cada una las consultas recuperaría (en el conjunto del directorio Web) unos 450 documentos (con una desviación típica de 904). En cambio, al restringir las búsquedas, de media cada consulta ha recuperado 42 documentos (con una desviación típica de 152).

Esto implica que mediante la utilización de las consultas restringidas, los usuarios son capaces de eliminar más de un 90% de los resultados, quedándose con aquellos más

relacionados con la temática de su consulta, lo que pone de manifiesto la gran versatilidad y utilidad de este tipo de búsquedas.

El presente trabajo de tesis doctoral se centra en la mejora del rendimiento de este tipo concreto de búsquedas, por lo que es fundamental determinar claramente el formato de las búsquedas realizadas por los usuarios. Tal y como se observa en este caso, las búsquedas se centran primordialmente en los primeros niveles del grafo, que contienen un menor número de categorías y a su vez engloban un mayor número de documentos. Esto implica que los esfuerzos para la mejora del rendimiento deben estar dirigidos especialmente hacia las categorías situadas en los primeros niveles.

2.5. Análisis de los accesos a documentos

El análisis de los documentos visitados se centra en los 157.667 accesos recibidos durante el período considerado. En los trabajos relacionados no existe información a este respecto, por lo que los datos aquí aportados constituyen la primera muestra sobre los documentos consultados por los usuarios de un servicio de búsqueda en el Web. Estos accesos se realizaron sobre 30.341 documentos diferentes, siendo visitado cada documento una media de 5,2 veces. Es importante destacar que el directorio examinado dispone de aproximadamente 60.000 documentos, lo que indica que existe una parte significativa de los documentos que no ha sido visitada, de hecho, considerando el total de documentos, la media de accesos a cada documento se reduce a la mitad, situándose en 2,6 visitas por documento.

Sin embargo, el aspecto más relevante se centra en la forma de la distribución de los accesos a documentos, que se aproxima notablemente a la distribución de las búsquedas.

Por una parte, la mediana se sitúa en la posición 1.746, lo que significa que únicamente 1.746 documentos son capaces de generar la mitad del total de las visitas (78.833 accesos). Por otra parte, en la Tabla 2-7 se muestran los porcentajes de repeticiones más bajos, en donde se observa que existe un porcentaje significativo de documentos que han sido visitados menos de 3 veces, concretamente un 15,72% de los documentos.

Esto provoca que la distribución sea muy similar al caso de las búsquedas, sobre todo en el caso de los documentos más visitados, mientras que en este caso la cola de documentos menos visitados es menor que el caso de las búsquedas. Esta similitud es razonable si se tiene en cuenta que una gran parte de los usuarios realizan las mismas búsquedas, por lo que obtendrán los mismos resultados, lo que implica que consultarán documentos muy similares. Por supuesto, siempre existe ese porcentaje importante de búsquedas repetidas una única vez, que sin embargo no producen como resultado un porcentaje tan elevado de documentos poco vistos probablemente por dos causas: el hecho de que en ese grupo de

Documentos	Porcentaje
Visitados 1 vez	8,5 %
Visitados 2 veces	7,22 %
Visitados 3 veces	5,6 %
Visitados + de 3 veces	78,67 %

Tabla 2-7: Porcentajes de repetición de documentos visitados

búsquedas se incluyen los errores tipográficos que no generarán documentos relevantes en sus resultados y por lo tanto no se consultarán, y el hecho de que dos búsquedas diferentes pueden producir resultados similares.

2.6. Análisis de las sesiones de usuarios

Se considera una sesión al conjunto de peticiones que realiza un usuario durante un intervalo de tiempo sobre un servicio de búsqueda, en este caso, en el Web. En esta parte del estudio, se han agrupado todas las peticiones (búsquedas, accesos a categorías y documentos examinados) realizadas por cada usuario durante una sesión.

Es importante destacar que la única información disponible para la diferenciación entre distintas sesiones es la dirección IP de la máquina de origen y los tiempos en los que se han realizado las distintas peticiones. Evidentemente, por medio de la dirección IP no es posible realizar una diferenciación unívoca de todas y cada una de las sesiones, debido a la posible existencia de equipos proxy o también por el uso de una misma máquina por varios usuarios simultáneamente (p.e. servidores UNIX). Sin embargo, se puede considerar una buena aproximación para la obtención de datos sobre las operaciones realizadas por los usuarios durante una conexión a un buscador.

El parámetro clave para la correcta determinación de las sesiones es el tiempo máximo de espera entre dos peticiones consecutivas. En trabajos previos, Silverstein et al. en su análisis sobre Altavista [Silverstein, 99] consideraban que un período de 5 minutos era suficiente. Sin embargo, en este caso se ha preferido aumentar este tiempo a 30 minutos para evitar pérdidas de peticiones que pudiese espaciarse en el tiempo.

Teniendo esto en cuenta, se compararon un total de 57.529 sesiones (ver Tabla 2-8). Los resultados muestran que en una sesión típica un usuario realiza 1,7 búsquedas, valor levemente inferior al obtenido en [Jansen, 98] y [Silverstein, 99] que era de aproximadamente 2,02 y 2,3, respectivamente. El hecho de que este valor sea tan bajo indica que el usuario típico se conecta al buscador, realiza una búsqueda y si los resultados no son satisfactorios intentará modificará la cadena búsqueda para intentar refinar, pero normalmente no sigue con el proceso de búsqueda de manera repetitiva.

Respecto a las categorías accedidas durante una sesión es necesario recalcar que aunque la media sea de 1,76, todos los usuarios acceden al buscador a través de la categoría principal, por lo que en realidad, esta media se convierte en 0,76, indicando que a pesar de tratarse de un directorio Web el sistema analizado, los usuarios no son muy propensos a navegar por las categorías.

Finalmente, otro dato importante es el número medio de documentos visitados durante una sesión: 2,46. Esto demuestra que los usuarios son muy exigentes, ya que de los resultados

Número de sesiones	57.529
Duración media sesiones	570,53 segs
Búsquedas por sesión	1,7037
Categorías por sesión	1,7622
Documentos por sesión	2,4606

Tabla 2-8: Datos de sesiones de usuario

consultados (normalmente diez o veinte), solamente consultarán aquellos que consideren relevantes, de ahí la importancia de los algoritmos de ordenación para facilitar la localización de los documentos relevantes a los usuarios.

Duración de sesiones	Porcentaje
0-500 segs	72,131%
500-1.000 segs	10,878%
1.000-1.500 segs	6,315%
1.500-2.000 segs	3,940%
2.000-2.500 segs	2,182%
2.500-3.000 segs	0,943%
3.000-3.500 segs	0,576%
3.500-4.000 segs	0,411%
Más	2,624%

Tabla 2-9: Duración de sesiones

Como se muestra en la Tabla 2-8 la duración media de una sesión es de 570,5 segundos, aproximadamente 9 minutos y medio, aunque en realidad, como se observa en la Tabla 2-9 la mayoría de las sesiones tienen un tiempo de duración inferior a los 500 segundos.

Esto refleja la forma de actuar del usuario característico, ya que indica que un usuario se conecta a un sistema de búsqueda en el Web para resolver una necesidad concreta y puntual de información, realiza una o dos búsquedas, consulta dos o tres documentos (los que el usuario considera más relevantes), y todo esto en un tiempo inferior a los 9 minutos. Esta forma de actuar hace que los buscadores se conviertan en herramientas que deben ofrecer a sus usuarios una capacidad de búsqueda puntual, precisa y rápida.

2.7. Distribuciones de búsquedas, categorías y documentos

En esta sección se describe la componente más novedosa e interesante del análisis realizado y su principal objetivo: determinar si los distintos tipos de solicitudes recibidos por un buscador se pueden ajustar a alguna distribución matemática y comprobar si existe alguna relación entre las distintas peticiones.

En los siguientes apartados se describen las características principales del análisis y las conclusiones más relevantes, si bien, en el Apéndice A se detallan los pasos de cada uno de los estudios de las distribuciones y sus relaciones.

2.7.1. Distribuciones de búsquedas

Este estudio se basa en las más de cien mil peticiones de búsqueda recibidas, con las respectivas fecha y hora de realización. A primera vista parece lógico intentar ajustar una distribución Exponencial sobre la diferencia de tiempos entre peticiones. Debido a que la precisión horaria se establece hasta niveles de segundos y en determinados casos se producen varias peticiones en el mismo segundo, es necesario cambiar la variable de estudio hacía una variable discreta que contabilice el número de búsquedas realizadas en un período significativo de tiempo, en este caso, un minuto, lo que se equipara a un

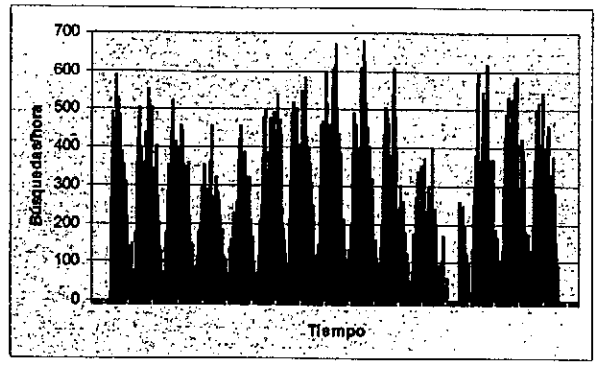

Figura 2-6: Búsquedas por hora durante el período analizado

proceso de Poisson de parámetro λ. En realidad, la distribución Exponencial es el equivalente continuo de un proceso discreto de Poisson.

El principal problema al intentar ajustar una distribución de Poisson consiste en que el parámetro λ, o lo que es lo mismo la media de la distribución es variable en el tiempo. Durante el período analizado, se realizaron 307,5 búsquedas por hora (o 5,12 búsquedas por minuto) de media, pero como se observa en la Figura 2-6, este dato es poco representativo ya que existe una gran variabilidad en los datos.

En este gráfico se observa como el número de búsquedas realizadas varía en función de la franja horaria analizada: los valores más elevados (entre 500 y 600 búsquedas por hora) se corresponden con la primera hora de la noche, mientras que los valores más bajos (sobre 100 búsquedas por hora) se corresponden con las horas de madrugada. A mayores, también se puede observar cierta fluctuación en función del día de la semana, ya que los fines de semana se reducen, a nivel general, las consultas realizadas.

Por lo tanto, esto sugiere realizar un análisis de los datos agrupados en series, en donde el parámetro λ permanezca estable. Obviamente, la duración de la serie es un parámetro

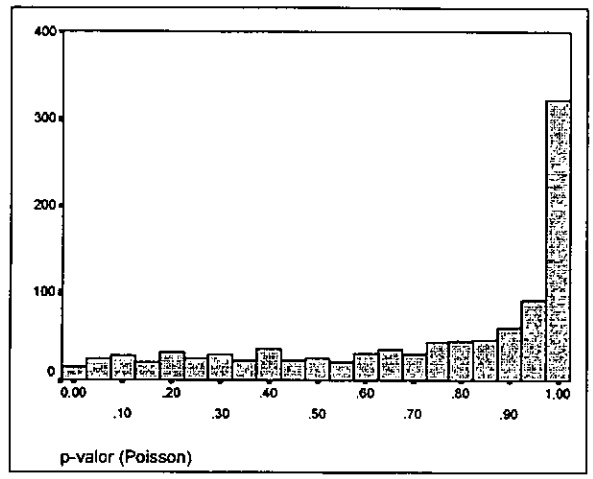

Figura 2-7: Histograma de p-valor obtenido al aplicar Kolmogorov-Smirnov a las series de 20 minutos de búsquedas

crítico que determinará si el valor de λ fluctúa o no.

Inicialmente, se ha considerado como duración de cada serie una hora, lo que genera 340 distribuciones con 60 elementos cada una de ellas. A cada una de estas distribuciones se le aplicó el test de Kolmogorov-Smirnov contrastando si la distribución original podría ser ajustada a una distribución de Poisson o a una distribución Normal. Los resultados reflejaban que en la gran mayoría de los casos se podía rechazar que las distribuciones originales se ajustasen a una distribución Normal. Sin embargo, aunque en la mayoría de las distribuciones estudiadas se podría aceptar que se ajustasen a una Poisson, los p-valores obtenidos no eran muy elevados e incluso, en un porcentaje no despreciable de casos, se debería rechazar esta hipótesis.

En consecuencia, esto lleva a replantear el tiempo considerado como duración de la serie, ya que probablemente, dentro de una distribución se estén produciendo variaciones en el parámetro λ que afecten a los tests. Por este motivo, se consideró un intervalo de duración de 20 minutos, obteniendo 1.020 distribuciones con 20 elementos cada una de ellas. Se repitió el procedimiento anterior, y en este caso la hipótesis de ajuste a una distribución de Poisson se apoyaba con más fuerza, ya que la amplia mayoría de las distribuciones (aproximadamente un 85%) se pueden considerar distribuciones de Poisson, con p-valores muy elevados (superiores al 90%), y como se muestra en la Figura 2-7 la gran mayoría de las series se ajustan con un p-valor del 100%.

De todas formas, y para confirmar esta primera parte del estudio, se decidió agrupar aquellas distribuciones con un comportamiento similar (intentado mantener el parámetro λ estable) y repetir los tests de Kolmogorov-Smirnov. De esta forma se obtuvieron 92 distribuciones que consideraban intervalos de tiempo mayores, de las cuales en un 95% se podría aceptar que se ajustasen a una distribución de Poisson.

Por lo tanto, se puede concluir que el número de búsquedas que generan los usuarios durante un minuto se ajusta a una distribución de Poisson, aunque con un parámetro λ variable a lo largo del tiempo. De hecho, el valor máximo de λ analizado ha sido de 11,3 búsquedas por minuto, frente al valor mínimo que ha sido 0 búsquedas por minuto.

2.7.2. Distribuciones de categorías

El caso de las categorías accedidas por los usuarios es muy similar al de las búsquedas. En primer lugar, se ha realizado una transformación de los datos con el objetivo de obtener una variable que mida el número de accesos a categorías realizados por minuto e intentar comprobar su ajuste a una distribución de Poisson o a una Normal.

Durante el período analizado se han producido una media de 258 accesos a categorías por cada hora (aproximadamente 4,3 visitas a categorías por minuto), aunque como se observa en la Figura 2-8 el número medio de categorías visitadas fluctúa a lo largo del tiempo, prácticamente de igual manera que las búsquedas realizadas por los usuarios.

Por este motivo, el procedimiento seguido es similar al anterior. Inicialmente se han agrupado los accesos a categorías en series de 1 hora de duración, obteniendo 340 series, compuesta cada una de ellas de 60 elementos. A continuación se aplicó el test de

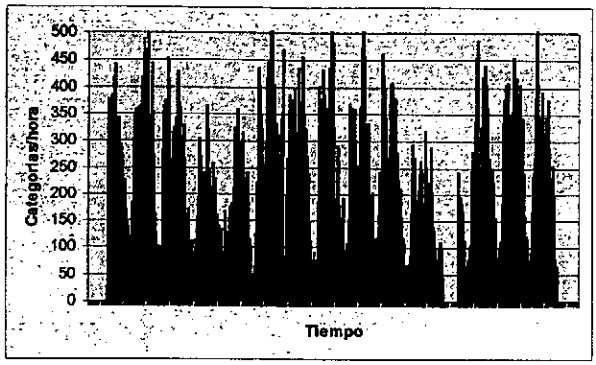

Figura 2-8: Accesos a categorías por hora durante el período analizado

Kolmogorov-Smirnov a todas y cada una de las series, contrastando si se podrían ajustar a una distribución Normal o de Poisson. Los resultados obtenidos no permitían ajustar las series a una distribución Normal ya que se rechazaba esta hipótesis en la mayoría de los casos, y aunque en el caso del proceso de Poisson el nivel de aceptación era superior, no se podía considerar suficientemente significativo.

En consecuencia, se realizó una agrupación de los datos en series de 20 minutos, generando 1.020 distribuciones con 20 elementos cada una de ellas, a las cuales se les aplicó el test de Kolmogorov-Smirnov para determinar su similitud a las distribuciones Normal y de Poisson. En este caso los resultados fueron definitivos, ya que aunque se podría aceptar en más de la mitad de los casos que las series se ajustasen a una Normal, como se observa en la Figura 2-9, la amplia mayoría de las series se ajustan con un p-valor superior al 90% a un proceso de Poisson.

Igualmente, y para mayor seguridad, se decidió formar agrupaciones de aquellas series con un comportamiento más homogéneo entre sí, con el objetivo de localizar períodos de tiempo mayores con una media estable. En este caso se crearon 114 grupos a los cuales se volvió a aplicar el test de Kolmogorov-Smirnov, obteniendo que en ninguno de los casos

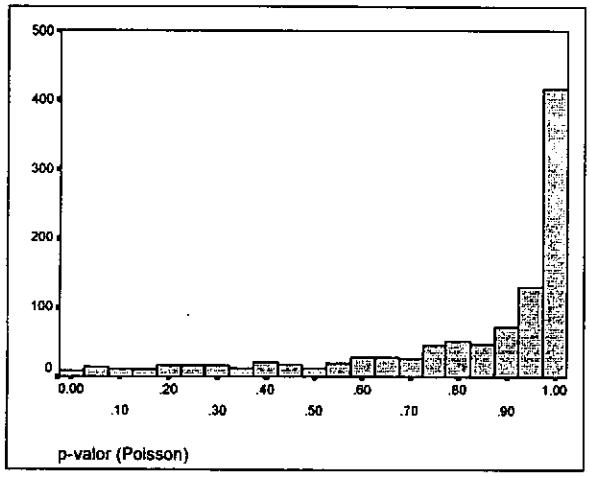

Figura 2-9: Histograma de p-valor obtenido al aplicar Kolmogorov-Smirnov a las series de 20 minutos de categorías

se podría rechazar la hipótesis de que se ajustasen a una distribución de Poisson.

En resumen, se puede concluir que el número de accesos a categorías realizados por los usuarios durante un minuto se ajusta a una distribución de Poisson, con λ variable en el tiempo, al igual que sucede con el caso de las búsquedas. Como dato interesante, indicar que el valor máximo de λ obtenido ha sido de 7,9 accesos a categorías durante un minuto.

2.7.3. Distribuciones de documentos

Al igual que en los casos anteriores, los documentos examinados por los usuarios parecen seguir una misma distribución, aunque la media es variable a lo largo del tiempo (ver Figura 2-10). En concreto, se han consultado una media de 458,3 documentos por hora (unos 7,6 documentos por minuto) aunque, como se observa en la Figura 2-10 y como sucedía en los casos anteriores, la media de documentos visitados varía en función de la franja horaria analizada.

En base a esto y a los casos previos, se han agrupado en series que engloban los accesos a documentos en un minuto durante 1 hora, obteniendo 340 series con 60 elementos. Al igual que en los casos anteriores, se ha aplicado el test de Kolmogorov-Smirnov para contrastar el ajuste a una distribución Normal o de Poisson. El resultado indica que se debería rechazar su ajuste a una distribución Normal en la gran mayoría de los casos, mientras que para el caso del proceso de Poisson, aunque se debe rechazar en un porcentaje importante de series, también se puede comprobar que hay un porcentaje significativo en donde se acepta con p-valores elevados, lo que apunta hacía la corrección del tiempo de duración de la serie.

En consecuencia se agruparon las series en colecciones de 20 elementos, y por lo tanto 20 minutos de duración. De igual forma, se les aplicó el test de Kolmogorov-Smirnov a las 1.020 distribuciones obtenidas, y en este caso, como se observa en la Figura 2-11, se acepta con un p-valor elevado que, en la mayoría de los casos, se pueden ajustar a un proceso de Poisson. En cambio, el ajuste a una distribución Normal aunque se podría aceptar en un amplio porcentaje no presenta unos p-valores tan elevados.

Por último, y como apoyo al ajuste de la distribución de Poisson se agruparon aquellas distribuciones con un comportamiento a nivel de media más semejante para crear

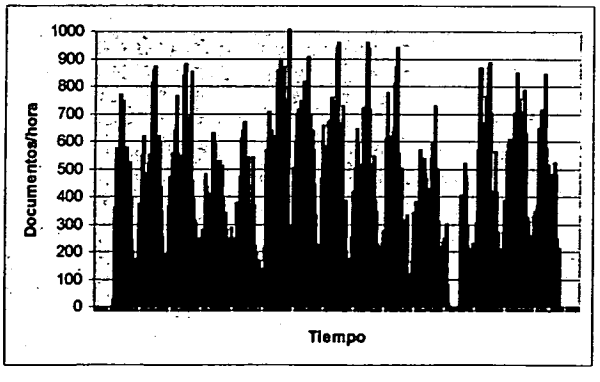

Figura 2-10: Documentos examinados por hora durante el período analizado

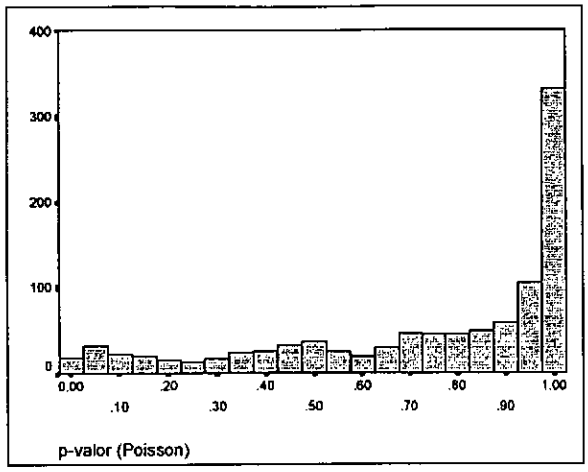

Figura 2-11: Histograma de p-valor obtenido al aplicar Kolmogorov-Smirnov a las series de 20 minutos de documentos examinados

distribuciones que engloben períodos de tiempos mayores. Se obtuvieron 74 grupos a los cuales se les volvió a aplicar el test de Kolmogorov-Smirnov obteniéndose que en ningún caso se podría rechazar la hipótesis de ajuste a un proceso de Poisson.

Finalmente, se puede concluir que el número de documentos examinados por los usuarios durante un minuto sigue el modelo de un proceso de Poisson, con λ variable en el tiempo. En este caso, el valor máximo de λ ha sido de 16 documentos visitados por minuto.

2.7.4. Relaciones entre búsquedas, categorías y documentos

En los apartados anteriores se han analizado las distribuciones a las que se ajustan las búsquedas realizadas en un directorio por minuto, las categorías accedidas por minuto y los documentos examinados por minuto, por el conjunto de todos los usuarios. La conclusión es clara: las tres distribuciones se ajustan a un proceso de Poisson con su media, λ, fluctuando en función de la franja horaria. Sin embargo, es necesario realizar un análisis más detallado que permita discernir si existe alguna relación entre las tres medias o parámetros, denominados λ_{busq}, λ_{cats} y λ_{docs}.

En principio, es razonable suponer que cuando se produzca un aumento en el número de búsquedas por minuto también se producirá un aumento en los documentos consultados y las categorías accedidas, simplemente porque se supone que hay un mayor número de usuarios conectados. No obstante, es importante identificar el tipo de relación existente y aproximar la relación para poder estimar y predecir valores.

Este estudio se abordará como un problema de regresión simple, considerando las búsquedas como la variable independiente y las categorías y los documentos como las variables dependientes. En una primera parte se analiza la relación entre las búsquedas y las categorías considerando las variables aleatorias discretas: "*Número de búsquedas por minuto*" y "*Número de accesos categorías por minuto*". Debido al gran volumen de datos tratados (más de 20.000 valores), y al tratarse de valores discretos en un rango de valores limitado no permite un análisis gráfico adecuado. En consecuencia, se decide tomar las

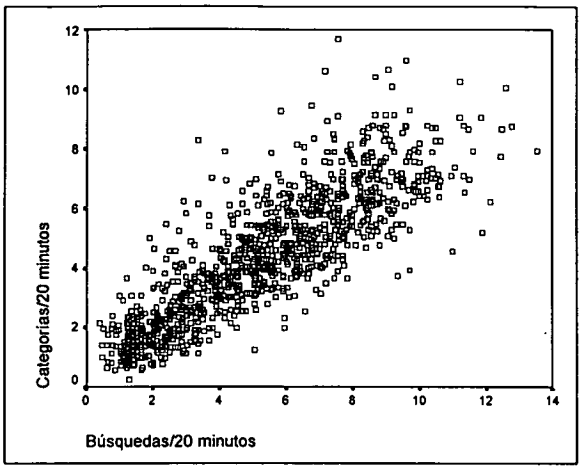

Figura 2-12: Relación búsquedas-categorías

medias de los valores cada 20 minutos, o lo que es lo mismo, los parámetros λ muestrales de las distribuciones analizadas en las secciones anteriores.

De esta forma se obtienen 1.020 pares de valores, que al representarlos gráficamente se obtiene la Figura 2-12.

En el gráfico se puede observar claramente la relación lineal existente entre las búsquedas realizadas por los usuarios y el número de categorías visitadas por los mismos. Sin embargo, al realizar un análisis de regresión sobre las dos variables ajustando un modelo lineal se comprueba que la hipótesis de normalidad de los residuos del modelo no es aceptada. La causa de este hecho se deriva de la fuerte relación de dependencia existente entre los datos observados, ya que claramente los datos muestrales se ajustan a un proceso autorregresivo de orden uno. Esto es, cuando en un instante de tiempo se producen un número elevado de búsquedas es muy probable que en el siguiente instante también se produzcan un número elevado de búsquedas, ya que el número de usuarios no habrá variado significativamente.

En este punto se plantean dos alternativas. En primer lugar, utilizar métodos de regresión dinámica para intentar combinar el estudio de regresión con la serie de tiempos, aunque

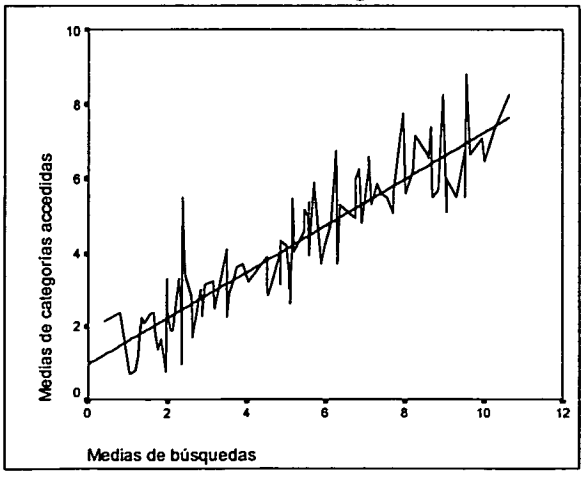

Figura 2-13: Modelo lineal para la relación búsquedas-categorías

este proceso es bastante complejo y los resultados obtenidos serían adecuados para predicción, lo cual no es el objetivo buscado en este caso. La siguiente alternativa consiste en aleatorizar los datos muestrales para eliminar la componente de dependencia entre ellos. Esta solución es posible ya que se dispone de un número elevado de datos y por consiguiente la pérdida de información es limitada.

Por lo tanto, aleatorizando la muestra permanecen un 10% de los datos originales, sobre los que se realiza el análisis de regresión de un modelo lineal. En la Figura 2-13 se observa el modelo lineal ajustado a la muestra.

Al aplicar el análisis regresivo se obtiene un coeficiente de correlación del 80% lo que confirma la hipótesis del modelo lineal, y además en este caso, los residuos se pueden considerar normales de media cero, homocedásticos e independientes.

El modelo lineal que relaciona el número de categorías visitadas con el número de búsquedas realizadas es el siguiente:

$$\lambda_{cats} = 0{,}626478\lambda_{busq} + 0{,}975086$$

Para confirmar la corrección del modelo se ha repetido el análisis tomando distintas aleatorizaciones, obteniéndose resultados similares.

El análisis de regresión para estudiar la relación entre las búsquedas y los documentos es similar a este último. En la Figura 2-14 se observa la clara relación lineal existente entre el número medio de búsquedas realizadas durante 20 minutos y el número medio de documentos accedidos en el mismo período.

Obviamente, en este caso también se presenta la dependencia existente entre los datos por lo que directamente se decide abordar el problema mediante la aleatorización de los mismos.

El modelo lineal ajustado en este caso se puede observar en la Figura 2-15. El modelo lineal obtenido posee un coeficiente de correlación elevado (del 73%) lo que confirma la relación existente entre las búsquedas y los documentos visitados.

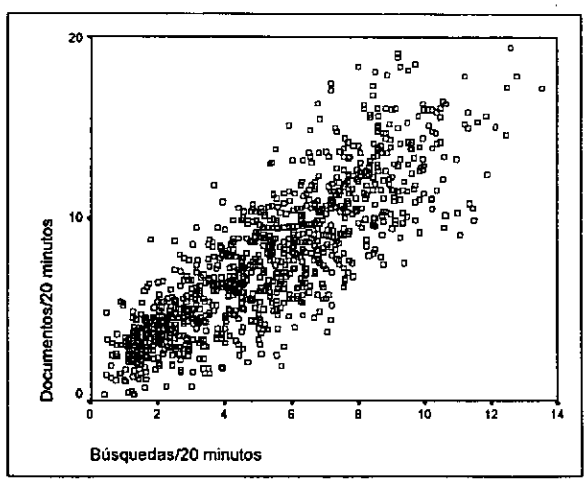

Figura 2-14: Relación búsquedas-documentos

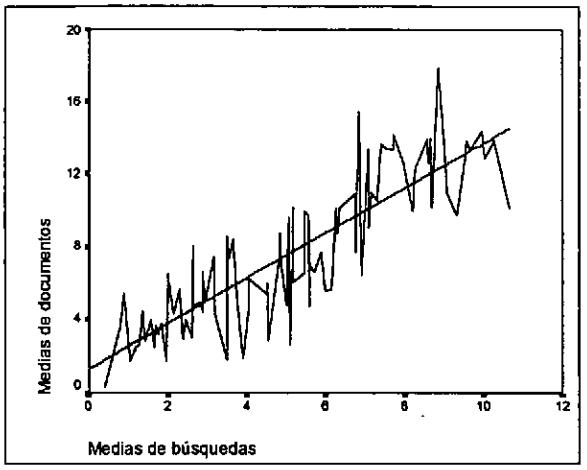

Figura 2-15: Modelo lineal para la relación búsquedas-documentos

El correcto ajuste del modelo se corrobora con el estudio de los residuos en donde se comprueba que se pueden ajustar a una Normal de media cero, se mantiene la hipótesis de homocedasticidad y la independencia de los residuos.

El modelo que relaciona el número de documentos visitados con las búsquedas realizadas se ajusta a la siguiente expresión:

$$\lambda_{docs} = 1{,}24202\lambda_{busq} + 1{,}294434$$

En este caso también se han repetido los análisis sobre distintas aleatorizaciones obteniendo resultados similares.

2.8. Conclusiones

En este capítulo se ha analizado en detalle el comportamiento de los usuarios frente a un sistema de búsqueda en Internet concreto. En la primera parte del estudio se han corroborado los estudios previos (básicamente [Jansen, 98], [Kirsch, 98] y [Silverstein, 99]) que indicaban que los usuarios de un sistema de recuperación de información en el Web difieren significativamente de los usuarios de un sistema de recuperación de información tradicional.

En concreto, los principales aspectos de diferenciación son los siguientes, algunos de los cuales ya han sido apuntados en los trabajos previos y otros reflejan aspectos novedosos no analizados previamente:

- Los usuarios Web usan muy pocas palabras en cada búsqueda, normalmente una o dos palabras, lo que provocará que los resultados obtenidos sean muy genéricos.
- Un aspecto no analizado hasta ahora se refiere al número de resultados obtenido en las búsquedas, que a nivel general es muy elevado, como consecuencia del punto anterior. De media, cada búsqueda ha obtenido más de 500 resultados (aproximadamente, el 0.85% del total de los documentos indexados en el directorio Web).

- Los usuarios Web consultan pocas pantallas de resultados, aproximadamente un 80% de los usuarios únicamente consultará las 2 primeras pantallas, lo que da una gran relevancia a los algoritmos de ordenación utilizados por los buscadores.
- Los usuarios Web tienden a buscar conceptos similares, como se comprueba a través de los términos de búsqueda, lo que apunta hacia la utilización de conceptos de caché para agilizar las búsquedas. Por otra parte, es importante destacar que también existe un porcentaje relevante de búsquedas que han sido realizadas una única vez.
- Los usuarios Web tienden a no emplear operadores en las búsquedas, ni lógicos ni simples. Esto denota que el usuario típico no tiene un gran conocimiento sobre la lógica Booleana. En cualquier caso, tienden a utilizar operadores lógicos en su lengua materna u operadores simples.
- Asimismo, como aspecto novedoso se establece que los usuarios tienden a consultar documentos similares, derivado del hecho de que realizan búsquedas similares.
- Anteriormente no habían sido analizados los accesos a un directorio Web, por lo que la información sobre los accesos al conjunto de categorías es trascendental. En concreto, a pesar de estar utilizando un directorio Web, la navegación por categorías es bastante reducida (de media, los usuarios ven 0,75 categorías a parte de la categoría inicial), y lo mismo sucede para las búsquedas restringidas a los documentos pertenecientes a una categoría. Además, la navegación por las categorías no suele realizarse de manera profunda.

Estas características a nivel general permiten definir el perfil de un usuario típico como alguien con una necesidad de información concreta y puntual (ya que la navegación por categorías es bastante reducida) y que accede a un sistema de búsqueda en el Web para intentar resolverla. Normalmente, realizará 1 ó 2 consultas, aunque constituidas por pocos términos de búsqueda, lo que producirá un gran número de resultados de los que examinará fundamentalmente los 20 primeros y seleccionará un número reducido (únicamente 2 ó 3) para su examen directo.

Sin embargo, el aspecto más novedoso e interesante se centra en el estudio sobre el conjunto de usuarios y los accesos al buscador. En concreto, se ha demostrado que el número de búsquedas realizadas por minuto, el número de categorías accedidas por minuto y el número de documentos examinados por minuto se ajusta a un modelo de Poisson. En los tres casos, el parámetro característico de la distribución, λ, es variable en el tiempo, aunque se demostró que existe una relación lineal entre los parámetros de las tres distribuciones.

Estos resultados son claves para la elaboración del siguiente capítulo en donde, en base a las conclusiones aquí obtenidas, se describe un simulador de usuarios de un sistema de búsqueda en Internet. Esta herramienta de simulación será fundamental para la correcta evaluación del rendimiento ofrecido por los buscadores, ante diferentes niveles de carga.

3. HERRAMIENTA PARA LA EVALUACIÓN DE SISTEMAS DE BÚSQUEDA EN EL WEB

3.1. Introducción

Antes de la implementación final de un sistema de recuperación de información es necesario realizar la evaluación del sistema. El tipo de evaluación considerado depende de los objetivos del sistema de búsqueda. Las medidas más habituales del rendimiento de un sistema son el tiempo y el espacio. Cuanto más corto sea el tiempo de respuesta y menor el espacio utilizado, mejor considerado será el sistema. Obviamente, existe un balance entre tiempo y espacio, el cual frecuentemente permite una compensación entre uno y otro. La determinación de las medidas de tiempo y espacio es lo que constituye la evaluación del rendimiento.

A pesar de que el núcleo central de este capítulo sea la evaluación del rendimiento de los sistemas de búsqueda en el Web, es necesario mencionar otras medidas que constituyen la evaluación del rendimiento de la recuperación, que se centran principalmente en la calidad de los documentos recuperados en la búsqueda.

En las siguientes secciones se describen los diferentes métodos de evaluación del rendimiento del sistema, como de la calidad de los documentos recuperados. A continuación, se describe una herramienta de simulación diseñada e implementada (en base al análisis descrito en el capítulo anterior) para la evaluación del rendimiento de sistemas de búsqueda en el Web ([Cacheda, 01b]).

3.1.1. Evaluación del rendimiento de la recuperación

Tradicionalmente las principales medidas de la calidad de los documentos recuperados por un sistema de búsqueda han sido la precisión y la exhaustividad. Estas medidas permiten realizar comparaciones a partir de un ejemplo de solicitud de información o documentos, I (usando normalmente como base una colección de referencia para las pruebas) y el conjunto de documentos relevantes para esa consulta, R. Asumiendo que una estrategia de búsqueda procesa la solicitud de información I y genera un conjunto de documentos de respuesta A, se define $|A|$ como el número de documentos en este conjunto y se define $|Ra|$ como el número de documentos en la intersección de los conjuntos R y A (ver Figura 3-1) [Baeza-Yates, 99a].

La precisión y la exhaustividad se definen como sigue:

- Precisión se define como la fracción de documentos recuperados (el conjunto A) que son relevantes:

$$precisión = \frac{|Ra|}{|A|}$$

- Exhaustividad se define como la fracción de documentos relevantes (el conjunto R) que han sido recuperados:

$$exhaustividad = \frac{|Ra|}{|R|}$$

La precisión y la exhaustividad asumen que la totalidad de los documentos en el conjunto A han sido examinados por el usuario. Sin embargo, un usuario no necesariamente tiene porque haber consultado la totalidad de los documentos del conjunto A. Por el contrario, se genera una lista con los documentos del conjunto A ordenados según su orden de relevancia aplicando un determinado algoritmo de ordenación o de ranking, que el usuario examinará en orden. En este caso, las medidas de precisión y exhaustividad varían en

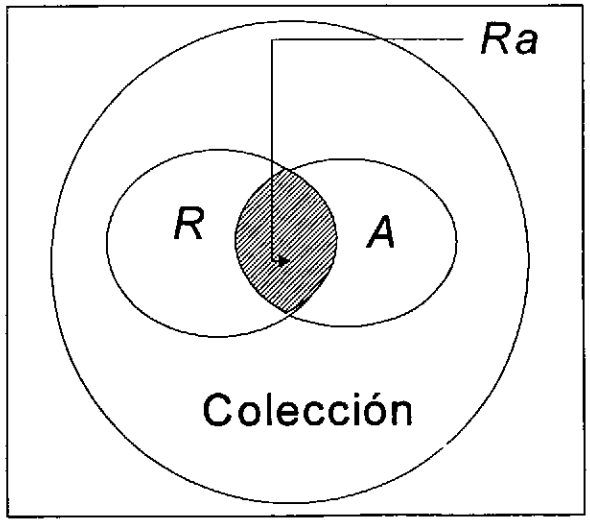

Figura 3-1: Precisión y exhaustividad para un ejemplo de solicitud de información

función de los documentos examinados por el usuario. Para paliar este problema se suelen utilizar gráficos que muestran la exhaustividad frente a la precisión [Korfhage, 97], en función del porcentaje de documentos analizados.

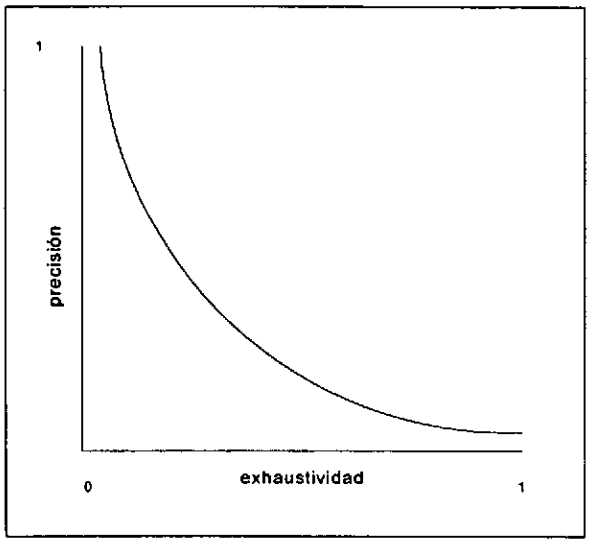

Figura 3-2: Curva exhaustividad-precisión típica

A pesar de la popularidad de estas medidas, han ido surgiendo diferentes medidas que intentan aportar nuevas perspectivas a la evaluación de los sistemas de recuperación de información. Por ejemplo, han surgido diversas medidas que combinan la precisión y la exhaustividad en un único valor como la media armónica [Shaw, 97] o la medida de evaluación E [Rijsbergen, 79]. La precisión y la exhaustividad se basan en la asunción de que el conjunto de documentos relevantes para una consulta es el mismo independientemente del usuario. Sin embargo, diferentes usuarios pueden tener diferentes interpretaciones sobre que documentos son relevantes y cuales no, por lo que se han propuesto nuevas medidas orientadas al usuario, como el porcentaje de cobertura, el porcentaje de novedad, la exhaustividad relativa y el esfuerzo de exhaustividad [Korfhage, 97].

3.1.1.1. Evaluación del rendimiento de la recuperación en el Web

Las características de la evaluación de la calidad de los buscadores en el Web difieren significativamente de la evaluación tradicional, precisamente por las características del propio World Wide Web:

- Dinamismo del Web y de los propios motores de búsqueda.
- Heterogeneidad de los documentos y de las consultas realizadas por los usuarios.
- La estructura de hiperenlaces existente entre las páginas Web.
- La gran cantidad de documentos obtenidos en una búsqueda, lo que hace prácticamente imposible la evaluación de la totalidad de los documentos recuperados.

En base a esto, la noción de relevancia en los documentos recuperados presenta ciertos matices al ser aplicada a páginas Web, principalmente debido a la presencia de enlaces

entre las páginas. Por lo tanto, se puede considerar una página como relevante si su contenido se considera significativo por el usuario, mientras que se puede considerar una página útil si contiene enlaces hacia páginas relevantes [Agosti, 01]. Esto genera diversas categorías de relevancia-utilidad tal y como se muestra en la Tabla 3-1.

PÁGINA RECUPERADA	PÁGINAS ENLAZADAS	
	relevante	**no relevante**
relevante	relevante y útil	relevante
no relevante	útil	no relevante e inútil

Tabla 3-1: Categoría de relevancia-utilidad de las páginas recuperadas

En consecuencia, las medidas de efectividad de sistemas de recuperación usadas en experimentos de laboratorio (precisión y exhaustividad) únicamente ofrecen una información parcial en la evaluación de motores de búsqueda, ya que no consideran la existencia de enlaces entre los documentos. Por lo tanto, se hace necesaria la utilización de nuevas medidas de esta efectividad, como las nuevas estimaciones de la exhaustividad propuestas en [Lowley, 00], o la novedad y el ruido. La novedad se define como la proporción de documentos relevantes obtenidas en una fase concreta, pero omitidos en fases anteriores [Salton, 69]. El ruido se define como la proporción de páginas que no son relevantes y son recuperadas en una fase concreta, pero que han sido omitidas en fases anteriores.

Además, la gran cantidad de documentos obtenidos invalida la posibilidad del cálculo de la precisión y la exhaustividad. Por lo tanto, diversos estudios ([Leighton, 97], [Rosenthal, 96], [Gauch, 96], [Leighton, 96], entre otros) han aplicado el cálculo de la precisión relativa a los primeros diez o veinte documentos recuperados (representada como P@10 ó P@20). Esto representa de una manera más adecuada el procedimiento realizado por los usuarios de un sistema de búsqueda en el Web, y permite obtener un valor, al menos aproximado, de la precisión del sistema.

Asimismo, la cobertura que ofrece cada motor de búsqueda del total de las páginas Web disponibles en Internet es un factor básico para la efectividad del sistema de búsqueda. En el estudio llevado a cabo por Bharat y Broder en [Bharat, 98] se demuestra que la cobertura que ofrecen varios motores de búsqueda ampliamente utilizados en Internet, en el mejor de los casos, no superaría el 50%. Mientras que la parte común entre los motores de búsqueda estudiados sería inferior al 1.4%.

3.1.1.2. Colecciones para la evaluación del rendimiento de la recuperación

En esta sección se describen las principales colecciones de documentos utilizadas para la evaluación de los sistemas de recuperación de información.

Uno de los primeros inconvenientes al que habían de hacer frente al desarrollar sistemas de búsqueda era la falta de puntos de referencia y tests a la hora de analizar el rendimiento del sistema. Inicialmente, la experimentación se basaba en colecciones relativamente reducidas que no reflejaban los entornos reales y no permitían establecer comparaciones entre experimentos válidas.

A principios de los años 90, surgieron unas conferencias anuales, denominadas TREC (Text Retrieval Conference, [TREC, 01]) dedicadas a la experimentación con grandes colecciones de documentos (inicialmente, más de un millón de documentos), diseñándose un conjunto de experimentos de referencia para cada conferencia.

La colección de TREC ha crecido a lo largo de los años, pasando de los 2 gigabytes en TREC-3 en 1.994 a los cerca de 6 gigabytes en TREC-6. Los documentos incluidos provienen de diferentes fuentes como el Wall Street Journal, US Patents y Financial Times, entre otros.

La colección incluye un conjunto de ejemplos de solicitudes de información o consultas que pueden ser empleados para la puesta a prueba de nuevos algoritmos de ordenación. Cada uno de estos ejemplos está especificado en lenguaje natural, y ha sido diseñado y escrito por expertos en la materia. Para cada uno de estas solicitudes se obtienen los documentos relevantes usando los documentos obtenidos por los sistemas de los participantes, lo que se conoce como el *método pooling* [Voorhees, 97] y [Harman, 95].

En las conferencias TREC se utilizan cuatro medidas de evaluación básicas: resumen de estadísticas, medias de exhaustividad-precisión, medias de niveles de documentos y el histograma de precisión media. A nivel global, a través de estas diferentes medidas se presentan los resultados de precisión y exhaustividad obtenidos para las diferentes solicitudes de información planteadas al sistema.

La conferencia TREC se considera la colección como referencia para sistemas de recuperación de información hoy en día, aunque existen otras colecciones como la CACM e ISI, ambas de dimensiones más reducidas y que consideran los documentos estructurados en campos. La colección CACM está formada por 3.204 artículos de las comunicaciones de la ACM (Association for Computing Machinery, [ACM, 01]), mientras la colección ISI está formada por 1.460 documentos del Institute of Scientific Information.

Sin embargo, según iba aumentando el desarrollo de sistemas de búsqueda orientados al Web, también las colecciones de documentos se fueron adaptando, creándose la TREC-8 Web track o WEB-TREC [Hawking, 99]. Esta colección está basada en el conjunto de datos del VLC2 (Very Large Collection, second edition) obtenidos a principios de 1.997 a partir del Internet Archive [Internet Archive, 01]. En total se obtuvieron aproximadamente 18,5 millones de páginas, constituyendo más de 100.000 gigabytes de datos y conformando una representación parcial del estado del World Wide Web. Los objetivos perseguidos se concentran tanto en la obtención de medidas de la eficiencia como de la velocidad obtenida en estos sistemas. Las medidas de efectividad se siguen ponderando en base a la precisión y la exhaustividad, tomando valores relativos al número de documentos analizados, si bien, se abre la posibilidad a la no-consideración de la exhaustividad ya que comúnmente se ha apreciado la poca importancia de la exhaustividad en los buscadores en Internet [Hawking, 99].

3.1.2. Evaluación del rendimiento del sistema

Otro de los parámetros básicos de un sistema de recuperación de información es el tiempo empleado en el proceso de búsqueda y el tiempo empleado en la construcción de las estructuras de datos necesarias para la realización de la búsqueda. De hecho, en

[Kobayashi, 00] se reconoce una relación entre las tres medidas típicas de los sistemas de recuperación de información: velocidad (o tiempo de respuesta), precisión y exhaustividad (ver Figura 3-3).

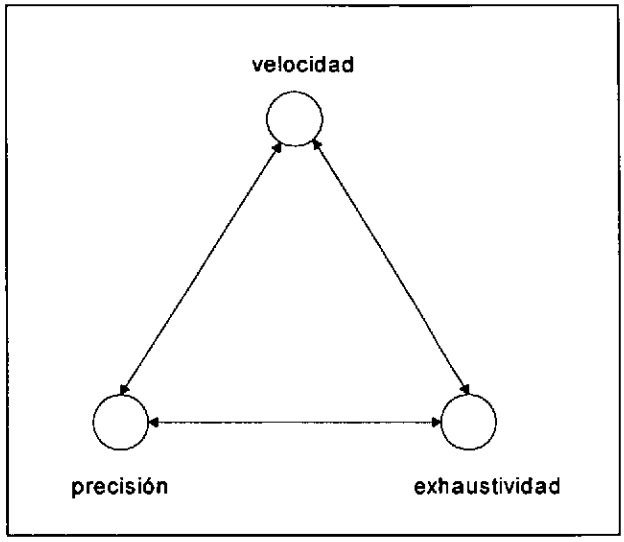

Figura 3-3: Relación entre velocidad, precisión y exhaustividad en un sistema de recuperación de información

El objetivo primordial de todo sistema de búsqueda consiste en obtener un equilibrio entre estas tres medidas, aspecto que se convierte en más complicado cuando se produce un aumento en el número de documentos y/o de usuarios del sistema.

Zobel, en [Zobel, 96], describe de forma detallada los criterios a tener en cuenta a la hora de realizar comparaciones entre varios sistemas de indexación, de los cuales los más relevantes son los siguientes:

- La escalabilidad del sistema, ya que la escala puede modificar el rendimiento relativo obtenido por algún componente del algoritmo, y especialmente en el caso de algoritmos que realicen uso intensivo de accesos a disco.
- Tiempo de respuesta ante una consulta, que constituye por sí mismo el parámetro principal de un sistema de búsqueda.
- Espacio de disco utilizado por las estructuras de datos.
- Tiempo de CPU, tráfico de disco y requerimientos de memoria.

A través de estos parámetros es posible realizar comparaciones entre distintos sistemas de indexado de manera equitativa y precisa. Sin embargo, como se ha comentado, el principal parámetro que permite validar por si solo a un sistema de recuperación de información es el tiempo de respuesta que ofrece ante una consulta ya que, independientemente del resto de parámetros, si éste es muy elevado, el sistema deberá ser descartado.

Aunque este parámetro debe ser medido, como se menciona en [Zobel, 96], no es una medida fácil de realizar, ya que depende de múltiples valores: velocidad de CPU, capacidad de disco, carga del sistema, entre otros.

Este mismo problema se plantea en el caso de los sistemas de búsqueda en Internet, y se complica por el entorno en el que operan, al estar sometidos a situaciones de cargas extremadamente diferentes. De hecho en el WEB-TREC, una de las medidas a establecer consiste en determinar los tiempos de respuesta obtenidos para las solicitudes de información de la colección sobre los documentos de VLC2. Sin embargo, a nivel general, todas las medidas de tiempos de respuesta realizadas sobre sistemas de búsqueda en el Web se suelen realizar considerando una situación de carga nula en el sistema, lo cual en la mayoría de los casos no representa una situación real. Además, el rendimiento que proporciona un sistema de búsqueda puede sufrir variaciones en función del nivel de carga del sistema en su conjunto.

Por lo tanto, se hace necesario la evaluación del rendimiento de un sistema de búsqueda en el Web considerando diferentes situaciones de carga y no únicamente la situación ideal de carga nula. Sin embargo, en un entorno experimental el sistema de búsqueda no dispone de usuarios reales que permitan reproducir las condiciones existentes en el entorno de producción. Con este objetivo, en la sección siguiente se describe una herramienta de simulación diseñada e implementada en base al estudio realizado en el capítulo 2 que se encarga de simular el comportamiento de múltiples usuarios frente a un buscador en Internet. Esta herramienta constituirá un componente crucial para la determinación del rendimiento de los sistemas de búsqueda del World Wide Web ante diferentes niveles de carga de usuarios.

3.2. USim: Herramienta de simulación

En base al estudio previo y a las investigaciones de Jansen [Jansen, 98], Silverstein [Silverstein, 99] y Kirsch [Kirsch, 98] se ha diseñado e implementado una herramienta que se encarga de simular el comportamiento de los usuarios frente a un servicio de búsqueda en el Web.

Esta herramienta de simulación se ha denominado USim (Users Simulator), diseñado para sustituir a los usuarios reales durante el desarrollo de servicios de búsqueda para Internet. USim deberá representar el papel de los usuarios realizando búsquedas, accediendo a categorías y consultando documentos.

En los siguientes apartados se describen los fundamentos matemáticos de su diseño e implementación, su modo de operación y varios casos prácticos a modo de ejemplo.

3.2.1. Diseño e implementación

La herramienta de simulación está compuesta de tres bloques básicos, asociados a los tres tipos de peticiones que se encarga de generar: búsquedas, accesos a categorías y consultas a documentos. Para la simulación de cada uno de estos bloques se han tenido en cuenta los resultados obtenidos en las secciones 2.7.1, 2.7.2 y 2.7.3.

El bloque correspondiente al módulo de la realización de búsquedas es el primero que se tratará, ya que desde un punto de vista lógico es el origen del proceso de simulación. Como se demostró en la sección 2.7.1, las búsquedas realizadas sobre un servicio de búsqueda en

el Web se pueden ajustar a un proceso de Poisson, de media λ (en este caso, medido en búsquedas por minuto), esto es, un proceso que nos indica el número de búsquedas que se van a realizar cada minuto. Sin embargo, para la simulación es más conveniente una distribución que proporcione el tiempo entre dos búsquedas consecutivas, por lo que se ha transformado el proceso de Poisson en la variable continua: "*Tiempo entre dos búsquedas consecutivas*" que se ajusta a una distribución Exponencial de media λ[‡], y cuya función de densidad y de distribución son las siguientes:

$$f(x)=\begin{cases}\lambda e^{-\lambda t} & \forall t>0\\ 0 & \forall t\le 0\end{cases}$$

$$F(t)=1-e^{-\lambda t}$$

En consecuencia, en base al método de inversión o de Montecarlo a partir de la función de distribución y de un parámetro λ concreto, el proceso de simulación es inmediato. Ahora bien, esto permite determinar el momento en el que se van a producir las búsquedas contra el servicio de búsqueda, pero es necesario establecer qué cadena de búsqueda se va a utilizar.

Para resolver este segundo aspecto del bloque de las búsquedas, es necesario tener en cuenta, tal y como se indicó en la sección 2.3, y según lo expuesto en [Jansen, 98], que el conjunto de las cadenas de búsquedas utilizadas por los usuarios de un buscador no se puede considerar aplicable a la ley de Zipf. Por lo tanto, y como en este caso no se dispone de un modelo matemático teórico que se pueda utilizar como base para la simulación de cadenas de búsquedas, se ha considerado conveniente utilizar la distribución empírica obtenida del análisis estadístico previo. Por consiguiente, la obtención de la cadena de búsqueda se calculará a partir de las 26.654 cadenas de búsqueda diferentes analizadas, junto con sus correspondientes frecuencias de repetición.

Otro factor que tiene una repercusión directa en las búsquedas realizadas por los usuarios es el punto de inicio del intervalo de documentos visualizados en los resultados. Como se ha demostrado en la sección 2.3, la mayoría de los usuarios se centran en las dos primeras páginas de resultados (esto es, la página que se inicia con el resultado número 1 y con el resultado número 11), sin embargo existe un porcentaje representativo de otras búsquedas que se inician en otras posiciones. Este hecho puede afectar directamente al rendimiento de un sistema de búsqueda, ya que, por ejemplo, el proceso de ordenación puede sufrir mayores penalizaciones. Para reflejar esto de forma adecuada en la herramienta de simulación se han tenido en cuenta los porcentajes de visualización de cada página de resultados (ver Tabla 2-1), para reproducir las distintas páginas de resultados consultadas por los usuarios. En este punto, es necesario comentar que, debido a que el simulador no está orientado a sesiones, no es posible realizar el proceso llevado a cabo por un usuario al realizar sucesivas búsquedas, si bien, se considera que el efecto que esto puede tener en el rendimiento del conjunto del sistema de búsqueda es mínimo.

[‡] La conversión entre un proceso discreto de Poisson y la distribución continua Exponencial es inmediato a partir de sus definiciones.

El resto de factores, como se ha expuesto en el capítulo anterior, no son modificados apenas por los usuarios, conservando en la mayoría de los casos los valores indicados por defecto. Por lo tanto, no se ha considerado necesario modificar el resto de valores del proceso de búsqueda, como el número de documentos recuperados por página o los campos en donde se realiza la búsqueda.

El siguiente bloque considerado es el encargado de generar los accesos a categorías. En base a la demostración de la sección 2.7.2 que probaba el ajuste a un proceso de Poisson, se realiza, al igual que en el caso anterior, una adaptación a una distribución Exponencial de parámetro λ. Por otra parte, el número de parámetros a considerar es considerablemente menor, ya que está reducido únicamente al identificador de la categoría que se desea consultar.

En este caso, y con el objetivo de ajustar un modelo que se adapte lo más posible al modo de operación real de un servicio de búsqueda, se ha estimado como aproximación más adecuada, la obtención de los identificadores de las categorías a visitar del propio funcionamiento del simulador. Para ello se ha considerado como la categoría inicial, la categoría principal, y a partir de este punto, todas las categorías obtenidas al realizar búsquedas o al consultar otras categorías se registran, formando un conjunto de categorías con una vida limitada. De esta forma, inicialmente se empieza con una categoría (la principal), a continuación se obtienen las categorías de nivel 1, y así, de manera sucesiva, en base a otros accesos a categorías y búsquedas se genera una base de categorías para los siguientes accesos. Obviamente, cada categoría tiene una vida limitada en este registro configurable, pero que típicamente se considerará como la duración de una sesión de usuario común (aproximadamente, 9 minutos y 30 segundos) desde el último acceso obtenido.

El módulo encargado de la generación de los accesos a documentos es muy similar al módulo generador de los accesos a categorías, descrito previamente. En este caso también se emplea una distribución Exponencial para determinar los instantes de tiempo en los que se generará el acceso a un documento. Para el establecimiento del identificador de documento a consultar se sigue un modelo muy similar al anterior. Inicialmente, se dispone de un conjunto vacío con lo cual no se pueden obtener identificadores; sin embargo, según el simulador va realizando búsquedas y consultando categorías, los identificadores de documentos obtenidos se van almacenando en el registro, por lo que al cabo de varias búsquedas y/o accesos a categorías ya se dispone de suficientes identificadores para una ejecución uniforme.

Sobre este último aspecto es necesario puntualizar que los documentos obtenidos no se consideran todos de igual importancia. Por el contrario, se hace una ponderación en función del lugar que ocupan en la lista, dándoles mayor importancia a los documentos de las primeras posiciones frente a los últimos. Aunque no es posible determinar de manera fiable qué documentos serían visitados por los usuarios, se ha considerado una buena aproximación aprovechar la ordenación generada por los algoritmos de ranking.

Finalmente, como nexo de unión entre los tres diferentes módulos se aplican los resultados obtenidos en la sección 2.7.4, en donde se demostraba que existía una relación lineal entre λ_{busq}, λ_{cats} y λ_{docs}, y se presentaba la correspondiente formulación. Por lo tanto, esto permite relacionar los tres módulos entre sí, de tal forma que a partir de un valor concreto de λ_{busq}

es posible calcular (al menos, de manera aproximada) los valores de λ_{cats} y λ_{docs}. Por ejemplo, para una simulación que genere 10 búsquedas por minuto, se deberían generar a su vez aproximadamente 7,23 accesos a categorías por minuto y 13,72 consultas a documentos por minuto.

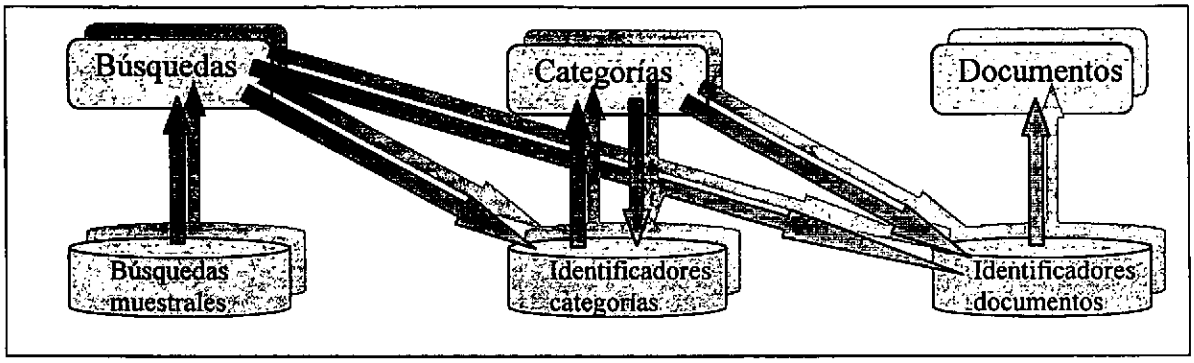

Figura 3-4: Bloques principales de USim

En la Figura 3-4 se muestran los principales módulos de la herramienta de simulación USim, y las interrelaciones existentes entre los distintos bloques tal y como se han descrito en los párrafos anteriores. Estos tres bloques operan de manera concurrente, enviando peticiones al servicio de búsqueda, cuyos resultados son analizados y almacenados para un posterior análisis.

En base a la descripción realizada, USim ha sido diseñado utilizando la metodología de la orientación a objetos y desarrollado íntegramente en Java para garantizar su ejecución en distintas plataformas y facilitar de este modo su utilización en diferentes entornos.

El objetivo final de USim es realizar medidas de los tiempos de ejecución que ofrece un sistema de búsqueda en Internet para los distintos servicios ofrecidos (buscador y directorio), aplicando diferentes niveles de carga al sistema. Por lo tanto, cada vez que es enviada una petición al buscador se almacenarán los siguientes parámetros de la respuesta recibida:

- La fecha y la hora en la que se recibe la respuesta de la petición del servicio de búsqueda.
- El identificador de la petición, que variará en función del tipo de petición entre: el literal de búsqueda, el identificador de la categoría o el identificador del documento solicitado.
- El tiempo transcurrido desde la solicitud de la petición hasta que la petición ha sido completamente atendida (descarga completa del documento HTML de respuesta, sin tener en cuenta las imágenes).
- El número de referencias a imágenes incluidas en la respuesta a la petición.
- El tiempo adicional de descarga de las imágenes.

Adicionalmente, para cada tipo concreto de petición, se realiza un análisis de la página de resultados obtenida para extraer información específica de cada uno de los tipos y almacenarla. En concreto, para el caso de las búsquedas, se obtiene la siguiente información:

- El número de categorías incluidas como resultado de la búsqueda.
- El número total de documentos obtenidos como resultado de la búsqueda.
- El número de documentos mostrados en esa página de resultados.
- La posición del primer documento mostrado, respecto del total.

Asimismo, en caso de que la petición se trate de un acceso a una categoría, durante el análisis de la página de resultados se extrae la siguiente información:

- El número de subcategorías obtenidas al visitar esa categoría.
- El número de documentos disponibles en esa categoría.
- El número de documentos mostrados en la primera página de esa categoría.

En el caso de las peticiones de documentos, no es almacenada ninguna información adicional, archivando únicamente los valores comunes a todas las peticiones.

Esta información es almacenada en tres archivos en formato texto, uno específico para cada tipo de petición, para su posterior análisis una vez finalizada la simulación, utilizando cualquier tipo de herramienta estadística o de procesamiento intensivo de datos.

3.2.2. Operación

En esta sección se describe de forma somera el modo de operación de USim con el objetivo de facilitar la compresión y funcionamiento de los casos prácticos descritos en el siguiente apartado. Si bien, en el Apéndice B se muestra el manual de usuario de USim en donde se realiza una descripción detallada y exhaustiva del modo de operación de USim.

En primer lugar es necesario indicar que la herramienta de simulación puede operar tanto con o sin interfaz gráfica, en base a ficheros de configuración propios. Por comodidad, y con carácter didáctico, se describen los principales parámetros de configuración disponibles a través de la interfaz gráfica, y el modo de operación en general.

En la Figura 3-5 se muestra la pantalla para la definición de los parámetros que afectan a

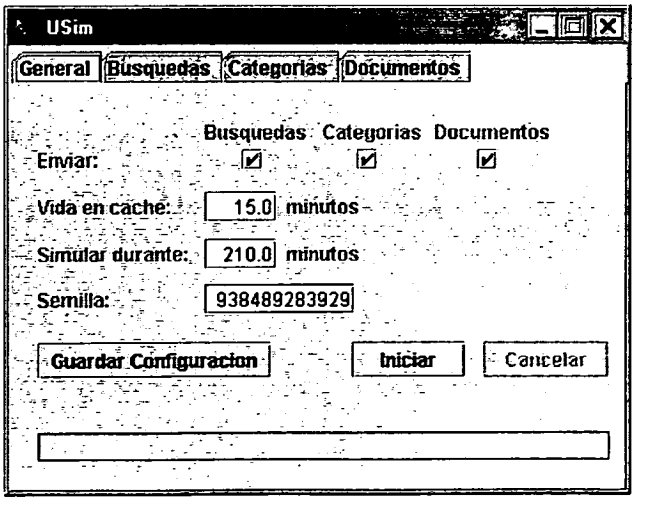

Figura 3-5: Configuración general de USim

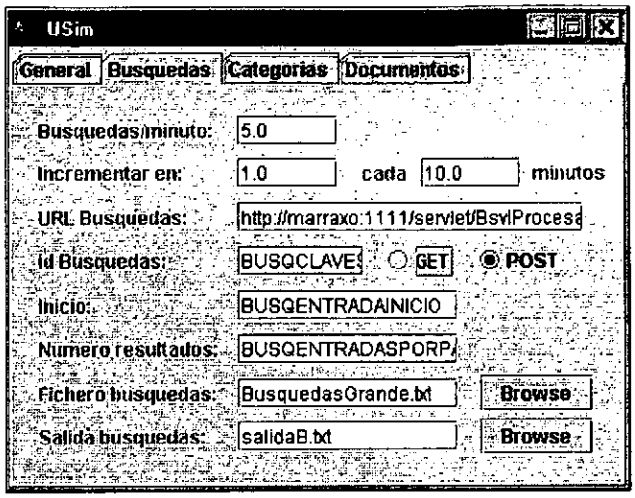

Figura 3-6: Configuración de búsquedas de USim

toda la simulación. En esta parte se indica la duración total de la simulación en minutos y el tiempo de vida de cada una de las categorías y documentos que son recogidos por USim durante la simulación (ver sección 3.2.1). Por otra parte también se puede indicar la semilla utilizada como inicio para la generación de números aleatorios de las distintas distribuciones Exponenciales empleadas durante el proceso simulativo. La configuración utilizada para la simulación puede ser almacenada, adoptando un formato de fichero propio de USim.

En el momento que se inicia la simulación se comprueban qué tipos de peticiones se deben enviar al sistema de búsqueda, y se inicia de manera independiente y concurrente la simulación de cada tipo de petición. El hecho de poder determinar el tipo de peticiones a enviar garantiza la generalidad de esta herramienta de simulación, y su validez para diferentes tipos de sistemas de búsqueda en el Web, como son los directorios (aplicando los tres tipos de peticiones) o los motores de búsqueda propiamente dichos (aplicando únicamente búsquedas y accesos a documentos), o incluso si el sistema no dispone de un servicio de acceso a los documentos se puede eliminar esta petición.

A través del resto de la interfaz gráfica se establecen los parámetros de la configuración de los distintos tipos de peticiones. Estos parámetros son relativamente similares, por lo que a continuación simplemente se describen los parámetros referentes a las peticiones de búsquedas (ver Figura 3-6).

Los principales valores a configurar son el número de búsquedas por minuto que se van a simular (equivalente al parámetro λ_{busq} de la distribución Exponencial asociada) y la URL del servicio de búsqueda con capacidad para soportar tanto CGIs, servlets o cualquier otra tecnología accesible a través del Web. Por otra parte, también es necesario indicar los nombres de los parámetros para la realización de una búsqueda, ya que son dependientes del servicio de búsqueda utilizado. Inicialmente se contemplan tres nombres de parámetros: el literal de búsqueda, la posición inicial del resultado a mostrar y el número de resultados a mostrar, que pueden ser transmitidos al servidor tanto utilizando un método GET como el POST. Mención especial merece el parámetro correspondiente al fichero de búsquedas, ya que se corresponde con el archivo que contiene la distribución empírica obtenida para las búsquedas realizadas por los usuarios (ver sección 3.2.1) y que será utilizado como base para la simulación de las búsquedas. Por último el archivo de salida

apunta al fichero utilizado para el almacenamiento de los resultados progresivos obtenidos de la simulación de este tipo concreto de peticiones.

Además, el valor de λ_{busq} no es estático sino que puede variar a lo largo de la simulación, ya que es posible indicar incrementos fijos de su valor a intervalos regulares de tiempo. Esto permite la evaluación de un sistema bajo diferentes cargas durante una misma simulación.

La configuración de las categorías y los accesos a documentos es similar. En este caso, se calcula automáticamente el valor de λ_{cats} y λ_{docs}, así como los incrementos a partir del valor indicado de λ_{busq} aplicando el modelo de relación lineal indicado en el apartado 2.7.4. En este caso también se indican las URLs para acceder a los servicios que muestran una categoría o un documento, con los correspondientes nombres de sus parámetros (identificador de la categoría o de documento, respectivamente).

Una vez que USim se encuentra debidamente configurado, la configuración se puede archivar en un fichero para futuros usos, pudiendo iniciarse la simulación directamente desde la interfaz gráfica, o en su defecto, iniciar la simulación desde línea de comandos sin necesidad de activar la interfaz gráfica, simplemente invocando una configuración previamente creada.

3.2.3. Casos prácticos

USim se puede considerar una herramienta de evaluación del rendimiento de un sistema de búsqueda en Internet que puede ser aplicada de diferentes maneras, consiguiendo diferentes resultados y medidas empíricas sobre un sistema real. En los siguientes apartados se describen dos de sus usos principales como herramienta de evaluación.

3.2.3.1. Medida del punto de saturación

Una de las principales medidas a la hora de poner en producción un servicio de búsqueda en Internet es el número máximo de peticiones que soportará por minuto. Es evidente que, a partir de un determinado umbral, el rendimiento ofrecido por un sistema cae bruscamente elevándose los tiempos de respuesta. Ese umbral se ha denominado *punto de saturación.*

El conocimiento del punto de saturación es fundamental, ya que permite establecer medidas preventivas basadas en técnicas de gestión de sistemas con el fin de evitar la superación de este umbral, con la consiguiente merma en el tipo de servicio ofrecido a los usuarios.

En este caso USim permite de manera sencilla someter al sistema al efecto producido por múltiples usuarios y de manera controlada. Para ilustrar esto se ha elaborado un experimento en donde se pretende determinar el punto de saturación de un servicio de búsqueda ofrecido por un prototipo de un directorio Web, disponible en una máquina Ultra Enterprise 250 con un procesador a 300 MHz y 768 MB de memoria.

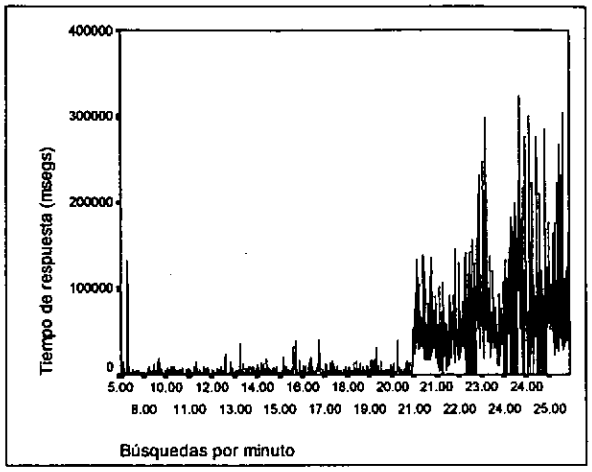

Figura 3-7: Tiempos de respuesta frente al número de búsquedas por minuto

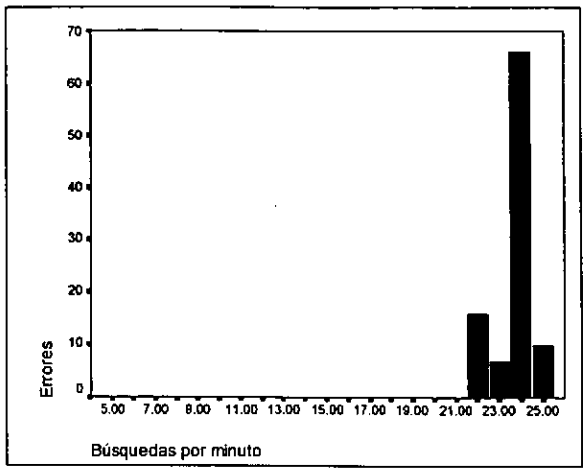

Figura 3-8: Fallos en el servicio frente a búsquedas por minuto

El modo de operación es muy simple, consiste únicamente en preparar USim para realizar una única simulación en donde el parámetro λ_{busq} variará de manera uniforme a lo largo del tiempo, y en consecuencia también los parámetros asociados λ_{cats} y λ_{docs}. Se inicia la simulación con un valor bajo, por ejemplo, 5 búsquedas por minuto (y 4.1 accesos a categorías por minuto y 7,5 consultas a documentos por minuto), con un incremento periódico de esta tasa. En este caso concreto, se ha decidido aumentar λ_{busq} en 1 búsqueda por minuto, cada 10 minutos. Es conveniente que el tiempo de incremento sea lo suficientemente holgado para permitir al sistema adaptarse convenientemente a la nueva carga ofrecida.

En la Figura 3-7 se muestran los tiempos de respuesta obtenidos al aplicar la simulación al servicio de búsqueda. Los resultados son bastante claros: a partir de aproximadamente una tasa de 21 búsquedas por minuto los tiempos de respuesta empiezan a incrementarse comprometidamente. En la Figura 3-8 se muestra el número de páginas de error retornadas por el sistema de búsqueda como respuesta a las peticiones. A través de este gráfico se confirma que el punto de saturación se sitúa en 21 búsquedas por minuto, a partir de donde el número de errores es significativo.

Este análisis ha sido realizado en base a los tiempos de respuesta obtenidos para las búsquedas. Como complemento se han efectuado los correspondientes análisis sobre los tiempos de respuesta derivados de la navegación por categorías y la visita a documentos. Los resultados y gráficos obtenidos son análogos a los del proceso de búsqueda, no aportando ningún aspecto novedoso y confirmando que el punto de saturación se sitúa en 21 búsquedas por minuto (y los correspondientes valores de categorías y documentos visitados).

3.2.3.2. Comparación de servicios de búsqueda

Sin embargo, uno de los aspectos fundamentales en la evaluación de sistemas de recuperación de información es la determinación de los tiempos de respuesta ofrecidos por el motor de búsqueda. En concreto, por medio de la comparación de distintos servicios de búsquedas se permite discernir y determinar las mejoras reales obtenidas. La comparación

de servicios de búsqueda puede venir determinada por el cambio de una parte del algoritmo de búsqueda empleado, o directamente por la comparación de dos servicios de búsqueda totalmente diferentes.

La mayoría de las evaluaciones de tiempos de respuesta realizadas en la literatura se basan en el sistema en reposo (o sea, con carga nula), sin embargo, es trascendental realizar la comparativa en diferentes niveles de carga del sistema para conseguir un estudio completo y representativo en la totalidad de situaciones posibles.

A continuación, se describe un caso práctico en el cual USim ha demostrado su valía a la hora de tomar medidas cuantitativas del buen funcionamiento de un motor de búsqueda. En este caso concreto, se debe determinar la supuesta mejora obtenida al utilizar un algoritmo de búsqueda optimizado. Para la realización de este experimento el motor de búsqueda se encontraba instalado en un equipo Ultra Sparc 1 con 128 MB de memoria RAM y un procesador a 167 MHz.

El objetivo del experimento consiste en comprobar el rendimiento ofrecido por un algoritmo de búsqueda estándar y una versión mejorada, en situación de carga elevada. Anteriormente, se ha comprobado la mejora del rendimiento aportada por el segundo algoritmo bajo una carga nula, sobre el algoritmo de búsqueda normal. Para verificar el comportamiento en situaciones de carga alta, se han realizado dos simulaciones de carga contra el motor de búsqueda (una por cada tipo de algoritmo a evaluar), cada una de 3 horas de duración, en donde se ha sometido al sistema a una media de 10 búsquedas por minuto.

Como resultado de ambas simulaciones se obtuvieron más de 1.700 datos sobre los tiempos empleados por ambos algoritmos en responder a cadenas de búsquedas reales. En base a los datos obtenidos de la simulación, se debe realizar un análisis de la varianza (ANOVA) que permita determinar los factores que influyen en la variabilidad del tiempo de respuesta. A priori, parece claro que el número de resultados recuperados en la consulta se trata de un factor influyente, y se debe establecer si también se puede considerar el tipo de algoritmo utilizado como factor de peso.

Para evitar los ya conocidos problemas de dependencia existentes en este tipo de series de datos, se ha aleatorizado la muestra permaneciendo aproximadamente un 10% del total de datos originales. Las consultas se agruparon en función de los resultados obtenidos, divididos en los siguientes grupos: consultas de 0 resultados, menos de 50, 150, 300, 500, 750, 1.000, 1.500, 2.000, 3.000, 3.500, 5.000, 7.500 y 10.000 resultados.

A continuación se realizó el análisis de la varianza (ANOVA) de estos datos, considerando los factores: el algoritmo de búsqueda utilizado y el número de resultados, como posiblemente influyentes, y tomando como variable respuesta el tiempo de realización de la búsqueda.

El resultado obtenido es interesante y sorprendente. Por una parte se debe rechazar que el tiempo de respuesta no depende del número de resultados obtenidos en esa búsqueda, como era previsible. En cambio, se debe aceptar que los dos algoritmos son equivalentes ya que no existen diferencias estadísticamente apreciables entre ambos. Sin embargo, la parte más interesantes se deriva del hecho de que existe interacción entre los dos factores, o lo que es lo mismo, ante un determinado tipo de consultas un algoritmo responde mejor

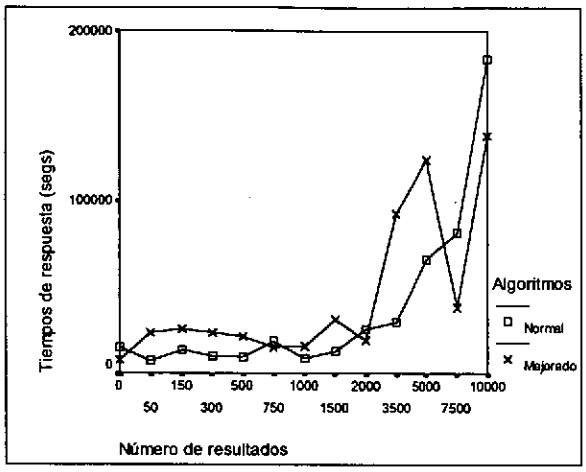

Figura 3-9: Comparativa de los tiempos de respuesta de los algoritmos de búsqueda normal y mejorado

que el otro. En la Figura 3-9 se observa que para niveles bajos de números de resultados se comportan de manera equiparable, no obstante cuando la búsqueda produce un número de resultados elevado el rendimiento ofrecido por el algoritmo mejorado empeora ligeramente.

En resumen, utilizando USim se ha podido estudiar y comparar el comportamiento de dos algoritmos de búsqueda en un motor de búsqueda simulando una situación de carga elevada. Esto simplemente constituye una parte del estudio, ya que el estudio completo se basa en el análisis sobre distintos niveles de carga (empezando por una carga inicial nula hasta la saturación del sistema). Como se ha comentado anteriormente, la versión mejorada del algoritmo ofrecía un mejor rendimiento en situaciones de carga nula, aunque esta mejora bajo una situación de carga elevada no era mantenida. Esto demuestra la importancia y la validez de este tipo de herramientas de simulación y evaluación para estimar el rendimiento en distintas situaciones.

3.3. Conclusiones

En el presente capítulo se han descrito a nivel general los principales aspectos a evaluar en un sistema de recuperación de información, con especial énfasis en la evaluación del rendimiento del sistema, frente a la evaluación del rendimiento de la recuperación.

En concreto, la evaluación del rendimiento de un sistema de búsqueda en el Web debe estudiarse considerando diferentes niveles de carga de usuarios, con el objetivo de una correcta evaluación de la velocidad de respuesta del sistema en todas las situaciones posibles.

En consecuencia, el diseño y desarrollo de una herramienta de simulación de los usuarios de un sistema de búsqueda en el Web se convierte en fundamental para la correcta evaluación del rendimiento de dichos sistemas.

Las conclusiones obtenidas en el capítulo anterior establecen la base matemática del proceso de simulación. En el capítulo previo se demostró que las búsquedas realizadas, las

categorías visitadas y los documentos consultados por los usuarios se ajustan a un proceso de Poisson (equivalente a una distribución Exponencial). Además se estableció una relación lineal entre las tres variables, que permite asociar variaciones en sus valores.

En base a esto, la principal aportación de este capítulo se centra en USim, una herramienta de simulación de los usuarios de un sistema de búsqueda en el Web genérico, al ser válido para robots, directorios y metabuscadores.

Las utilidades de USim se centran principalmente en dos puntos. En primer lugar, para la determinación del punto de saturación de cualquier sistema de recuperación de información en el Web, lo que es crucial para dimensionar correctamente el hardware y el software del sistema, así como para evitar situaciones de carga extremas en donde el rendimiento del sistema se ve drásticamente afectado.

Si bien, su principal aplicación radica en la medida y comparación de diferentes sistemas o algoritmos de búsqueda, en base a los tiempos de respuesta (y otros factores, como el número de resultados) obtenidos bajo diferentes situaciones de carga. Tal y como se ha demostrado, una comparación en una situación ideal de carga nula puede conducir a conclusiones erróneas, por lo que es necesario estudiar el comportamiento de los sistemas analizados ante diferentes situaciones de carga para un estudio completo y exhaustivo.

4. ARQUITECTURA DE DATOS PROPUESTA PARA UN DIRECTORIO WEB

4.1. Introducción

En este capítulo se describe la arquitectura de datos propuesta para un sistema de recuperación de información en Internet organizado en base a una jerarquía de categorías, esto es, un directorio Web.

La descripción del modelo se centra especialmente en las estructuras de datos y la organización asociada, con el objetivo primordial de mejorar el rendimiento ofrecido a los usuarios, fundamentalmente en los aspectos de búsqueda, con especial hincapié en las búsquedas restringidas a una zona de la ontología, característica peculiar de este tipo de sistemas de recuperación de información ([Cacheda, 01e], [Cacheda, 02a], [Cacheda, 02b]).

La arquitectura básica de un directorio Web está ampliamente basada en las estructura central de un robot, sin embargo es necesario tener en cuenta las necesidades especiales y características de estos sistemas. No es frecuente encontrar en la literatura la descripción de la arquitectura o las estructuras de datos empleadas por los directorios Web, no obstante, en la sección 1.5.2.2 se muestra una visión general de los componentes estructurales de estos sistemas de búsqueda, proponiendo las soluciones básicas y similares a un robot de búsqueda en Internet.

Sin embargo, las características de robots y directorios son sensiblemente diferentes, y se hacen necesarias estructuras específicas para cada sistema con el objetivo de priorizar las necesidades de cada sistema. A continuación se describen las principales diferencias estructurales entre ambos sistemas:

- En ambos sistemas se requiere un fichero invertido para la búsqueda eficiente de los documentos asociados con cada palabra (estructura típica en los sistemas de recuperación de información actuales [Baeza-Yates, 99b]). Sin embargo, en el caso de un robot este componente constituye el núcleo básico y único (junto con el vocabulario) sobre el que se asienta el sistema. En cambio, en un directorio esta estructura de datos se centra en los procesos de búsqueda, sin relación alguna con la estructura de la ontología propiamente dicha.
- En el caso de los robots, un aspecto fundamental durante la elaboración de índice invertido lo constituyen los tiempos de respuesta obtenidos y el espacio de almacenamiento, debido al gran volumen de información a gestionar. Por el contrario, en el caso de los directorios Web en su arquitectura se primará especialmente el tiempo de respuesta, ya que la cantidad de páginas Web tratadas en estos sistemas es sensiblemente menor. Sin embargo, los tiempos de respuesta no se limitan exclusivamente a búsquedas, sino también a búsquedas restringidas y navegación a través de la jerarquía.
- Las estructuras de datos de los robots se centran en dos conceptos: palabras clave y documentos, y básicamente deben representar la relación existente entre ambos. Para esto es necesario tener en cuenta que el origen de la búsqueda estará siempre marcado por una o más palabras clave, a partir de donde se localizarán los documentos asociados, para lo cual la estructura de fichero invertido es la más adecuada. En cambio, en el caso de un directorio Web existen tres conceptos involucrados en el sistema de recuperación de información: palabras clave, documentos y categorías, con los correspondientes pares de relaciones existentes (palabras clave-documentos, palabras clave-categorías y categorías-documentos). Además, los accesos a estas estructuras no son independientes, ya que para una búsqueda restringida a una zona de la jerarquía es necesario combinar las estructuras de palabras clave-documentos y categorías-documentos de manera eficiente.
- En los directorios Web las categorías conforman el punto básico de navegación y se requiere una estructura adecuada para tal efecto, ya que la ontología se representa por medio de grafo dirigido acíclico, estructura compleja aunque semejante al concepto tradicional de árbol. Por el contrario, un robot carece totalmente de esta estructura, centrándose única y exclusivamente en la localización de información en un gran volumen de datos.

4.2. Arquitectura básica

Dentro de esta sección se describen las estructuras de datos específicas para los componentes de un directorio Web, estableciendo un modelo básico que permitirá constituir una base y una nomenclatura para el modelo propuesto, así como la localización de los aspectos clave de diseño de este sistema de recuperación de información.

En primer lugar, la característica diferenciadora de un directorio la constituye la ontología sobre la que se clasifican los documentos (ver sección 1.5.2.2). Típicamente, cada categoría constará de una serie de información básica (nombre, una breve descripción), e información sobre las categorías hijas que posee y a su vez los diferentes padres de los que hereda su posición en la ontología. Además, cada categoría podrá disponer de una serie de documentos directamente ligados a la misma.

Cada categoría debe estar siempre registrada por medio de un identificador de categoría (identificador único para cada categoría). De esta manera, es necesario disponer de un fichero de categorías en donde se almacena la información básica de cada categoría, e independientemente, utilizando únicamente los identificadores de categorías, una estructura que represente el grafo dirigido acíclico que constituye la ontología.

Típicamente, los accesos al grafo se realizan sobre una categoría concreta para obtener los descendientes directos asociados. Por lo tanto, la estructura adecuada consiste en la construcción de una estructura de grafo en base a punteros, doblemente enlazados, que asocien categorías padre con sus respectivos nodos hijo. En caso necesario, se puede considerar la existencia de varios padres, y entre ellos un padre "*por defecto*", que será el empleado en los recorridos hacia el nodo raíz en cada caso.

Las características de espacio requeridas por esta estructura de grafo son mínimas, ya que el número de categorías disponibles en un directorio Web suele ser bastante limitado (típicamente, varias decenas de miles). Por otra parte, los tiempos de respuesta son adecuados si los accesos se realizan siempre desde el nodo raíz, por lo que puede ser conveniente disponer de un puntero directamente a dicha estructura desde el fichero de categorías (ver Figura 4-1).

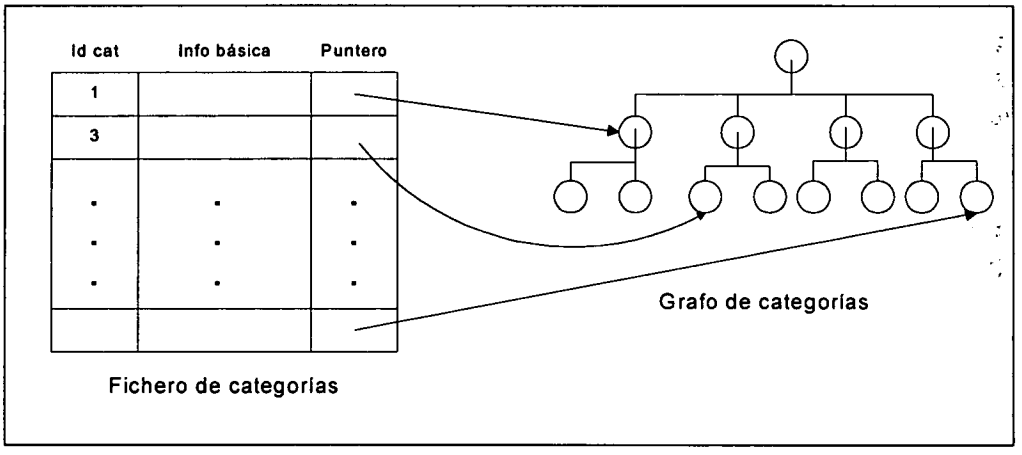

Figura 4-1: Estructura de datos correspondiente a las categorías de un directorio Web

En segundo lugar, se analizan las estructuras de datos referentes a las palabras o vocablos indexados por el sistema. Si se tiene en cuenta únicamente el proceso de indexación referente a los documentos, parece claro que la estructura más adecuada consistirá en la utilización de un fichero invertido (ver sección 1.4.2).

La estructura de fichero invertido se divide en dos partes: el vocabulario y la lista invertida. Por una parte se almacenará el vocabulario, pudiendo emplearse cualquiera de las técnicas expuestas previamente: arrays ordenados, árboles B, estructuras PAT, etc. A este respecto no existe ninguna limitación que establezca mejoras de una técnica frente al resto, a parte de las específicas de cada método ya analizadas en la sección 1.4.2. El almacenamiento de la lista invertida se deberá realizar de la manera convencional, utilizando los identificadores de documento que se indicarán a continuación, para mantener los punteros hacia las ocurrencias de los vocablos.

El criterio de ordenación de la lista invertida constituye un aspecto importante, como se especifica en la sección 1.4.2. Una ordenación según el identificador de documento facilita la combinación de varias listas; mientras que una ordenación según un criterio de importancia de los documentos convierte en triviales las búsquedas simples [Brin, 98]. La elección de una ordenación u otra (o incluso alguna modalidad híbrida) constituye una decisión de diseño del sistema, si bien, el modelo más sencillo y efectivo para consultas típicas consiste en la ordenación por identificador de documento.

En cualquier caso, este aspecto constituye un punto fundamental dentro de las estructuras de datos de un directorio Web respecto a las posibles repercusiones en el rendimiento. Por lo tanto, es conveniente que la arquitectura propuesta no presuma ningún tipo de ordenación, sino que sea flexible en este punto.

Sin embargo, dentro de un directorio Web suele ser común que las categorías presenten ciertas palabras clave o descripciones bajo las que son indexadas, para posteriormente poder aplicar sobre éstas el proceso de búsqueda. En este caso, obviamente, la estructura más idónea es un fichero invertido, si bien, existen dos posibilidades: almacenar conjuntamente la lista de documentos y categorías invertidas, o utilizar estructuras separadas.

El almacenamiento conjunto de ambas listas, en un único fichero invertido presenta como principal ventaja que es necesario un único acceso para la obtención de ambas listas. Sin embargo, es necesario articular mecanismos que permitan diferenciar ambos tipos de identificadores (por ejemplo, asignando rangos diferentes). Por otra parte, ambas listas presentan un comportamiento diferente, ya que las listas de documentos presentan cierto dinamismo por la incorporación de diferentes documentos; frente al caso de las categorías cuyo comportamiento es más estático, salvo algunas actualizaciones esporádicas.

Por otra parte, el almacenamiento separado implica la necesidad de dos accesos a dos estructuras diferentes para la obtención de ambas listas. Este hecho aumentará ligeramente el tiempo de respuesta ante una búsqueda, el volumen de información leída será exactamente equivalente y simplemente será necesario realizar un nuevo posicionamiento de la cabeza lectora. De esta manera, se mantienen separados dos objetos que conceptualmente son diferentes (con lo cual no es necesario emplear ningún mecanismo para diferenciarlos), y se garantiza que la búsqueda de categorías es independiente de la búsqueda de documentos. Esta independencia permite la realización de búsquedas en categorías sin necesidad de leer la lista de documentos asociada a las palabras clave y viceversa, con la consiguiente mejora en el tiempo de respuesta.

En la Figura 4-2 se muestran las estructuras de datos de un directorio Web asociadas con las palabras indexadas y su relación con documentos y categorías. La técnica de fichero invertido se basa en un array ordenado y se ha adoptado la solución que separa ambas listas invertidas.

Finalmente, el tercer elemento básico de un directorio Web lo constituyen los propios documentos o páginas Web. Cada documento se identifica por medio de un identificador de documento (análogo al identificador de categorías) que registra a cada documento unívocamente. Al mismo tiempo, de cada documento se archiva cierta información básica que generalmente estará constituida por: la URL del documento Web, el título de dicho documento y una breve descripción de sus contenidos. Típicamente, se define un fichero

de documentos que contiene la información referente a todos los documentos, que se representa utilizando un array ordenado por identificador de documento (los accesos a esta información se realizarán siempre a partir de un identificador) ya que de esta manera se permiten las búsquedas binarias en la estructura (ver Figura 4-3).

Asociado con los documentos existen dos estructuras que los relacionan con las palabras indexadas y con las categorías asociadas. Ya se ha comentado previamente que un fichero invertido es la estructura más adecuada para el almacenamiento de los vocablos asociados a cada documento, con la consiguiente lista invertida de documentos (ver Figura 4-2). Sin embargo, la estructura de datos asociada a los documentos y sus categorías asociadas, o viceversa requiere una consideración especial, ya que por una parte esta estructura será accedida en el proceso de navegación y en el proceso de búsqueda restringida a una zona del grafo.

En el proceso de navegación, a partir de una categoría se obtendrán los diferentes nodos hijo asociados (utilizando para ello la estructura que representa el grafo de categorías, según el modelo de la Figura 4-3) y también es necesario obtener los diferentes documentos asociados. Obviamente, el formato más adecuado consiste en almacenar una lista de documentos invertida asociada con cada categoría, que representará a los documentos ligados con dicha categoría. Como en todos los ficheros invertidos, el criterio de ordenación de las listas invertidas es fundamental para un correcto rendimiento. Por una parte, una ordenación de la lista según los identificadores de documentos garantiza una mejor combinación con otras listas, mientras que una ordenación según la importancia de los documentos facilita el proceso de navegación. Teniendo en cuenta únicamente el proceso de navegación, la alternativa que ofrece un mejor rendimiento es aquella que mantiene ordenados en la lista los documentos según su importancia, ya que esto elimina la

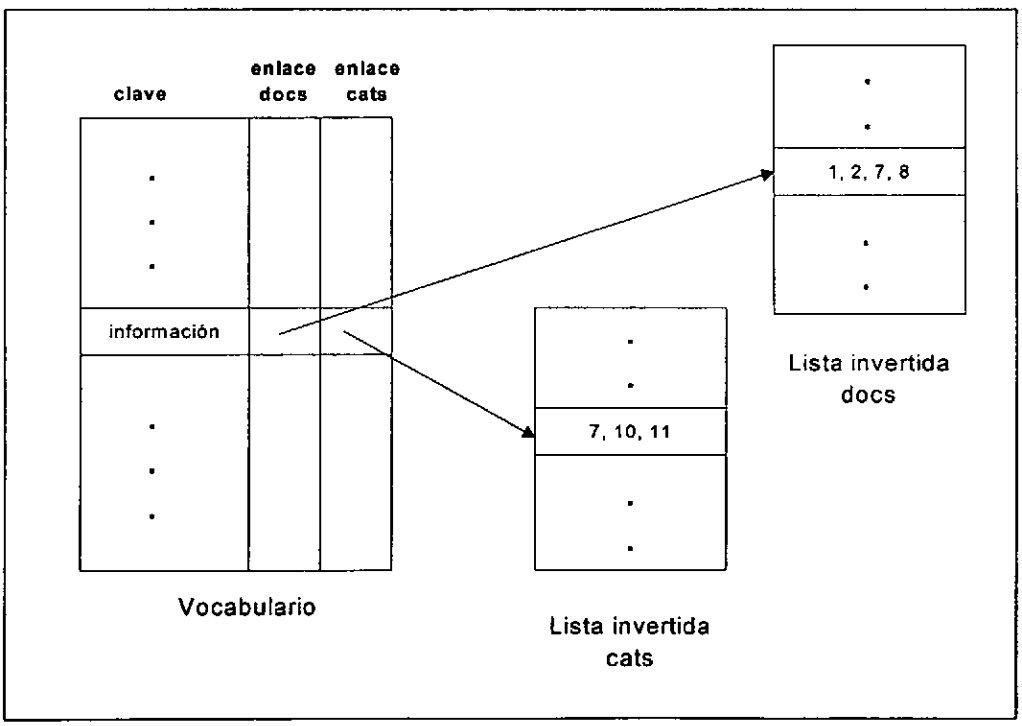

Figura 4-2: Estructura de datos correspondiente a las palabras clave y sus relaciones con categorías y documentos de un directorio Web

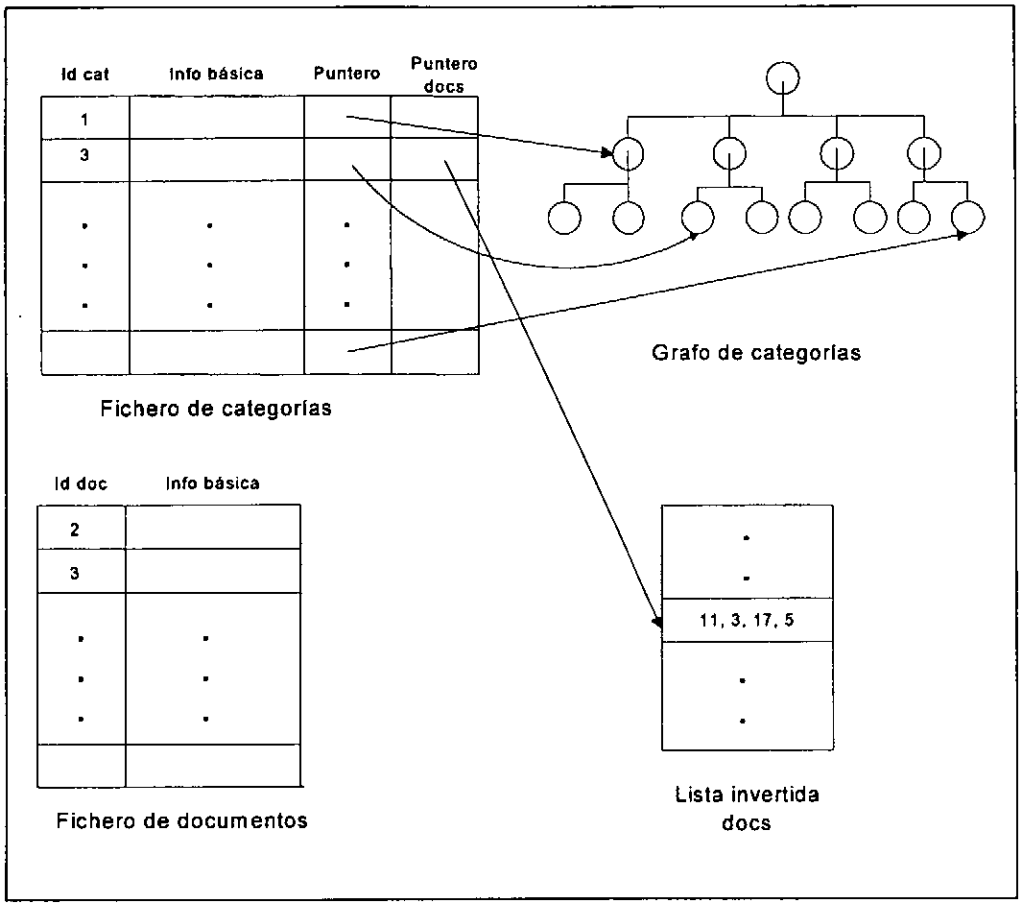

Figura 4-3: Estructura de datos correspondiente a los documentos de un directorio Web

etapa de ordenación en este proceso.

El proceso de búsqueda restringido a una zona del grafo, o lo que es lo mismo, a una categoría y todas sus categorías descendientes con los correspondientes documentos asociados, requiere un acceso más complejo a esa información. Por una parte, se realiza la búsqueda accediendo al fichero invertido de palabras y combinando las listas de documentos de la manera usual, de lo que se obtiene como resultado una lista (probablemente ordenada por importancia de los documentos). El paso clave consiste en determinar qué documentos de esa lista pertenecen a la zona del grafo especificada. Para el filtrado de esta lista se define dos alternativas.

La primera de ellas consiste en la obtención de la lista de documentos asociados a esa zona del grafo para posteriormente combinar la lista de resultados con la lista de documentos asociados a las categorías. La obtención de la lista de documentos asociados a las categorías es un proceso medianamente costoso. En primer lugar, partiendo del nodo base que delimita la zona de restricción se debe recorrer el grafo, obteniendo para cada nodo la lista de documentos asociados (accediendo al fichero invertido). Y en segundo lugar, una vez obtenidas todas las listas invertidas es necesario realizar una combinación de todas ellas para formar la lista de documentos asociados a las categorías.

En este punto, merece especial atención el criterio de ordenación dentro de las listas invertidas, ya que en este caso es prioritario una ordenación por identificador de

documento, para una rápida combinación de las listas (se consigue una combinación eficiente de dos listas, si previamente han sido ordenadas [Joyanes, 98]). Una vez que la lista se ha construido simplemente es necesario realizar una combinación con la lista de resultados. A su vez, este proceso depende del criterio de ordenación existente en la lista de resultados, ya que normalmente los resultados se obtienen ordenados según la relevancia de la búsqueda. En cualquier caso, este proceso de combinación o intersección puede ser muy costoso si ambas listas son de un tamaño considerable, lo cual puede suceder en un porcentaje importante de los casos.

La segunda de las alternativas consiste en la obtención de la lista de categorías de la zona de restricción (proceso más sencillo y menos costoso que la obtención de la lista de documentos), y realizar un chequeo secuencial de la lista de resultados comprobando si los documentos se encuentran en al menos uno de los nodos de la lista de categorías. La obtención de la lista de categorías (ordenada según el identificador de categorías) simplemente requiere un recorrido por el grafo, almacenando los identificadores de categorías en una lista ordenada. El siguiente paso, el recorrido secuencial de la lista de resultados, puede ser especialmente costoso si la lista de resultados es extensa. Además, es necesaria una estructura auxiliar que indique las categorías asociadas a cada documento, lo que podría estar constituido por un fichero invertido en donde a partir del fichero de documentos se obtuviesen las categorías en donde se localiza cada documento (ver Figura 4-4).

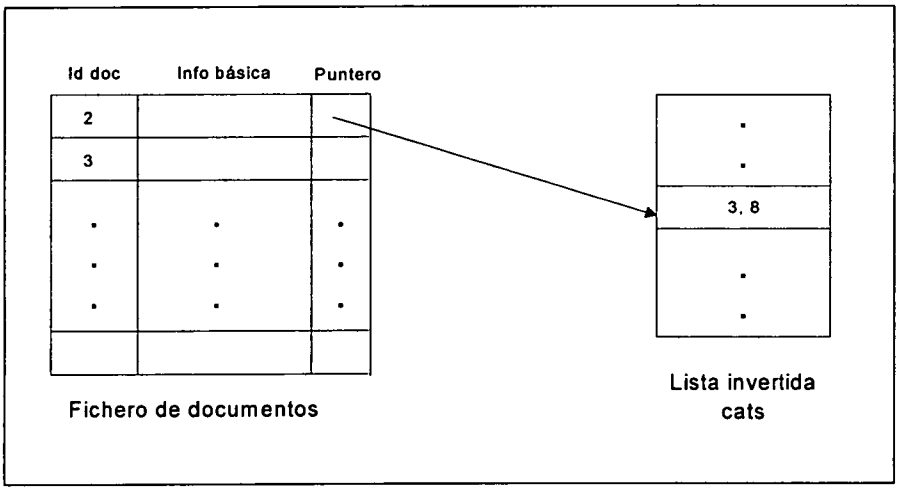

Figura 4-4: Estructura de datos adicional para lista invertida de categorías y documentos

El primero de los métodos expuestos presenta la principal ventaja de que no se requiere ningún tipo de estructura de datos adicional para resolver las búsquedas restringidas a una zona del grafo de categorías. En cambio, el principal inconveniente se centra en el tiempo de respuesta ante este tipo de consultas, ya que hay varias partes del proceso que requieren un tiempo de procesamiento elevado:

- La obtención de la lista de documentos de las categorías especificadas requiere por una parte el recorrido de una zona del grafo (proceso que se considera despreciable) y la recuperación todas y cada una de las listas invertidas asociadas a esas categorías. Este proceso de lectura del fichero invertido es muy costoso ya que implica el posicionamiento y la lectura de los bloques asociados con dichas listas.

Además, el tamaño de las listas invertidas variará del orden de varias decenas hasta miles de documentos.

- El siguiente paso consiste en la combinación de todas las listas recuperadas. El número de listas recuperadas es variable, dependiendo de la profundidad de la categoría restringida, pero puede representar un porcentaje importante del total del grafo (limitando la consulta a una categoría de primer nivel, supone abarcar entre un 5% y un 10% del total del grafo). Sin embargo, el principal problema se deriva de la ordenación de las listas, ya que normalmente se encontrarán ordenadas según la importancia de los documentos para agilizar el proceso de navegación, por lo que el proceso de combinación implica la ordenación de las listas y la posterior combinación.
- El paso final consiste en la combinación de ambas listas (de resultados y de documentos de las categorías) para la realización de la intersección. Este proceso, suponiendo que ambas listas están ordenadas por identificador de documento, será especialmente costoso en el caso de que ambas listas tengan un tamaño considerable.

En consecuencia, este método se adapta adecuadamente a aquellas consultas en las que la zona del grafo restringida es reducida (restricciones a categorías de los niveles inferiores), ya que el número de categorías (y por lo tanto de listas de documentos a combinar) involucradas es menor. Sin embargo, para aquellas consultas restringidas a categorías de los primeros niveles el coste asociado puede ser demasiado elevado, en especial en el caso de consultas con un número elevado de resultados. Además, este método entra en conflicto con determinadas estructuras de datos en el proceso de navegación, por lo que la obtención de una solución de compromiso es compleja.

Respecto a la segunda alternativa, la principal desventaja que presenta es la necesidad de una estructura de datos redundante con el fichero invertido de categorías y documentos, adaptado para agilizar la localización de las categorías asociadas a cada documento. Sin embargo, es necesario tener en cuenta que en los directorios Web el número de documentos está bastante limitado, y que suelen estar asociados a un número reducido de categorías (normalmente una o dos), ya que cada documento Web suele tener asociada una temática clara y la posibilidad de ubicar una categoría en diferentes zonas del grafo evita la necesidad de duplicar los documentos.

Asimismo, esta alternativa requiere una lectura secuencial de los resultados (por lo tanto el orden en el que se obtienen no es importante) y un posterior acceso al índice que asocia cada documento con sus categorías. Esto hace que este método sea eficiente en aquellos casos en los que el número de documentos en los resultados es reducido, ya que el número de accesos al índice supondrá una carga aceptable. En cambio, cuando el número de resultados es elevado el tiempo de respuesta proporcionado se verá incrementado, ya que el número de lecturas del índice será mayor, con el consiguiente retardo.

En la Figura 4-5 se muestra el conjunto de las estructuras de datos que conforman el modelo de datos de un directorio Web básico.

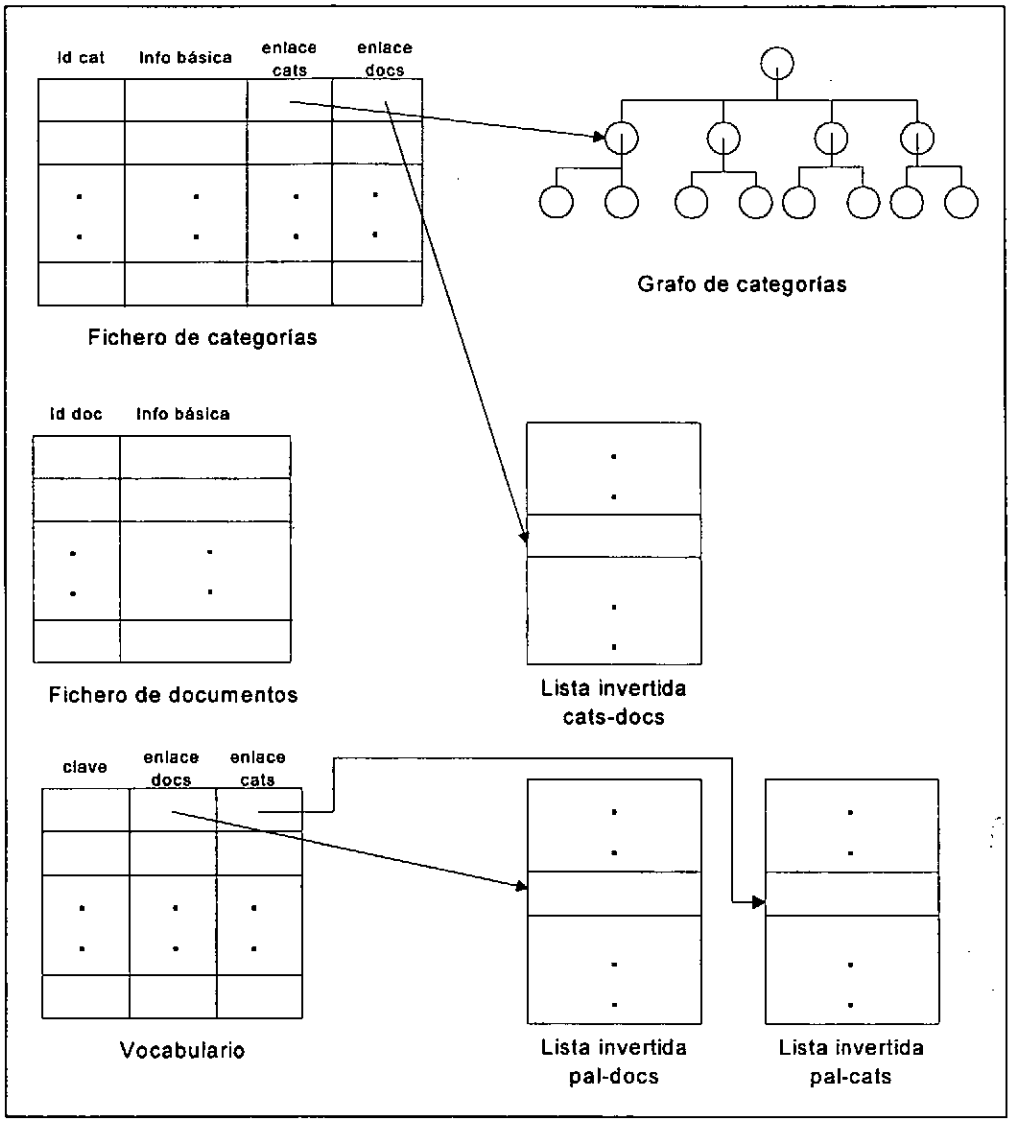

Figura 4-5: Estructura de datos del modelo básico para un directorio Web

En resumen, según un modelo de arquitectura basada en la técnica de fichero invertido se plantea un problema para la resolución eficiente de las búsquedas restringidas a una zona del grafo. En concreto, se han planteado dos alternativas. La primera de ellas únicamente permite resolver eficientemente aquellas consultas restringidas a una zona reducida del grafo, y además entra en conflicto en aspectos de rendimiento con el proceso de navegación. Por otra parte, la segunda alternativa resuelve adecuadamente aquellas consultas restringidas a una zona del grafo que recuperan un número reducido de resultados.

Sin embargo, una gran parte de las consultas se caracterizan por obtener un gran número de resultados (ver sección 2.3) y al mismo tiempo, los usuarios al restringir las búsquedas a una categoría suele emplear categorías de las posiciones superiores del grafo (ver sección 2.4.1). En consecuencia, es necesario la elaboración de una nueva arquitectura de datos adecuada que permita resolver eficientemente este tipo de consultas, aspecto que es tratado en las siguientes secciones.

4.3. Arquitectura híbrida con información total

El desarrollo de la arquitectura propuesta para las estructuras de datos adecuadas para un directorio Web se centrará especialmente en los componentes que afectan a las búsquedas restringidas a una zona del grafo de categorías. Para el caso del resto de componentes, su definición y constitución se ha tratado ampliamente en la sección previa, destacando la técnica del fichero invertido como la más adecuada para la construcción de esos índices.

En cambio, las búsquedas restringidas a una categoría presentan una problemática asociada que no puede ser solucionada fácilmente empleando únicamente la técnica de fichero invertido.

4.3.1. Descripción

El modelo de estructuras de datos propuesto se basa en la segunda alternativa descrita para la resolución de una búsqueda restringida. Esta solución se basa en la obtención en primer lugar de la lista de categorías sobre la que se restringe la búsqueda, para a continuación realizar un recorrido secuencial de la lista de resultados (por lo que no es necesario que estén sujetos a ningún tipo de ordenación previo), comprobando para cada documento si alguna de las categorías asociadas se encuentra en la lista de categorías restringidas.

Para un funcionamiento competente de esta solución se hace necesario un índice invertido que permita recuperar eficientemente las categorías asociadas a cada documento. Sin embargo, el principal problema de eficiencia de esta solución se localiza en el recorrido secuencial de los resultados para determinar si pertenece o no a alguna de las categorías restringidas.

En consecuencia, la idea sobre la que se asienta el modelo de datos propuesto se basa en la aplicación de un filtro inexacto sobre los resultados de la búsqueda, de tal forma que se puedan eliminar una gran mayoría de los resultados no asociados a las categorías restringidas. De esta forma, para la realización de un filtrado exacto bastaría con examinar secuencialmente los documentos restantes, cuyo número se habrá visto sensiblemente reducido. Este concepto se basa en lo expuesto en la sección 2.4.1, en donde se indica que únicamente permanecen, de media, un 10% de los resultados, al ser restringida una consulta a una categoría.

Como se ha descrito en el capítulo 1, los ficheros de firmas son una técnica de filtrado inexacto que se basa en el acceso secuencial a los datos. El principal problema que plantean es su pobre rendimiento para volúmenes grandes de datos, sin embargo, este hecho no supone un inconveniente en este caso, ya que el filtrado secuencial se realizará únicamente sobre los resultados obtenidos de la búsqueda, nunca sobre la totalidad dè los documentos disponibles.

El punto clave de la arquitectura propuesta consiste en que cada documento deberá poseer una firma en la que se representarán todas y cada una de las categorías a las que pertenece, incluyendo tanto la categoría a la que se encuentra asociado, como todos los ancestros de la misma. No obstante, esto no implica la existencia de un fichero de firmas de todos los

documentos. Por el contrario, la incorporación de los ficheros de firmas en la estructura de datos permite crear un esquema mixto de fichero invertido y fichero de firmas.

El almacenamiento de las firmas de los documentos en un fichero de firmas independiente, o incluso como parte del fichero de documentos, no es posible por razones de eficiencia. En este sistema, cada consulta generará dinámicamente su propio fichero de firmas (de los documentos obtenidos como resultado), por lo que no sería competente el acceso posterior para la recuperación únicamente de la firma asociada a cada documento.

Por el contrario, en la estructura de datos híbrida propuesta, para un acceso eficiente, el fichero de firmas se encuentra embebido dentro del fichero invertido, de tal manera que cada lista invertida es a su vez un fichero de firmas (ver Figura 4-6). De esta manera se permite la realización del primer filtrado fácilmente.

Teniendo en cuenta esta estructura de datos, el esquema de funcionamiento para las búsquedas restringidas será el siguiente:

1. En primer lugar se obtiene la lista de resultados de la manera usual, ordenados de la manera más conveniente (típicamente por orden de importancia).

2. En segundo lugar, a partir del fichero de firmas asociado a la lista de resultados, se realiza el filtrado para la categoría sobre la que se restringe la búsqueda.

3. En tercer lugar, se realiza el filtrado definitivo sobre los resultados restantes, comprobando secuencialmente si las categorías asociadas coinciden con alguna de las categorías seleccionadas.

Es importante destacar que el tercer paso es opcional. Dependiendo de la calidad del filtrado inexacto puede no considerarse necesario la realización de un tercer paso que probablemente eliminase un número mínimo de resultados (en caso de eliminar alguno),

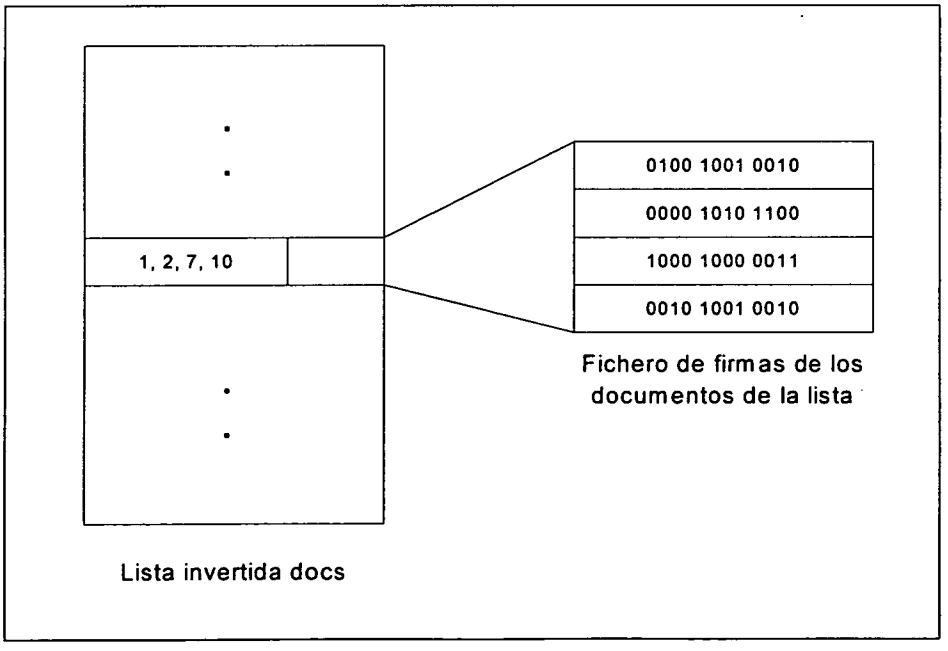

Figura 4-6: Estructura de datos híbrida de fichero invertido y fichero de firmas

bajo el peso del aumento del tiempo de respuesta.

Sin embargo, es necesario concretar algunos aspectos sobre la combinación de ficheros invertidos y ficheros de firmas, como son la validez de los ficheros de firmas y los códigos de superposición para la representación de la información de las categorías asociadas a un documento, así como el almacenamiento conjunto de ambas estructuras.

Respecto al último aspecto, en la Figura 4-6 se muestra un esquema de almacenamiento posible, en donde se almacenan de forma separada la lista de documentos invertida y el fichero de firmas asociado a esa lista. El principal problema que presenta esta estructura es que las operaciones realizadas sobre la lista invertida no repercuten sobre el fichero de firmas. De esta manera, al realizar operaciones de combinación entre varias listas (comúnmente uniones o intersecciones) que podrían eliminar determinados documentos (duplicados o por intersección), los ficheros de firmas asociados requerirán la repetición de las operaciones.

Por este motivo se ha considerado más adecuado la conversión de las firmas de los documentos en parte del identificador del documento. La idea bajo la que se sustenta este concepto consiste en que varios documentos estarán agrupados bajo las mismas categorías (porque pertenecen a las mismas categorías) por lo que constituyen elementos de un mismo conjunto y por lo tanto numerables, mientras que otro grupo de documentos pertenecerá a otras categorías, por lo que constituirán un conjunto diferente, igualmente numerable. De esta forma, un identificador de documento podría subdividirse en una firma de categorías y un identificador de documento local (relativo al conjunto de categorías), tal y como se representa en la Figura 4-7.

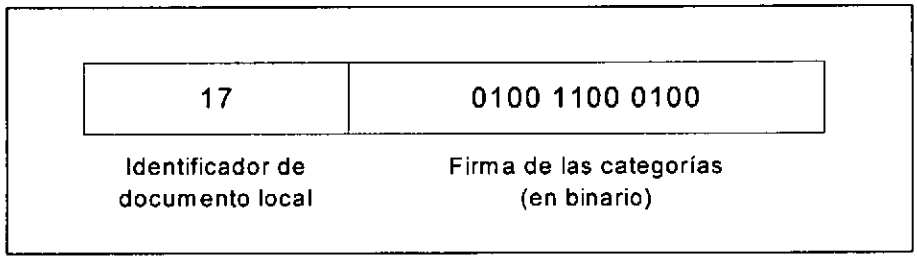

Figura 4-7: Identificador de documento compuesto

El espacio que se asigna a cada una de las partes es variable para cada diseño concreto, y deberá permanecer estable para todos los documentos de la colección que conforman el directorio Web. Por otra parte, la obtención de la firma de las categorías a las que pertenece el documento se discute en la siguiente sección, mientras que el identificador de documento local simplemente sigue la numeración consecutiva de los documentos asociados a esa firma.

De esta manera, se consigue representar a través del identificador de documento la firma de las categorías a las que se encuentra, directa o indirectamente, asociado. Por lo tanto, el esquema híbrido de fichero invertido y fichero de firmas, quedaría como se muestra en la Figura 4-8.

A través de este esquema, los documentos se encuentran identificados unívocamente y todas las operaciones de combinación realizadas sobre los identificadores de documentos se realizan simultáneamente sobre las firmas asociadas sin coste alguno. Así, al obtener la

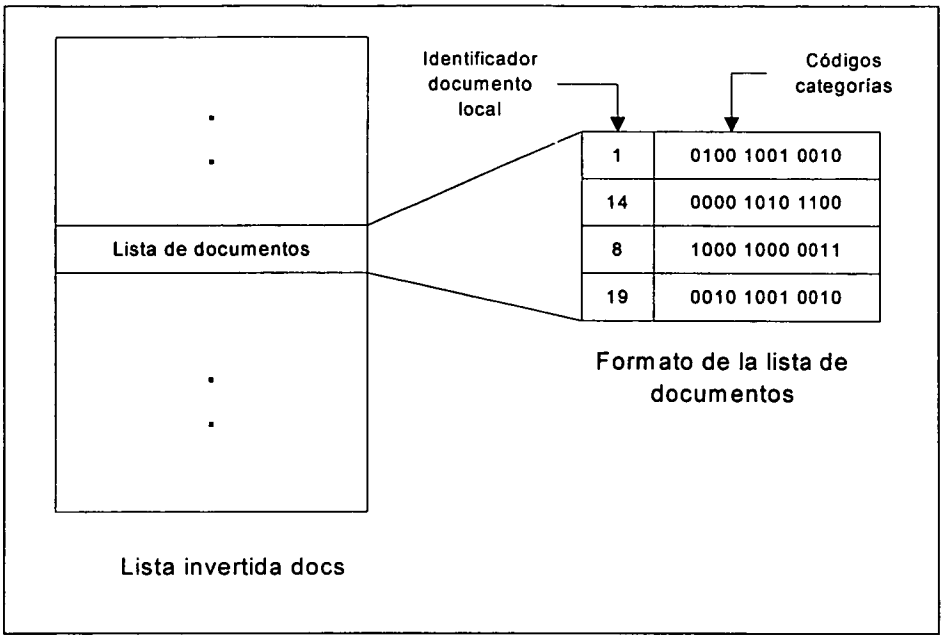

Figura 4-8: Estructura de datos híbrida de fichero invertido y fichero de firmas, con identificadores de documento compuestos

lista de identificadores de documentos final, indirectamente se dispone del fichero de firmas asociado a ese conjunto de resultados, por lo que basta con realizar un chequeo secuencial sobre la parte correspondiente a la firma de categorías que constituye el fichero de firmas.

El principal inconveniente derivado de este método es el aumento del tamaño utilizando por el identificador de documento, lo que implica un aumento de tamaño del índice invertido asociado. Sin embargo, como se verá a continuación es posible mantener controlado el tamaño de la firma de los documentos, por lo que el aumento de tamaño final no supondrá una carga excesiva para el conjunto del sistema de recuperación de información.

De esta forma, con el esquema de identificador de documento compuesto se permite la obtención dinámicamente de ficheros de firmas asociados a cualquier conjunto de resultados obtenidos a través de una consulta, sin importar el número ni el tipo de operaciones de combinación realizadas previamente. El hecho de obtener de manera sencilla y rápida el fichero de firmas asociado con un conjunto de resultados, agiliza la realización del filtrado inexacto, todo ello bajo un coste reducido de espacio de almacenamiento.

Los efectos que presenta el modelo híbrido sobre las operaciones de actualización son mínimas. Por una parte, la inserción de una categoría simplemente implica la asignación de una firma (siendo válida cualquier firma no asignada previamente) y el cálculo de su código genético, mediante la superposición de las firmas de sus ancestros. Las modificaciones y borrados de categorías no presentan cambios.

Por otra parte, la inserción de un documento requiere la asignación de un identificador de documento válido. Para ello, basta con calcular la firma de sus categorías asociadas y localizar el primer identificador local libre (dinámicamente o mediante el uso de una lista

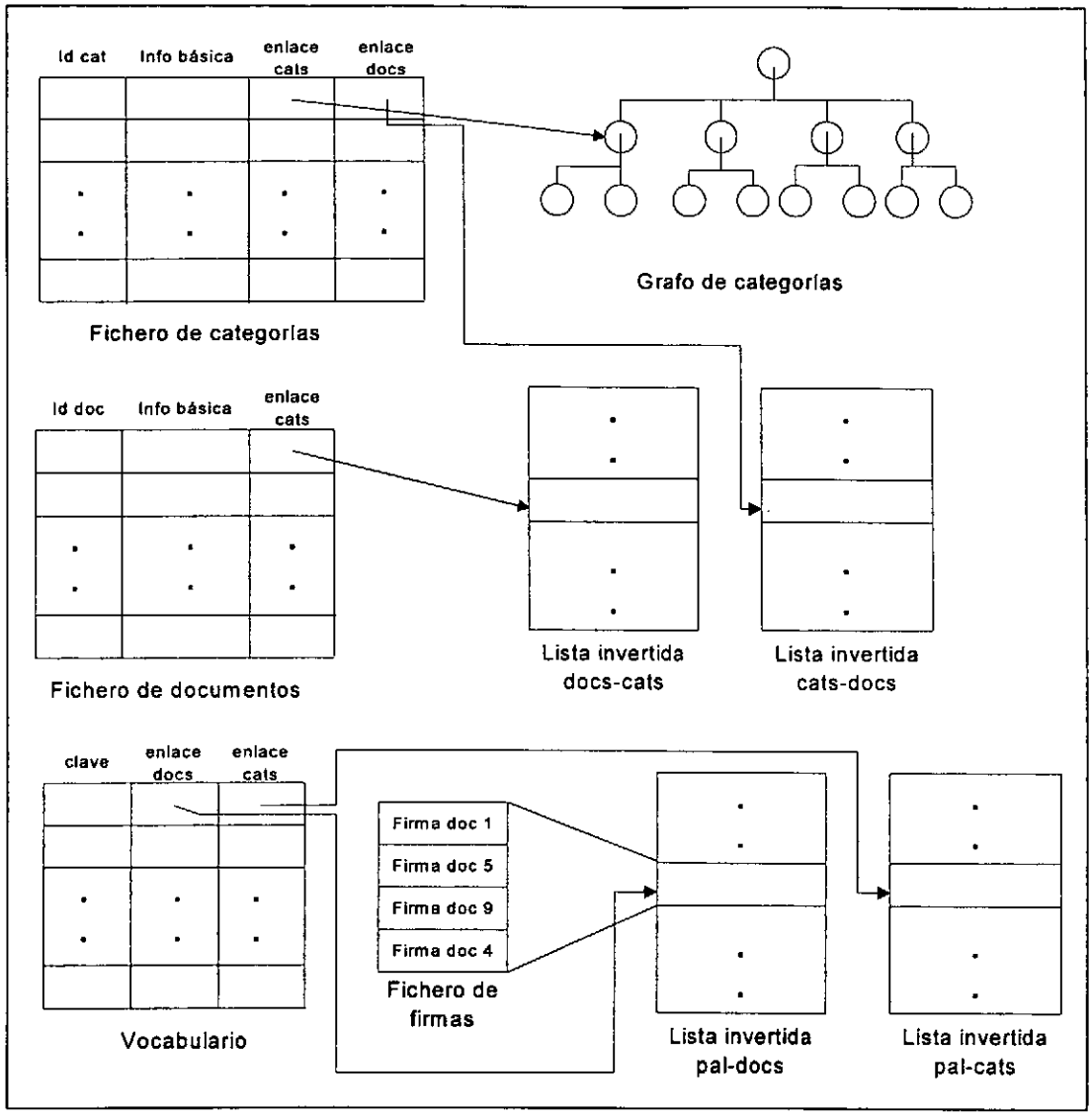

Figura 4-9: Estructura de datos de la arquitectura propuesta

de identificadores libres), mientras que el resto del proceso no sufre cambios. Respecto a la actualización, únicamente se requiere un tratamiento en el caso de producir alguna modificación (alta o baja) en las categorías asociadas a un documento, ya que esto modifica su firma. En este caso, la solución más simple y efectiva consiste en crear una réplica de este documento con el nuevo identificador correspondiente y eliminar el antiguo, principalmente porque este tipo de actualización es muy poco frecuente.

A modo de resumen, en la Figura 4-9 se muestran el conjunto de las estructuras de datos que conforman la arquitectura propuesta. El siguiente paso consiste en concretar y comprobar la validez de los ficheros de firmas basados en la técnica de los códigos de superposición para el tipo de información que se desea representar, aspecto que es examinado en detalle en la siguiente sección.

4.3.2. Ficheros de firmas y códigos de superposición

Un aspecto clave dentro del desarrollo de la arquitectura de datos híbrida de fichero invertido y fichero de firmas propuesta para la resolución eficiente de las consultas restringidas a una categorías, pasa por la utilización de la técnica de ficheros de firmas y los consiguientes códigos de superposición para la representación de las categorías asociadas a cada documento [Cacheda, 01e].

Los códigos de superposición constituyen la técnica de generación de firmas más empleada para el método de ficheros de firmas [Faloutsos, 84], y su utilidad a quedado patente en este tipo de método en diferentes implementaciones y con sucesivas mejoras, como se describe detalladamente en la sección 1.4.3.

Los códigos de superposición conllevan asociados una cierta probabilidad de falsos aciertos, F_d, definida como:

$$F_d = \frac{número\ falsos\ aciertos}{N - número\ aciertos\ reales}$$

Y a su vez, Stiassny en [Stiassny, 60] expone que el valor óptimo de w (el peso de cada firma) será aquel que permita igualar el número de "1"s y "0"s en la firma de cada bloque de datos, de lo cual se deduce la siguiente fórmula que aproxima el valor óptimo de w, junto con la probabilidad de falsos aciertos asociada:

$$w = \frac{b \ln 2}{D} \qquad (1)$$

$$F_d = 2^{-w} \qquad (2)$$

Sin embargo, estas fórmulas son aplicables siempre y cuando se cumplan dos restricciones básicas. La primera de ellas consiste en que el número de firmas por bloque, D, sea constante, ya que de esta manera se garantiza que el número de firmas a superponer es constante por lo que se puede obtener una estimación el número resultante de "1"s que tendrá la firma de bloque. Y la segunda restricción consiste en garantizar que el número de bits de las firmas, b, es mucho mayor que el peso de la firma, w [Kitagawa, 97]. En realidad, los valores típicos de b son centenares de bits, mientras que w toma normalmente valores inferiores a diez "1"s por firma.

Como se ha comentado en la sección anterior, por medio de los ficheros de firmas se pretende representar todas las categorías a las que está asociado un documento. En la Figura 4-10 se expone un grafo dirigido acíclico simplificado de un directorio Web. En dicho gráfico se puede observar que el documento 1 está asociado a las categorías 5 y 3, e indirectamente a todos sus ancestros. Por lo tanto el documento 1 pertenece, directa o indirectamente a las categorías: 5, 3, 2 y 1. De tal forma que si la búsqueda es restringida a alguna de esas categorías el documento 1 se encontraría dentro de los resultados posibles.

La técnica de ficheros de firmas aplicada al grafo dirigido acíclico de un directorio Web consiste en asociar una firma diferente para cada categoría (cada nodo del grafo) y cada documento generará la firma de bloque como la superposición de las firmas de las

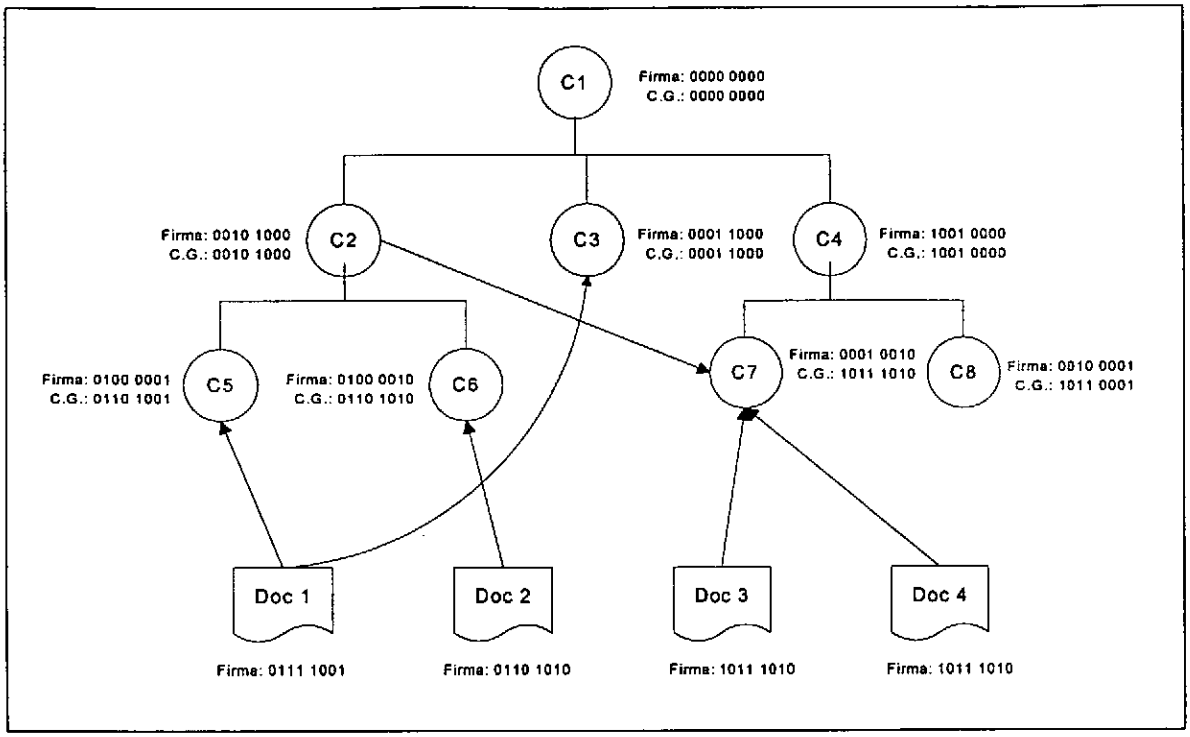

Figura 4-10: Códigos de superposición aplicados a un grafo dirigido acíclico

categorías a las que se encuentra asociado directa o indirectamente. Por ejemplo, considerando que las firmas de las categorías 5, 3, 2 y 1 son, respectivamente (se asume que $b=8$ y $w=2$), 0100 0001, 0001 1000, 0010 1000, 0000 0000 (la categoría raíz no requiere firma ya que todos los documentos indirectamente están asociados a esta categoría), la firma del documento 1 sería: 0111 1001. En la Figura 4-10 se muestran las firmas de cada categoría y las firmas de cada documento.

En el proceso de búsqueda restringido a una zona del grafo se debe indicar la categoría que marca el inicio o raíz de dicha zona. Asociado a cada categoría se encuentra lo que se ha denominado *código genético* (representado como C.G. en la Figura 4-10), consistente en un código resultante de la superposición de los códigos de dicha categoría y todos sus ancestros. De esta manera se consigue maximizar la cantidad de información disponible en la firma de la categoría para la posterior localización de los documentos. La búsqueda en el fichero de firmas se realiza en base a dicho código genético, localizando aquellos documentos cuya firma lo contenga. Por ejemplo, considerando una búsqueda restringida a la categoría 5 (y todas las posibles subcategorías) cuyo código genético es 0110 1001, el documento 1 se clasificaría, como era de prever.

Como se observa en el ejemplo, todas las categorías del directorio tienen una firma que los identifica. Por este motivo, el modelo propuesto se denomina arquitectura híbrida con información total, ya que todas y cada una de las categorías que forman parte del grafo disponen de una firma asociada, por lo que aportan información a su código genético (y al de sus descendientes).

A partir del ejemplo previo se pueden destacar las principales diferencias del método propuesto respecto a la superposición de firmas tradicional. En los ficheros de firmas tradicionales, cada bloque estaba constituido por un número constante de palabras, D, lo

que garantizaba un número constante de superposiciones. En cambio, en este caso cada documento puede pertenecer a un número variable de categorías, por lo que el número de superposiciones a la hora de generar la firma del documento también será variable. Por ejemplo, el documento 1 pertenece a cuatro categorías, frente al documento 2 que se incluye en tres categorías.

La segunda diferencia se centra en los valores de D, b y w, ya que en la superposición para los ficheros de firmas se asume que D toma un valor elevado, al igual que b, mientras que w debe tomar un valor mucho menor que b para dar validez a las fórmulas simplificadas del cálculo de la probabilidad de falsos aciertos. En cambio, en este caso, el valor de D, al estar directamente relacionado con la profundidad del grafo, nunca tomará valores elevados, y además, por motivos de optimización del espacio de la estructura de datos no se considerará la utilización de valores grandes de b (al menos no tan elevados como en las aplicaciones tradicionales de los ficheros de firmas).

Estas diferencias implican que las fórmulas (1) y (2), que permitían una aproximación al valor óptimo de w, no puedan ser directamente aplicadas en este caso, sino que es necesario emplear las ecuaciones básicas de la superposición de códigos. El problema de calcular la probabilidad de falsos aciertos en la utilización de códigos de superposición, se puede considerar como un problema relativamente complejo de combinatoria. En [Stiassny, 60] y [Tsichritzis, 83], entre otros, se presentan las ecuaciones básicas del problema de la probabilidad de falsos aciertos, a partir de las cuales se pueden derivar la siguiente fórmula aplicando teoría combinatoria [Cacheda, 01e]:

$$F_d = \frac{\sum_{i=1}^{\min(b,w*D)} \binom{b-w}{i} R_{w+i} - \binom{n-1}{D-1}}{\binom{n}{D} - \binom{n-1}{D-1}} \quad (3)$$

$$R_{w+i} = \frac{\binom{b}{w+i}\binom{n'_i}{D} - \sum_{j=1}^{i-1} R_{w+i-j}\binom{w+i}{j}}{\binom{b}{w+i}}$$

$$n = \binom{b}{w} \qquad n'_i = \binom{w+i}{i}$$

La ecuación (3) permite el cálculo de la probabilidad de falsos aciertos para cualquier valor de D, b y w. En concreto, esta fórmula permite calcular el valor exacto de la probabilidad de que dada una firma se produzca un falso acierto. Por lo tanto, mediante esta expresión su puede calcular la probabilidad de que se produzca un falso acierto para aquellas consultas en las que la búsqueda se ha restringido a una categoría del primer nivel, en donde el código genético de la categoría coincide con su firma. Sin embargo, en los niveles inferiores el código genético aporta más información, por lo que esta fórmula no es aplicable en estos casos.

El cálculo de una fórmula genérica que permita el cálculo de las probabilidades de falsos aciertos a partir de códigos genéticos (en lugar de únicamente firmas, que constituyen un

caso particular de los códigos genéticos) se escapa del objetivo perseguido en este trabajo. Además, existen otros parámetros que pueden repercutir en la probabilidad de falsos aciertos y cuyo análisis matemático es inviable. Por lo tanto, se ha determinado la implementación de un prototipo de la estructura híbrida de fichero invertido y fichero de firmas aplicado al grafo dirigido acíclico de un directorio Web, con el objetivo de estudiar la probabilidad de falsos aciertos ante diferentes situaciones del entorno.

En concreto, la primera tarea consiste en la estimación de los valores de F_d para categorías de diferentes niveles en el grafo. Cada proceso de simulación genera 10 grafos diferentes formados por aproximadamente 2.000 nodos cada uno, estructurados en 6 niveles de profundidad (lo que constituye el número de superposiciones de los códigos) y trabajando con una colección de 50.000 documentos. A cada nodo se le asigna una firma de 52 bits de longitud con un peso de 6 bits (valor óptimo para $D=6$ y $b=52$). Para cada grafo obtenido se simulan un grupo de consultas restringidas a una categoría del grafo para todos los niveles del grafo (se denomina D_q al nivel asignado a la categoría de la consulta). Cada grupo de consultas está formado por 1.000 consultas, recuperando cada una de ellas un total de 1.000 documentos.

Estos valores constituyen la base para el resto de simulaciones descritas a continuación. Los valores obtenidos de probabilidad de falso acierto en función de la profundidad de la consulta se pueden observar en la Figura 4-11.

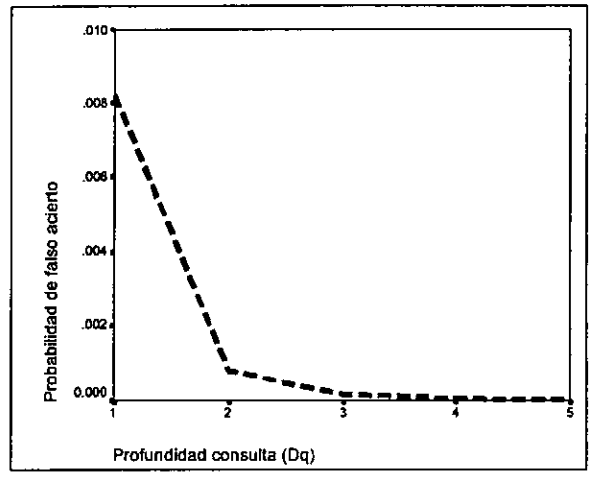

Figura 4-11: Probabilidad de falso acierto en función de la profundidad de la consulta

El valor teórico de la probabilidad de falso acierto para una consulta de nivel 1 (representado como $F_d(D_q=1)$) aplicando la ecuación (3) para los valores de D, b y w de la simulación es aproximadamente de 0,007. El valor obtenido en la simulación es muy próximo al valor teórico, 0,008. Sin embargo, el aspecto más importante de la simulación está en el hecho de que el valor estimado de F_d es decreciente según aumenta la profundidad a la que se restringe la consulta. Esto se deriva del hecho de utilizar los códigos genéticos de las categorías, frente a las firmas, ya que la información aportada en el código genético es más precisa, por lo que permite precisar mejor las búsquedas y reducir por consiguiente el número de falsos aciertos.

Además la reducción producida es muy significativa, aproximadamente de un orden de magnitud por cada nivel de profundidad. Este hecho confirma la facilidad de adaptación y

el buen rendimiento ofrecido por los códigos de superposición para la representación de información jerárquica. Sin embargo, es necesario realizar un estudio detallado sobre los parámetros del modelo con el objetivo de determinar aquellos que presentan un efecto significativo sobre el rendimiento, medido en base a la probabilidad de falsos aciertos.

4.3.2.1. Parámetros de la estructura jerárquica

Uno de los parámetros básicos de la estructura en grafo es su profundidad, y su efecto en la probabilidad de falsos aciertos. Debido a que la profundidad establece el número de superposiciones que se realizarán en las firmas de los documentos, es obvio que constituye un parámetro a tener en cuenta en el diseño del sistema de códigos de superposición. De hecho, se han simulado diferentes entornos aumentando paulatinamente la profundidad del grafo (manteniendo los parámetros b y w constantes) y el empeoramiento en F_d es patente. Esto implica que a la hora del diseño del sistema la profundidad del grafo es un factor clave en la determinación de los valores óptimos de b y w.

Asimismo, en las simulaciones generadas los árboles y grafos obtenidos son estructuras perfectamente balanceadas, mientras que en un entorno real normalmente el grafo dirigido acíclico no presenta esta característica. Sin embargo, el hecho de que un grafo no sea balanceado tiene un efecto positivo en la probabilidad de falso acierto, mejorando ligeramente los valores simulados. Esto es debido a que en algunos casos el número de superposiciones de las firmas de los documentos será menor, al asociarse con categorías no balanceadas y de menor profundidad.

Por otra parte, las características particulares de la estructura sobre la que se aplican los códigos superpuestos permiten una mayor flexibilidad a la hora de la creación y asignación de las firmas de cada categoría. Tradicionalmente el número de firmas empleados en un fichero de firmas era muy elevado (una por cada palabra del vocabulario) utilizando incluso funciones hash que podían asignar firmas iguales a diferentes palabras. En cambio en este caso, únicamente cada categoría requiere una firma diferente (el número de categorías es sensiblemente inferior al de posibles palabras indexadas).

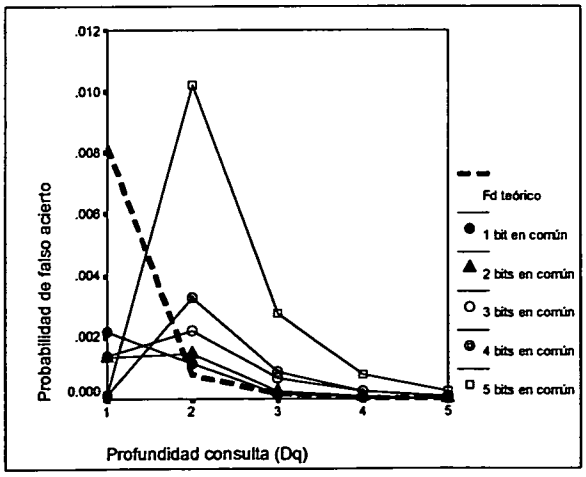

Figura 4-12: Probabilidad de falso acierto para firmas con distintos bits en común con sus ancestros

Por otra parte, dada una categoría con la correspondiente firma es posible determinar con qué firmas será superpuesto su código, debido a que la estructura jerárquica posee una gran componente estática.

Estas dos características incitan al estudio respecto a la asignación de las firmas de las categorías respecto a las de sus "vecinos" (nodos padres y nodos hermanos). En concreto, se ha analizado el efecto de la similitud de las firmas respecto a los nodos ancestros y a los hermanos en distintas simulaciones.

En la primera simulación se han generado las firmas de cada categoría con 1, 2, 3, 4 y 5 bits a "1" en común con las firmas de sus ancestros. Los resultados obtenidos se muestran en la Figura 4-12 (se mantiene el valor teórico de F_d obtenido en la Figura 4-11).

En este gráfico se observan claramente que según aumenta el parecido de las firmas de una categoría con la de sus ancestros, menor es la probabilidad de falsos aciertos en el primer nivel debido a que todos los nodos descendientes (y por lo tanto los documentos asociados) presentan firmas similares, lo que hace más difícil la confusión entre dos códigos diferentes. En cambio, la probabilidad en el segundo nivel empeora progresivamente, ya que al aumentar el parecido con los ancestros es más difícil la diferenciación entre dos nodos hermanos. Sin embargo, es importante localizar el punto de equilibrio entre ambos efectos, que se produce cuando las firmas presentan 2 bits en común, ya que el valor $F_d(D_q=1)$ desciende hasta 0,00133, mientras que en el resto de niveles el empeoramiento es poco significativo.

En una segunda simulación se ha chequeado el efecto de la similitud entre las firmas de nodos hermanos. Para ello se han realizado diferentes experimentos considerando 0, 1, 2 y 3 bits en común entre las firmas, tal y como se muestra en la Figura 4-13. En este caso se observa como cuanto menor es el número de bits en común entre los nodos hermanos mejor es el rendimiento que se obtiene en todos los niveles, sin embargo, el hecho de reducir el número de bits en común también reduce drásticamente el número de códigos posibles a emplear (por ejemplo, con $b=52$ se pueden generar únicamente 8 códigos diferentes con 0 bits en común entre sí). Por lo tanto, al emplear esta cualidad es necesario tener en cuenta el número de códigos necesarios (o sea, el número de categorías del grafo)

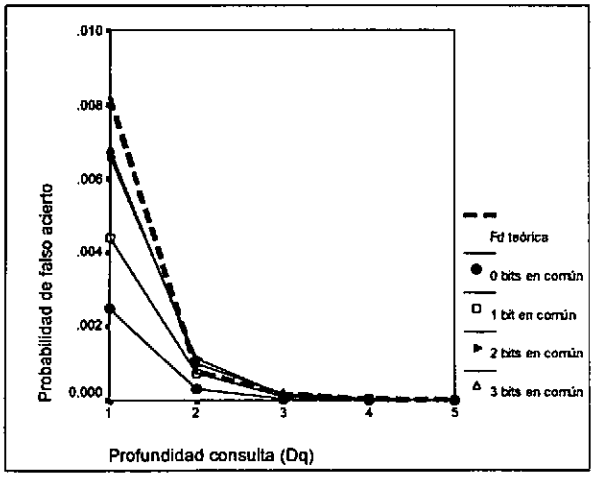

Figura 4-13: Probabilidad de falso acierto para firmas con distintos bits en común con los nodos hermanos

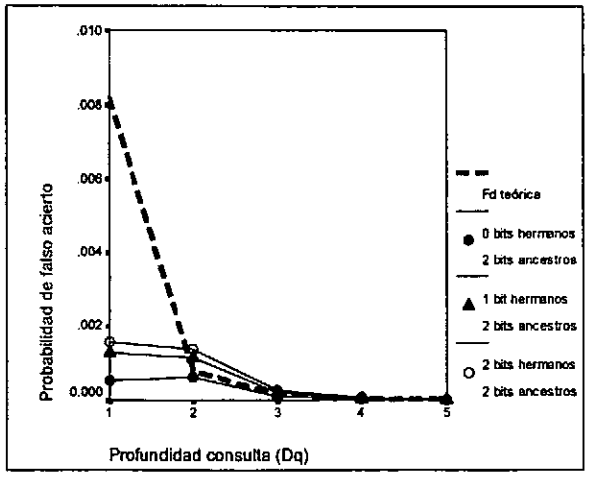

Figura 4-14: Probabilidad de falso acierto para firmas con 2 bits en común con sus ancestros y distintos con los nodos hermanos

y el número de códigos disponibles.

Asimismo, por medio de la combinación de ambas cualidades se pueden obtener nuevas mejoras en los valores de la probabilidad de falsos aciertos. Para ello se han realizado múltiples simulaciones que combinaban 1, 2, 3 y 4 bits en común con los nodos ancestros y 0, 1 y 2 bits en común con los nodos hermanos. En la Figura 4-14 se muestra la mejor serie de valores obtenidos que se corresponden con firmas que posean 2 bits en común con el código genético de sus ancestros, y 0, 1 ó 2 bits en común entre los nodos hermanos.

Los resultados obtenidos de esta primera parte de la investigación realizada permiten una reducción de los valores de F_d en todos los niveles, siendo la más importante la que se produce en el primer nivel al pasar de un valor inicial de 0,008 a un valor de 0,0005 (un orden de magnitud inferior). Sin embargo, esto únicamente constituye un primer paso y en futuros trabajos será necesario profundizar en nuevas mejoras posibles en el diseño y elaboración de los códigos de superposición para estructuras jerárquicas.

4.3.2.2. Parámetros de documentos

El primer parámetro analizado referente a los documentos clasificados dentro de la jerarquía se corresponde con el número de documentos incluidos dentro de la colección. Inicialmente las simulaciones partían de una base de 50.000 documentos, que se ha ampliado en distintos tramos hasta alcanzar los 500.000 documentos. En cada caso, se han estimado los valores obtenidos para los distintos niveles de profundidad de F_d.

Tal y como se observa en la Figura 4-15, la independencia entre el número de documentos de la colección y la probabilidad de falsos aciertos es manifiesta, al permanecer prácticamente constante F_d en función del número de documentos. Asimismo, se contrastó por medio del test ANOVA que confirmó con un 99,999% de seguridad que los distintos niveles de F_d eran independientes del número de documentos en la colección. Esto garantiza que el tamaño de la colección puede aumentar sin repercusión alguna en la probabilidad de falsos aciertos del sistema.

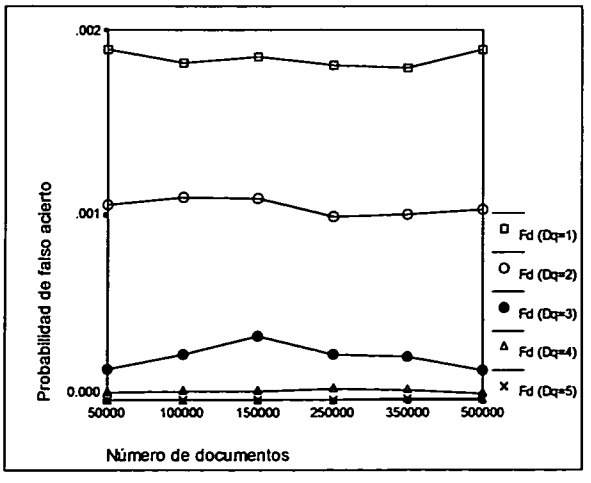

Figura 4-15: Probabilidad de falso acierto en función del número de documentos en la colección

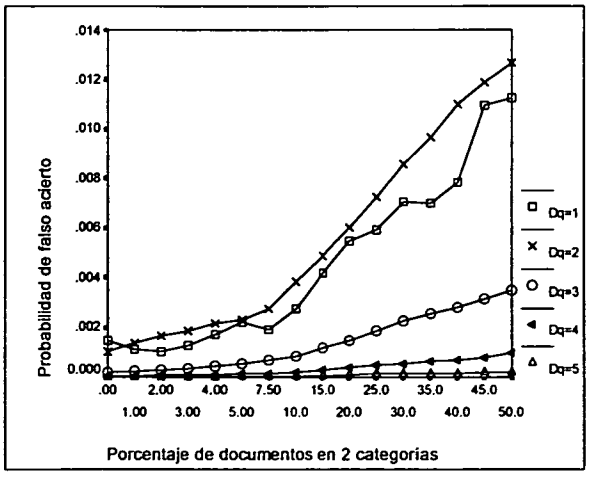

Figura 4-16: Probabilidad de falso acierto en función del porcentaje de documentos asociados a 2 categorías

Por otra parte, otro parámetro interesante, a efectos de simulación, es el número de documentos recuperados en una consulta simulada, por lo que es necesario chequear la independencia entre F_d y el número de resultados de una consulta simulada. Bajo este objetivo se realizaron diferentes simulaciones recuperando entre 100 y 10.000 resultados. Posteriormente, se aplicó el test ANOVA para contrastar si los diferentes valores de F_d (para cada nivel de profundidad) se podrían considerar constante, opción confirmada con un 99,999% de probabilidad (se obvia el gráfico asociado por su similitud con el anterior).

Hasta este momento las simulaciones representando los documentos se ajustaban a un modelo ideal, en el que cada documento está asociado a una única categoría lo que provoca que el número de superposiciones sea constante. Sin embargo, en un entorno real los documentos pueden estar asociados a varias categorías. Este hecho aplicado a los códigos de superposición implica que la firma de un documento pasará de estar formada por la superposición de D firmas de categorías, a estar formada, en el peor de los casos, por la superposición de $D+D$ firmas. Esto, evidentemente, afectará directamente a la probabilidad de falsos aciertos del sistema.

Con el objetivo de determinar tal efecto se han simulado múltiples entornos con diferentes porcentajes de documentos asociados a dos categorías realizando un estudio detallado de los resultados obtenidos. En la Figura 4-16 se puede observar como el empeoramiento producido en F_d es más acusado en los niveles 1 y 2, y mucho más sutil en los niveles inferiores. De hecho, si la mitad de los documentos se asociasen a dos categorías el valor de F_d empeoraría en dos órdenes de magnitud.

Esto hace patente el hecho de que los parámetros de diseño de los códigos de superposición no son adecuados para esta situación. Inicialmente se calcularon los valores óptimos de los parámetros para $D=6$. El hecho de que en un porcentaje significativo de los casos $D=12$ modifica drásticamente las opciones del diseño, y por lo tanto sería necesario recalcular una aproximación para w. Esto da gran importancia al hecho de determinar a priori el porcentaje de documentos asociados a múltiples categorías ya que su efecto en la probabilidad de falsos aciertos es drástico. De todas formas, este efecto que puede ser minimizado con los parámetros de diseño adecuados.

4.3.2.3. Parámetros de categorías

El primer parámetro importante respecto a las categorías del grafo consiste en determinar si el número de categorías influye en la probabilidad de falsos aciertos. Para ello se han simulado diferentes árboles con aproximadamente entre 1.000 y 21.000 categorías, analizando los valores de F_d en cada caso. Los resultados se muestran en la Figura 4-17, en donde se observa que F_d permanece estable independientemente del número de categorías del árbol. Además, a través del test ANOVA se confirma (con una probabilidad del 88%) que ambos factores son independientes.

El segundo parámetro consiste en determinar la repercusión que tendrá en F_d la incorporación de categorías con varios padres en el grafo. Obviamente, el hecho de incorporar varios padres a un nodo aumenta el número de superposiciones necesarias para el cálculo de su código genético, y consecuentemente el de todas las categorías y documentos inferiores. Sin embargo, el aumento en el número de superposiciones depende

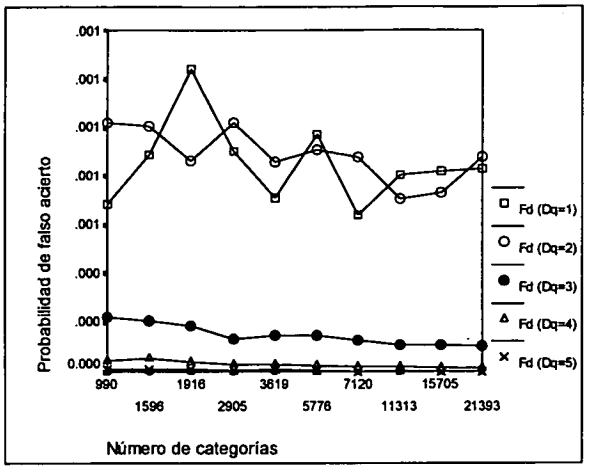

Figura 4-17: Probabilidad de falso acierto en función del número de categorías en el grafo

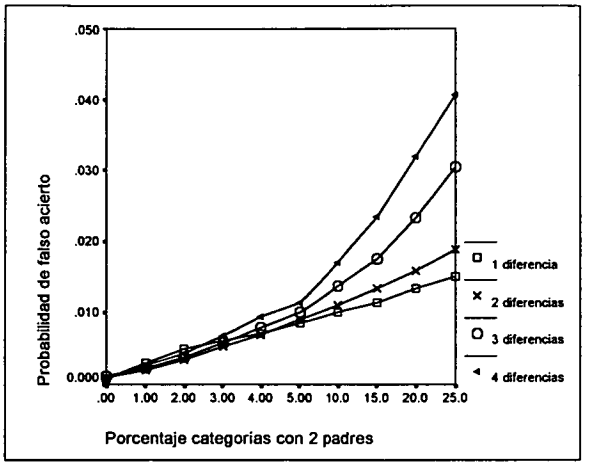

Figura 4-18: Probabilidad de falso acierto en función del porcentaje de categorías con 2 padres

de las ubicaciones de los nodos padre (independientemente de su profundidad). Esto se debe a que los distintos padres pueden presentar partes de su jerarquía en común, por lo que el número final de superposiciones sería menor.

Este aspecto ha sido analizado por medio de la asociación de un nuevo padre a las categorías del último nivel, teniendo en cuenta las diferencias de códigos genéticos entre el nuevo padre y el padre "real". Para un análisis completo se introdujeron diferentes porcentajes de categorías con dos padres (desde un 2% hasta un 25% del total) estimando los valores de F_d. En la Figura 4-18 se muestran los valores obtenidos en función de las diferencias entre ambos padres para $F_d(D_q=1)$ (ya que el primer nivel es el que ofrece una mayor repercusión respecto al aumento de la probabilidad de falsos aciertos). Además se muestran cuatro gráficos correspondientes a los casos en los que los padres presenten 1, 2, 3 y 4 nodos diferentes, respectivamente.

Como se observa en el gráfico la repercusión del número de categorías con varios padres es muy elevada, produciéndose un incremento de la probabilidad de falsos aciertos muy acuciada para un 25% de categorías con varios padres. Además, el efecto del número de nodos diferentes entre los padres es significativo pasando de una probabilidad de falsos aciertos del 4% en el peor de los casos (con cuatro nodos diferentes) a una mejora del 1% en el mejor de los casos (con un nodo diferente).

En resumen, la distribución de las categorías en la jerarquía presenta una gran repercusión en el rendimiento del sistema. Por una parte, la probabilidad de falsos aciertos es independiente del número de categorías, sin embargo, el porcentaje de categorías con varios padres sí produce un efecto directo en F_d. Este efecto viene marcado por las diferencias existentes entre los diferentes padres de un nodo, independientemente del nivel del grafo en el que se sitúen. Por lo tanto, es preferible la asignación de varios padres jerárquicamente similares a un nodo de posiciones elevadas en el grafo, que la asignación de varios padres a nodos de posiciones bajas, a pesar de que en este caso el número de documentos afectados sea menor.

Por último se ha comprobado que la interacción entre documentos y categorías con varios padres simplemente presenta un efecto aditivo, con preponderancia de las categorías. Por ejemplo, en un sistema con un 10% de categorías con dos padres (lo que implica $F_d=0,011$)

y con un 30% de documentos asignados a dos categorías (lo que implica F_d=*0,007*), el valor de F_d resultante de la combinación empeora ligeramente pasando a ser de 0,012. En cualquier caso, al igual que sucedía en el caso de los documentos, es necesario replantear el cálculo de los parámetros de diseño de los códigos de superposición con el objetivo de minimizar el efecto sobre la probabilidad de falsos aciertos.

4.3.3. Parámetros de la arquitectura

La arquitectura híbrida de fichero invertido y fichero de firmas con información total, se basa en la utilización de identificadores de documentos compuestos, en donde una parte se encuentra constituida por la superposición de las firmas de todas aquellas categorías a las que se encuentra asociado el documento directa o indirectamente.

En la sección anterior se ha analizado en detalle la adaptación de la técnica de los códigos de superposición a la estructura del grafo de categorías de un directorio Web típico, estudiando los efectos de los parámetros principales de dicha estructura. A modo de resumen las conclusiones principales indican que el número de documentos y de categorías del sistema no repercuten en la probabilidad de falsos aciertos, mientras que determinados parámetros estructurales tienen un efecto claro en el rendimiento al modificar el número medio de superposiciones de las firmas de los documentos. En concreto, la profundidad del grafo, el porcentaje de categorías con varios padres y el porcentaje de documentos asociados a más de una categoría presentan un efecto negativo que exige su consideración a la hora del establecimiento de los parámetros de la componente del fichero de firmas.

En concreto, los parámetros básicos asociados se derivan de la componente del fichero de firmas: D, b y w. En este caso, el número de superposiciones por bloque (D) es un parámetro variable y viene impuesto por la propia estructura del grafo (profundidad, categorías y documentos con varios padres). Sin embargo, los valores asignados a b y w tendrán una repercusión directa en el rendimiento ofrecido por el sistema, por lo que se requiere una buena aproximación. El valor asignado a b repercutirá directamente en el tamaño final que tomará la estructura de datos en su conjunto, mientras que w, para un valor concreto de b, deberá minimizar el valor de F_d.

Desde un punto de vista teórico, una buena aproximación para el valor de w puede ser obtenida a partir de la ecuación (1) partiendo de un valor medio estimado de D y del valor asignado a b. Aunque también se pueden emplear técnicas de simulación para intentar confirmar y contrastar el valor obtenido.

Técnicamente, el modo de operación normal consistirá en a partir del valor medio estimado de D y de una probabilidad de falsos aciertos objetivo, combinar los valores de b y w para alcanzar el valor de F_d óptimo. Obviamente, cuanto mayor sea el valor de b, menor será la probabilidad de falsos aciertos y en consecuencia, el proceso de filtrado será más preciso. Por otra parte, el aumento de b también repercute negativamente en el espacio de almacenamiento requerido por los índices. En consecuencia, en la implementación del sistema se debe lograr una situación de compromiso ante ambos parámetros que afectan directamente al rendimiento del sistema.

4.4. Arquitectura híbrida con información parcial

En esta sección se describe una variante propuesta sobre la arquitectura anterior que, a través de una sencilla modificación permite mejorar el rendimiento ofrecido por el sistema de recuperación de información ante las consultas restringidas a una zona del grafo.

Comúnmente, la manera habitual de reducir F_d consiste en aumentar el valor de b (ver ecuación (1)), sin embargo, también es posible reducir el valor de la probabilidad de falsos aciertos reduciendo el valor de D, o lo que es lo mismo, reduciendo el número de superposiciones en las firmas. En las aplicaciones tradicionales de los ficheros de firmas esto no era posible ya que este valor era un parámetro fijo de diseño.

En cambio, en la arquitectura propuesta si es posible realizar una reducción en el número de superposiciones. En el modelo anterior la superposición se lleva a cabo en todos los niveles del grafo, en cambio, es posible aplicar la técnica de los códigos de superposición únicamente a ciertos niveles del grafo, y dejar que el resto de nodos simplemente hereden las firmas de los niveles superiores. De esta manera, se reduce el número medio de superposiciones y en consecuencia el valor de D. En la Figura 4-19 se muestra una modificación del ejemplo anterior, en donde únicamente las categorías del nivel 0 y 1 disponen de firma, mientras que las de nivel 2 heredan directamente, reduciendo el número de superposiciones finales en los documentos.

A esta variante de la arquitectura propuesta se le denomina arquitectura híbrida con información parcial, ya que en este caso únicamente una parte de las categorías disponen de firma propia y por lo tanto aportan información a los códigos genéticos, mientras que el resto simplemente heredan la información de los códigos de sus ancestros.

De manera más detallada, en esta alternativa únicamente las categorías de los primeros

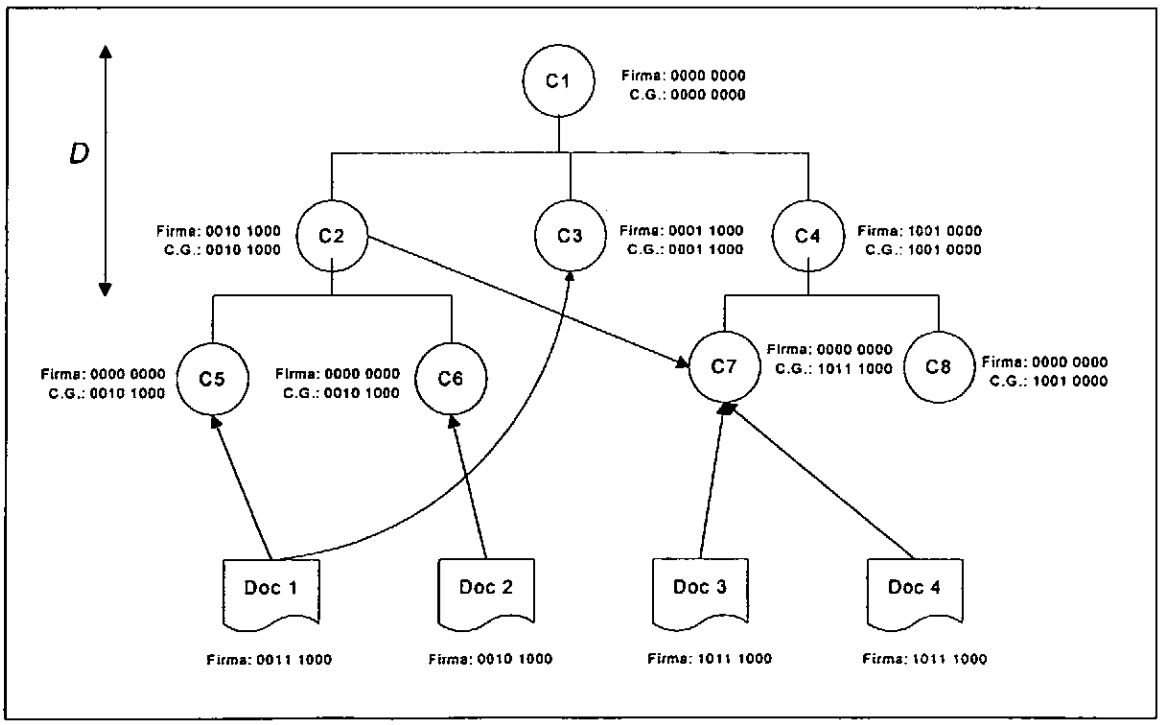

Figura 4-19: Códigos de superposición aplicados a un grafo dirigido acíclico, con reducción de *D*

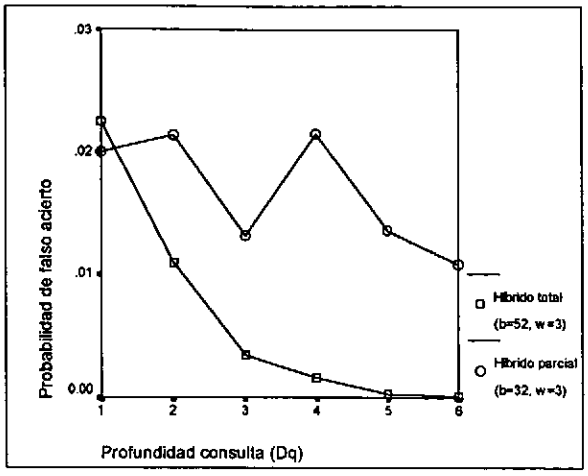

Figura 4-20: Probabilidad de falso acierto estimada para los dos modelos propuestos

niveles (típicamente, los tres primeros niveles) disponen de firma asociada, mientras que los nodos de los niveles inferiores no aportan información propia, disponiendo únicamente del código genético heredado de sus ancestros. De esta manera se reduce el valor de D, permitiendo para valores menores de b (y por consiguiente utilizando un menor espacio de almacenamiento) obtener valores similares para la probabilidad de falsos aciertos en los primeros niveles.

El único inconveniente que presenta esta alternativa se centra en las consultas restringidas a los niveles inferiores, ya que al carecer de firma propia, la calidad del filtrado disminuye al coincidir el código genético de varias categorías de un mismo nivel (en realidad, aquellas categorías con ancestros comunes). Esto, se traduce en un aumento de la probabilidad de falsos aciertos para los niveles inferiores, que en todo momento debe estar controlado a fin de evitar niveles que impliquen una disminución del rendimiento del sistema.

Sin embargo, los efectos de este inconveniente se ven minimizados por dos factores. En primer lugar, la mayoría de las búsquedas restringidas a categorías se centran en los primeros niveles del grafo. De hecho, en el análisis realizado los tres primeros niveles concentraron más del 81% del total de consultas (ver sección 2.4.1). Y en segundo lugar, los valores típicos de F_d para los niveles inferiores son tres o cuatro órdenes de magnitud inferiores (ver Figura 4-11), mientras que el aumento producido no es demasiado acusado, haciendo que los valores de F_d sean homogéneos para todos los niveles.

En la Figura 4-20 se muestran los valores estimados para F_d según los dos modelos propuestos. En el primer modelo (arquitectura híbrida con información total) aplicado a la totalidad del grafo el diseño óptimo se obtiene con $b=52$ bits y $w=3$ bits, mientras que en el segundo modelo (arquitectura híbrida con información parcial) aplicado únicamente a los tres primeros niveles del grafo los parámetros óptimos son $b=32$ bits y $w=3$ bits (con una generación de firmas con similitud a los ancestros).

Como se observa en el gráfico, el primer modelo presenta un mejor comportamiento a nivel general, especialmente para las consultas restringidas a los niveles inferiores del grafo. Por otra parte, los valores de la probabilidad de falsos aciertos para el segundo

modelo presentan un comportamiento más homogéneo para diferentes profundidades de las consultas.

Sin embargo, es importante destacar que los tiempos de respuesta ante las consultas restringidas en estos modelos están determinados por dos factores: el número de falsos aciertos y el tiempo de recuperación/obtención de los ficheros de firmas. Si bien, un valor de F_d elevado puede hacer caer drásticamente el rendimiento del sistema, para valores similares la repercusión puede ser mínima. Por otra parte, el segundo factor está asociado directamente con el tamaño del índice invertido, que a su vez depende del tamaño de las firmas. Por lo tanto, el tamaño de las firmas constituye un factor crucial en el tiempo de lectura del índice, y en consecuencia también en el tiempo de respuesta de una consulta restringida.

Para el correcto análisis de la repercusión de ambos efectos, en el capítulo 5 se aborda la implementación y evaluación, tanto del modelo híbrido con información total como del modelo híbrido con información parcial, frente al modelo básico. La evaluación se centra en los aspectos del rendimiento del sistema, usando para ello la herramienta de simulación USim, descrita en el capítulo 3.

4.5. Trabajos relacionados

Existen diversos trabajos publicados que utilizan estructuras híbridas de datos con el objetivo de mejorar los tiempos de respuesta ante búsquedas en diferentes sistemas de búsqueda. La mayor parte de estos trabajos se centran en sistemas de almacenamiento de texto (aplicado directamente a bases de datos o incluso sobre mecanismos de recuperación de imágenes). Aunque ninguno de los trabajos se centra en los grafos dirigidos acíclicos, si muestran las posibilidades que pueden ofrecer este tipo de sistemas.

En concreto, Dervos et al. en [Dervos, 96] y [Dervos, 97] exponen una nueva estructura denominada S-Index (Signature-Index) que es el resultado de combinar la técnica de ficheros invertidos (con resultados eficientes en el procesamiento de consultas, aunque con una sobrecarga de almacenamiento elevada) y los ficheros de firmas basados en códigos de superposición (que presentan unos menores requerimientos de espacio).

La estructura S-Index se caracteriza por presentar un rendimiento que puede ser ajustado, obteniendo resultados que se encuentran entre los dos extremos. Por una parte, S-Index se puede comportar como un fichero de firmas, con un rendimiento superior a los ficheros de firmas tradicionales. Y por otra parte, S-Index puede ser ajustado para comportarse como un sistema de fichero invertido.

Los autores proponen la utilización de este sistema para consultas de recuperación de texto, en donde aquellas partes más frecuentemente consultadas son indexadas utilizando la técnica de fichero invertido con el objetivo de obtener una mejora en el rendimiento; mientras que el grueso de la base de datos, que no será frecuentemente accedida por los usuarios, es almacenada utilizando la técnica de ficheros de firmas al ofrecer una mejor tasa de compresión.

En otro trabajo realizado por Faloutsos et al. [Faloutsos, 92b] se estudia como la utilización de un sistema híbrido puede mejorar los tiempos de respuesta, teniendo en cuenta la

naturaleza sesgada de la distribución de frecuencia de los términos indexados. Partiendo de la base de que una única técnica no permitirá obtener resultados adecuados en todas las situaciones, se plantea la utilización de una aproximación híbrida para la indexación de texto.

En concreto, identifican el problema de espacio de la técnica de ficheros invertidos en el almacenamiento de las listas invertidas, especialmente en el caso de aquellos términos con un gran número de ocurrencias. Por este motivo, en el sistema propuesto, aquellos términos más frecuentes sustituyen la lista invertida de ocurrencias por un vector de bits. El conjunto de todos los vectores de bits se almacenan conjuntamente formando un mapa de bits. De esta manera, en el caso de realizarse una consulta sobre un término frecuente únicamente es necesario recuperar una columna del mapa de bits para identificar a todos aquellos documentos asociados con este término.

Las conclusiones del trabajo demuestran que tanto en entornos estáticos como dinámicos el sistema híbrido mejora el rendimiento en todas las medidas de interés: espacio, tiempo de repuesta y tiempo de inserción. Sin embargo, el sistema ha sido probado en entornos reducidos, y sería necesario aumentar la experimentación hacia entornos con un mayor volumen de información.

En otro artículo publicado por Cha et al. en [Cha, 99] describen un mecanismo para la recuperación de imágenes en base a su contenido. El contenido de una imagen se define en base a tres tipos de rasgos: cuantitativos (que describen la información visual), y no cuantitativos (que describen la información semántica y ciertas palabras clave que describen información más abstracta).

En su sistema, definen tres tipos de índices diferentes considerando cada uno de los rasgos a almacenar. Los tres índices están basados en una estructura definida por ellos mismos y denominada HG-tree [Cha, 98]. En concreto, en la representación de las palabras clave emplean la técnica de ficheros de firmas, al ser un sistema de calidad probada para el almacenamiento de texto y la recuperación en base a múltiples atributos.

4.6. Conclusiones

En este capítulo se ha analizado como puede ser abordada la realización de consultas restringidas a una zona de la jerarquía de categorías de un directorio Web.

Considerando el modelo básico se definen dos alternativas para la resolución de este tipo de consultas. La primera opción se comporta adecuadamente en aquellas consultas restringidas a una zona reducida de la jerarquía (incorporando además una posible penalización al proceso de navegación por las categorías del grafo). Por otra parte, la segunda alternativa presenta un resultado eficiente únicamente en aquellas consultas que recuperan un número reducido de resultados. Obviamente, ninguna de ambas alternativas ofrece una solución eficiente ante este tipo específico de consultas.

La principal aportación del presente capítulo se centra en la definición de un modelo de arquitectura híbrida de fichero invertido y fichero de firmas, inicialmente con información total, con el objetivo de mejorar el rendimiento de las búsquedas restringidas.

La arquitectura propuesta se basa en una estructura de datos híbrida, constituida por un fichero invertido en donde cada una de las listas invertidas contiene embebido su propio fichero de firmas. Para ello se ha definido un identificador de documento compuesto (que contiene las firmas de todas sus categorías asociadas), de tal manera que por medio del propio proceso de búsqueda se obtiene de manera dinámica el fichero de firmas asociado con cada consulta.

Sobre este fichero de firmas se realiza un primer filtrado inexacto que permite eliminar la mayoría de los documentos que no se clasificarán, para después aplicar de manera eficiente un filtrado exacto según la segunda alternativa (que ofrece un rendimiento adecuado para este caso).

Una segunda aportación en este trabajo consiste en la adaptación de la técnica de los códigos de superposición para representar las categorías a las que está asociado un documento. Para ello, se ha examinado en detalle la compleja estructura del grafo de categorías de un directorio Web, identificando aquellos parámetros (profundidad del grafo, categorías con varios padres, etc.) con una mayor repercusión en el rendimiento del sistema, esto es, en la probabilidad de falsos aciertos.

Por otra parte, se plantea una variante del modelo propuesto denominada arquitectura híbrida con información parcial que aplica la técnica de los códigos de superposición únicamente a una parte del grafo. De esta manera se consigue una reducción del espacio de almacenamiento, manteniendo estables los valores de F_d de los primeros niveles y con un empeoramiento controlado en los niveles inferiores que permite conseguir valores homogéneos en todos los niveles.

A nivel general la arquitectura híbrida propuesta, en sus dos variantes, permite obtener las siguiente ventajas:

- Desde un punto de vista teórico, esta arquitectura permitirá mejorar el rendimiento ofrecido por los directorios Web ante búsquedas restringidas a una zona de la ontología. Es tarea del siguiente capítulo determinar el porcentaje de mejora que se obtiene en cada modelo, respecto al sistema básico.
- La arquitectura propuesta se centra en aplicar la técnica de los ficheros de firmas sobre un atributo para su posterior filtrado. Las características peculiares de un grafo dirigido acíclico de categorías permiten que la flexibilidad de la técnica de los códigos de superposición se adapte adecuadamente a este entorno. Si bien, el modelo propuesto es válido de manera genérica, y aunque su arquitectura e implementación han sido definidas para este problema concreto, su aplicación es adecuada para cualquier filtrado sobre un fichero invertido, en donde la técnica de combinación de listas invertidas no sea adecuada.
- Debido a las características intrínsecas del filtrado, no se requiere ningún tipo de ordenación previa de las listas. Esto permite su adaptación e implantación en cualquier sistema de ficheros invertidos sin restricciones sobre las listas invertidas. Este es un aspecto importante, ya que el rendimiento de un sistema de búsqueda suele depender en gran medida de la ordenación de las listas invertidas (ver sección 1.5.2.1).

- En ambos modelos propuestos, se completan las búsquedas restringidas mediante la realización de un filtrado exacto. Este paso puede ser eliminado, principalmente en dos casos, con un ahorro significativo en el tiempo de respuesta. La eliminación del filtrado exacto simplemente producirá la aparición de documentos resultantes de la búsqueda que no se encuentran realmente en la zona restringida. En un primer caso, si la probabilidad de falsos aciertos es suficientemente reducida el número de falsos aciertos será prácticamente nulo. En segundo lugar, en el caso de que se hayan clasificado una gran cantidad de documentos (porque el número de aciertos reales es muy elevado) el porcentaje de falsos aciertos puede ser despreciable, y además sus efectos pueden ser minimizados por el algoritmo de ordenación.
- Además, el rendimiento del sistema es independiente del número de documentos y de categorías incluidos en el directorio Web, mientras que los parámetros que afectan directamente al rendimiento (profundidad, porcentaje de categorías con varios padres y porcentaje de documentos asociados a varias categorías) se caracterizan por permanecer estables a lo largo de la vida del sistema de búsqueda. Por lo tanto, los modelos híbridos se adaptan adecuadamente a los cambios típicos producidos en los directorios Web, sin un empeoramiento en el rendimiento de las consultas restringidas.
- Finalmente, asociado con el buen rendimiento de las búsquedas restringidas a una zona del grafo de categorías se encuentran múltiples posibilidades de explotación. Por ejemplo, se posibilitaría la aparición de restricciones inversas (restringir una búsqueda a todo el grafo excepto una zona concreta), o incluso la combinación de ambas restricciones (normal e inversa).

5. IMPLEMENTACIÓN DE LA ARQUITECTURA PROPUESTA

5.1. Introducción

En este capítulo se presenta la implementación de la arquitectura híbrida de fichero invertido y fichero de firmas en sus dos variantes (con información total y con información parcial), junto con el modelo básico, para la evaluación del rendimiento ofrecido por cada uno de ellos ([Cacheda, 02a], [Cacheda, 02b]).

Las diferentes implementaciones realizadas consisten en el desarrollo de un prototipo de un directorio Web, basado en un entorno real. En concreto, el prototipo consta de un grafo de categorías compuesto de 888 categorías distribuidas en 7 niveles de profundidad, en el cual se han clasificado e indexado más de 51.000 páginas Web diferentes. Tanto en las categorías como en los documentos se encuentran casos con múltiples padres asociados, en concreto, un 16% de las categorías presentan varios padres, mientras que un 27% de los documentos se encuentran asociados a varias categorías. Los datos han sido obtenidos en base al contenido de un directorio Web español, BIWE [Biwe, 01], por lo que constituyen un entorno real para la validación de los distintos modelos propuestos.

El desarrollo de los prototipos se basa en una arquitectura de tres capas, especialmente utilizada en los sistemas reales del World Wide Web. La primera de las capas se centra en el almacenamiento y recuperación de los datos, la capa central realiza el procesamiento sobre los datos y por último la capa cliente [Cacheda, 99].

Para la implementación de los diferentes prototipos, se ha considerado la utilización de un sistema gestor de bases de datos para la implementación de la capa inferior de la arquitectura, mientras que las capas superiores se basan en Java, lenguaje de programación orientado a objetos y multiplataforma y que ha alcanzado una gran popularidad en los desarrollos Web [Gosling, 96]. La utilización de un sistema gestor de bases de datos garantiza el correcto y óptimo almacenamiento de los datos en el disco. En concreto, el gestor de bases de datos empleado es Oracle, ya que a través de la organización de los datos por medio de clusters e índices facilita la implementación de las diferentes estructuras de datos de los modelos propuestos.

Básicamente las principales diferencias entre los tres prototipos se encuentran en las estructuras de datos empleadas, especialmente aquellas relacionadas con las búsquedas restringidas a una zona del grafo. En consecuencia, los detalles de implementación expuestos a continuación se centrarán explícitamente en dichas estructuras de datos, describiendo especialmente los parámetros de diseño, así como la implementación propiamente dicha. Se asume que el resto de componentes del directorio presenta un diseño e implementación común a los tres diferentes modelos expuestos.

Las implementaciones presentadas a continuación de los dos modelos propuestos están basadas en el modelo de datos básico definido en la sección 4.2. El modelo básico está basado en la utilización de estructuras de ficheros o índices invertidos, que pueden ser directamente definidos e implementados a través de cualquier sistema gestor de bases de datos, por lo que los detalles de implementación son obviados. Las capas superiores, como se ha comentado anteriormente, han sido implementadas en su totalidad en Java y son comunes a los diferentes prototipos, lo que permite realizar las comparaciones de tiempos de respuesta con mayor fiabilidad.

Los desarrollos han sido realizados sobre una máquina Ultra Enterprise 250 con 768 MB de memoria, un único procesador a 300 MHz y un disco ultra-SCSI de 18 GB. La evaluación del rendimiento de cada uno de los sistemas propuestos se realizará sobre esta máquina, estando en cada caso únicamente operativos los diferentes procesos asociados con el sistema de búsqueda correspondiente, sin ningún tipo de carga adicional sobre la máquina, para garantizar la objetividad de los tests realizados.

A continuación se presentan las implementaciones realizadas de la arquitectura híbrida tanto de información total como de información parcial, siendo el objetivo final una evaluación del rendimiento ofrecido por los diferentes sistemas frente al modelo básico, centrado especialmente en las búsquedas restringidas a una zona del grafo. De todas formas, también se comprobará el correcto rendimiento de los sistemas propuestos en los procesos de búsqueda normales.

5.2. Implementación arquitectura híbrida con información total

El modelo de arquitectura híbrida con información total se caracteriza porque todas y cada una de las categorías del grafo (sin importar su nivel de profundidad) tienen asociada una firma, y en consecuencia aportan una cierta información a su código genético.

A continuación se describen los valores óptimos para los parámetros del sistema y su repercusión en las estructuras de datos asociadas.

5.2.1. Parámetros de diseño

El primer paso a la hora del diseño e implementación de un modelo híbrido de fichero invertido y fichero de firmas, es la determinación de los parámetros correspondientes al fichero de firmas y los códigos de superposición empleados, *D*, *b* y *w*. En el caso concreto de la aplicación de los códigos de superposición a un grafo dirigido acíclico la estimación de un valor aproximado para *D* es una tarea compleja, ya que diferentes factores repercuten en el valor final. En concreto, el número de categorías con varios padres y el número de documentos asociados a varias categorías aumentan directamente este valor, mientras que los documentos asociados a categorías que no se encuentran en el último nivel del grafo reducen ligeramente el número de superposiciones a realizar.

En consecuencia, como la estimación de un valor para *D* sería compleja y su utilización para el cálculo de *w* podría dar lugar a errores, al emplear valores medios, se ha preferido realizar diferentes simulaciones para obtener el valor óptimo de *w* para el caso concreto analizado. Las simulaciones realizadas siguen el modelo de las definidas en la sección 4.3.2.

Las simulaciones se basan en la estructura del directorio a implementar, en donde se ha reproducido el grafo de categorías y los documentos asociados, para disponer de una representación exacta de las superposiciones a realizar, y por lo tanto de *D*. Por otra parte, el valor de *b* ha sido fijado con antelación en 52 bits. El número de bits asignados a las firmas ha sido previamente establecido con el objetivo de restringir al máximo el aumento en el espacio de almacenamiento necesario por las nuevas estructuras de datos. De esta forma, se debe estimar un valor adecuado para *w* (número de "1" asignados en cada firma), para que el número de falsos aciertos por consulta no sea demasiado elevado y no repercuta en los tiempos de respuesta finales obtenidos en las búsquedas restringidas a categorías.

Las primeras simulaciones realizadas estudiaron el rendimiento a nivel de probabilidad de falsos aciertos obtenido en función del número de "1" asignados a diferentes firmas. En la Figura 5-1 se observa claramente que cuanto mayor es el valor de *w* peor es el rendimiento que se obtiene, y en concreto los mejores valores se obtuvieron con *w=3* y *w=4*.

En estos casos, es conveniente escoger el menor valor de *w* ya que esto garantiza que en caso de aumentar el número de superposiciones en el futuro (por un aumento en el número de categorías con varios padres o en el número de documentos asociados a varias categorías), el hecho de que los códigos dispongan de un menor número de "1"s repercutirá en menor medida en la probabilidad de falsos aciertos obtenida. Por otra parte, también es importante tener en cuenta el número de códigos o firmas diferentes que se pueden obtener en cada caso, ya que cada categoría requiere una firma única en todo el grafo. En concreto, si *w=3* se pueden generar 22.100 códigos diferentes (*C(52, 3)=22.100*), mientras que si *w=4* se generarán 270.725 códigos diferentes (*C(52, 4)=270.725*). Obviamente, en la implementación realizada el número de códigos obtenidos para *w=3* es más que suficiente para el grafo de categorías considerado y permite un

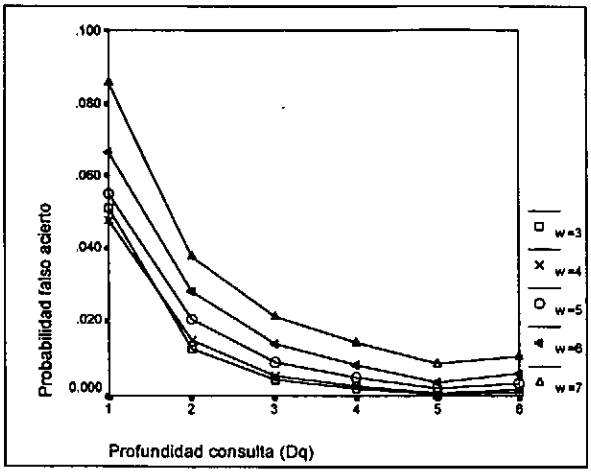

Figura 5-1: Probabilidad de falso acierto para diferentes valores de *w*

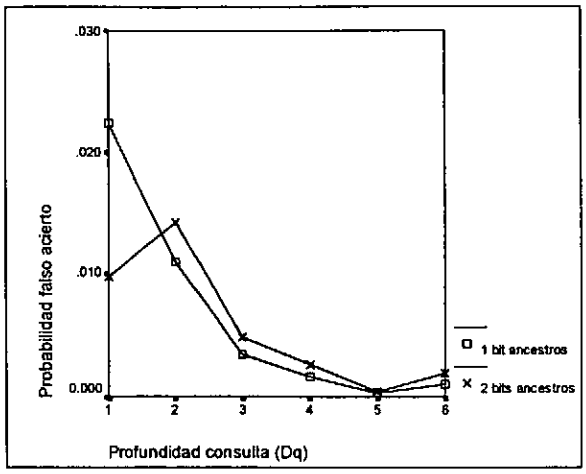

Figura 5-2: Probabilidad de falso acierto para *w=3* con varios bits en común con sus ancestros

aumento considerable del número de categorías, sin una disminución del rendimiento del sistema, como se ha demostrado previamente.

Esto constituye el primer paso, sin embargo como se mostró en la sección 4.3.2.1 es posible mejorar el rendimiento estudiando la asignación de las firmas a las categorías en función de las firmas de sus ancestros (o lo que es lo mismo, del código genético) y de los nodos hermanos. Para este caso concreto, se han analizado las diferentes variaciones en función de la similitud de cada categoría con sus ancestros y con sus hermanos, y los resultados indican que el mejor rendimiento se obtiene en el caso en el que cada nodo presentaba una cierta similitud con sus ancestros, independientemente de los nodos hermanos.

En la Figura 5-2 se muestran las gráficas de las probabilidades de falsos aciertos obtenidas según la similitud con sus ancestros. En caso de que las firmas consten de dos bits en común con sus ancestros, el número de falsos aciertos en las consultas de primer nivel se reduce sensiblemente a más de la mitad, mientras que se produce un ligero repunte en el resto de niveles, frente al caso que considera un único bit en común. En cualquier caso, el rendimiento que ofrecen cualquiera de las dos alternativas es similar, aunque para el diseño se ha seleccionado el modelo con dos bits en común.

En resumen, en la arquitectura híbrida con información total se generarán firmas de 52 bits de longitud, de los cuales 3 bits estarán a uno en cada firma, siendo generadas de tal manera que haya dos bits en común entre la firma de un nodo y la de sus ancestros.

5.2.2. Estructuras de datos

En base a los parámetros de diseño estimados, a continuación se definen las estructuras de datos necesarias para la implementación del modelo híbrido de fichero invertido y fichero de firmas, con especial detalle en el espacio de almacenamiento requerido en cada caso.

En primer lugar, cada categoría estará identificada unívocamente a través de su firma asociada, con un tamaño de 52 bits. La firma de la categoría podría constituir el identificador de categoría utilizado en las diversas listas invertidas como puntero hacia la

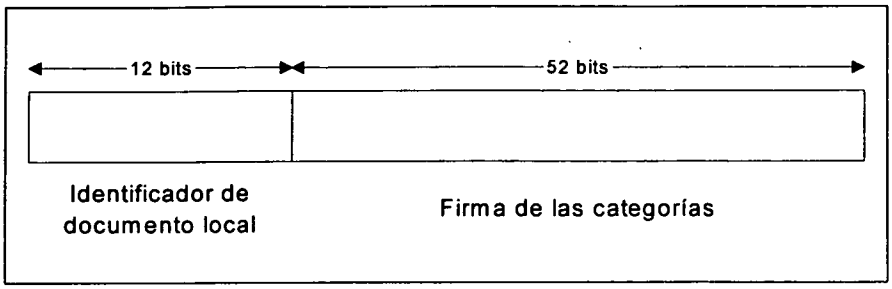

Figura 5-3: Identificador de documento compuesto para la implementación de la arquitectura híbrida con información total

estructura correspondiente. Esto plantea el inconveniente de un aumento en el espacio de almacenamiento necesario, y facilitaría la realización de filtrados en base a listas de categorías. Sin embargo, como los filtrados se centran única y exclusivamente en las listas de documentos no se requiere esta utilidad, por lo que es conveniente la utilización de un identificador de categoría convencional, con el consiguiente ahorro en espacio de almacenamiento.

Por lo tanto, asociado a cada categoría (en el fichero de categorías) se ubicarán dos campos de 52 bits de longitud que contendrán la firma de esa categoría y el código genético correspondiente. Conviene destacar que el código genético podría calcularse dinámicamente a partir de las firmas de cada categoría con sencillas operaciones OR. No obstante, el recorrido del árbol penalizaría estos cálculos, por lo que, teniendo en cuenta el reducido espacio necesario para el almacenamiento de todos los códigos genéticos, es aconsejable el almacenamiento de la información genética de cada categoría.

Por otra parte, el identificador de documentos en base al modelo propuesto se ha considerado como un valor de 64 bits, estructurados tal y como se indica en la Figura 5-3.

De esta manera se asignan 12 bits para los identificadores de documentos locales, lo que permite un máximo de 4.096 documentos por cada firma de categorías diferente. Es conveniente destacar el concepto asociado con el campo de firma de categorías, ya que éste establecerá un límite para el número máximo de documentos aceptados en el directorio Web.

El campo de firma de categorías estará formado por la superposición de todas las categorías a las que se asocie un documento directa o indirectamente. Por ejemplo, en el caso de que un documento esté asociado con una única categoría denominada *C*, este campo almacenará la superposición de los códigos de todas las categorías desde *C* hasta la raíz, esto es, el código genético de *C*. Y todos aquellos documentos que pertenezcan únicamente a la categoría *C* compartirán el valor del campo firma de categorías y se diferenciarán a través del identificador de documento local. En cambio, si un documento está asociado a dos categorías, el campo de firma se corresponderá con la superposición de ambos códigos genéticos.

Es decir, el campo de firma de categorías agrupa a aquellos documentos asociados con las mismas categorías, diferenciándose a través del identificador de documento local. En este punto, según la probabilidad de falsos aciertos, podría causar que para un número reducido de documentos las firmas de categorías coincidiesen considerándose del mismo grupo, aunque esto no supone ningún problema para el sistema en su conjunto.

Por lo tanto, en base a los tamaños de los campos se puede obtener un valor aproximado del número total de documentos que aceptará el sistema. Teniendo en cuenta el número máximo de firmas que se podrían generar, y asumiendo que aproximadamente cada firma diferente generará un código genético diferente, el sistema soportará 4.096 documentos por categoría, lo que ofrece un total de más de 90 millones de documentos. Esta cifra constituye un valor más que considerable para cualquier directorio Web. Además, este tipo de sistemas suele caracterizarse por disponer de un grafo de categorías balanceado, por lo que la limitación máxima por categoría difícilmente será superada, y en caso de serlo bastaría con realizar una división del contenido en varios nodos.

Respecto al aumento en el espacio de almacenamiento requerido por las diferentes estructuras de datos, este aumento se encuentra concentrado básicamente en el índice invertido de palabras-documentos. Típicamente, los identificadores de documentos se representarán empleando valores numéricos de un máximo de 32 bits, por lo que el tamaño de dicho índice prácticamente se verá duplicado. Esto presenta connotaciones a la hora de la realización de búsquedas normales, y en consecuencia en las búsquedas restringidas, ya que un parámetro importante de los tiempos de respuesta se deriva de los tiempos de lectura de las listas invertidas de dicho índice.

También se producirá un aumento en el espacio de almacenamiento del índice que relaciona las categorías con los documentos asociados, aunque en este caso el tamaño del índice es muy reducido por lo que el aumento no repercute especialmente en los tiempos de navegación de los usuarios.

Simplemente comentar la posibilidad de mantener dicho índice en su tamaño original mediante la utilización de identificadores dobles (tradicionales y compuestos) para cada documento, lo que implica el aumento de tamaño en el fichero de documentos. En este caso, el aumento de espacio sería más reducido, si bien la complejidad incorporada para la obtención de los ficheros de firmas dinámicamente para cada consulta sería considerablemente mayor, por lo que ha sido descartado.

Por último, para una recuperación eficiente de las búsquedas restringidas a categorías es necesario la utilización de un índice invertido adicional, para facilitar la recuperación de las categorías asociadas con cada documento. Este índice simplemente contiene la misma información que la correspondiente que permite recuperar los documentos asociados con una categoría.

En cambio, el tamaño de este índice no se ve influido por el aumento de tamaño de los identificadores de documentos, ya que este índice invertido únicamente contiene identificadores de categorías, que como se ha comentado anteriormente, pueden mantener perfectamente los tamaños y valores del modelo básico.

A modo de resumen, se detallan las repercusiones que presenta la implementación de la arquitectura híbrida con información total sobre las estructuras de datos del modelo básico, con especial hincapié en los aumentos de tamaño de almacenamiento requerido para el prototipo de arquitectura híbrida con información total. En la Figura 5-4 se destacan con línea punteada las estructuras que requieren algún tipo de modificación.

En primer lugar, cada categoría requiere dos campos adicionales de 52 bits para almacenar su firma y su código genético, que será almacenados en el fichero de categorías (ver Figura 5-4(a)). Esto es totalmente transparente para los índices asociados, ya que se conservan los identificadores básicos. Esto implica un aumento de 104 bits (13 bytes) por categoría, y provoca un aumento total ligeramente superior a 10 kilobytes en el fichero de categorías. Además, los accesos a esta información simplemente repercuten en el rendimiento de las búsquedas restringidas, sin mayores implicaciones.

En cambio, el aumento de tamaño de los identificadores de documentos provoca un aumento de tamaño generalizado en varias estructuras de datos. Por una parte, considerando que los documentos empleaban identificadores de 32 bits, este tamaño debe ser duplicado para dar cabida a los 64 bits de los identificadores compuestos. Esto provoca un aumento de unos 200 kilobytes en el propio fichero de documentos (ver Figura 5-4(b)).

Respecto al índice invertido que relaciona categorías con documentos, básicamente se

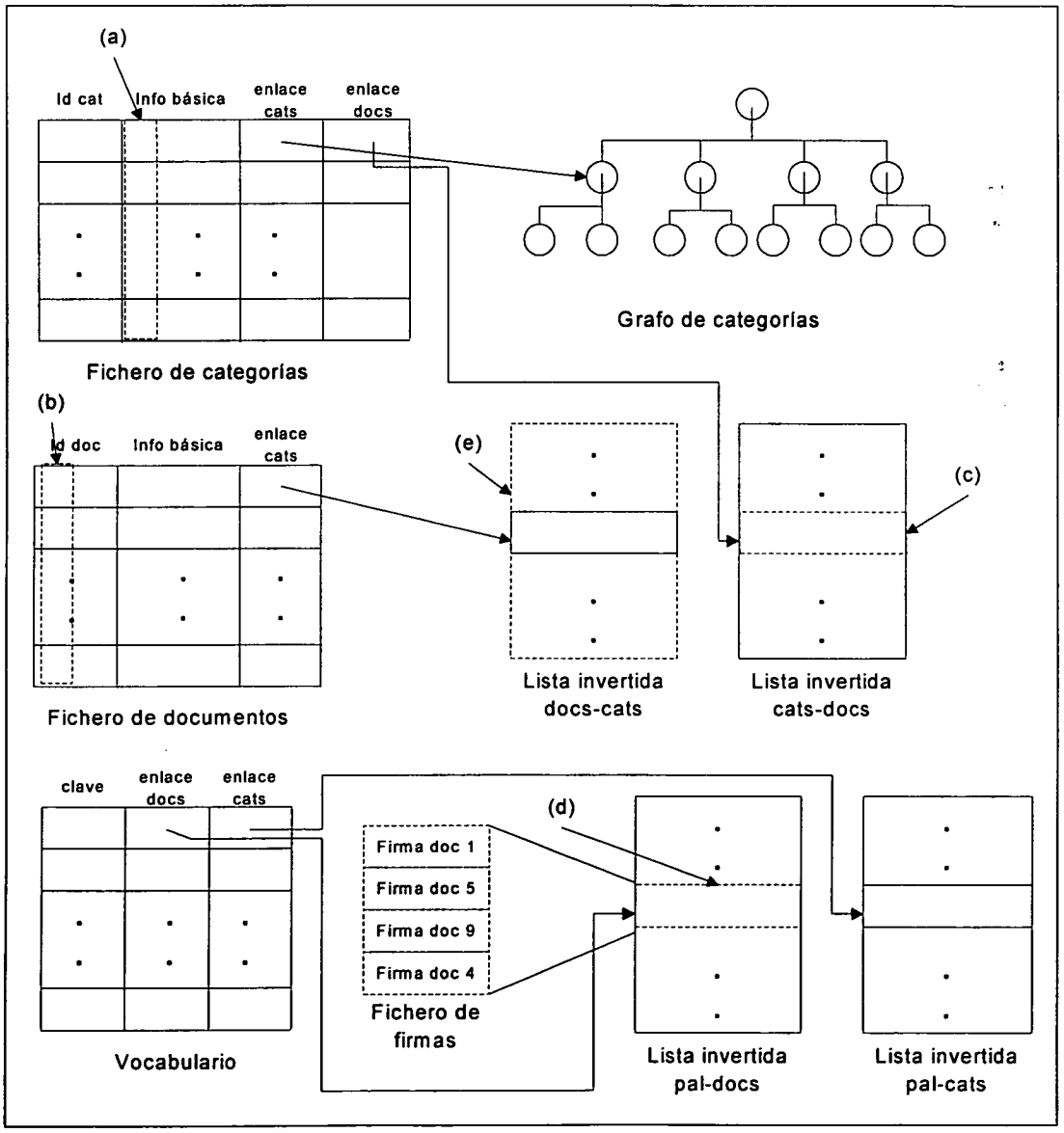

Figura 5-4: Estructuras de datos correspondientes a la implementación de la arquitectura híbrida con información total

produce una duplicación del espacio de almacenamiento (ver Figura 5-4(c)), si bien el espacio requerido por este fichero invertido no es demasiado elevado, teniendo en cuenta que cada categoría consta de unos 77 documentos de media. Por lo tanto, el tamaño del fichero invertido completo pasaría de aproximadamente unos 275 kilobytes a unos 550 kilobytes. Este índice invertido es utilizado únicamente en las operaciones de navegación del usuario, sin embargo, dada las reducidas dimensiones de las listas invertidas la repercusión en el rendimiento ofrecido será mínima.

Por el contrario, la repercusión en el índice invertido que relaciona palabras y documentos es más drástica (ver Figura 5-4(d)). Este índice se caracteriza por ser de grandes dimensiones al haber indexado los documentos del directorio Web. De hecho, el vocabulario está constituido por aproximadamente unas 250.000 palabras clave diferentes, cada una de ella asociada con 6 documentos diferentes, lo que genera un total de 1.500.000 identificadores de documentos empleados en este índice. Esto implica que dicho fichero invertido pasará de unos 6 megabytes, al doble, aproximadamente unos 12 megabytes.

A este respecto, el principal problema radica en el hecho de que este índice constituye la base para el proceso de búsqueda, por lo que un aumento de tamaño de las listas invertidas implica un aumento de los tiempos de lectura y en consecuencia un aumento en los tiempos de respuesta. En la sección 5.4.2 se determinará el alcance del aumento del tiempo de respuesta y si es aceptable por el sistema de recuperación de información.

Por último, la estructura de fichero invertido que asocia cada documento con las categorías asociadas produce un aporte de espacio de almacenamiento reducido, pero que en el modelo básico no tiene porque estar presente (ver Figura 5-4(e)). En concreto, esta estructura es equivalente al fichero invertido que relaciona categorías con documentos, con un tamaño de aproximadamente unos 275 kilobytes, derivado del hecho de que cada documento está asociado con una media de 1,34 categorías (se asume un identificador de categoría de 32 bits, aunque podría incluso ser de un tamaño inferior).

En resumen, el principal aumento de tamaño se localiza en el índice invertido que asocia cada palabra clave con sus documentos y que podría repercutir negativamente en los tiempos de respuesta. Por otra parte, el aumento en el resto de estructuras de datos es mínimo.

5.3. Implementación arquitectura híbrida con información parcial

En esta sección se describen los detalles de implementación asociados con la variante de la arquitectura propuesta que asigna firmas únicamente a las categorías de los primeros niveles del grafo.

Al igual que en el caso anterior, se describen los valores óptimos de los parámetros de diseño y su repercusión en las estructuras de datos.

5.3.1. Parámetros de diseño

La primera característica a establecer en este tipo de sistemas es el número concreto de niveles del grafo de categorías a los que se les asignarán firmas. Para la determinación de este valor es fundamental tener en cuenta que aquellas consultas restringidas a las categorías inferiores sufrirán una cierta penalización temporal, al no disponer de una firma y código genético propios.

El estudio estadístico expuesto en el capítulo 2 proporciona información trascendental para la determinación de los niveles consultados por los usuarios. En concreto, en la sección 2.4.1 se detalla que un 81% de las consultas están restringidas a categorías de los tres primeros niveles del grafo. Por lo tanto, considerando únicamente los tres primeros niveles del grafo (del nivel 1 al 3, teniendo en cuenta que la raíz se vincula con el nivel 0), se tendrán cubiertas la mayoría de las consultas restringidas. En estos tres niveles están incluidas aproximadamente 500 categorías, frente al total de 888 del caso anterior.

En base a este valor, se deben establecer los valores adecuados para el resto de parámetros del modelo. Teniendo en cuenta que el valor de D es de un cálculo complejo, al igual que en el modelo anterior, también se utilizará la técnica de la simulación para obtener una mejor aproximación a un valor aceptable de w. De esta manera, el número de superposiciones será establecido empíricamente según la propia estructura del grafo de categorías del directorio Web.

La principal ventaja de una arquitectura híbrida con información parcial es la reducción del tamaño de las firmas al haberse reducido el número de superposiciones. En concreto, en este caso, se plantea la utilización de firmas de 32 bits ($b=32$ bits). Inicialmente es necesario estimar la probabilidad de falsos aciertos obtenida para el valor óptimo de w, y determinar si se considera un valor aceptable. En caso contrario, bastaría con replantear el sistema aumentando el valor de b hasta obtener un valor de F_d adecuado.

Para determinar el valor óptimo de w se han realizado diferentes simulaciones, a partir de las cuales se han obtenido las probabilidades de falsos aciertos, tal y como se muestra en la Figura 5-5.

Al igual que sucedía en el modelo anterior, la probabilidad de falsos aciertos disminuye sensiblemente según el valor de w es menor. En concreto, para $w=3$ bits se obtiene el rendimiento óptimo en este caso, si bien, como se observa, la probabilidad de falsos aciertos es ligeramente superior a un 6%, lo cual, en comparación con el modelo híbrido con información total resulta un valor relativamente elevado.

Resaltar que con $w=2$ se podrían haber obtenido mejores valores, sin embargo el número de combinaciones posibles es muy reducido (en concreto, $C(32, 2)=496$), no siendo suficiente para el número total de categorías a indexar. En cambio, generando firmas con $w=3$ el número de combinaciones posibles aumenta hasta 4.960 ($C(32, 3)=4.960$).

Respecto a la Figura 5-5 destacar como en las categorías inferiores (niveles 4, 5 y 6) se produce un ligero repunte de la probabilidad de falsos aciertos. Esto es característico del modelo con información parcial, ya que únicamente las categorías de los primeros niveles disponen de firmas, por lo que pueden ser diferenciadas entre sí y por lo tanto la probabilidad de falsos aciertos disminuye según la pauta normal. En cambio, en los niveles

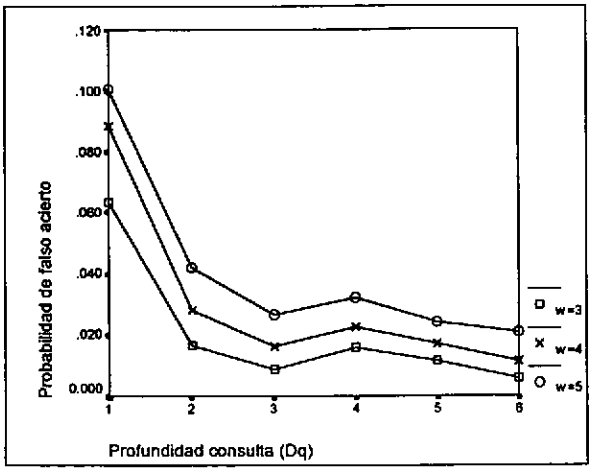

Figura 5-5: Probabilidad de falso acierto para diferentes valores de *w*, con *b=32* bits.

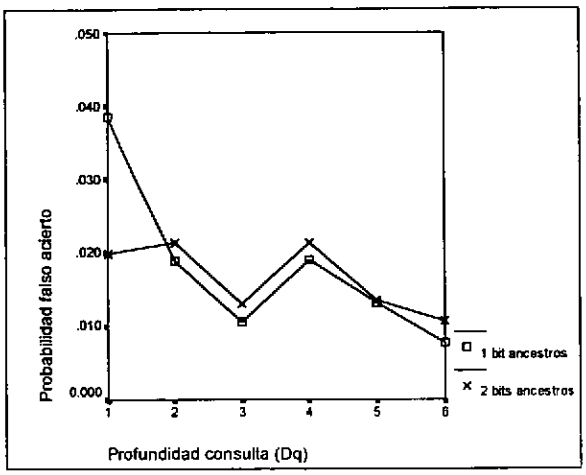

Figura 5-6: Probabilidad de falso acierto para *b=32* y *w=3* con varios bits en común con sus ancestros

inferiores aquellas categorías con los mismos padres presentarán el mismo código genético (al carecer de firma) y por lo tanto es imposible diferenciarlas. Lo que se consigue es que, a parte de los fallos inherentes al modelo de códigos de superposición, las categorías hermanas también producirán falsos aciertos entre sí.

Por otra parte, considerando *w=3* bits se han estudiado diferentes alternativas de similitud de los códigos generados de una categorías con sus ancestros y con sus nodos hermanos. En la Figura 5-6 se muestran los mejores resultados obtenidos, en concreto generando los códigos en función de los códigos de sus ancestros. En cambio, la similitud con respecto a los códigos de hermanos no aportaba ninguna mejoría, probablemente porque cada firma dispone de un peso muy reducido.

Por una parte, considerando que únicamente un bit (de los tres bits a "1" de cada firma) se ubica en una posición correspondiente al código genético de sus ancestros se obtiene una mejora de la probabilidad de falsos aciertos, especialmente en el primer nivel. En cambio, considerando dos bits de similitud con sus ancestros, el rendimiento para el primer nivel es sensiblemente mejor, descendiendo $F_d(D_q=1)$ hasta un 2%, aunque se produce un aumento de la probabilidad de falsos aciertos en los niveles inferiores, especialmente en el segundo nivel.

De todas formas, para la implementación realizada se ha considerado más eficiente la primera alternativa, ya que aunque la probabilidad de falsos aciertos en el primer nivel es superior, se ve compensada por la disminución en el resto de niveles. Además, la asignación de firmas se realizará buscando un modelo óptimo, por lo que es de esperar una reducción en la probabilidad del primer nivel.

En resumen, la implementación de la arquitectura híbrida con información parcial se centrará en los tres primeros niveles del grafo de categorías. Para lo cual se generarán firmas de 32 bits de longitud estableciendo 3 bits a uno, teniendo en cuenta que uno de estos bits ocuparán posiciones de las firmas de los ancestros de la categoría.

5.3.2. Estructuras de datos

Las estructuras de datos empleadas en la implementación de la arquitectura híbrida con información parcial se basan en el modelo anterior, con unos ligeros cambios que afectan especialmente al espacio de almacenamiento del sistema. Por lo tanto, a continuación se describen los principales aspectos de cambio, sin entrar en detalles sobre los aspectos ya expuestos en la sección anterior.

Referente a las categorías del directorio, en el fichero de categorías se almacenarán dos campos de 32 bits que representarán la firma y el código genético de cada categoría. En este caso, la firma de cada categoría no podría ser empleada como identificador primario ya que las categorías pertenecientes a niveles inferiores carecen de dicha firma. En consecuencia, se hace necesario la utilización de un identificador de categorías independiente de los códigos de superposición, tal y como se ha expuesto en el modelo híbrido con información total.

Respecto al identificador de documentos, se ha estructurado en un campo de 46 bits como se muestra en la Figura 5-7, con una notable reducción del tamaño frente a los 64 bits del caso anterior.

El concepto de identificador de documento local se modifica ligeramente respecto al modelo con información total. En el caso anterior, este campo se empleaba para identificar cada documento dentro de cada categoría. En cambio, en este caso se agrupan conjuntamente todos aquellos documentos asociados, de manera indirecta probablemente, con una categoría de los primeros niveles (normalmente, el tercero). Por este motivo es necesario un aumento del campo de identificador de documento local hasta 14 bits, lo que permite un máximo de 16.384 documentos diferentes para un mismo código.

El campo de firma de categorías se sigue construyendo a partir de la superposición de los códigos genéticos de las categorías a las que se asocia cada documento, pero por ser un modelo con información parcial, únicamente se recogerán las categorías de los tres primeros niveles.

El número total de documentos aceptados por el sistema se mantiene en el orden del caso anterior. Asumiendo que existe un máximo de 4.960 firmas diferentes, y que asociado a cada firma se pueden incluir más de 16.000 documentos, un límite teórico para el número de documentos máximo se sitúa entorno a los 81 millones de documentos, límite más que suficiente para cualquier directorio Web operativo hoy en día.

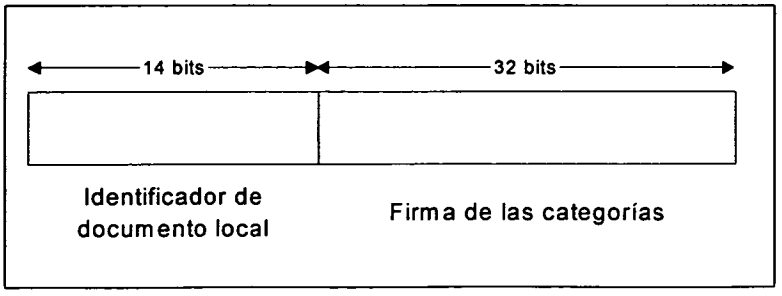

Figura 5-7: Identificador de documento compuesto para la implementación de arquitectura híbrida con información parcial

Respecto al número máximo de categorías del sistema, puntualizar que se podrían crear un máximo de 4.960 categorías en los tres primeros niveles (limitado por el número de firmas disponibles). Sin embargo, para los niveles inferiores no existe ningún tipo de limitación, al contrario de lo que sucedía en el modelo anterior. Por lo tanto, teóricamente, este modelo no impone limitación alguna respecto al número de categorías del directorio Web.

Las estructuras de datos empleadas en este modelo son exactamente las mismas que las descritas en la Figura 5-4 correspondientes a la arquitectura híbrida con información total. Simplemente se producirán alteraciones en el espacio de almacenamiento requerido por las nuevas estructuras, que se detallan a continuación.

En cuanto al incremento de tamaño producido en el fichero de categorías por el almacenamiento de las firmas y códigos genéticos de cada categoría (ver Figura 5-4(a)), se deben aumentar 64 bits a cada categoría, lo que implica un aumento total 56 kilobytes, aproximadamente un 50% inferior al modelo anterior.

En este caso los identificadores de documentos presentan un tamaño de 46 bits, lo que conlleva un aumento del fichero de documentos de aproximadamente 134 kilobytes (ver Figura 5-4(b)). El aumento producido en el índice que asocia categorías con documentos se sitúa en 126 kilobytes respecto al tamaño original (ver Figura 5-4(c)), siendo sensiblemente inferior al aumento producido por el modelo inferior.

Respecto a la estructura que más efecto presenta en el proceso de búsqueda, el índice que relaciona palabras clave con documentos, el aumento en este caso es más moderado ya que el tamaño aproximado de dicho índice en base a los nuevos identificadores de documentos se sitúa en 8,25 megabytes (ver Figura 5-4(d)).

Sobre el índice invertido que se muestra en la Figura 5-4(e) no se produce variación alguna en el tamaño total, siendo de aproximadamente 275 kilobytes. Este índice permanece invariable debido a que está basado en identificadores de categorías normales, independientes de los códigos de superposición.

En líneas generales, por medio de la arquitectura híbrida con información parcial se consigue una considerable reducción del espacio para el almacenamiento de las estructuras de datos asociadas a este modelo. El contrapunto lo presentan las consultas restringidas a categorías inferiores en las que la tasa de falsos aciertos es superior al modelo con información total, lo que puede repercutir en los tiempos de respuesta, aspecto que será concretado, medido y examinado en detalle en la siguiente sección.

5.4. Evaluación del rendimiento

En esta sección se realiza una evaluación de los tiempos de respuesta que ofrecen los dos modelos propuestos frente al modelo básico, especialmente ante consultas de recuperación de información restringidas a una zona del grafo de categorías de un directorio Web.

La evaluación del rendimiento se centra en un primer momento sobre las búsquedas no restringidas (realizadas sobre la totalidad de los documentos del directorio Web), con el objetivo de determinar si existe algún tipo de pérdida en el rendimiento ofrecido por los nuevos modelos frente al modelo básico ante las búsquedas normales.

Y a continuación, se expone la evaluación del rendimiento de las búsquedas restringidas a una zona del grafo, analizando las mejoras aportadas por cada uno de los modelos propuestos sobre el modelo básico.

Para una correcta evaluación del rendimiento se siguen unas pautas definidas que se detallan seguidamente. El objetivo consiste en realizar un análisis del comportamiento de este tipo de sistemas lo más exhaustivo posible.

5.4.1. Metodología

Los sistemas de recuperación de información en el Web se caracterizan por soportar unos niveles de carga variables a lo largo del tiempo, pasando de períodos de inactividad hasta situaciones de carga elevada o extrema. Es tarea de los administradores del sistema dimensionar correctamente los equipos disponibles para dar servicio a todos los usuarios en aquellas situaciones estimadas como de carga más elevada.

Obviamente, el rendimiento ofrecido por un sistema de estas características depende drásticamente de las condiciones de carga que soporte en cada momento. Por este motivo, una de las partes básicas de la evaluación del rendimiento de los diferentes modelos propuestos será la evaluación bajo diferentes supuestos de carga, utilizando para tal efecto la herramienta de simulación USim, cuya base se encuentra detalladamente expuesta en el capítulo 3.

Para la evaluación del rendimiento se considerarán cinco situaciones de carga diferentes: nula, baja, media, alta y saturación. Para el establecimiento de los diferentes puntos de carga se ha tenido en cuenta el estudio del punto de saturación de un prototipo del directorio Web expuesto en la sección 3.2.3.1. En dicho apartado, se realiza la medida del punto de saturación del modelo básico del prototipo de directorio Web estudiado, sobre el hardware previamente descrito.

La principal conclusión de dicho análisis sitúa el punto de saturación a partir de 21 búsquedas por minuto, con los correspondientes procesos de navegación por categorías y visitas a documentos asociados en los niveles correspondientes.

Por lo tanto, en base a este estudio previo, se calculan los valores de las diferentes situaciones de carga como:

- Nula: 0 búsquedas por minuto, y sin accesos a categorías ni documentos.
- Baja: 5 búsquedas por minuto, lo que implica 4,1 accesos a categorías por minuto y 7,5 documentos visitados por minuto.
- Media: 12 búsquedas por minuto, lo que implica 8,5 accesos a categorías por minuto y 16,2 documentos visitados por minuto.
- Alta: 19 búsquedas por minuto, lo que implica 12,9 accesos a categorías por minuto y 24,9 documentos visitados por minuto.
- Saturación: 23 búsquedas por minuto, lo que implica 15,4 accesos a categorías por minuto y 29,9 documentos visitados por minuto.

Si bien, los valores asignados dependen en gran medida de los equipos disponibles, los resultados y conclusiones obtenidos son totalmente generalizables para diferentes entornos ya que, aunque los puntos de carga puedan ser diferentes, las implicaciones sobre el sistema serán equiparables.

La evaluación del rendimiento consiste en simular un entorno real de utilización por medio de USim bajo los diferentes niveles de carga descritos. Inicialmente se establece un período de estabilización del sistema, a partir del cual se inician las medidas de los tiempos de respuesta ante consultas.

Por lo tanto, la evaluación consta de dos procesos. Por una parte USim se encarga de establecer una situación de carga real sobre el sistema de búsqueda. Y por otra parte, y de manera independiente, se lanzan consultas de distintos tipos al sistema midiendo los tiempos de respuesta obtenidos. En base a estos tiempos de respuesta se realizarán los análisis pertinentes para la correcta evaluación.

Las consultas enviadas se garantiza que constituyen un abanico suficientemente amplio de aquellos factores que representan (o pueden representar) una alteración sobre el tiempo de respuesta. El principal de estos factores está constituido por el número de resultados recuperados en la consulta (cuanto mayor es el número de resultados mayor es el tiempo de respuesta), y en el caso de las consultas restringidas, la profundidad de la categoría o el número de documentos asociados a dicha categoría.

5.4.2. Búsquedas no restringidas

En esta sección se determinará si los modelos híbridos propuestos para intentar mejorar el rendimiento de las consultas restringidas, tienen algún efecto sobre el rendimiento ofrecido por el proceso de búsqueda normal, no restringido.

Para ello se han analizado los tiempos de respuesta ante búsquedas normales (esto es, sobre todos los documentos del directorio Web) ofrecidos por cada modelo en cada una de las situaciones descritas en la sección anterior. El análisis se basa en un estudio del proceso de búsqueda de los tres modelos (básico, híbrido con información total e híbrido con información parcial) midiendo los tiempos de respuesta ante diferentes consultas que recuperan diferentes cantidades de resultados.

Para la determinación de la influencia de cada uno de los modelos se ha realizado un análisis ANOVA estudiando dos factores: el número de resultados en la consulta y el tipo de modelo en el que se ha realizado la consulta, frente al tiempo de respuesta obtenido. Obviamente, el número de resultados constituye un parámetro influyente y el objetivo es determinar hasta que punto es influyente el tipo de modelo.

A continuación se describen los resultados más reseñables correspondientes a las situaciones de carga baja, media y alta. Las situaciones de carga nula y en saturación aportan poca información por diferentes efectos derivados de la propia experimentación. Por una parte, en una situación de carga nula, las consultas realizadas presentan una gran dependencia entre sí (una consulta con tiempo de respuesta elevado también aumentará el tiempo de respuesta de las siguiente) por lo que los resultados del análisis no son aplicables. Y por otra parte, bajo una situación de saturación la variabilidad en los tiempos

de respuesta se debe mayormente a la propia saturación del sistema, no tanto a los diferentes modelos o consultas.

Los detalles respecto este análisis se encuentran en el Apéndice C, describiendo a continuación únicamente los aspectos más relevantes del mismo.

La principal conclusión de los diferentes análisis es que los tres modelos estudiados se comportan de la misma manera ante una consulta estándar, independientemente del número de resultados que vaya a recuperar. En la Figura 5-8 se observan los resultados obtenidos en el caso del análisis realizado bajo una situación de carga baja. En primer lugar, se observa una tendencia ascendente en los tiempos de respuesta de los tres modelos según aumenta el número de resultados obtenidos tras la consulta. Esto era de esperar ya que se deriva del hecho de que el tiempo del proceso de ordenación depende directamente del número de resultados a ordenar. De hecho, en todos los análisis realizados se acepta que el número de resultados es un parámetro influyente con una probabilidad del 99,9%.

Como se observa en la Figura 5-8 el rendimiento obtenido es similar para los tres modelos. Se puede comprobar la existencia de ligeras fluctuaciones en los tiempos de respuesta en algunos puntos, sin embargo no son estadísticamente significativos, ya que a través del análisis de la varianza se acepta que los tres modelos son equivalentes con una probabilidad del 84%. Además, el test ANOVA refleja que no existe interacción entre ambos factores (lo que implicaría que un modelo se comportaría mejor o peor ante consultas con un determinado número de resultados) con una probabilidad del 82,2%.

Bajo una situación de carga media los resultados son similares. Como se observa en la Figura 5-9 se mantiene la tendencia ascendente respecto al número de resultados, y a pesar de ligeras fluctuaciones puntuales, el test ANOVA indica que con un 75% de probabilidad los tres modelos se comportan de manera similar y que no se puede asumir que exista interacción entre los dos factores (con un 65,5% de probabilidad).

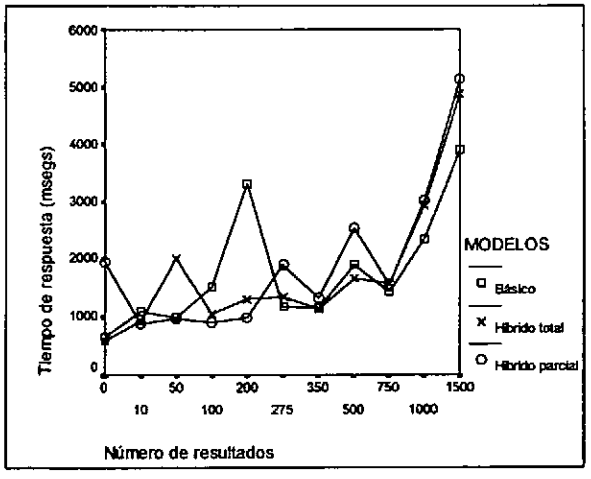

Figura 5-8: Tiempos de respuesta estimados en función del número de resultados en consultas no restringidas (carga baja)

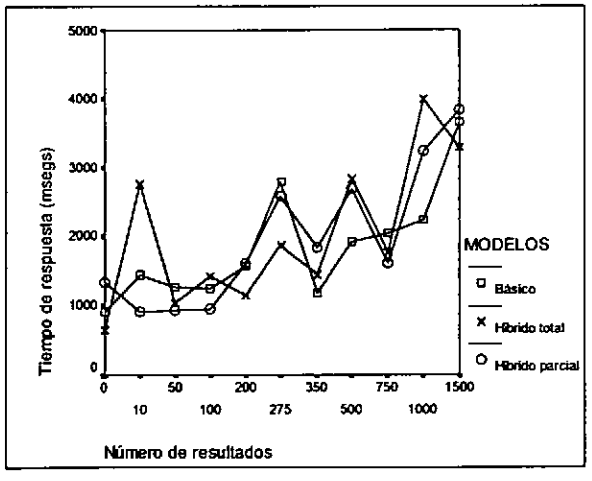

Figura 5-9: Tiempos de respuesta estimados en función del número de resultados en consultas no restringidas (carga media)

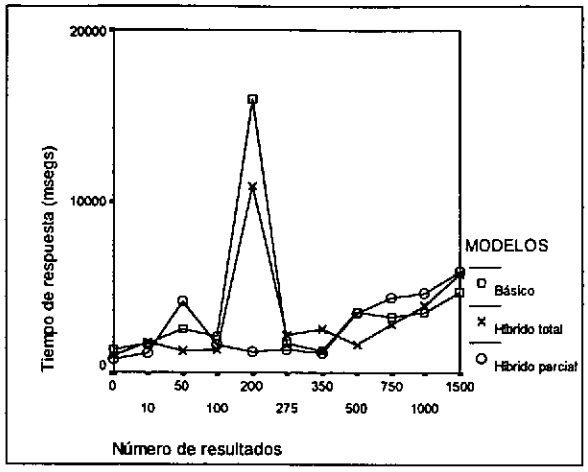

Figura 5-10: Tiempos de respuesta estimados en función del número de resultados en consultas no restringidas (carga alta)

Los diferentes modelos se comportan de manera similar en situaciones de carga elevada (ver Figura 5-10). En este caso, el análisis de la varianza permite aceptar que el tipo de modelo no es un factor influyente en los tiempos de respuesta de consultas no restringidas, con una probabilidad del 75%. Igualmente, tampoco se considera que exista interacción entre ambos factores, con una probabilidad del 85,3%.

La principal conclusión derivada de esta primera evaluación del rendimiento de los modelos híbridos propuestos, frente al modelo básico, permite afirmar que los tres modelos ofrecen tiempos de respuesta similares ante consultas no restringidas. Esto garantiza que los modelos híbridos con información total y parcial no afectan negativamente a las búsquedas no restringidas, aspecto fundamental de cualquier mejora sobre el modelo básico.

5.4.3. Búsquedas restringidas

En esta sección se describe la última fase en la evaluación del rendimiento de los modelos híbridos con información total y parcial, frente al modelo básico, ante consultas restringidas a una zona del grafo de categorías de un directorio Web.

La evaluación de los tiempos de respuesta obtenidos se divide en dos partes. En primer lugar se analizan los tiempos de respuesta de los tres modelos ante consultas restringidas a categorías de los tres primeros niveles, para determinar las mejoras aportadas por cada uno de los modelos propuestos en las consultas más comunes. Y en segundo lugar, se examinan los tiempos de respuesta para categorías más profundas con el objetivo de determinar si alguno de los modelos (especialmente el modelo híbrido con información parcial) presenta un empeoramiento en los tiempos de respuesta en este caso.

En cualquier caso, tal y como se ha descrito en la sección 5.4.1, los tiempos de respuesta serán medidos en diversas situaciones de carga del sistema. En el Apéndice C se describen de manera detallada los diferentes análisis realizados en cada caso, ya que a continuación únicamente se hace referencia a los aspectos más relevantes.

En las primeras pruebas realizadas, se han ejecutado en cada uno de los prototipos una serie de consultas restringidas a categorías de los niveles 1, 2 y 3, midiendo el tiempo de respuesta bajo diferentes situaciones de carga.

La primera de las situaciones simulada considera al sistema con carga nula (sin usuarios conectados). En la Figura 5-11 se muestran los tiempos medios de respuesta estimados para los tres modelos en función del número de resultados que recupera la consulta, mientras que la Figura 5-12 muestra los tiempos de respuesta estimados para los tres modelos en función del número de documentos asociados (directa e indirectamente) a la categoría sobre la que se ha restringido la búsqueda.

Destacar que el número de documentos asociados con una categoría es inversamente proporcional a la profundidad a la que se encuentra (cuanto menor sea la profundidad de la categoría, mayor será el número de documentos asociados). Sin embargo, como no todas las categorías están igualmente distribuidas se considera más exacto el estudio sobre el número de documentos asociados.

En la Figura 5-12 se observa el aumento del tiempo de respuesta según aumenta el número de documentos asociados a la categoría de restricción (o lo que es lo mismo, según la categoría pertenece a niveles superiores del grafo), y especialmente en el caso del modelo básico, frente a la destacada mejoría del rendimiento de los modelos híbridos propuestos.

Sin embargo, es en la Figura 5-11 en donde se puede observar más detalladamente el comportamiento de los diferentes modelos. Inicialmente, cuando el número de resultados de la consulta no restringida es reducido, los tres modelos se comportan igual. En cambio, según aumenta el número de resultados, los modelos híbridos presentan un mejor comportamiento, especialmente a partir de consultas que obtienen 500 resultados en donde los algoritmos propuestos mejoran el rendimiento en un 50%, reduciendo a la mitad los tiempos de respuesta.

Por medio del análisis de la varianza realizado se confirma la importancia del tipo de modelo como un factor influyente (junto con el número de resultados de la consulta no restringida y el número de documentos asociados a la categoría de restricción) con un

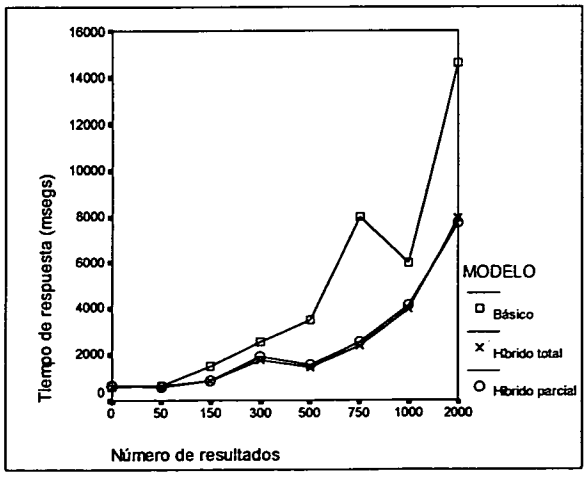

Figura 5-11: Tiempos de respuesta estimados en función del número de resultados para los tres modelos (carga nula)

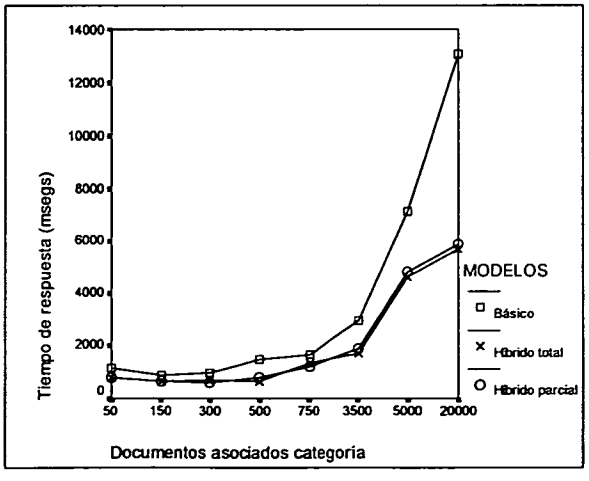

Figura 5-12: Tiempos de respuesta estimados en función del número de documentos asociados a la categoría (carga nula)

99,9% de probabilidad.

Además, como parte del estudio se ha aplicado el test ANOVA únicamente sobre los datos referentes a los modelos híbridos propuestos para comprobar si ambos modelos se comportan de manera similar. En concreto, en este caso, tal y como era de esperar, se confirma que ambos modelos propuestos son equivalentes y no afectan al tiempo de respuesta (entre sí), bajo una situación de carga nula.

Sin embargo, una situación de carga nula representa unas condiciones ideales. Por lo tanto, tomando como base el experimento anterior se repite el proceso sometiendo al sistema a una carga baja de usuarios. Los resultados se muestran en la Figura 5-13 y en la Figura 5-14. Como se ha indicado anteriormente, en el Apéndice C se exponen de manera exhaustiva todos los gráficos y tests ANOVA asociados a cada estudio.

El aspecto más relevante respecto a una situación de carga nula se centra en un ligero aumento del tiempo de respuesta a nivel general para los tres modelos producido, obviamente, por el aumento de la carga del sistema.

Por otra parte, las conclusiones derivadas del caso anterior son igualmente válidas. Se mantiene la mejoría de los modelos híbridos frente al modelo básico (con una mejora aproximada del 50%), y en el análisis de la varianza realizado se mantiene la similitud entre los dos modelos propuestos frente al modelo básico.

Ante una situación de carga media las respuestas de los modelos varían ligeramente. Obviamente, se produce un aumento en el tiempo medio de respuesta y el modelo básico continúa ofreciendo un peor rendimiento en líneas generales, si bien, surgen algunas diferencias entre los dos modelos propuestos, aunque no se pueden considerar estadísticamente significativas.

En la Figura 5-15 y la Figura 5-16 se muestran los tiempos de respuesta en función del número de resultados de la consulta y del número de resultados asociados a la categoría restringida. El análisis de la varianza realizado sobre los tiempos obtenidos y los diferentes factores analizados prueba que el tipo de modelo es un factor influyente, al igual que en los

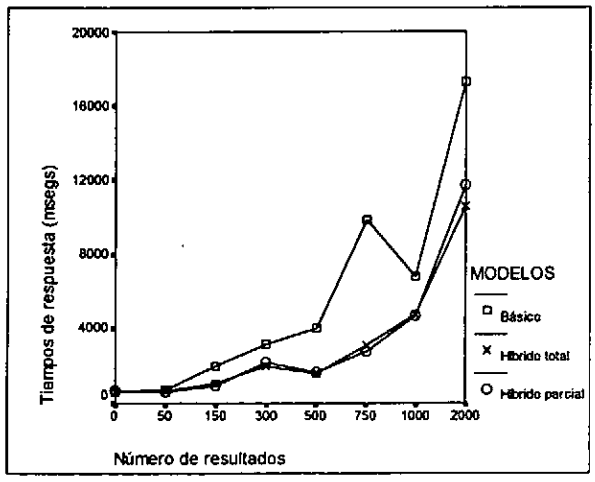

Figura 5-13: Tiempos de respuesta estimados en función del número de resultados para los tres modelos (carga baja)

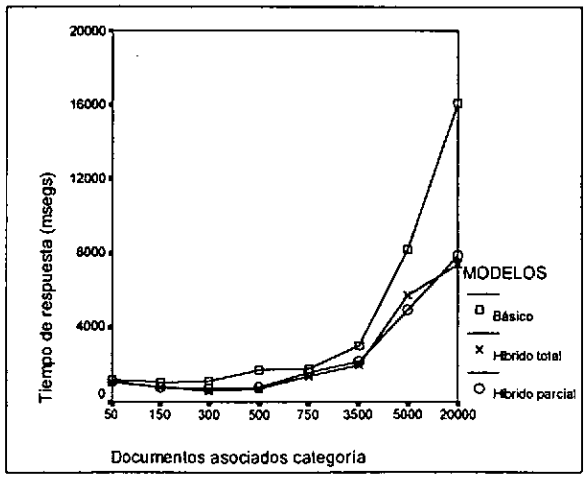

Figura 5-14: Tiempos de respuesta estimados en función del número de documentos asociados a la categoría (carga baja)

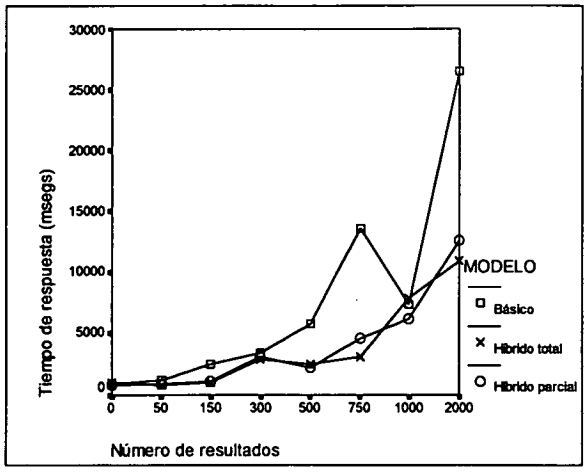

Figura 5-15: Tiempos de respuesta estimados en función del número de resultados para los tres modelos (carga media)

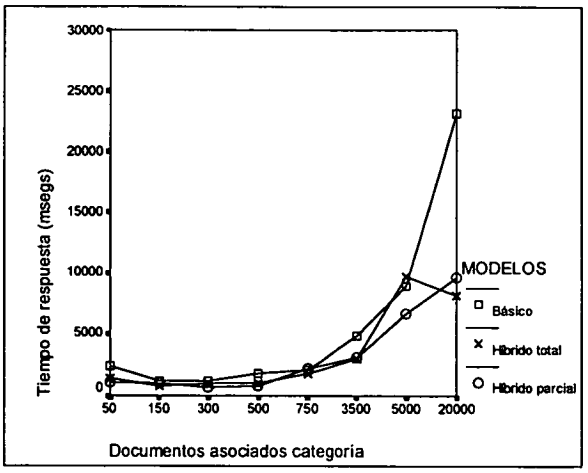

Figura 5-16: Tiempos de respuesta estimados en función del número de documentos asociados a la categoría (carga media)

casos anteriores, lo que mantiene a los modelos híbridos por encima del modelo básico. En el segundo test ANOVA realizado, centrado únicamente en los modelos propuestos, se debe aceptar que ambos modelos se comportan de la misma manera, aunque con un nivel de significación del 47%, sensiblemente inferior a los casos anteriores.

Esto sugiere la existencia de diferencias entre los dos modelos propuestos. Estas ligeras diferencias se presentan debido a las diferencias de implementación de ambos modelos que afectan básicamente al espacio de almacenamiento, y que repercuten directamente en los tiempos de respuesta. Será en las situaciones de mayor carga en donde se definan de manera más clara las diferencias entre los modelos propuestos. En la Figura 5-15 y la Figura 5-16 se observa que las diferencias de comportamiento se localizan especialmente en los intervalos superiores.

Por este motivo, es especialmente interesante el análisis realizado bajo una situación de carga elevada en donde se pueden observar claramente las diferencias entre los tres modelos, tal y como se muestran en la Figura 5-17 y en la Figura 5-18. En este caso, el incremento de los tiempos de respuesta entre la situación de carga media y alta es muy reducido, especialmente teniendo en cuenta que el incremento de búsquedas por minuto entre los diferentes niveles es equivalente. En cambio, las diferencias de comportamiento entre los tres modelos, y substancialmente entre los dos modelos híbridos, son patentes.

El análisis de la varianza confirma que los modelos influyen en el tiempo de respuesta ante las consultas restringidas a una subcategoría, aunque los resultados más interesantes se obtienen al analizar únicamente los dos modelos propuestos. Al contrario de lo que sucedía en los casos anteriores, el test ANOVA confirma que los modelos híbridos con información total y parcial se comportan de manera diferente.

Esto es patente en la Figura 5-17, en donde se observa como los tiempos de respuesta del modelo híbrido con información total son equiparables al modelo básico, mientras que el modelo híbrido con información parcial mantiene una mejoría en los tiempos de respuesta del 50%.

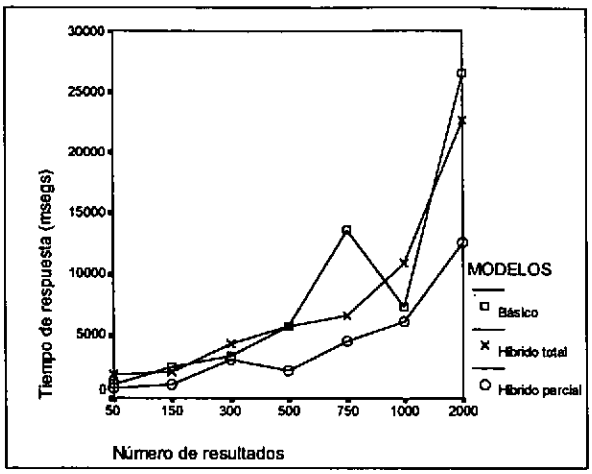

Figura 5-17: Tiempos de respuesta estimados en función del número de resultados para los tres modelos (carga alta)

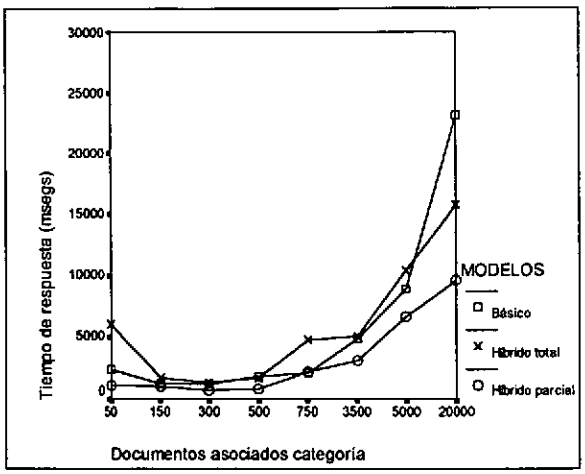

Figura 5-18: Tiempos de respuesta estimados en función del número de documentos asociados a la categoría (carga alta)

Las causas del mejor funcionamiento del modelo híbrido con información parcial bajo situaciones de carga elevada se basan en el menor tamaño del índice híbrido empleado por este sistema. Una situación de carga elevada implica un elevado número de búsquedas que acceden al índice y por lo tanto un mayor acceso al disco para las operaciones de lectura.

En el caso del modelo híbrido con información total, tal y como se describe en los detalles de implementación, el tamaño del índice duplica al tamaño básico. Esto implica que el proceso de búsqueda requiere más operaciones de lectura de disco para leer la misma información, lo que conlleva un mayor tiempo de procesamiento de la búsqueda, que afecta directamente a los tiempos de respuesta de las búsquedas restringidas. En cambio, el modelo híbrido con información parcial, aunque presentaba un aumento del tamaño del índice, no presentaba las proporciones del modelo con información total (aproximadamente, presenta un aumento del tamaño del 35%).

En ambos casos, la aplicación del filtrado inexacto asociado a los ficheros de firmas permite reducir los tiempos de respuesta ante las consultas restringidas. Sin embargo, en el caso del modelo híbrido con información total el tiempo requerido por el proceso de búsqueda minimiza el efecto positivo del filtrado inexacto. En cambio, en el caso del modelo híbrido con información parcial, el tiempo de búsqueda es menor (al requerir un menor número de accesos a disco), junto con los beneficios del filtrado inexacto, lo que permite mantener la mejoría de los tiempos de respuesta en un 50% respecto al modelo básico.

Respecto a una situación de saturación, los experimentos no permitieron aportar información concluyente. Esto es debido a que bajo una situación de saturación, los tiempos de respuesta que ofrece el sistema de búsqueda se encuentran muy degradados y varían drásticamente en función de las búsquedas realizadas. Esto se ve confirmado con un coeficiente de correlación del 53% obtenido durante el análisis ANOVA, lo que indica que los factores propuestos únicamente explican la mitad de la variabilidad de los datos, mientras que el resto se debe ajustar a causas fuera del análisis realizado. En cualquier caso, en el Apéndice C se describe el análisis realizado en una situación de saturación de los prototipos de directorio Web implementados.

Hasta este punto se ha demostrado que el modelo híbrido con información parcial presenta un mejor rendimiento que el modelo básico en todas las situaciones posibles, e incluso mejora al modelo híbrido con información total en situaciones de carga elevada. Sin embargo, tal y como se describe en los detalles de implementación, el modelo híbrido con información parcial, al no asignar firmas a los niveles inferiores del grafo, puede estar penalizando las búsquedas restringidas a los nodos inferiores.

Por este motivo, se ha analizado la repercusión de los diferentes modelos en función del número de resultados de la consulta y de la profundidad de la categoría a la que se restringe la búsqueda. Destacar que existe una relación directa entre profundidad y número de documentos asociados a dicha categoría, sin embargo, para este análisis en concreto es conveniente la determinación de la profundidad ya que esta repercute en el número de superposiciones de los códigos, especialmente en el caso del modelo híbrido con información parcial.

En este caso, se ha realizado un análisis de la varianza sobre los tiempos de respuesta en función de tres factores: número de resultados de la consulta, profundidad de la categoría y el tipo de modelo (básico, híbrido con información total e híbrido con información parcial). Al igual que en el caso anterior, los procesos se han repetido en diferentes tipos de situaciones de carga. El primer factor ha sido restringido únicamente a búsquedas con más de 750 resultados, ya que es a partir de este punto en donde se observan las mayores diferencias de rendimiento entre los tres modelos.

El modelo híbrido con información parcial bajo situaciones de carga nula o baja (ver Figura 5-19 y Figura 5-20, respectivamente) presenta un comportamiento similar al modelo híbrido con información total para todos los niveles de profundidad. La aplicación del análisis de la varianza confirma el hecho de que los tres factores (número de resultados, profundidad y modelo) tienen una influencia directa en el tiempo de respuesta, ya que como era de esperar, el modelo básico presenta un rendimiento inferior a los dos modelos propuestos.

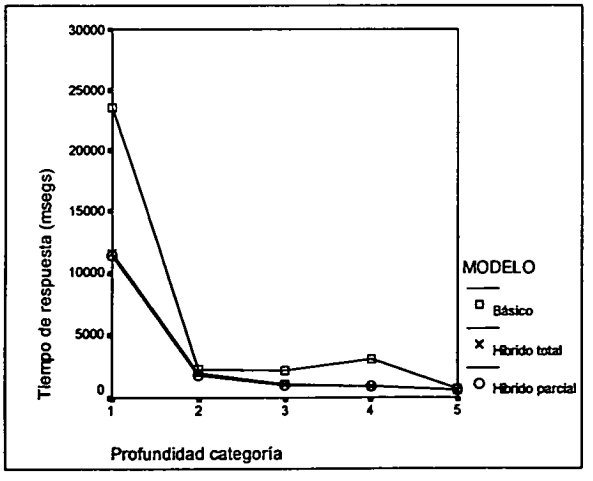

Figura 5-19: Tiempos de respuesta estimados en función de la profundidad de la categoría restringida (carga nula)

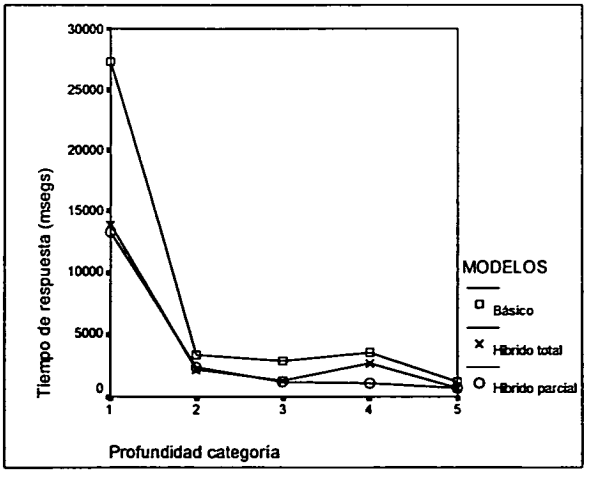

Figura 5-20: Tiempos de respuesta estimados en función de la profundidad de la categoría restringida (carga baja)

Bajo una situación de carga media del sistema (ver Figura 5-21) el análisis de la varianza permite observar claramente como el modelo básico ofrece un rendimiento inferior a los demás. Aunque también se puede observar un ligero empeoramiento del modelo híbrido con información parcial en los niveles inferiores, especialmente en las categorías de nivel 5.

En cambio, bajo una situación de carga elevada (ver Figura 5-22), el modelo híbrido con información parcial mejora el rendimiento de los otros dos modelos, en especial en los primeros niveles, pero también en niveles inferiores. En esta situación, el tercer modelo no sufre un empeoramiento en los niveles inferiores, debido a que aunque se produce un aumento en la probabilidad de falsos aciertos en los niveles inferiores, esta tasa queda estabilizada entorno a los valores obtenidos para los primeros niveles. De esta manera, aunque se estén produciendo un mayor número de falsos aciertos en el modelo híbrido con información parcial, estos quedan compensados por el menor tiempo de lectura necesario para recuperar de disco los ficheros de firmas embebidos en el fichero invertido.

Respecto a una situación de saturación del sistema, al igual que en el caso anterior, el test ANOVA explica un porcentaje poco significativo de la variabilidad de los tiempos de respuesta. En el Apéndice C se exponen todos los tests ANOVA realizados para el análisis de la repercusión de la profundidad de la búsqueda en los diferentes modelos propuestos.

A nivel general se puede concluir que el modelo híbrido con información parcial no presenta un empeoramiento del rendimiento para los niveles inferiores del grafo. Esto es debido a que con este modelo la probabilidad de falsos aciertos toma valores homogéneos para todos los niveles (alrededor del 1%). De esta manera, el aumento en la tasa de falsos aciertos es compensada por la reducción en el espacio de almacenamiento. Además, la tasa obtenida es un valor adecuado para las prestaciones requeridas por el sistema.

En cambio, en el caso del modelo híbrido con información total, la probabilidad de falsos aciertos en los niveles superiores toma valores cercanos al 1%, mientras que en los niveles inferiores es tres o cuatro órdenes de magnitud inferior, a costa de un elevado espacio de almacenamiento para la estructura de datos híbrida. La falta de equilibrio entre los distintos niveles podría ser aprovechada mediante la eliminación del filtrado exacto en aquellas

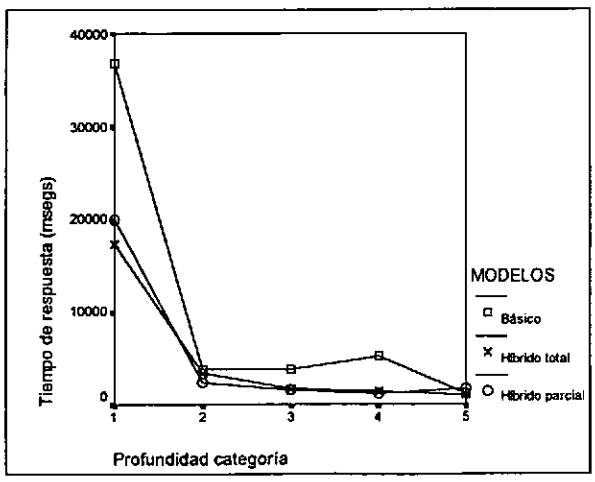

Figura 5-21: Tiempos de respuesta estimados en función de la profundidad de la categoría restringida (carga media)

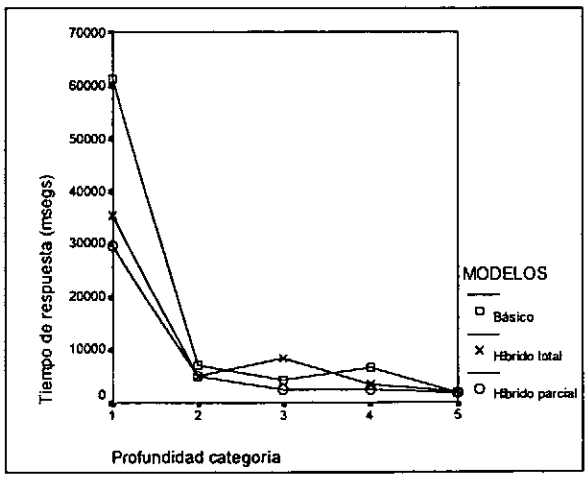

Figura 5-22: Tiempos de respuesta estimados en función de la profundidad de la categoría restringida (carga alta)

consultas restringidas a niveles inferiores, lo que por un lado complica ligeramente el modelo y da lugar a la posible aparición de falsos aciertos entre los resultados (aunque con una muy baja probabilidad). Además, el pobre rendimiento de este modelo bajo situaciones de carga alta sigue estando presente.

De esta forma, por medio del modelo híbrido con información parcial las probabilidades de falsos aciertos de todos los niveles se encuentran compensadas, requiriendo además un menor incremento en el espacio de almacenamiento, lo que permite obtener una mejora en los tiempos de respuesta bajo todas las situaciones de carga y para consultas restringidas a cualquier zona del grafo (independientemente de su profundidad).

5.5. Conclusiones

El presente capítulo ha abordado la implementación de los modelos híbridos propuestos, tanto la variante con información total como la variante con información parcial, con el objetivo de evaluar el rendimiento ofrecido por cada uno de ellos para contrastarlo con un modelo básico de directorio Web.

Respecto a la implementación de los modelos híbridos, destacar que en este caso es necesario determinar los valores para los parámetros asociados a la técnica de ficheros de firmas. Obviamente, en el caso del modelo básico, al estar construido en base a índices invertidos, no se requiere la determinación de ningún parámetro previo a la implementación.

Los valores asignados a estos parámetros deben ser optimizados teniendo en cuenta varios factores: el espacio de almacenamiento requerido para la estructura de datos híbrida, probabilidad de falsos aciertos y límites de los identificadores (de categorías y documentos). Para las implementaciones se ha buscado una probabilidad de falsos aciertos del orden del 1%, procurando minimizar el espacio requerido y generando un modelo flexible respecto al número posible de categorías y documentos.

En las implementaciones de los dos modelos híbridos, el sistema tiene capacidad para albergar del orden de 80 millones de documentos, cifra un orden de magnitud mayor que el número de documentos estimados en Yahoo!.

Respecto al crecimiento posible en el grafo de categorías ambos modelos se diferencian. Por una parte, la implementación del modelo híbrido con información total soporta un máximo de 22.100 categorías en base a los parámetros obtenidos y que da soporte holgadamente al prototipo desarrollado. Destacar que con un ligero cambio en los parámetros del sistema (asignado $w=4$ bits) el límite se amplia hasta 270.000, con capacidad para albergar casi el doble del número de categorías estimadas en Yahoo!.

Por otra parte, en el caso del modelo híbrido con información parcial se impone una limitación de 4.960 categorías, pero para los tres primeros niveles del grafo. Respecto a los niveles inferiores el número de categorías podría aumentar indefinidamente. Resaltar que el prototipo implementado consta de aproximadamente 500 categorías en los tres primeros niveles, por lo que el límite considerado proporciona suficiente flexibilidad al sistema.

Una vez analizados los detalles de implementación y habiendo comprobado la flexibilidad inherente a ambos modelos, se pasan a destacar las principales conclusiones derivadas de la evaluación del rendimiento de los modelos híbridos frente al modelo básico.

Como primer punto destacar que los modelos híbridos no suponen una penalización respecto a las búsquedas no restringidas a una zona del grafo de categorías (analizando el sistema en todas las situaciones de carga posible). Este es un aspecto clave para su correcta implantación, independientemente de la mejora del rendimiento para las búsquedas restringidas.

Respecto al rendimiento ante búsquedas restringidas a una zona del grafo, bajo una situación de carga nula, baja o media, los modelos híbridos mejoran el rendimiento en un 50% respecto al modelo básico.

En cambio, bajo una situación de carga alta el modelo híbrido con información total pierde efectividad y se equipara con el bajo rendimiento ofrecido por el modelo básico, debido al importante aumento en el espacio de almacenamiento requerido por el índice que asocia palabras clave con documentos. Por el contrario, el modelo híbrido con información parcial mantiene una mejora del rendimiento del 50% sobre el modelo básico (y en consecuencia también sobre la variante con información total).

Además, se ha demostrado que el modelo híbrido con información parcial responde adecuadamente a todas las consultas restringidas, independientemente de su profundidad en el grafo. Esto es debido a la obtención de unas probabilidades de falsos aciertos homogéneas y equilibradas para todos los niveles, junto con un menor tamaño de los ficheros de firmas embebidos en las listas invertidas.

En resumen, el modelo de arquitectura híbrida de fichero invertido y fichero de firmas con información parcial permite una mejora del rendimiento del 50% ante consultas restringidas a una zona del grafo de categorías, bajo todas las situaciones de carga del sistema; manteniendo estables los tiempos de respuesta ante búsquedas normales. Por otra parte, la implementación propuesta ha demostrado ser suficientemente flexible respecto al número de documentos capaz de soportar el sistema (más de 80 millones) así como respecto al número de categorías del directorio (aproximadamente 5.000 en los tres primeros niveles, e ilimitado para el resto).

6. CONCLUSIONES

En este capítulo se realiza un resumen de las conclusiones y aportaciones más relevantes derivadas de este trabajo. En primer lugar, se han examinado en detalle las principales técnicas de indexación, con especial énfasis en la técnica de ficheros invertidos y en la técnica de ficheros de firmas, al constituir la base sobre la que se asientan los actuales sistemas de búsqueda en el Web.

Asimismo, dentro de la diversidad existente entre los sistemas de búsqueda en el Web, los directorios Web se caracterizan por disponer de un conjunto reducido de información, pero de gran calidad (por contraposición al caso de los motores de búsqueda en donde prima la cantidad sobre la calidad). Las estructuras de datos gestionadas en un directorio Web engloban a los documentos, las categorías y las palabras clave, así como las interrelaciones entre ellos, lo que le confiere al conjunto una complejidad intrínseca.

El proceso de búsqueda en los sistemas de recuperación de información en el Web suele estar basado en una estructura de fichero invertido, al ofrecer un buen rendimiento frente al espacio de almacenamiento requerido. En cambio, el caso de los directorios incluye el valor añadido de las búsquedas restringidas a una zona concreta del grafo de categorías, que presentan ciertos problemas de eficiencia respecto a la estructura de fichero invertido. En concreto, se derivan de la complejidad de las listas invertidas a la hora de combinar los resultados de varias listas, especialmente si el número de elementos es elevado.

Con el objetivo de la mejora del rendimiento ante este tipo de consultas, se diseña e implementa USim, una herramienta de simulación de los usuarios de un servicio de búsqueda en el Web, con el propósito de realizar una evaluación del rendimiento completa y exhaustiva, analizando los tiempos de respuesta ante diferentes niveles de carga de un sistema de búsqueda. Esta herramienta está basada en la posibilidad de ajustar el comportamiento de los usuarios reales (búsquedas realizadas, categorías accedidas y

documentos consultados) a un modelo matemático (en concreto, a un proceso de Poisson o distribución Exponencial), principal aportación del análisis estadístico sobre los accesos realizados a un directorio Web.

En dicho análisis también se incluyen la confirmación de determinadas diferencias de comportamiento entre los usuarios de sistemas de búsqueda Web y tradicionales, así como la aportación de nuevas diferenciaciones. Dentro de éstas cabe destacar el gran número de resultados obtenido de media por cada consulta, derivado directamente de la realización de consultas poco concretas. Además, existe un conjunto reducido de documentos que son examinados repetidas veces por múltiples usuarios, relacionado directamente con aquellas búsquedas más reiterativas. Igualmente, al analizar un directorio Web por primera vez, se comprueba como la navegación por categorías es bastante reducida y restringida básicamente a los niveles superiores del grafo.

Sin embargo, la principal aportación del presente trabajo de tesis doctoral se centra en la arquitectura híbrida propuesta, que permite la mejora en el rendimiento ante consultas restringidas. Esta arquitectura se basa en una estructura de datos híbrida, constituida por un fichero invertido en donde cada una de las listas invertidas contiene embebido su propio fichero de firmas. Para ello ha sido definido un identificador de documento compuesto (que contiene las firmas superpuestas de todas sus categorías asociadas), de tal manera que a través del propio proceso de búsqueda se obtiene dinámicamente el fichero de firmas asociado a los resultados de la consulta.

En base a este fichero de firmas resultante, se realiza un primer filtrado inexacto que permite eliminar la gran mayoría de los documentos que no se clasificarán, para aplicar a continuación de manera eficiente el filtrado exacto, únicamente sobre los resultados restantes.

Para la definición e implementación de esta estructura híbrida se ha realizado un estudio exhaustivo sobre la utilización de los códigos de superposición (vinculados a la técnica de los ficheros de firmas) para representar la información del grafo de categorías y los documentos asociados. De este estudio se desprende la flexibilidad de los códigos de superposición para la representación de las características peculiares de un grafo dirigido acíclico de categorías, destacando la independencia frente al número de documentos y de categorías del sistema; mientras que los parámetros que afectan directamente al rendimiento (profundidad del grafo, porcentaje de categorías y documentos con varios padres) se caracterizan por permanecer estables a lo largo de la vida del sistema de búsqueda.

Dentro de la arquitectura de datos propuesta se definen dos variantes. La primera de las variantes, denominada arquitectura híbrida con información total, se caracteriza porque todas y cada una de las categorías del grafo disponen de su propia firma, y por lo tanto, aportan información a sus documentos asociados. Esta alternativa requiere un aumento en el espacio de almacenamiento considerable, mientras que las consultas restringidas a los niveles inferiores del grafo presentan una menor probabilidad de falsos aciertos.

Por otro lado, la variante del modelo propuesto denominada arquitectura híbrida con información parcial se caracteriza por aplicar la técnica de los códigos de superposición únicamente a las categorías pertenecientes a los tres primeros niveles del grafo. De esta manera se consigue una reducción en el espacio de almacenamiento, haciendo que la

probabilidad de falsos aciertos tome valores homogéneos para los diferentes niveles de profundidad.

Respecto a la implementación de ambos modelos, destacar en primer lugar la gran flexibilidad existente en ambos casos respecto al número de categorías y al número de documentos capaz de gestionar el sistema. En ambos casos, el número de documentos y categorías soportados se encuentra muy por encima del número práctico de documentos gestionados por un directorio Web típico. En el caso del modelo híbrido con información parcial este límite únicamente afecta a los primeros niveles del grafo, lo que permite un crecimiento ilimitando en las categorías inferiores del grafo sin repercusión alguna en el rendimiento del sistema de búsqueda.

En segundo lugar, en cuanto al rendimiento obtenido por los modelos híbridos frente a un modelo básico, señalar que los modelos híbridos no suponen una penalización respecto a las búsquedas no restringidas, aspecto clave para su correcta implantación independientemente de la mejora del rendimiento para las búsquedas restringidas.

El rendimiento que ofrecen los modelos híbridos ante búsquedas restringidas a una zona del grafo mejora el rendimiento en un 50% respecto al modelo básico, bajo situaciones de carga nula, baja y media del sistema. En cambio, bajo una situación de carga alta el modelo híbrido con información parcial mantiene esta mejora en el rendimiento, mientras que la variante con información total sufre una pérdida significativa de efectividad debido a la diferencia en el tamaño del índice invertido empleado en el proceso de búsqueda.

Además, el modelo híbrido con información parcial responde adecuadamente a todas las consultas restringidas, independientemente de su profundidad en el grafo. En consecuencia, la variante con información parcial ofrece una mejora del 50% en el rendimiento bajo todas las situaciones de carga del sistema, con una gran flexibilidad en el número de documentos y categorías a incluir en el sistema.

Como trabajo futuro en este desarrollo está la profundización en la utilización de los códigos de superposición aplicados a un grafo dirigido acíclico de categorías. En el presente trabajo se han apuntado diferentes formatos de generación de códigos de superposición, que afectan de manera directa a la probabilidad de falsos aciertos del sistema. Por este motivo, parece claro que la utilización de la técnica de generación adecuada permitirá una mejora en el rendimiento del sistema.

Otro aspecto interesante, se centra en la repercusión de la probabilidad de falsos aciertos de la componente de ficheros de firmas, en el rendimiento final del sistema de búsqueda, con especial atención en la localización del umbral a partir del cual el rendimiento es significativo. De esta manera, sería posible ajustar de manera adecuada la probabilidad de falsos aciertos (y por lo tanto, la mejora en el rendimiento final) con el incremento en el espacio de almacenamiento requerido. Directamente relacionado con el aspecto anterior, se encuentra la posibilidad de evitar la realización del filtrado exacto, con el consiguiente ahorro en tiempo de ejecución, en el caso de que la probabilidad de falsos aciertos sea lo suficientemente reducida como para garantizar unos errores despreciables.

Asimismo, otra alternativa para la mejora de los sistemas híbridos consiste en la aplicación parcial del filtrado inexacto, hasta obtener un número suficiente de potenciales resultados para la posterior aplicación del filtrado exacto. Esta alternativa está especialmente indicada

en aquellas consultas que recuperan un gran volumen de documentos, pero en donde el filtrado eliminará un pequeño porcentaje de los mismos, evitándose así el recorrido completo de todos los resultados.

Otra futura línea de trabajo se deriva de la herramienta de simulación USim, a través de la cual se ha podido determinar la caída en el rendimiento del modelo híbrido con información total bajo situaciones de carga elevada. Sobre USim se abren diferentes aspectos de mejora centrados en la creación de una herramienta con un mayor nivel de generalidad ante diferentes sistemas de recuperación de información y la definición de nuevas utilidades derivadas, todas ellas enfocadas hacia la evaluación del rendimiento de sistema de recuperación de información en el Web.

Finalmente, destacar la inclusión de la presente tesis doctoral dentro del proyecto CICYT "*Arquitecturas Distribuidas para Técnicas de Búsqueda en Internet*", con referencia TIC2001-0547. Esto permite el estudio y evolución del modelo híbrido propuesto en una arquitectura distribuida, así como su generalización para cualquier sistema de búsqueda, en donde el empleo únicamente de la técnica de ficheros invertidos no es adecuado.

REFERENCIAS

[ACM, 01] *ACM, Association for Computing Machinery*, página principal. http://www.acm.org, 2001.

[Agosti, 01] M. Agosti, M. Melucci, "*Information Retrieval on the Web*". En M. Agosti, F. Crestani, G. Pasi, editores, "Lectures on Information Retrieval: Third European Summer-School", ESSIR 2000. Revised Lectures, Springer-Verlag, Berlin/Heidelberg, 2001, pp. 242-285.

[Agrawal, 97] R. Agrawal, S. Chakrabarti, B. Dom, R. Raghavan, "*Using taxonomy, discriminants, and signatures for navigating in text databases*". The 23rd International Conference on Very Large Databases, VLDB 1997, pp. 446-455.

[Agrawal, 98] R. Agrawal, P. Raghavan, S. Chakrabarti, B. Dom, "*Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies*". VLDB Journal, vol. 7, no. 3, pp. 163-178, Agosto 1998.

[Aho, 75] A. V. Aho, M. J. Corasick, "*Fast pattern matching: an aid to bibliographic search*". Communications of ACM (CACM), 18(6): 333-340, Junio 1975.

[Altavista, 01] *Altavista*, página principal. http://www.altavista.com, 2001.

[Avram, 75] H. D. Avram, "*MARC: Its history and implications*". Washington: Library of Congress, ISBN 0-8444-0176-5, 1975.

[Baeza-Yates, 92] R. A. Baeza-Yates, "*Introduction to data structures and algorithms related to information retrieval*". En W. Frakes y R. Baeza-Yates, editores, "Information

Retrieval: Data structures and algorithms", capítulo 2, pp 13-27. Prentice-Hall, Englewood Cliffs, NF, USA, 1992, ISBN 0-13-463837-9, 1992.

[Baeza-Yates, 99a] R. Baeza-Yates, B. Ribeiro-Neto, "*Retrieval Evaluationl*". En R. Baeza-Yates, B. Ribeiro-Neto, "Modern Information Retrieval", capítulo 3, pp 73-97. Addison Wesley, ISBN 0-201-39829-X, 1999.

[Baeza-Yates, 99b] R. Baeza-Yates, B. Ribeiro-Neto, "*Searching the Web*". En R. Baeza-Yates, B. Ribeiro-Neto, "Modern Information Retrieval", capítulo 13, pp 367-395. Addison Wesley, ISBN 0-201-39829-X, 1999.

[Baeza-Yates, 99c] R. Baeza-Yates, B. Ribeiro-Neto, "*Modeling*". En R. Baeza-Yates, B. Ribeiro-Neto, "Modern Information Retrieval", capítulo 2, pp 19-71. Addison Wesley, ISBN 0-201-39829-X, 1999.

[Baeza-Yates, 99d] R. Baeza-Yates, B. Ribeiro-Neto, "*Introduction*". En R. Baeza-Yates, B. Ribeiro-Neto, "Modern Information Retrieval", capítulo 1, pp 1-18. Addison Wesley, ISBN 0-201-39829-X, 1999.

[Bayer, 77] R. Bayer, K. Unterauer, "*Prefix B-Trees*". ACM Transactions on Database Systems, vol. 2, no. 1, pp. 11-26, 1977.

[Bell, 93] T. Bell, A. Moffat, C. Nevill-Manning, I. Witten, J. Zobel, "*Data compression in full-text retrieval systems*". Journal of the American Society for Information Science, vol. 44, no. 9, pp. 508-531, 1993.

[Berners-Lee, 94] T. Berners-Lee, R. Cailliau, A. Luotonen, H. F. Nielsen, A. Secret, "*The World Wide Web*". Communication of the ACM, vol. 37, no. 8: 76-82, 1994.

[Bharat, 98] K. Bharat, A. Broder, "*A technique for measuring the relative size and overlap of public Web search engines*". The 7th International World Wide Web Conference, pp. 379-388, Brisbane, Australia, Abril 1998.

[Biwe, 01] *BIWE (Buscador en Internet de Webs en Español)*, página principal: http://www.biwe.es, 2001.

[Bowman, 94] C. M. Bowman, P. B. Danzing, D. R. Hardy, U. Manber, M. F. Schwartz, "*The Harvest information discovery and access system*". The 2nd International World Wide Web Conference, pp. 763-771, Octubre 1994.

[Boyer, 77] R. S. Boyer, J. S. Moore, "*A fast string searching algorithm*". Communications of ACM (CACM), vol. 20, no. 10, pp. 762-772, Octubre 1977.

[Briandais, 59] R. de la Briandais, "*File searching using variable length keys*". AFIPS Western Joint Computer Conference, pp. 295-298, San Francisco, USA, Marzo 1959.

[Brin, 98] S. Brin, L. Page, "*The anatomy of a large-scale hypertextual web search engine*". The 7th International World Wide Web Conference, pp.102-117, Brisbane, Australia, Abril 1998.

[Broder, 97] A. Broder, S. Glassman, M. Manasse, G. Zweig, "*Syntactic clustering of the Web*". The 6th International World Wide Web Conference, pp. 391-404, Santa Clara, CA USA, Abril 1997.

[Cacheda, 99] F. Cacheda, A. Pan, L. Ardao, "*A Layered Architecture based on Java for Internet and Intranet Information Systems*". International Journal of e-Business Strategy Management, vol. 1, 123-129. Noviembre/Diciembre 1999.

[Cacheda, 01a] F. Cacheda, A. Viña, "*Experiencies retrieving information in the World Wide Web*". 6th IEEE Symposium on Computers and Communications, ISBNs: 0-7695-1177-5, 0-7695-1178-3 (case), 0-7695-1179-1 (microfiche), pp. 72-79. Julio del 2001, Tunisia.

[Cacheda, 01b] F. Cacheda, A. Viña, "*Simulación para la Evaluación de Sistemas de Recuperación de Información en el WWW*". Primer Congreso Iberoamericano de Telematica (CITA 2001), ISBN: 958-9475-19-1, pp. 6 (Memorias en CD). Agosto de 2001, Cartagena de Indias, Colombia.

[Cacheda, 01c] F. Cacheda, M. Ponte, A. Viña, "*Análisis de las búsquedas realizadas, categorías accedidas y documentos vistos en un directorio Web*". III Jornadas de Ingeniería Telemática (JITEL 2001), ISBN: 84-7653-783-2, pp. 325-331. Septiembre 2001, Barcelona.

[Cacheda, 01d] F. Cacheda, A. Viña, "*Understanding how people use search engines: a statistical analysis for e-Business*". e-2001 (e-Business and e-Work Conference and Exhibition), ISBNs: 1-58603-205-4 (IOS Press) y 4-274-90469-5-C3055 (Ohmsha), Volumen 1, pp. 319-325. Octubre 2001, Venecia, Italia.

[Cacheda, 01e] F. Cacheda, A. Viña, "*Superimposing Codes Representing Hierarchical Information in Web directories*". 3rd International Workshop on Web Information and Data Management (WIDM 2001) en Tenth International Conference on Information and Knowledge Management (ACM-CIKM 2001), ACM ISBN: 1-58113-444-4, pp. 54-60. Noviembre 2001, Atlanta, USA.

[Cacheda, 02a] F. Cacheda, A. Viña, "*Inverted files and dynamic signature files for optimisation of Web directories*". The 11th International World Wide Web Conference (WWW2002). ISBN: 1-880672-20-0, Póster (Memorias en CD). Mayo 2002, Hawai, USA.

[Cacheda, 02b] F. Cacheda, A. Viña, "*Optimización de Directorios Web mediante Estructuras de Datos Híbridas*". Aceptado para las I Jornadas de Tratamiento y Recuperación de la Información (JOTRI 2002). Julio 2002, Valencia, España.

[Campbell, 93] M. Campbell, "*The design of text signatures for text retrieval systems*". Technical Report C93/23, School of Computing & Mathematics, Technical Reports Computing Series, Deakin University, 1993.

[Cha, 98] G. H. Cha, C. W. Chung, "*A New Indexing Scheme for Content-Based Image Retrieval*". Multimedia Tools and Applications, vol. 6, no. 3, pp. 263-288, Mayo 1998.

[Cha, 99] G. H. Cha, C. W. Chung, "*An Indexing and Retrieval Mechanism for Complex Similarity Queries in Image Databases*". Journal of Visual Communication and Image Representation, vol. 10, pp. 268-290, 1999.

[Cho, 98] J. Cho, H. García-Molina, L. Page, "*Efficient crawling through URL ordering*". The 7th International World Wide Web Conference, pp. 161-172, Brisbane, Australia, Abril 1998.

[Christodoulakis, 84] S. Christodoulakis, C. Faloutsos, "*Design considerations for a message file server*". IEEE Transaction on Software Engineering, vol. 10, no. 2, pp. 201-210, 1984.

[Clarke, 95] C. Clarke, G. Cormack, F. Burkowski, "*An algebra for structured text search and a framework for its implementation*". The Computer Journal, vol. 38, no. 1, pp. 43-56, 1995.

[Croft, 80] W. B. Croft, "*A model of cluster searching based on classification*". Information Systems, vol. 5, pp. 189-195, 1980.

[Croft, 88] W. B. Croft, P. Savino, "*Implementating ranking strategies using text signatures*". ACM Transactions on Office Information Systems, vol. 6, no. 1, pp. 42-62, 1988.

[Cutting, 90] D. Cutting, J. Pedersen, "*Optimizations for dynamic inverted index maintenance*". The 13th Internacional ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 405-411, 1990.

[Deppisch, 86] U. Deppisch, "*S-tree: A dynamic balanced signature index for office retrieval*". The 9th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 77-87, Septiembre 1986.

[Dervos, 96] D. Dervos, P. Linardis, Y. Manolopoulos, "*Perfect Encoding: a Signature Method for Text Retrieval*". Proceedings 4th Conference on Advanced Databases and Information Systems (ADBIS 96), vol. 1, pp. 176-182, Septiembre 1996.

[Dervos, 97] D. Dervos, P. Linardis, Y. Manolopoulos, "*S-Index: a Hybrid Structure for Text Retrieval*". Proceedings 5th Conference on Advanced Databases and Information Systems (ADBIS 97), pp 204-209, 1997.

[Eastman, 89] C. M. Eastman, "*Handling incrementally specified Boolean queries: a comparision of inverted and signature file organizations*". Information Processing and Management, vol. 25, no. 1, pp. 27-38, 1989.

[Elias, 75] P. Elias, "*Universal codeword sets and representations of the integers*". IEEE Transactions on Information Theory, vol. IT-21, pp. 194-203, Marzo 1975.

[Excite, 01] *Excite*, página principal. http://www.excite.com, 2001.

[Faloutsos, 84] C. Faloutsos, S. Christodoulakis, "*Signature files: an access method for documents and its analytical performance evaluation*". ACM Transactions on Office Information Systems, vol. 2, no. 4, pp. 267-288, Octubre, 1984.

[Faloutsos, 85a] C. Faloutsos, "*Access methods for text*". Computing Surveys, vol. 17, no. 1, pp. 49-74, Marzo, 1985.

[Faloutsos, 85b] C. Faloutsos, "*Signature files: Design and performance comparison of some signature extraction methods*". ACM SIGMOD International Conference on Management of Data and Symposium on Principles of Database Systems, pp. 63-82, 1985.

[Faloutsos, 87] C. Faloutsos, S. Christodoulakis, "*Description and performance analysis of signature file methods*". ACM Transactions on Office Information Systems, vol. 5, no. 3, pp. 237-257, Julio, 1987.

[Faloutsos, 88] C. Faloutsos, R. Chan, "*Fast text access methods for optical and large magnetics disks: Designs and performance comparison*". The 14th International Conference on Very Large DataBase, pp. 280-293, 1988.

[Faloutsos, 92a] C. Faloutsos, "*Signature files*". En W. Frakes y R. Baeza-Yates, editores, "Information Retrieval: Data structures and algorithms, capítulo 4, páginas 44-65. Prentice-Hall, Englewood Cliffs, NF, USA, 1992.

[Faloutsos, 92b] C. Faloutsos, H. V. Jagadish, "*Hybrid index organizations for text databases*". Proceedings of the 3rd International Conference on Extending Database Technology (EDBT), pp. 310-327, 1992.

[Faloutsos, 95] C. Faloutsos, D. Oard, "*A survey of Information Retrieval and filtering methods*". Technical Report CS-TR3514, Department of Computer Science, University of Maryland, 1995.

[Files, 69] J. R. Files, H. D. Huskey, "*An information retrieval system based on superimposed coding*". Fall Joint Computer Conference, AFIPS Press, vol. 35, pp. 423-432, 1969.

[Fox, 91] E. A. Fox, W. C. Lee, "FAST-INV: A fast algorithm for building large inverted files". Technical Report TR-91-10, Virginia Polytechnic Institute Department of Computer Science, Marzo 1991.

[Fredkin, 60] E. Fredkin, "*Trie memory*". Communications of the ACM, vol. 3, no. 9, pp. 490-499, 1960.

[Friedman, 71] S. R. Friedman, J. A. Maceyak, S. F. Weiss, "*A relevance feedback system based on document transformation*". En G. Salton, "The SMART retrieval system - Experiments in automatic document processing", Ed. Prentice-Hall, 1971.

[Galaxy, 01] *Galaxy*, página principal. http://www.galaxy.com/, 2001.

[Gauch, 96] S. Gauch, G. Wang, "*Information Fusion with ProFusion*". The World Conference of the Web Society, WebNet'96, pp. 174-179, Octubre, 1996.

[Gonnet, 83] G. Gonnet, "*Unstructured data bases or very efficient text searching*". ACM Principles of Database Systems, vol. 2, pp. 117-124, 1983.

[Gonnet, 92] G. H. Gonnet, R. A. Baeza-Yates, T. Snider, "*New Indices for text: PAT trees and PAT arrays*". En W. Frakes y R. Baeza-Yates, editores, "Information Retrieval: Data structures and algorithms, capítulo 5, páginas 66-82. Prentice-Hall, Englewood Cliffs, NF, USA, 1992.

[Google, 01] *Google*, página principal. http://www.google.com/, 2001.

[Gosling, 96] J. Gosling, H. McGilton, "*The Java language environment: A white paper*". Sun Microsystems, 1996.

[Gustafson, 71] R. A. Gustafson, "*Elements of the randomised combinatorial file structure*". ACM SIGIR Symposium on Information Storage and Retrieval, pp. 163-174, 1971.

[Harman, 90] D. K. Harman, G. Candela, "*Retrieving records from a gigabyte of text on a minicomputer using statistical ranking*". Journal of the American Society for Information Science, vol. 41, no. 8, pp. 581-589, 1990.

[Harman, 92] D. K. Harman, E. Fox, R. Baeza-Yates, W. Lee, "*Inverted files*". En W. Frakes y R. Baeza-Yates, editores, "Information Retrieval: Data structures and algorithms, capítulo 3, páginas 28-43. Prentice-Hall, Englewood Cliffs, NF, USA, 1992.

[Harman, 95] D. K. Harman, "*Overview of the third text retrieval conference*". The 3rd Text Retrieval Conference (TREC-3), pp. 1-19, National Institute of Standards and Technology Special Publication 500-207, 1995.

[Harper, 80] D. J. Harper, "*Relevance Feedback in Automatic Document Retrieval Systems: An Evaluation of Probabilistic Strategies*". Disertación doctoral, Jesus College, Cambridge, England.

[Harrison, 71] M. C. Harrison, "*Implementation of the substring test by hashing*". Communications of the ACM, vol. 14, no. 12, pp. 777-779, 1971.

[Haskin, 81] R. L. Haskin, "*Special-purpose processors for text retrieval*". Database Engineering, vol. 4, no. 1, pp. 16-29, Septiembre, 1981.

[Hawking, 99] D. Hawking, N. Craswell, P. Thistlewaite, D. Harman, "*Results and challenges in Web search evaluation*". The 8th World Wide Web Conference, pp. 243-252, Mayo, 1999.

[Heaps, 78] J. Heaps, "*Information Retrieval – Computational and Theoretical Aspects*". Academic Press, New York, 1978.

[Hirschberg, 90] D. Hirschberg, D. Lelewer, "*Efficient decoding of prefix codes*". Communications of the ACM, vol. 33, no.4, pp. 449-459, Abril, 1990.

[Hollaar, 83] L. A. Hollaar, K. F. Smith, W. H. Chow, P. A. Emrath, R. L. Haskin, "*Architecture and operation of a large, full-text information-retrieval system*". Advanced Database Machine Architecture, pp. 256-299. Ed. Prentice-Hall, 1983.

[HotBot, 01] *HotBot*, página principal. http://www.hotbot.com, 2001.

[Huang, 00] L. Huang, "*A survey on web information retrieval technologies*". Research Proficiency Exam Report, Experimental Computer System Lab, 2000.

[Hyman, 89] R. J. Hyman, "*Information Access: Capabilities & Limitations of Printed & Computerized Sources*", American Library Association, ISBN 0-838-90512-9, 1989.

[Indyk, 98] P. Indyk, S. Chakrabarti, B. Dom, "*Enhanced hypertext categorization using hyperlinks*". ACM SIGMOD International Conference on Management of Data, pp. 307-318, 1998.

[Infoseek, 01] *Infoseek*, página principal. http://www.infoseek.com, 2001.

[Internet Archive, 01] *Internet Archive*. Building a digital library for the future, 2001, http://www.archive.org/

[Jansen, 98] B. Jansen, A. Spink, J. Bateman, T. Saracevic, "*Real Life Information Retrieval: A Study Of User Queries On The Web*". SIGIR Forum, vol. 32, no. 1, pp. 5-17, 1998.

[Joyanes, 98] L. Joyanes, I. Zahonero, "*Ordenación, búsqueda y mezcla*". En L. Joyanes, I. Zahonero "Estructura de datos. Algoritmos, abstracción y objetos", capítulo 15, páginas 501-552, Ed. McGraw-Hill, ISBN 85-481-2042-6.

[Kahle, 97] B. Kahle, "*Archiving the Internet*". Scientific American, Marzo, 1997.

[Kautz, 64] W. Kautz, R. C. Singleton, "*Nonrandom binary superimposed codes*". IEEE Transactions on Information Theory, vol. 10, pp. 363-377, 1964.

[Kent, 88] A. Kent, R. Sacks-Davis, K. Ramamohanarao, "*A superimposed coding scheme based on multiple block descriptor files for indexing very large data bases*". The 14th Very LargeDataBase Conference, pp. 351-359, 1988.

[Kirsch, 98] S. Kirsch, "*Infoseek's experiences searching the Internet*". ACM SIGIR Forum, vol. 32, no. 2, pp. 3-7, 1998.

[Kitagawa, 97] H. Kitagawa, Y. Ishikawa, "*False drop analysis of set retrieval with signature files*". IEICE Transactions on Information and Systems, vol. E80-D, no. 6, pp. 653-664, Junio, 1997.

[Knuth, 77] D. E. Knuth, J. H. Morris, V. R. Pratt, "*Fast pattern matching in strings*". SIAM Journal of Computing, vol. 6, no. 2, pp. 323-350, Junio, 1977.

[Kobayashi, 00] M. Kobayashi, K. Takeda, "*Information Retrieval on the Web*". ACM Computing Surveys, vol. 32, no. 2, pp. 144-173, Junio, 2000.

[Korfhage, 97] R. Korfhage, "*Information Storage and Retrieval*". John Wiley & Sons, Inc., ISBN 0-471-14338-3, 1997.

[Koster, 94] M. Koster, "*A standard for robot exclusion*". Disponible en http://www.robotstxt.org/wc/norobots.html, 1994.

[Kowalski, 00] G. J. Kowalski, M. T. Maybury, "*Information storage and retrieval systems: theory and implementation*". Ed. Kluwer Academic Publishers, ISBN: 0-7923-7924-1, 2000.

[Larson, 83] P. A. Larson, "*A method for speeding up text retrieval*". ACM SIGMOD Conference: Databases for Business and Office Applications, pp. 117-123, Mayo, 1983.

[Labrou, 99] Y. Labrou, T. Finin, "*Yahoo! as an ontology – Using Yahoo! categories to describe documents*". The 8th International Conference on Information Knowledge Management, pp. 180-187, Noviembre, 1999.

[Lee, 89] D. L. Lee, C. Leng, "*Partitioned signature files: design issues and performance evaluation*". ACM Transactions on Office Information Systems, vol. 7, no. 2, pp. 158-180, 1989.

[Lee, 90] D. L. Lee, C. Leng, "*A partitioned signature file structure for multiattribute and text retrieval*". The 6th International Conference on Data Engineering, pp. 389-397, Febrero, 1990.

[Lee, 95] D. K. Lee, Y. M. Kim, G. Patel, "*Efficient signature file methods for text retrieval*". IEEE Transactions on Data and Knowledge Engineering, vol. 7, no. 3, pp. 423-435, Junio, 1995.

[Leighton, 96] H. V. Leighton, "*Performance of four World Wide Web (WWW) index services: Infoseek, Lycos, WebCrawler, and WWWWorm*". Disponible en http://www.winona.msus.edu/library/webind.htm, 1996.

[Leighton, 97] H. V. Leighton, J. Srivastava, "*Precision among World Wide Web Search Services (Search Engines): AltaVista, Excite, Hotbot, Infoseek, Lycos*". Disponible en http://www.winona.msus.edu/is-f/library-f/webind2/webind2.htm, 1997.

[Lifantsev, 98] M. Lifantsev, OpenGrid (Open Global Ranking Search Engine and Directory). Disponible en http://www.ecsl.cs.sunysb.edu/~maxim/OpenGRiD/, 1998.

[Lin, 88] Z. Lin, C. Faloutsos, "*Frame sliced signature files*". IEEE Transactions on Knowledge and Data Engineering, vol. 4, no. 3, pp. 281-289, Junio, 1992.

[Lowley, 00] S. Lowley, "*The evaluation of WWW search engines*". Journal of Documentation, vol. 56, no. 2, pp. 190-211, 2000.

[McIlroy, 82] M. D. McIlroy, "*Development of a spelling list*". IEEE Transactions on Communications, vol. 30, no. 1, pp. 91-99, 1982.

[Moffat, 92] A. Moffat, J. Zobel, "*Parameterised compression for sparse bitmaps*". ACM-SIGIR International Conference on Research and Development in Information Retrieval, pp. 274-285, Junio 1992.

[Moffat, 94] A. Moffat, J. Zobel, "*Compression and fast indexing for multi-gigabyte text databases*". Australian Computer Journal, vol. 26, no. 1, pp. 1-9, Febrero, 1994.

[Mooers, 49] C. Mooers, "*Application of Random Codes to the Gathering of Statistical Information*". Bulletin 31, Zator Co., Cambridge, USA. Basado en M. S. thesis, MIT, Enero 1949.

[Morrison, 68] D. Morrison, "*PATRICIA-Practical Algorithm To Retrieve Information Coded in Alphanumeric*". Journal of the ACM, vol. 15, no. 4, pp. 514-534, Octubre, 1968.

[Navarro, 98] G. Navarro, "*Approximate text searching*". PhD thesis, Department of Computer Science, Universidad de Chile, Diciembre 1998.

[Navarro, 99] G. Navarro, "*Indexing and Searching*". En R. Baeza-Yates, B. Ribeiro-Neto, "Modern Information Retrieval", capítulo 8, páginas 191-228. Addison Wesley, ISBN 0-201-39829-X, 1999.

[NetSizer, 01] *NetSizer*, página principal. http://www.netsizer.com/, 2001.

[Norris, 69] D. M. Norris, "*A History of Cataloguing and Cataloguing Methods 1100-1850*". Ed. Detroit: Gale, 1969.

[ODP, 01] *Open Directory Project*, página principal. http://www.dmoz.org, 2001.

[Page, 98] L. Page, S. Brin, R. Motwani, T. Winograd "*The pagerank citation ranking: Bringing order to the web*". Annual Meeting of the American Society for Information Science, ASIS'98, 1998.

[Pfaltz, 80] J. L. Pfaltz, W. H. Berman, E. M. Cagley, "*Partial match retrieval using indexed descriptor files*". Communication of the ACM, vol. 23, no. 9, pp. 522-528, Septiembre, 1980.

[Rijsbergen, 79] C. J. van Rijsbergen, "*Information Retrieval*". Butterworths, Londres, 1979.

[Roberts, 79] C. S. Roberts, "*Partial-match retrieval via the method of superimposed codes*". Proceedings of the IEEE, vol. 67, no. 12, pp. 1624-1642, Diciembre, 1979.

[Rocchio, 71] J. J. Rocchio, "*Relevance feedback in information retrieval*". En G. Salton, "The SMART retrieval system - Experiments in automatic document processing", Ed. Prentice-Hall, 1971.

[Rosenthal, 96] M. Rosenthal, H. Chu, "*Search engines for the World Wide Web: A comparative study and evaluation methodology*". American Society for Information Science, ASIS 1996, pp. 127-135, Octubre, 1996.

[Sacks-Davis, 83] R. Sacks-Davis, K. Ramamohanarao, "*A two level superimposed coding scheme for partial match retrieval*". Information Systems, vol. 8, no. 4, pp. 273-280, 1983.

[Sacks-Davis, 87] R. Sacks-Davis, A. Kent, K. Ramamohanarao, "*Multikey access methods based on superimposed coding techniques*". ACM Transactions on Database Systems, vol. 12, no. 4, pp. 655-696, 1987.

[Salton, 69] G. Salton, "*Evaluation problems in interactive information retrieval*". Information Storage & Retrieval, vol. 6, no. 1, pp. 29-44, 1969.

[Salton, 71] G. Salton, "*Relevance feedback and the optimisation of retrieval effectiveness*". En G. Salton, "The SMART retrieval system - Experiments in automatic document processing", Ed. Prentice-Hall, 1971.

[Salton, 78] G. Salton, A. Wong, "*Generation and search of clustered files*". ACM Transactions on Database Systems, vol. 3, no. 4, pp. 321-346, 1978.

[Salton, 83] G. Salton, M. J. McGill, "*Introduction to modern information retrieval*". McGraw-Hill, ISBN 0-07-054484-0, 1983.

[Shaw, 97] W. M. Shaw Jr., R. Burgin, P. Howell, "*Performance standards and evaluation in IR test collections: Cluster-based retrieval models*". Information Processing & Management, vol. 33, no. 1, pp. 1-14, 1997.

[Silverstein, 99] C. Silverstein, M. Henzinger, H. Marais, M. Moricz. "*Analysis of a Very Large Web Search Engine Query Log*". SIGIR Forum, vol. 33, no. 1, pp. 6-12, 1999.

[Stiassny, 60] S. Stiassny, "*Mathematical analysis of various superimposed coding methods*". American Documentation, vol. 11, no. 2, pp. 155-169, 1960.

[Sullivan, 98] D. Sullivan, "*Search engine sizes*". Disponible en http://www.searchenginewatch.com/reports/sizes.html, 1998.

[Sunday, 90] D. M. Sunday, "*A very fast substring search algorithm*". Communications of the ACM, vol. 33, no. 8, pp. 132-142, Agosto, 1990.

[TREC, 01] *TREC NIST* (Text Retrieval Conference, National Institute of Standards and Technologies), página principal. http://trec.nist.gov/, 2001.

[Tsichritzis, 83] D. Tsichritzis, S. Christodoulakis, "*Message files*". ACM Transactions on Office Information Systems, vol. 1, no. 1, pp. 88-98, Enero, 1983.

[Voorhees, 97] E.M. Voorhees, D. K. Harman, "*Overview of the 6th text retrieval conference (TREC-6)*". The 6th Text Retrieval Conference, pp. 1-24. National Institute of Standards and Technology Special Publication, 1997.

[Witten, 94] I. H. Witten, A. Moffat, T. C. Bell, "*Managing gigabytes: Compressing and indexing documents and images*". Van Nostrand Reinhold, New York, 1994.

[Wu, 92] S. Wu, U. Manber, "*Agrep – a fast approximate pattern searching tool*". USENIX Conference, pp. 153-162, Enero, 1992.

[Yahoo!, 01] *Yahoo!*, página principal. http://www.yahoo.com/, 2001.

[Yu, 77] C. T. Yu, W. S. Luk, "*Analysis of effectiveness of retrieval in clustered files*". Journal of the ACM, vol. 24, no. 4, pp. 607-622, 1977.

[Zahn, 71] C. T. Zahn, "*Graph-theoretical methods for detecting and describing gestalt clusters*". IEEE Transactions on Computers, vol. 20, no. 1, pp. 68-86, 1971.

[Zipf, 49] G. Zipf, "*Human behaviour and the principle of least effort*". Ed. Addison-Wesley, 1949.

[Zobel, 96] J. Zobel, A. Moffat, K. Ramamohanarao, "*Guidelines for Presentation and Comparison of Indexing Techniques*". ACM SIGMOD Record, vol. 25, no. 3, pp. 10-15, Septiembre, 1996.

[Zobel, 98] J. Zobel, A. Moffat, K. Ramamohanarao, "*Inverted files versus signature files for text indexing*". ACM Transactions On Database Systems, vol. 23, no. 4, pp. 453-490, Diciembre, 1998.

Apéndice A: ESTUDIO DE LOS ACCESOS A UN DIRECTORIO WEB

En este apéndice se incluyen los detalles del análisis estadístico realizado en el capítulo 2 sobre los accesos a un directorio Web, centrado especialmente en la investigación alrededor de las distribuciones a las que se ajustan las peticiones de búsqueda, accesos a categorías y visitadas de documentos y las relaciones existentes entre ellas.

A.1. Distribución de las búsquedas

El problema que se plantea en esta sección consiste en comprobar si los instantes de tiempo en los que fueron recibidas las 105.786 búsquedas a lo largo de 16 días se pueden ajustar a algún modelo matemático. Inicialmente, la distribución Exponencial parece ser el modelo a seguir, sin embargo, debido a que la precisión del registro de tiempos es a nivel de segundos se plantean problemas, ya que en algunos casos se pueden producir varias búsquedas simultáneamente, lo que hace que el tiempo entre dos eventos sea de 0 segundos. Esto invalida la utilización de esta distribución al ser continua, no obstante, es inmediata la transformación en la variable aleatoria discreta: "*Número de búsquedas en un minuto*", que permite un estudio más adecuado.

A partir de esta variable aleatoria, los modelos matemáticos que mayor probabilidad tienen de ajustarse correctamente son el proceso de Poisson y una distribución Normal, por lo que el desarrollo siguiente se centrará en la comprobación un ajuste adecuado a alguna de las dos distribuciones.

En este punto, el principal problema que se plantea se deriva del hecho que a lo largo del tiempo los parámetros (básicamente, la media) fluctúan, como se puede apreciar

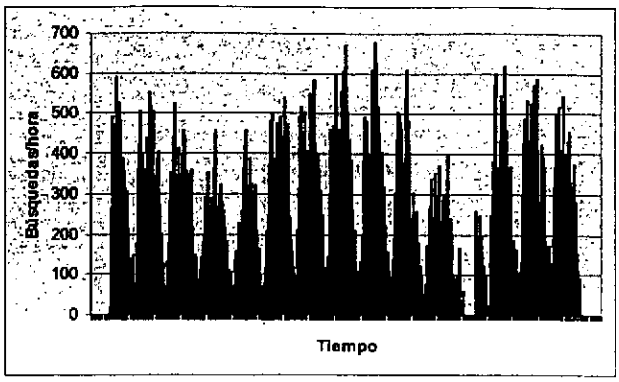

Figura A-1: Búsquedas por hora durante el período analizado

claramente en la Figura A-1. Por lo tanto, aunque la media de toda la serie sea de 5,2 búsquedas por minuto (ó 312 búsquedas por hora), no es posible un análisis de todos los datos en conjunto ya que los tests rechazarían el ajuste a cualquier distribución debido a la variabilidad de sus parámetros.

Inicialmente, el análisis parte de la descomposición de los 20.400 datos de la variable aleatoria "*Número de búsqueda por minuto*" (obtenidos de las 340 horas analizadas) en grupos de 1 hora, en donde presumiblemente los parámetros de las distribuciones permanecerán constantes. Esto genera 340 variables aleatorias con 60 elementos cada una de ellas. A cada una de estas variables se les ha aplicado el test de Kolmogorov-Smirnov para comprobar su ajuste a una distribución Normal y a un proceso de Poisson.

Por motivos de espacio no es posible mostrar el resultado de todos los tests realizados, por lo que se ha decidido mostrar un histograma de los p-valores obtenidos para las 340 variables aleatorias. Es importante recordar que un p-valor elevado apoya firmemente la tesis de que la variable aleatoria se ajusta a la distribución comprobada, mientras que es necesario un p-valor bajo (p.e. menor de 0,1) para rechazar que la variable aleatoria se ajusta al modelo en cuestión.

En la Figura A-2 se muestran los resultados obtenidos al intentar ajustar las 340

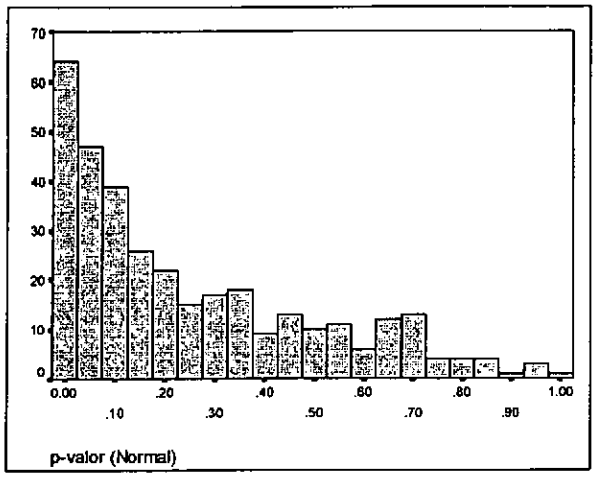

Figura A-2: Histograma p-valores test Kolmogorov-Smirnov para la Normal

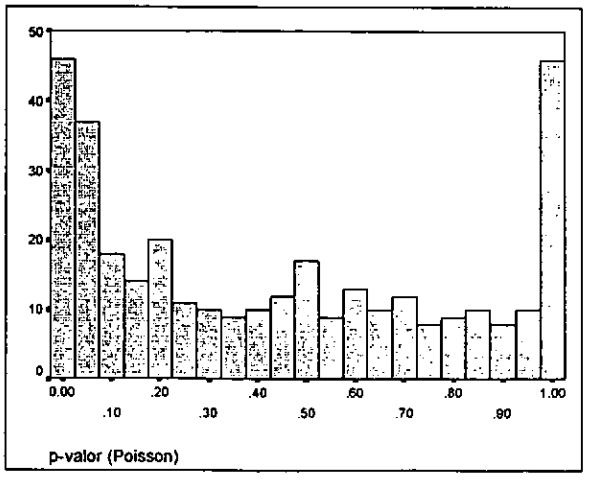

Figura A-3: Histograma p-valores test Kolmogorov-Smirnov para la Poisson

distribuciones a una distribución Normal. Como se observa la mayor parte de los tests producen resultados cercanos al 0, en muy pocos casos se podría aceptar con una probabilidad superior al 90%. De hecho, la media obtenida es de aproximadamente 0,25.

Por otra parte, en la Figura A-3 se muestran los resultados al repetir la operación pero en este caso ajustando las distribuciones a un proceso de Poisson. Como se observa, aún sigue habiendo un porcentaje importante de distribuciones en donde se debe rechazar el ajuste a un proceso de Poisson, sin embargo, al contrario de lo que sucedía en el caso anterior, sí hay un grupo importante de distribuciones que se ajustan a una Poisson, de hecho la media obtenida es el doble que en el caso anterior, de 0,51.

Este primer estudio indica, por una parte que el ajuste de las distribuciones analizadas se adapta mejor a un proceso de Poisson, y por otra parte, la causa de que exista un gran número de distribuciones en donde se debe rechazar este ajuste, probablemente sea debida a que durante la hora de duración de cada variable aleatoria se estén produciendo variaciones en la media, o lo que es lo mismo, el parámetro λ está fluctuando. Por lo tanto, se hace necesario realizar un segundo estudio, en donde se consideran 1.020 variables aleatorias compuestas de 20 elementos cada una de ellas, en donde se han considerado intervalos temporales de 20 minutos, con el objetivo de que la media de la distribución permanezca estable.

A cada una de las 1.020 variables aleatorias, al igual que en el caso anterior, se les ha aplicado el test de Kolmogorov-Smirnov para comprobar su ajuste a una distribución Normal y a un proceso de Poisson. Los resultados se muestran en las siguientes figuras.

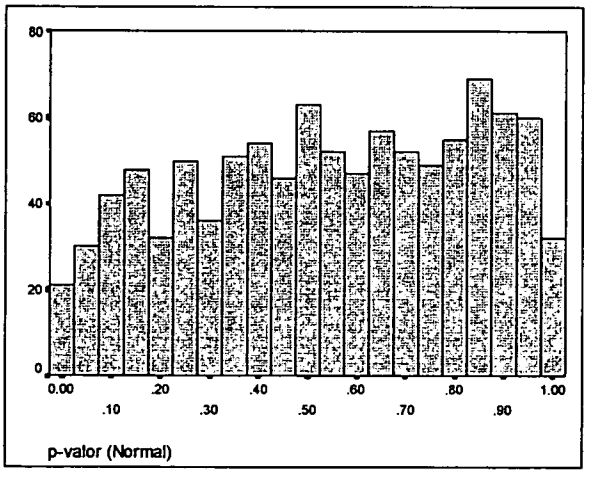

Figura A-4: Histograma p-valores test Kolmogorov-Smirnov para la Normal

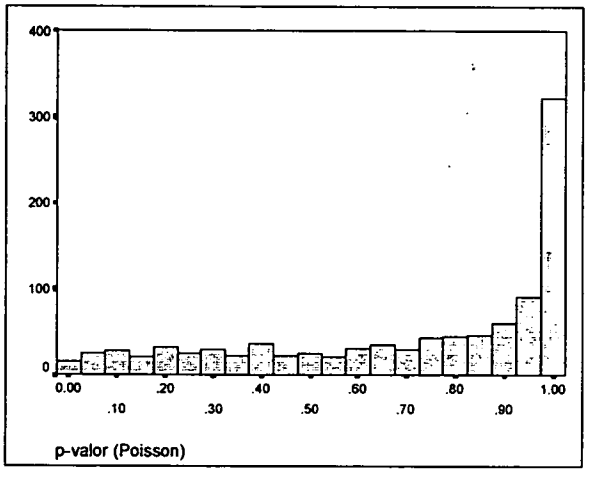

Figura A-5: Histograma p-valores test Kolmogorov-Smirnov para la Poisson

En el caso del ajuste a una distribución Normal (Figura A-4) se observa que en la mayoría de las variables analizadas no se puede rechazar el ajuste, pero tampoco se obtienen unos p-valores elevados que indiquen una clara relación. Por otra parte, en el gráfico del ajuste a una distribución de Poisson (Figura A-5) se observa claramente que existe una gran mayoría de distribuciones que pasan el ajuste con p-valores por encima del 90% (de hecho, más de 300 variables aleatorias obtienen un p-valor superior 97,5%). Y aunque existen casos en donde el ajuste no es tan claro, y en algunos casos se podría incluso rechazar, debido probablemente a fluctuaciones de la media en el intervalo analizado, su reducido

número no afecta al conjunto de las distribuciones, en donde la media obtenida de un nivel de aceptación 0,71 indica rotundamente el correcto ajuste a un proceso de Poisson.

Finalmente, y como confirmación del ajuste a un proceso de Poisson, se han agrupado algunas variables aleatorias con el objetivo de obtener grupos homogéneos respecto a la media, y con un número mayor de datos. De hecho, se han formado cerca de 90 grupos de los cuales, en ningún caso se puede rechazar la hipótesis de que se ajusten a una distribución de Poisson.

Esto pone de manifiesto el hecho de que la variable aleatoria "*Número de búsquedas por minuto*" sigue un proceso de Poisson, en donde, su parámetro λ_{busq} es variable a lo largo del tiempo.

A.2. Distribución de las categorías

El caso de los accesos a categorías realizados durante los 16 días del período investigado es totalmente análogo al caso de las búsquedas, por lo que la exposición se centrará principalmente en los resultados obtenidos.

Al igual que en el caso anterior, se ha definido la variable aleatoria: "*Número de accesos a categorías por minuto*" para facilitar su estudio y la aplicación de los tests de ajuste.

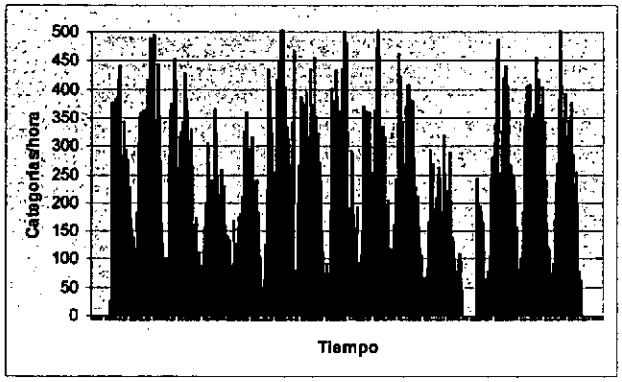

Figura A-6: Accesos a categorías por hora durante el período analizado

El principal problema existente es el hecho de que la media de la distribución varía en función de la franja horaria considerada (ver Figura A-6). Por lo tanto, la solución adoptada es semejante al caso anterior, y aunque sería correcto (y posiblemente más razonable) agrupar el total de los 20.400 datos sobre categorías en intervalos de 20 minutos, ya que ha quedado patente su adaptación a la hora de mantener en un estado homogéneo la media de cada una de las variables aleatorias, se ha preferido, por coherencia, realizar un primer estudio de las variables agrupadas en intervalos de 1 hora. En las siguientes figuras se muestran los resultados obtenidos al aplicar el test de Kolmogorov-Smirnov sobre las 340 variables aleatorias, considerando el ajuste a una distribución Normal y de Poisson.

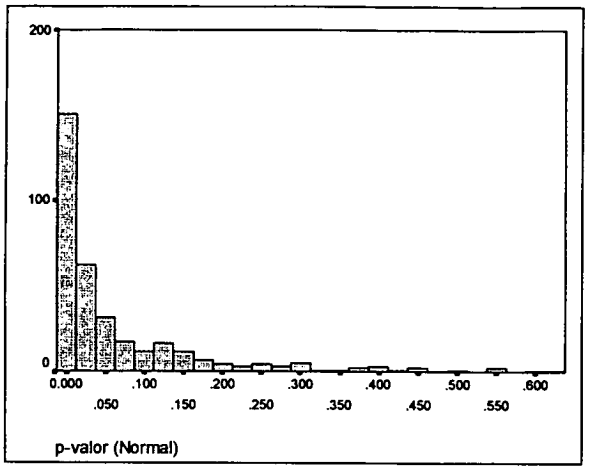

Figura A-7: Histograma p-valores test Kolmogorov-Smirnov para la Normal

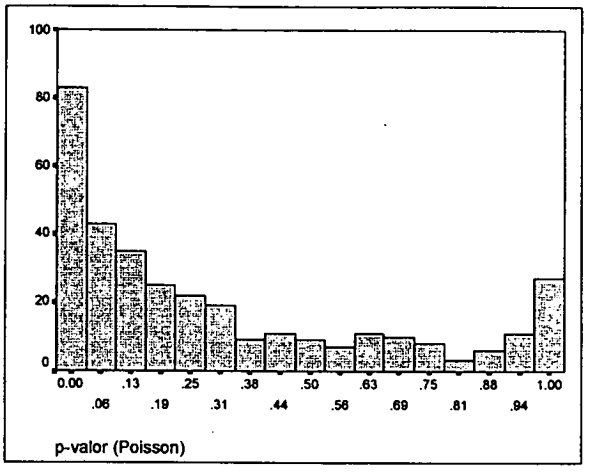

Figura A-8: Histograma p-valores test Kolmogorov-Smirnov para la Poisson

En la Figura A-7 se observa manifiestamente que en la mayoría de los casos se rechaza con p-valores inferiores al 0,05 que las variables analizadas se ajusten a una Normal, asimismo, no existe ningún valor superior al 55% de aceptación de esta hipótesis. En cambio, en la Figura A-8 destaca el hecho de que haya un porcentaje reducido de variables en donde se acepta con valores elevados su ajuste a un proceso de Poisson. Por otra parte, al igual que en el caso de las búsquedas, existe un porcentaje significativo de rechazos, lo que indica la necesidad de modificar el intervalo considerado para intentar mantener la media estable. En los siguientes gráficos se muestran los resultados al aplicar los tests considerando 1.020 variables aleatorias con 20 elementos, es decir, al agrupar los datos en intervalos de 20 minutos de duración.

En la Figura A-9 se muestran los resultados obtenidos para el ajuste a una distribución Normal. Como se observa, aunque a nivel general se podría aceptar el ajuste a esta distribución, los p-valores obtenidos no son especialmente elevados, de hecho se obtiene un p-valor medio de 0,5. Por el contrario, en el ajuste a un proceso de Poisson (Figura A-10) más de 400 variables pasan el test con un nivel de aceptación superior al 95%. También en este caso existen unos porcentajes residuales de variables en donde los niveles de aceptación son reducidos, pero probablemente sea debido a variaciones en el parámetro

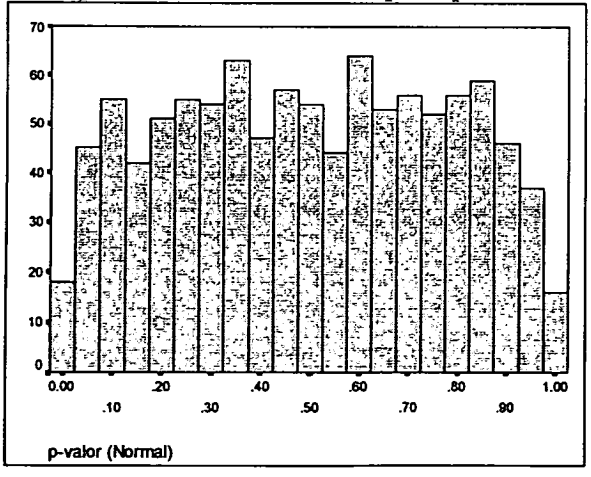

Figura A-9: Histograma p-valores test Kolmogorov-Smirnov para la Normal

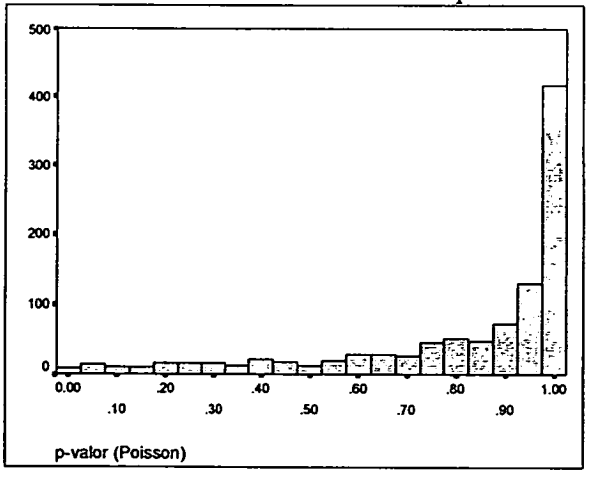

Figura A-10: Histograma p-valores test Kolmogorov-Smirnov para la Poisson

λ de las distribuciones. Con el objetivo de contrastar estos resultados, se han agrupado aquellas variables aleatorias con valores medios similares, generando 110 grupos y en ningún caso se rechazaría la hipótesis del ajuste a un proceso de Poisson.

Por lo tanto, se puede concluir que la variable aleatoria "*Número de categorías accedidas por minuto*" sigue un proceso de Poisson, en donde, su parámetro λ_{cats} es variable a lo largo del tiempo.

A.3. Distribución de los documentos

El estudio de la distribución a la que se ajustan los accesos a documentos de un directorio Web es totalmente análogo a los dos casos ya presentados, por lo que la exposición se ajustará a los resultados.

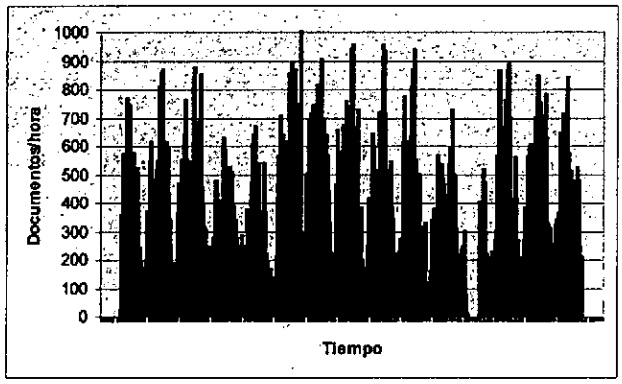

Figura A-11: Documentos examinados por hora durante el período analizado

Al igual que en los dos casos anteriores, el primer paso consiste en la construcción de la variable aleatoria "*Número de documentos consultados por minuto*", convirtiendo en discretos los datos originales. A continuación, y teniendo en cuenta que los datos disponibles presentan una gran variación en la media a lo largo de todo el período

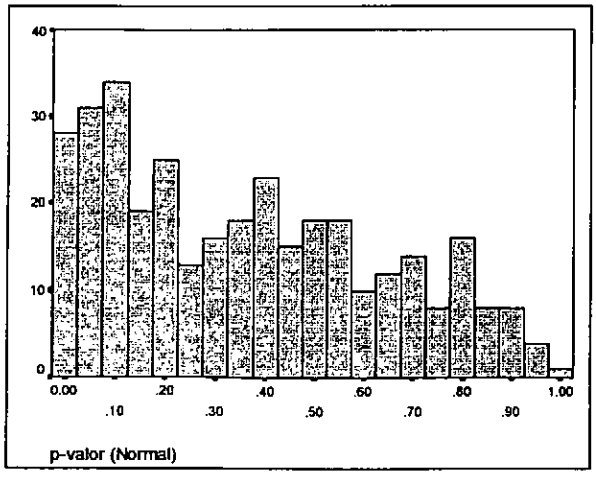

Figura A-12: Histograma p-valores test Kolmogorov-Smirnov para la Normal

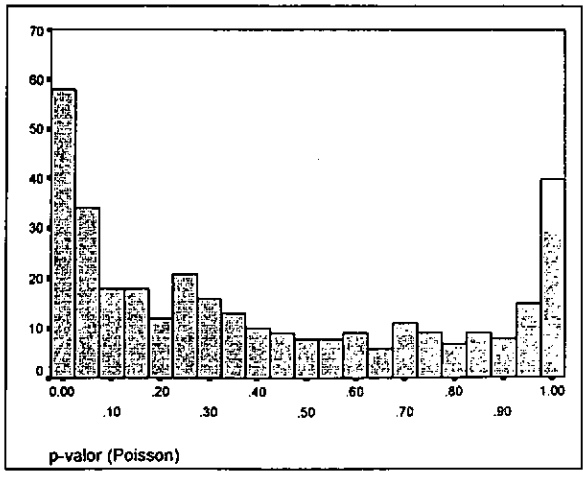

Figura A-13: Histograma p-valores test Kolmogorov-Smirnov para la Poisson

analizado (ver Figura A-11), la primera parte del estudio se centrará en la división de los datos en variables aleatorias formadas por los datos correspondientes a una 1 hora, confeccionando 340 series de 60 elementos cada una de ellas. A cada una de estas series se les aplicó el test de Kolmogorov-Smirnov para contrastar su ajuste a una distribución Normal o Poisson.

Los resultados obtenidos son totalmente equivalentes a los obtenidos en los dos casos previos. Por una parte, el ajuste a una distribución Normal (Figura A-12) a nivel global tiende hacia niveles de rechazo, con un p-valor medio de reducido (0,36). Por otra parte, el ajuste a un proceso de Poisson (Figura A-13) aunque presente un p-valor medio ligeramente superior (0,41), el hecho de que exista un grupo de variables aleatorias en donde el nivel de aceptación sea superior al 95% ya apunta hacia el hecho de que la distribución se pueda ajustar a una Poisson. Por consiguiente, se realiza la agrupación en intervalos de 20 minutos para estabilizar el valor medio dentro de cada variable aleatoria, obteniéndose los resultados de las siguientes figuras.

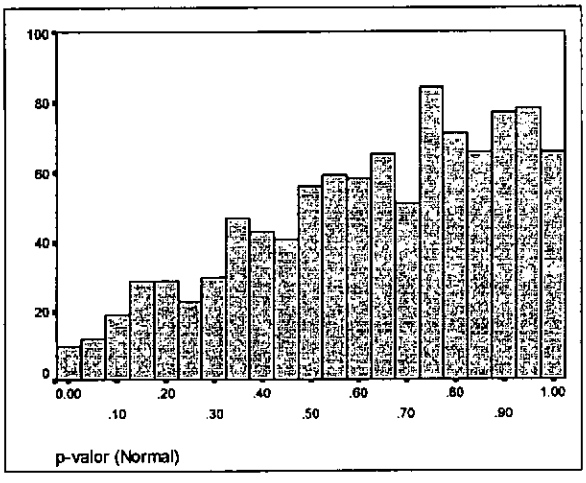

Figura A-14: Histograma p-valores test Kolmogorov-Smirnov para la Normal

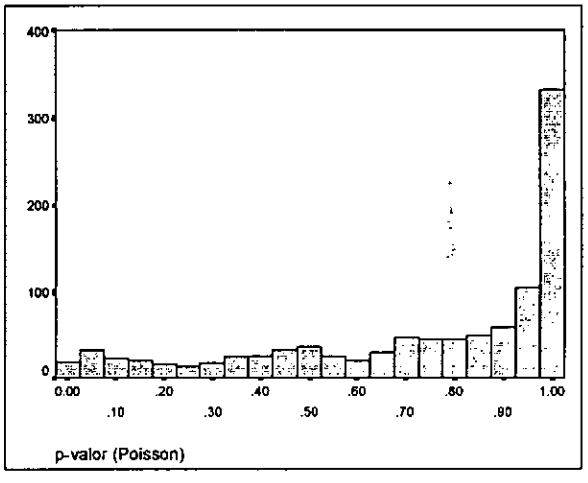

Figura A-15: Histograma p-valores test Kolmogorov-Smirnov para la Poisson

En la Figura A-15 se obtienen los resultados esperados, en donde una gran mayoría de las variables examinadas superan el test de ajuste con p-valores muy elevados, mientras que en la Figura A-14 los niveles de aceptación para el ajuste a una Normal, a pesar de ser razonablemente aceptables, no consiguen alcanzar los niveles obtenidos en el caso del ajuste a un proceso de Poisson. Para mantener la uniformidad en el análisis, también se han agrupado las variables aleatorias de accesos a documentos en función de sus valores medios. En este caso se han generado 76 grupos, y en ninguno de los casos se rechaza la hipótesis nula de ajuste a un proceso de Poisson.

En conclusión, la variable aleatoria "*Número de documentos consultados por minuto*" sigue un proceso de Poisson, en donde, su parámetro λ_{docs} es variable a lo largo del tiempo.

A.4. Relaciones entre búsquedas, categorías y documentos

En las secciones previas se ha demostrado que las búsquedas realizadas, las categorías accedidas y los documentos consultados por los usuarios son procesos de Poisson, con λ

variable en el tiempo. El siguiente paso consiste en determinar si existe alguna relación entre las medias de las tres distribuciones ya que esto permitiría aproximar el valor del resto a partir de una de ellas.

A priori parece razonable considerar que existe una relación entre las tres variables, ya que a mayor número de búsquedas, probablemente el número de documentos consultados sea mayor y se accederá a un mayor número de categorías, sencilla y básicamente porque el número de usuarios habrá aumentado. Sin embargo, es importante determinar con exactitud el tipo de relación existente e intentar formular una expresión que la defina.

El primer paso en el estudio de las relaciones existentes consiste en determinar cuál de las tres variables se considera como variable independiente. En realidad, cualquiera de las tres podría ocupar ese lugar, no obstante se ha considerado la variable aleatoria: "*Número de búsquedas por minuto*" como la más adecuada ya que desde un punto de vista lógico las búsquedas son el núcleo de un sistema de búsqueda, y por lo tanto el resto de acciones se derivan de esta (aunque el caso de los accesos a categorías no tiene porque ser considerado de esta manera). En consecuencia, se analizarán dos relaciones por separado: búsquedas-categorías y búsquedas-documentos.

Para iniciar el estudio de la relación entre búsquedas y categorías accedidas se podrían haber tomado los 20.400 puntos disponibles de cada una de las distribuciones y mostrar los pares, pero como se trata de valores discretos y debido al gran volumen su tratamiento sería complejo y además poco productivo, ya que se genera una nube de puntos con múltiples puntos apilados que aportan poca información. Por lo tanto, y precisamente al disponer de una gran cantidad de datos se ha decido tomar como base para el análisis de regresión las 1.020 medias disponibles de cada una de las variables aleatorias analizadas en las secciones anteriores.

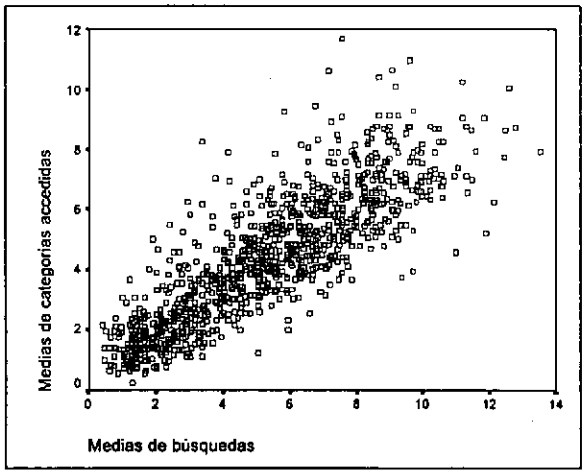

Figura A-16: Relación entre búsquedas realizadas y categorías accedidas

Esto genera una nube de puntos formada por 1.020 pares λ_{busq}-λ_{cats} de las distribuciones de búsquedas y categorías accedidas (Figura A-16). A partir de la figura se vislumbra la existencia de una relación lineal entre ambas variables, lo que puede ser analizado y determinado siguiendo un análisis de regresión simple.

Sin embargo, es necesario tener en cuenta el hecho de que existe una fuerte dependencia de los datos, que hace que se puedan asociar perfectamente a un proceso autorregresivo. Desde un punto de vista lógico, esto se produce porque si en un instante (minuto) determinado se producen 10 búsquedas es muy poco probable que en el siguiente instante se produzcan únicamente 2 búsquedas, mientras que lo razonable es que se produzcan valores cercanos a 10 búsquedas por minuto. Esto se basa en el número de usuarios conectados, ya que este número no varía de manera brusca, sino todo lo contrario. En las siguientes figuras se muestran los coeficientes de autocorrelación simple y parcial para los 1.020 valores medios de búsquedas por minuto.

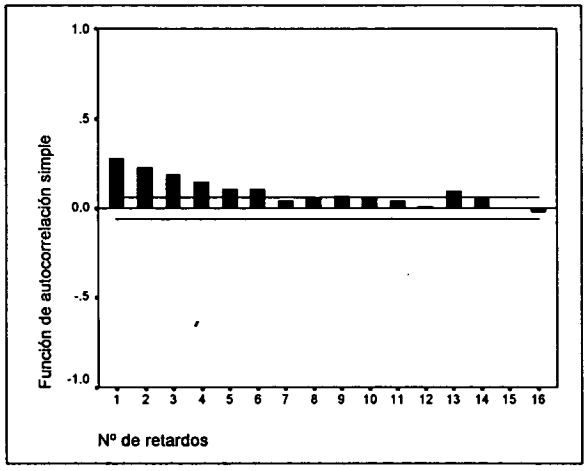

Figura A-17: Función de autocorrelación simple para las búsquedas

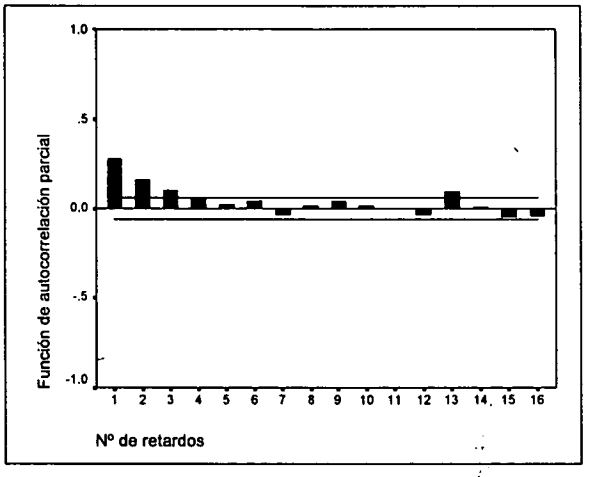

Figura A-18: Función de autocorrelación parcial para las búsquedas

En base a la Figura A-17 y a la Figura A-18 se comprueba la existencia de una fuerte relación de dependencia entre los datos analizados, en concreto y sin entrar en un análisis más profundo, se observa la presencia de un proceso autorregresivo de orden 1 ó 2.

El hecho de que los datos sean dependientes anula directamente el análisis de regresión simple, ya que los contrastes realizados dejan de tener validez. En este punto se plantean dos alternativas: utilizar métodos de regresión dinámica para intentar combinar el estudio de regresión con la serie de tiempos, aunque este proceso es bastante complejo y los resultados obtenidos serían adecuados para predicción, lo cual no es el objetivo buscado en este caso. La siguiente alternativa consiste en aleatorizar los datos muestrales para eliminar la componente de dependencia entre ellos. Esta solución es posible ya que se dispone de un número elevado de datos y por consiguiente la pérdida de información es mínima. Por lo tanto, se realiza una aleatorización de la muestra y se fijan un 10% de los datos originales, sobre los que se realiza el análisis de regresión de un modelo lineal que se describe a continuación.

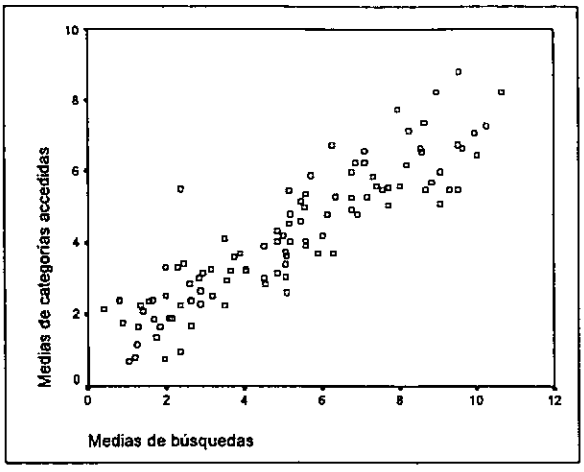

Figura A-19: Relación entre búsquedas realizadas y categorías accedidas, después de la aleatorización

En la Figura A-19 se muestra la nube de puntos que permanece después de la aleatorización y que conforma un subconjunto de la representada en la Figura A-16. En esta figura sigue percibiéndose la existencia de una clara relación lineal, por lo que el siguiente paso consiste en la realización del análisis de regresión simple sobre estos datos.

A través de este análisis se comprueba si realmente existe una relación lineal entre los datos, y se estiman los parámetros de la recta:

$$y = \beta_0 + \beta_1 x$$

En donde, en este caso, x representa λ_{busq} e y representa λ_{cats}. La hipótesis nula será que no existe una relación lineal entre ambas variables, que se chequea a través de la Tabla A-1 de análisis de la varianza (ANOVA):

	Suma de cuadrados	Grados de libertad	Cociente
Variabilidad explicada	299,19475	1	299,19475
Variabilidad no explicada	70,80015	102	0,69412
	F = 431,04238	**Signif. F = 0,0000**	

Tabla A-1: ANOVA para relación búsquedas-categorías

Como se observa en la tabla el nivel de significación de la F es inferior a 1 milésima, por lo que se debe rechazar la hipótesis nula, o lo que es lo mismo, se acepta que haya una relación lineal entre las búsquedas por minuto y las categorías visitadas por minuto. Además, otro factor importante para medir la relación existente es el coeficiente de correlación[§], que en este caso toma un valor muy elevado, concretamente del 80%.

El siguiente paso consiste en la estimación de los parámetros β_0 y β_1, que se muestra en la Tabla A-2, junto con el gráfico de ajuste a la nube de puntos (Figura A-20).

[§] El coeficiente de correlación da una medida del grado de importancia de la relación en la variabilidad de los datos. Cuanto mayor sea el valor del coeficiente de correlación mayor porcentaje de la variabilidad es explicado por la relación existente.

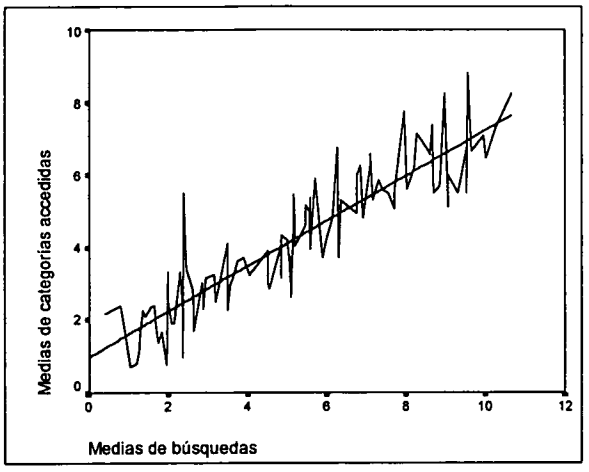

Figura A-20: Relación lineal de búsquedas realizadas y categorías accedidas

	Valor	T	Signif. T
β_1	0,626478	20,762	0,0000
β_0	0,975086	5,557	0,0000

Tabla A-2: Resolución modelo de regresión lineal búsquedas-categorías

Como se observa en la tabla el valor estimado de β_1 es de 0,626 y el valor estimado para la constante es de 0,98. La última columna de la tabla indica hasta que punto se pueden considerar distintas de cero las estimaciones, en este caso, la probabilidad de que alguna de las estimaciones sea cero es inferior a una milésima.

Por lo tanto, se puede concluir que existe una relación lineal entre las búsquedas realizadas por los usuarios y las categorías visitadas, y que dicha relación se ajusta a la siguiente expresión:

$$\lambda_{cats} = 0{,}626478\lambda_{busq} + 0{,}975086$$

Finalmente, para ultimar el análisis de regresión simple de búsquedas y categorías es necesario realizar un estudio de los residuos obtenidos durante la estimación, comprobando que se mantienen las hipótesis de normalidad, homocedasticidad e independencia.

En primer lugar se le ha aplicado el test de Kolmogorov-Smirnov a los residuos obtenidos del ajuste lineal entre búsquedas y categorías, obteniendo un p-valor del 56%, por lo que se acepta la hipótesis nula de que se ajustan a una normal. Además, en la Figura A-21 se muestra el histograma de los residuos, junto con la curva de la normal, en donde se puede comprobar que se encuentran centrados en el cero.

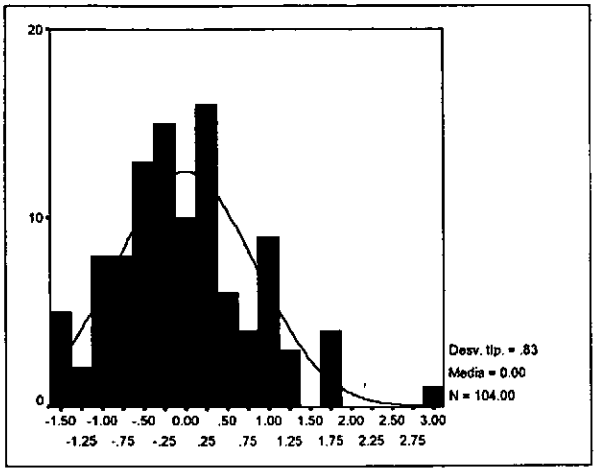

Figura A-21: Histograma de los residuos del modelo relacional búsquedas-categorías

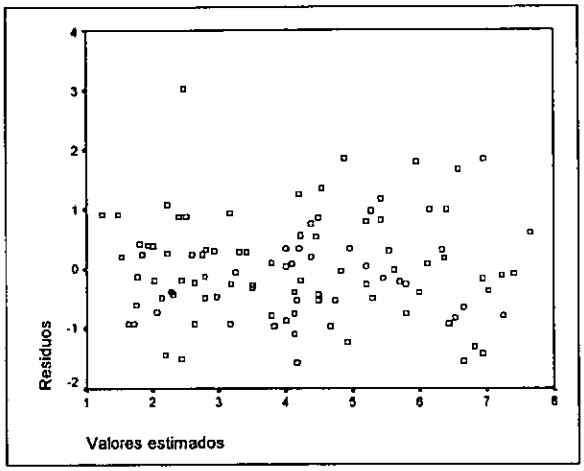

Figura A-22: Residuos frente a valores estimados (búsquedas-categorías)

A través de Figura A-22 se observa que los residuos están correctamente centrados en el cero, y que no se detecta heterocedasticidad, ya que la variabilidad en los residuos frente a los valores estimados es constante en toda la secuencia. Por último, se han analizado los coeficientes de correlación simple y parcial y todos los valores se encuentran por debajo de los límites (no se muestran los gráficos para no abrumar al lector con información redundante), por lo que los residuos se pueden considerar normales de media cero, de varianza constante e independientes, completando el análisis de regresión simple de búsquedas y categorías.

Como punto final a este primer análisis, indicar que se ha repetido el procedimiento considerando distintos porcentajes de aleatorización (del 5%, 10% y 15%) que confirman la existencia de la relación lineal y su ajuste a la expresión obtenida.

La segunda parte del estudio se centra en la relación existente entre búsquedas y documentos, siguiendo la pauta establecida por el análisis de regresión anterior. En primer lugar se muestra la nube de puntos de pares búsquedas-documentos consultados (Figura A-23), para las medias de las 1.020 distribuciones analizadas en las secciones anteriores.

Debido al problema de dependencia identificado en el caso anterior, se realiza directamente la aleatorización antes de iniciar el análisis de regresión simple. En la Figura A-24 se muestra la nube de puntos una vez realizada la aleatorización, sobre la que se estimará la relación lineal existente entre búsquedas y documentos examinados.

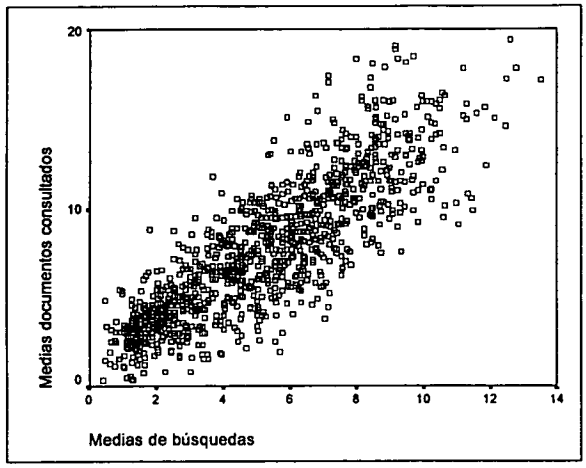

Figura A-23: Relación entre búsquedas realizadas y documentos consultados

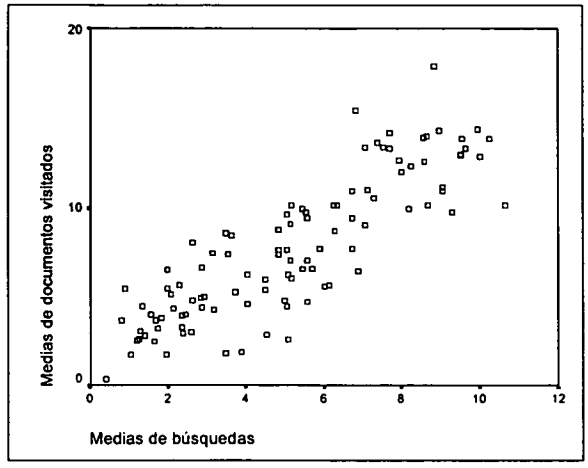

Figura A-24: Relación entre búsquedas realizadas y documentos consultados, tras la aleatorización

Al igual que en el caso anterior, el primer paso del análisis regresión simple consiste en determinar la existencia de una relación lineal entre ambas variables, utilizando para ello la Tabla A-3. En esta tabla ANOVA se comprueba si no existe ninguna relación entre ambas variables (hipótesis nula), que debe ser rechazada con una probabilidad menor de una milésima.

	Suma de cuadrados	Grados de libertad	Cociente
Variabilidad explicada	1.175,9776	1	1.175,9776
Variabilidad no explicada	433,7446	102	4,2524
	F = 276,54455	**Signif. F = 0,0000**	

Tabla A-3: ANOVA para relación búsquedas-documentos

En este caso, el valor del coeficiente de correlación es de un 73%, lo que confirma la relación lineal existente. Por consiguiente, se realiza la estimación de los parámetros, obteniendo los valores mostrados en la Tabla A-4. Como resultado, se puede afirmar que existe una relación lineal entre las búsquedas realizadas por los usuarios y los documentos consultados que se ajusta a la siguiente expresión matemática:

$$\lambda_{docs} = 1{,}242020\lambda_{busq} + 1{,}294434$$

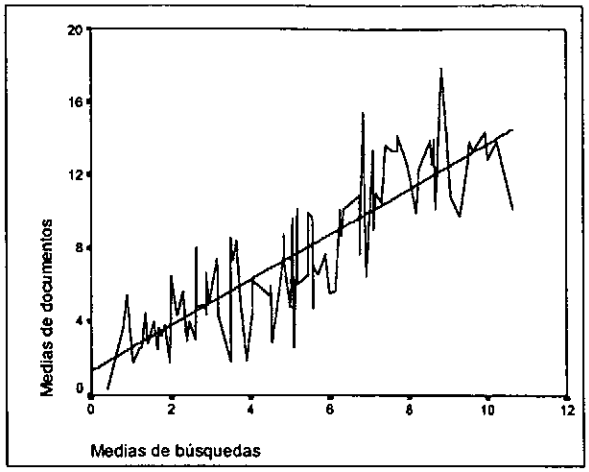

Figura A-25: Relación lineal de búsquedas realizadas y documentos consultados

	Valor	T	Signif. T
β_1	1,242020	16,630	0,0000
β_0	1,294434	2,98	0,00360

Tabla A-4: Resolución modelo de regresión lineal búsquedas-documentos

En último lugar, se comprobará que los residuos del modelo ajustado mantienen las hipótesis de normalidad, homocedasticidad e independencia para confirmar los resultados obtenidos.

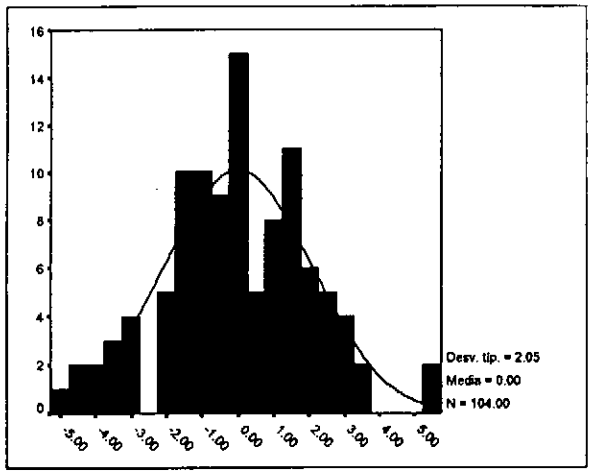

Figura A-26: Histograma de los residuos del modelo relacional búsquedas-documentos

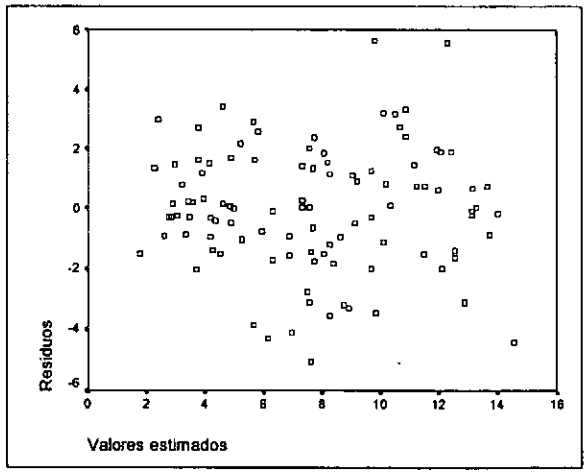

Figura A-27: Residuos frente a valores estimados (búsquedas-documentos)

En la Figura A-26 se muestra el histograma de los residuos obtenidos, en donde se puede comprobar que la media obtenida es cero. Por otra parte, el test de Kolmogorov-Smirnov confirma el ajuste a una Normal (con un p-valor superior al 97%). En la Figura A-27 se comprueba que los residuos tienen varianza constante al no darse fluctuaciones en la varianza en función de los valores estimados, y por último, las funciones de

autocorrelación simple y parcial se ajustan dentro de los límites, garantizando la independencia de los residuos analizados.

En este caso también se ha repetido el análisis de regresión considerando diferentes aleatorizaciones, y en todos los casos se confirmaba la relación lineal existente y el modelo matemático obtenido.

En resumen, se ha demostrado en primer lugar que las tres variables aleatorias discretas: "*Número de búsquedas por minuto*", "*Número de categorías accedidas por minuto*" y "*Número de documentos consultados por minuto*" se ajustan a un proceso de Poisson, o lo que es lo equivalente, que las variables aleatorias continuas: "*Tiempo entre búsquedas*", "*Tiempo entre accesos a categorías*" y "*Tiempo entre consultas a documentos*" se ajustan a una distribución Exponencial. Y en ambos casos, los parámetros λ_{busq}, λ_{cats} y λ_{docs} son variables a lo largo del tiempo.

Y en segundo lugar se ha demostrado que existe una relación lineal entre los parámetros λ_{busq}, λ_{cats} y λ_{docs} definida según las siguientes expresiones matemáticas que las relacionan:

$$\lambda_{cats} = 0.626478\lambda_{busq} + 0.975086$$

$$\lambda_{docs} = 1.242020\lambda_{busq} + 1.294434$$

Apéndice B: GUÍA DE USUARIO USIM

En este manual de usuario se muestra el modo de utilizar la herramienta de simulación desarrollada para la evaluación de sistemas de recuperación de información en Internet. Se detallan cada una de las distintas opciones que se encuentran disponibles en el visor.

B.1. Arranque de la aplicación

El programa se debe lanzar como una aplicación *standalone* Java, siendo necesario disponer del JDK 1.2, o superior. La línea que hay que teclear para el inicio de la aplicación es: java USim

La aplicación dispone de dos opciones que se activan desde la línea de comandos al arrancar el programa:

- *<fichero de configuración>*: se puede indicar el nombre de un fichero de configuración a partir del cual se leerá la información contenida en el mismo. Esta configuración será la base utilizada para ser mostrada en la interfaz gráfica (en caso de estar activada) y/o iniciar el proceso de simulación.
- *-start*: indica a la aplicación que no inicie interfaz gráfica y se comience inmediatamente el proceso de simulación. Debe ser empleada con la anterior opción, para establecer la configuración de la simulación a realizar.

En caso de ser iniciada sin interfaz gráfica, la aplicación finaliza una vez que ha finalizado el tiempo asignado a la simulación. En las siguientes secciones se describen cada uno de los componentes de la interfaz gráfica.

B.2. Ventana principal de USim

Al iniciar la aplicación se muestra la siguiente ventana (ver Figura B-1), con la etiqueta de configuración general activada por defecto. En caso de que no se haya indicado ningún fichero de configuración cada una de las opciones tomarán los valores por defecto, mientras que en caso de haber indicado un fichero de configuración aparecerán los valores especificados.

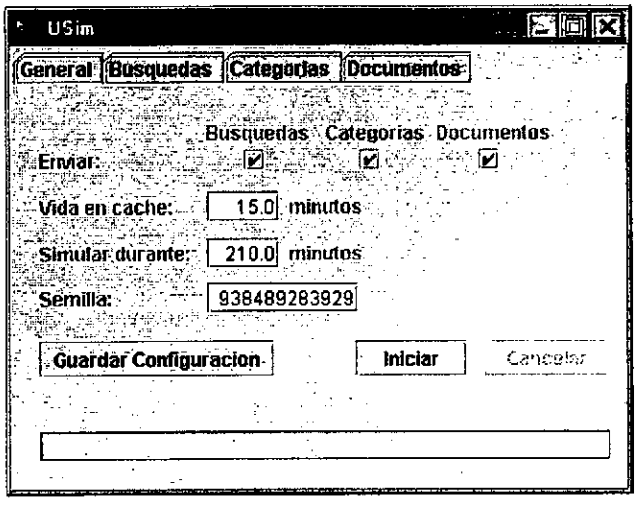

Figura B-1: Configuración general de USim

B.2.1. Configuración general

En la Figura B-1 se muestra el apartado de configuración general. A continuación se describen en detalle cada una de las opciones disponibles en el mismo.

B.2.1.1. Enviar

La opción *Enviar* permite determinar el tipo de peticiones que se van a generar durante la simulación. Obviamente, inicialmente se encuentran activadas las tres opciones (Búsquedas, Categorías y Documentos) pudiendo activar o desactivar cualquiera de las combinaciones posibles. Cada opción que es desactivada implica que también será desactivada la etiqueta correspondiente para impedir la modificación de los parámetros asociados. En caso de no seleccionar ninguna opción, el botón de inicio se desactiva impidiendo la ejecución de la simulación al no disponer de peticiones a enviar.

B.2.1.2. Vida en cache

Esta opción permite especificar el tiempo que permanecerán los identificadores de categorías y documentos en el registro respectivo durante la simulación (ver sección 3.2.1). El valor por defecto que toma este campo es el valor aproximado de la duración de una sesión media de usuario, aproximadamente 9 minutos y 30 segundos.

B.2.1.3. Simular durante

La opción *Simular durante* indica el tiempo que durará la simulación en minutos. Una vez que se inicia la simulación se activa un cronómetro encargado de examinar el tiempo transcurrido hasta el cumplimiento del tiempo especificado. Además, en la parte inferior se puede observar una barra de progreso que muestra la evolución del proceso de simulación.

B.2.1.4. Semilla

Indica la semilla base que se utilizará para la generación de todos los valores aleatorios empleados durante la simulación.

B.2.1.5. Guardar Configuración

Esta opción abre un cuadro de diálogo en donde se puede seleccionar el directorio y el nombre del archivo en donde se desea guardar la configuración actual.

B.2.1.6. Iniciar

El botón *Iniciar* se encarga de iniciar el proceso de simulación. Sólo estará activo en caso de que al menos una de las tres opciones de *Enviar* se encuentre seleccionada. Al comenzar la simulación se desactivan todos los componentes de la interfaz gráfica, excepto el botón Cancelar. Además en la barra de progreso de la parte inferior se muestra el avance y el tiempo restante de la simulación.

B.2.1.7. Cancelar

El botón Cancelar se encuentra desactivado hasta que da comienzo la simulación. A partir de ese instante, se desactiva el resto de la interfaz gráfica salvo este botón que permite cancelar la simulación en cualquier momento.

La cancelación no es inmediata, sino que en ese instante se envía una señal para que no dirigir más peticiones y se espera hasta el remate de todas las peticiones pendientes de finalización.

B.2.2. Configuración de búsquedas

En la Figura B-2 se muestra la parte que permite configurar las opciones correspondientes a la simulación de las búsquedas enviadas al sistema de recuperación de información.

A continuación se describen de forma detallada cada una de las opciones disponibles.

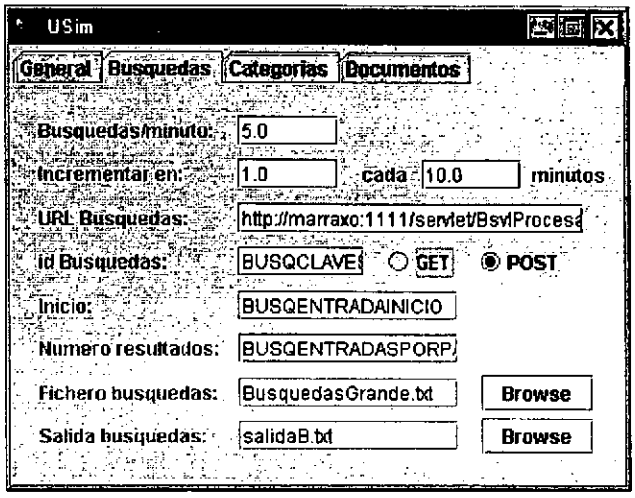

Figura B-2: Configuración de búsquedas de USim

B.2.2.1. Búsquedas/minuto

Esta opción permite establecer el valor inicial de λ_{busq}, esto es, el número de búsquedas por minuto que se van a generar. El valor indicado puede permanecer estático durante toda la simulación o bien variar periódicamente en función de los valores de la siguiente opción.

B.2.2.2. Incrementar en ... cada

Esta opción permite especificar cuánto se va a incrementar el valor inicial de λ_{busq} y cada cuanto tiempo. En caso de que el tiempo indicado sea de cero minutos no se producirá incremento alguno (independientemente del valor indicado en *Incrementar en*) y el valor de λ_{busq} permanecerá estático durante todo el proceso de simulación.

B.2.2.3. URL Búsquedas

Este campo representa la URL en donde se encuentra el servicio de búsquedas. El servicio puede estar implementado en cualquier tecnología accesible a través del World Wide Web, típicamente CGIs o Java Servlets.

Las siguientes opciones establecen los nombres de los parámetros que recibirá la URL anterior, mientras que los valores serán los generados durante la simulación.

B.2.2.4. id Búsquedas

Representa el nombre que recibe el parámetro en donde se indica la cadena de búsqueda.

B.2.2.5. Inicio

Representa el nombre que recibe el parámetro en donde se indica la posición de inicio de los resultados que se van a obtener en la búsqueda.

B.2.2.6. Número resultados

Representa el nombre del parámetro en donde se indica el número de resultados que se van a obtener en la búsqueda.

B.2.2.7. Método: GET o POST

Permite establecer la forma en la que serán pasados los parámetros al servicio de búsqueda. La herramienta de simulación soporta los dos tipos básicos: GET y POST.

B.2.2.8. Fichero búsquedas

En este atributo se indica el nombre del archivo que contiene la distribución empírica de las búsquedas que se van a utilizar durante la simulación. El archivo consta de una lista de cadenas de búsqueda y frecuencias separadas por el carácter ';'. En función del total de cadenas de búsqueda y frecuencias procesadas, USim se encarga de generar de manera aleatoria las cadenas de búsqueda empleadas en la simulación.

A continuación se muestra un ejemplo de una línea de un fichero de búsquedas:

```
;gran hermano;663
```

A través del botón *Browse* se abre un cuadro de diálogo que permite seleccionar cómodamente un archivo para este campo.

B.2.2.9. Salida búsquedas

Este atributo representa el nombre del fichero en donde se almacenará la salida producida por el simulador sobre las peticiones de búsquedas generadas y la información obtenida en la respuesta.

A continuación se muestra una línea obtenida durante una simulación:

```
13:44:13 /servlet/BsvlProcesaBusqueda gran+hermano 1 10 0 10 1210 4 2122
5 45
```

La información archivada para cada petición es la siguiente (se indica por orden de aparición):

- La hora en la que ha sido recibida la respuesta a la petición.
- La URL del servicio que atendió a la petición.

- La cadena de búsqueda enviada (en formato adecuado para su procesamiento por un servidor Web).
- La posición de inicio solicitada para los resultados.
- El número de resultados solicitados.
- El número de categorías obtenidas en la respuesta a la petición.
- El número de resultados mostrados en esa pantalla de resultados.
- El número de resultados totales coincidentes con la petición enviada.
- El número de imágenes incluidas en la página Web de respuesta.
- El tiempo (en milisegundos) en obtener la página Web de respuesta, sin tener en cuenta las imágenes.
- El tiempo empleado en el análisis de la página de respuesta.
- El tiempo empleado en la descarga de todas las imágenes incluidas en la página Web de respuesta.

El botón *Browse* abre un cuadro de diálogo que permite seleccionar cómodamente un nombre de archivo para este campo.

B.2.3. Configuración de categorías

La pantalla que permite realizar la configuración de los accesos a categorías se muestra en la Figura B-3.

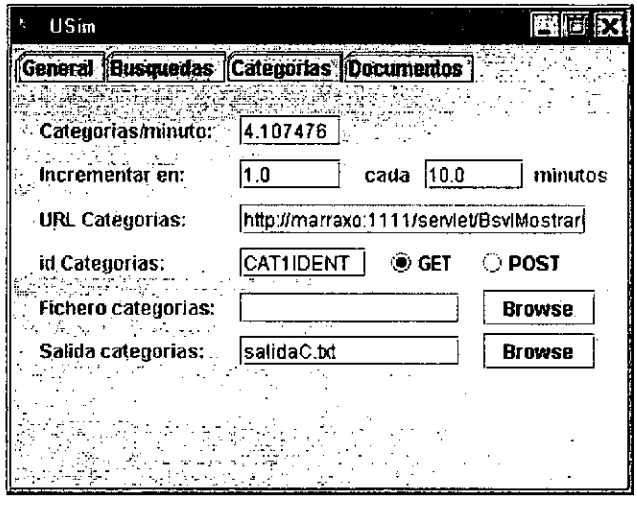

Figura B-3: Configuración de accesos a categorías de USim

A continuación se describen en detalle cada uno de los atributos.

B.2.3.1. Categorías/minuto

Esta opción indica el valor inicial de λ_{cats} y se calcula automáticamente a partir del valor establecido para λ_{busq}, a partir de la relación lineal establecida en la sección 2.7.4. En caso de querer emplear un valor diferente basta con situarse sobre la caja de texto y escribir el nuevo valor.

B.2.3.2. Incrementar en ... cada

Esta opción permite especificar cuánto se va a incrementar el valor inicial de λ_{cats} y cada cuanto tiempo. Los valores tomados en cada uno de los campos son obtenidos directamente de los indicados en los campos correspondientes de la configuración de búsquedas (ver sección B.2.2.2.).

B.2.3.3. URL Categorías

Este campo representa la URL en donde se encuentra el servicio de navegación por categorías. El servicio puede estar implementado en cualquier tecnología accesible a través del World Wide Web, típicamente CGIs o Java Servlets.

B.2.3.4. id Categorías

Representa el nombre que recibe el parámetro en donde se indica el identificador de la categoría a la que se desea acceder.

B.2.3.5. Método: GET o POST

Permite establecer la forma en la que serán pasados los parámetros al servicio de navegación por categorías. Se dispone de los dos tipos principales: GET y POST.

B.2.3.6. Fichero categorías

Este atributo normalmente no se considerará, ya que los identificadores de las categorías empleados en la simulación se obtendrán de los diferentes accesos realizados tanto a búsquedas como a otras categorías, de donde se obtendrán nuevos identificadores de categorías sobre los que realizar consultas.

B.2.3.7. Salida categorías

Este atributo almacena el nombre del fichero en donde se almacenará la salida producida por el simulador sobre las peticiones de accesos a categorías generadas y la información obtenida en la respuesta.

A continuación se muestra una línea obtenida durante una simulación:

```
13:39:23 /servlet/BsvlMostrarCategoria?CAT1IDENT=412 74 20 80 2 2014 9 18
```

La información archivada para cada petición es la siguiente (se indica por orden de aparición):

- La hora en la que ha sido recibida la respuesta a la petición.
- La URL del servicio que atendió a la petición (como se puede observar, en caso de utilizar el método GET los parámetros son insertados directamente en la URL).
- El número de categorías obtenidas en la respuesta a la petición.
- El número de documentos mostrados en esa pantalla de resultados.
- El número de documentos totales pertenecientes a la categoría.
- El número de imágenes incluidas en la página Web de respuesta.
- El tiempo (en milisegundos) en obtener la página Web de respuesta, sin tener en cuenta las imágenes.
- El tiempo empleado en el análisis de la página de respuesta.
- El tiempo empleado en la descarga de todas las imágenes incluidas en la página Web de respuesta.

Al igual que en el resto de casos, el botón *Browse* abre un cuadro de diálogo para poder elegir un nombre de archivo para este campo.

B.2.4. Configuración de documentos

La Figura B-4 muestra los atributos, que se describen a continuación, que permiten configurar los accesos a documentos de la simulación.

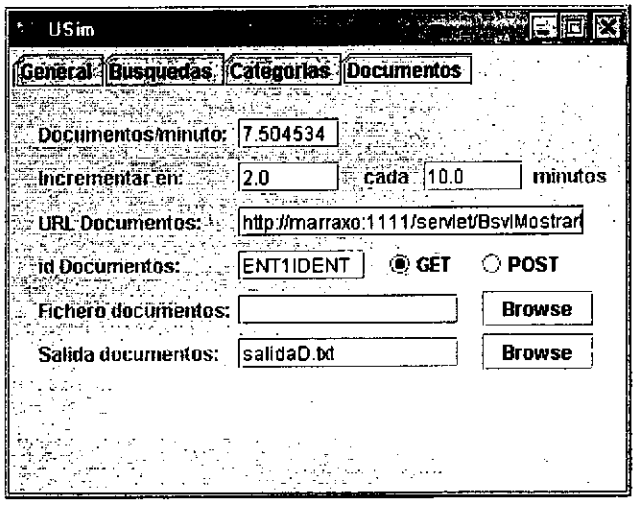

Figura B-4: Configuración de accesos a documentos de USim

B.2.4.1. Documentos/minuto

Esta opción indica el valor inicial de λ_{docs} y se calcula automáticamente a partir del valor establecido para λ_{busq}, a partir de la relación lineal establecida en la sección 2.7.4. Para modificar el valor calculado por defecto simplemente se teclea en la caja de texto el nuevo y éste queda archivado.

B.2.4.2. Incrementar en ... cada

Esta opción permite especificar cuánto se va a incrementar el valor inicial de λ_{docs} y cada cuanto tiempo. Los valores de los dos campos son obtenidos directamente de los indicados en los campos correspondientes de la configuración de búsquedas (ver sección B.2.2.2.).

B.2.4.3. URL Documentos

Este campo representa la URL en donde se encuentra el servicio de acceso a documentos. El servicio puede estar implementado en cualquier tecnología accesible a través del World Wide Web, típicamente CGIs o Java Servlets.

Es importante destacar que no todos los servicios de búsqueda en Internet disponen de un sistema para acceder a los documentos, y en cambio conectan al usuario directamente con la página final. En ese caso no es necesario realizar la configuración de documentos ya que no se está produciendo ningún tipo de carga sobre el sistema en estudio.

B.2.4.4. id Documentos

Representa el nombre que recibe el parámetro en donde se indica el identificador del documento que se desea consultar.

B.2.4.5. Método: GET o POST

Permite establecer la forma en la que serán pasados los parámetros al servicio de navegación por categorías. Se dispone de los dos tipos principales: GET y POST.

B.2.4.6. Fichero documentos

Este atributo normalmente no se considerará, ya que los identificadores de documentos empleados en la simulación se obtendrán de las diferentes páginas de resultados obtenidas tanto en búsquedas como en accesos a categorías, de donde se extraerán nuevos identificadores de documentos sobre los que realizar consultas.

B.2.4.7. Salida documentos

Este atributo almacena el nombre del fichero en donde se almacenará la salida producida por el simulador sobre las peticiones de accesos a documentos y la información obtenida en la respuesta.

A continuación se muestra una línea obtenida durante una simulación:

```
13:38:24 /servlet/BsvlMostrarEntrada?ENT1IDENT=36522 2 245 1 17
```

La información archivada para cada petición es la siguiente (se indica por orden de aparición):

- La hora en la que ha sido recibida la respuesta a la petición.
- La URL del servicio que atendió a la petición (como se puede observar, en caso de utilizar el método GET los parámetros son insertados directamente en la URL).
- El número de imágenes incluidas en la página Web de respuesta.
- El tiempo (en milisegundos) en obtener la página Web de respuesta, sin tener en cuenta las imágenes.
- El tiempo empleado en el análisis de la página de respuesta.
- El tiempo empleado en la descarga de todas las imágenes incluidas en la página Web de respuesta.

Al igual que en los casos anteriores, el botón *Browse* abre un cuadro de diálogo para poder elegir un nombre de archivo para este campo.

Apéndice C: ANÁLISIS DE TIEMPOS DE RESPUESTA

En este apéndice se presentan los análisis realizados para la evaluación del rendimiento de los diferentes modelos analizados, para los diferentes tipos de consultas analizados.

C.1. Búsquedas no restringidas

El análisis de las búsquedas no restringidas se basa en los diferentes niveles de carga analizados en la sección 5.4.1. Para cada uno de los niveles de carga se han realizado una serie de consultas, cada una de las cuales recupera un número de documentos diferente del sistema.

Los análisis realizados se basan en un análisis de la varianza en donde se examina si los factores: número de resultados de la consulta y modelo (básico, híbrido con información total y con información parcial), afectan a la variable respuesta "*Tiempo de respuesta de la consulta*". Para cada test ANOVA se ha realizado un análisis de los residuos para comprobar las hipótesis de normalidad, heterocedasticidad e independencia. En todos los casos se conservaban dichas hipótesis (aunque en la situación de carga nula se presentan ligeras anomalías), por lo que no se detallarán a continuación.

C.1.1. Carga nula

TABLA ANOVA					
	Suma de cuadrados	G.L.	Media cuadrática	F	Signficación
Modelo corregido	216799465,657	29	7475843,643	1,322	0,172
Intersección	498788071,345	1	498788071,345	88,208	0,000
Número resultados (1)	151243610,101	9	16804845,567	2,972	0,005
Modelo (2)	29959579,532	2	14979789,766	2,649	0,078
Interacción (1) y (2)	34442495,960	18	1913471,998	0,338	0,994
Error	390172100,000	69	5654668,116		
Total	1111178600,000	99			
Total corregido	606971565,657	98			
Coeficiente de correlación: 0,357					

Tabla C-1: Tabla ANOVA para búsquedas no restringidas bajo carga nula

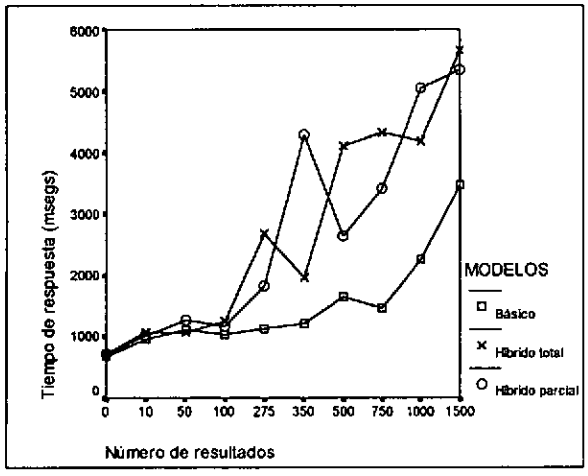

Figura C-1: Tiempos de respuesta estimados en función del número de resultados en consultas no restringidas (carga nula)

C.1.2. Carga baja

TABLA ANOVA					
	Suma de cuadrados	G.L.	Media cuadrática	F	Signficación
Modelo corregido	124557024,242	32	3892407,008	2,586	0,001
Intersección	329239309,091	1	329239309,091	218,730	0,000
Número resultados (1)	103267890,909	10	10326789,091	6,861	0,000
Modelo (2)	532145,455	2	266072,727	0,177	0,838
Interacción (1) y (2)	20756987,879	20	1037849,394	0,689	0,822
Error	99345266,667	66	1505231,313		
Total	553141600,000	99			
Total corregido	223902290,909	98			
Coeficiente de correlación: 0,556					

Tabla C-2: Tabla ANOVA para búsquedas no restringidas bajo carga baja

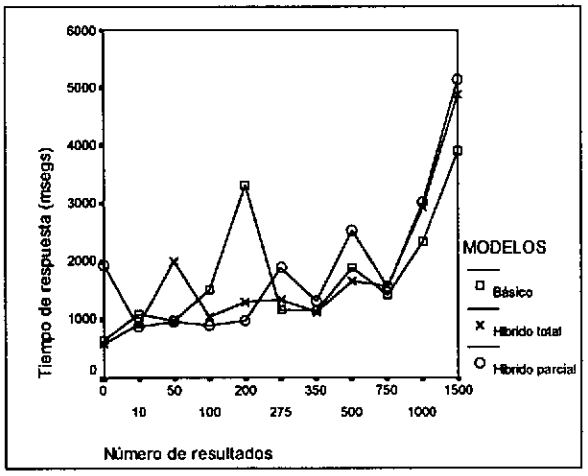

Figura C-2: Tiempos de respuesta estimados en función del número de resultados en consultas no restringidas (carga baja)

C.1.3. Carga media

TABLA ANOVA					
	Suma de cuadrados	G.L.	Media cuadrática	F	Signficación
Modelo corregido	83638472,727	32	2613702,273	2,843	0,000
Intersección	373295127,273	1	373295127,273	406,095	0,000
Número resultados (1)	67616206,061	10	6761620,606	7,356	0,000
Modelo (2)	525896,970	2	262948,485	0,286	0,752
Interacción (1) y (2)	15496369,697	20	774818,485	0,843	0,655
Error	60669200,000	66	919230,303		
Total	517602800,000	99			
Total corregido	144307672,727	98			
Coeficiente de correlación: 0,58					

Tabla C-3: Tabla ANOVA para búsquedas no restringidas bajo carga media

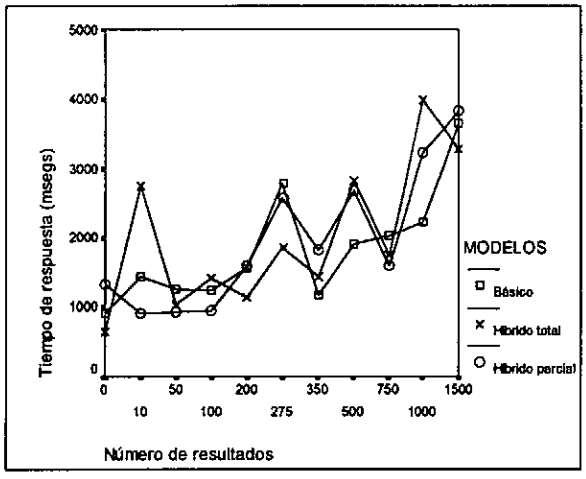

Figura C-3: Tiempos de respuesta estimados en función del número de resultados en consultas no restringidas (carga media)

C.1.4. Carga alta

TABLA ANOVA					
	Suma de cuadrados	G.L.	Media cuadrática	F	Signficación
Modelo corregido	905074468,687	32	28283577,146	1,044	0,430
Intersección	1036219031,313	1	1036219031,313	38,239	0,000
Número resultados (1)	530047624,242	10	53004762,424	1,956	0,053
Modelo (2)	19476765,657	2	9738382,828	0,359	0,699
Interacción (1) y (2)	355550078,788	20	17777503,939	0,656	0,853
Error	1788505600,000	66	27098569,697		
Total	3729799100,000	99			
Total corregido	2693580068,687	98			
Coeficiente de correlación: 0,336					

Tabla C-4: Tabla ANOVA para búsquedas no restringidas bajo carga alta

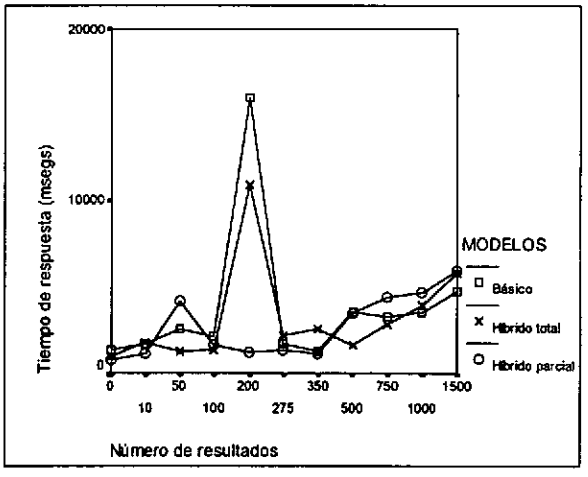

Figura C-4: Tiempos de respuesta estimados en función del número de resultados en consultas no restringidas (carga alta)

C.1.5. Carga saturada

TABLA ANOVA					
	Suma de cuadrados	G.L.	Media cuadrática	F	Signficación
Modelo corregido	6647165941,414	32	207723935,669	0,873	0,657
Intersección	4811739758,586	1	4811739758,586	20,225	0,000
Número resultados (1)	2445187119,192	10	244518711,919	1,028	0,430
Modelo (2)	9915638,384	2	4957819,192	0,021	0,979
Interacción (1) y (2)	4192063183,838	20	209603159,192	0,881	0,610
Error	15701728000,000	66	237904969,697		
Total	27160633700,000	99			
Total corregido	22348893941,414	98			
Coeficiente de correlación: 0,297					

Tabla C-5: Tabla ANOVA para búsquedas no restringidas bajo carga saturada

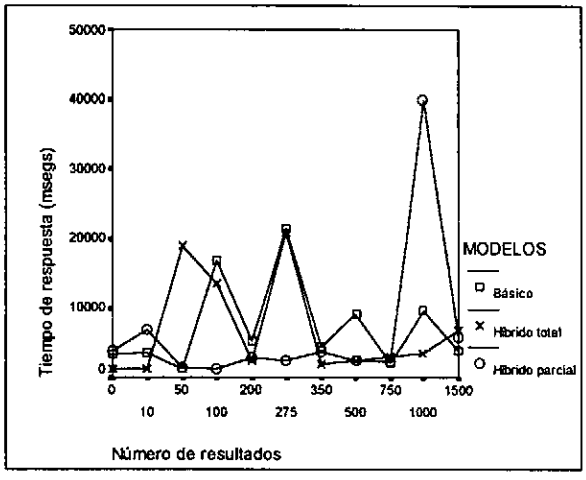

Figura C-5: Tiempos de respuesta estimados en función del número de resultados en consultas no restringidas (carga saturada)

C.2. Búsquedas restringidas

El análisis del rendimiento de las búsquedas restringidas a una categoría se centra en los cinco niveles de carga descritos: nula, baja, media, alta y saturación. Para cada uno de estos niveles de carga, se han realizado una serie de consultas restringidas a diferentes categorías.

Los experimentos han sido diseñados de tal forma que las consultas realizadas recuperasen diferentes números de documentos para estudiar los efectos de los distintos modelos bajo dichas situaciones. Y asimismo, las consultas estaban restringidas a categorías de diferentes niveles (inicialmente, los tres primeros niveles) que contenían a su vez diferentes cantidades de documentos asociados directa e indirectamente.

Los análisis realizados constan de un análisis de la varianza en donde se estudia si los factores, número de resultados de la consulta básica, número de documentos asociados a la

categoría y modelo (considerando básico, híbrido con información total e híbrido con información parcial), repercuten en la variable "*Tiempo de respuesta de la consulta*". Asimismo, se hace un análisis de varianza centrado únicamente en los modelos propuestos para los mismos factores.

En los siguientes apartados se describen los resultados obtenidos para las diferentes situaciones de carga. Simplemente, destacar que el análisis de los residuos ha sido realizado en todos los tests ANOVA realizados y en todos los casos se mantenían las hipótesis de normalidad, heterocedasticidad e independencia sobre los residuos, por lo que no se detallarán a continuación.

C.2.1. Carga nula

Análisis de la varianza considerando los modelos básicos, híbrido con información total e híbrido con información parcial.

TABLA ANOVA					
	Suma de cuadrados	G.L.	Media cuadrática	F	Signficación
Modelo corregido	13887131263,709	80	173589140,796	527,515	0,000
Intersección	3228886527,447	1	3228886527,447	9812,178	0,000
Número resultados (1)	3577033604,437	7	511004800,634	1552,879	0,000
Documentos asociados categoría (2)	4128558346,309	7	589794049,473	1792,309	0,000
Modelo (3)	280380331,092	2	140190165,546	426,020	0,000
Interacción (1) y (2)	4344942914,120	12	362078576,177	1100,311	0,000
Interacción (1) y (3)	519762146,236	14	37125867,588	112,821	0,000
Interacción (2) y (3)	754849725,360	14	53917837,526	163,849	0,000
Interacción (1), (2) y (3)	624914777,244	24	26038115,719	79,127	0,000
Error	136234694,400	414	329069,310		
Total	17736800816,000	495			
Total corregido	14023365958,109	494			
Coeficiente de correlación: 0,988					

Tabla C-6: Tabla ANOVA para búsquedas restringidas y los tres modelos bajo carga nula

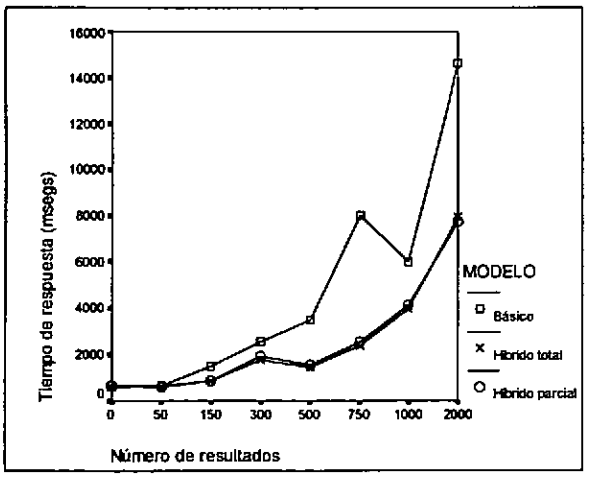

Figura C-6: Tiempos de respuesta estimados en función del número de resultados para los tres modelos (carga nula)

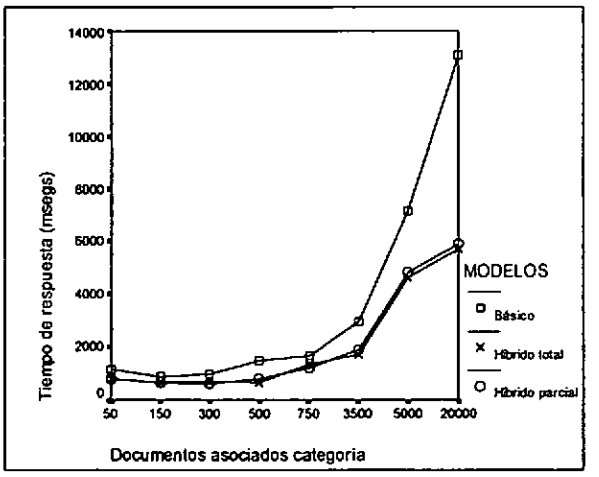

Figura C-7: Tiempos de respuesta estimados en función del número de documentos asociados a la categoría (carga nula)

Análisis de la varianza de los modelos híbridos con información total y con información parcial.

TABLA ANOVA					
	Suma de cuadrados	G.L.	Media cuadrática	F	Signficación
Modelo corregido	4362707311,124	53	82315232,285	733,178	0,000
Intersección	1354866383,922	1	1354866383,922	12067,732	0,000
Número resultados (1)	1394586846,045	7	199226692,292	1774,503	0,000
Documentos asociados categoría (2)	1372720725,294	7	196102960,756	1746,680	0,000
Modelo (3)	189077,273	1	189077,273	1,684	0,195
Interacción (1) y (2)	1702118463,635	12	141843205,303	1263,391	0,000
Interacción (1) y (3)	964819,788	7	137831,398	1,228	0,287
Interacción (2) y (3)	978504,203	7	139786,315	1,245	0,278
Interacción (1), (2) y (3)	1073806,451	12	89483,871	0,797	0,653
Error	30987027,000	276	112271,837		
Total	5899672615,000	330			
Total corregido	4393694338,124	329			
Coeficiente de correlación: 0,992					

Tabla C-7: Tabla ANOVA para búsquedas restringidas y los modelos híbridos bajo carga nula

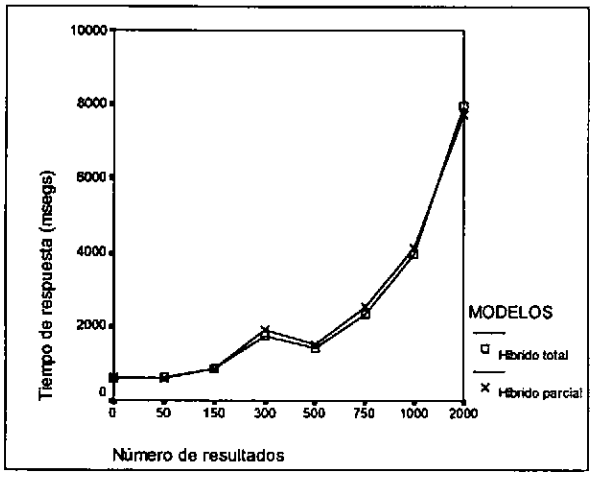

Figura C-8: Tiempos de respuesta estimados en función del número de resultados para los modelos propuestos (carga nula)

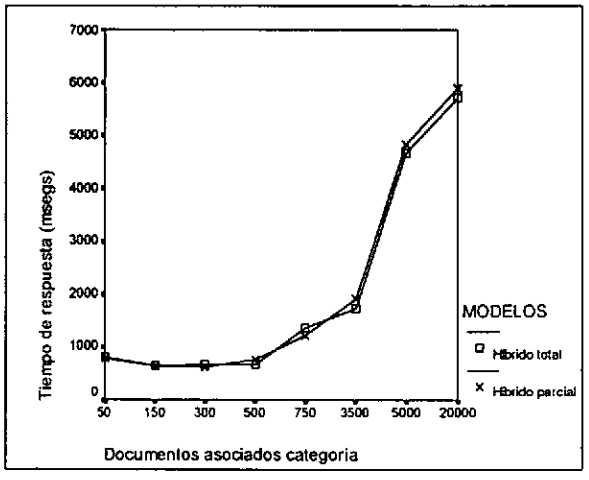

Figura C-9: Tiempos de respuesta estimados en función del número de documentos asociados a la categoría para los modelos propuestos (carga nula)

C.2.2. Carga baja

Análisis de la varianza considerando los modelos básicos, híbrido con información total e híbrido con información parcial.

TABLA ANOVA					
	Suma de cuadrados	G.L.	Media cuadrática	F	Signficación
Modelo corregido	22253816776,731	80	278172709,709	74,523	0,000
Intersección	4705235930,361	1	4705235930,361	1260,550	0,000
Número resultados (1)	6112068141,683	7	873152591,669	233,921	0,000
Documentos asociados categoría (2)	6695072468,170	7	956438924,024	256,234	0,000
Modelo (3)	338557401,891	2	169278700,945	45,350	0,000
Interacción (1) y (2)	7287077798,174	12	607256483,181	162,686	0,000
Interacción (1) y (3)	543043156,641	14	38788796,903	10,392	0,000
Interacción (2) y (3)	1044149734,636	14	74582123,903	19,981	0,000
Interacción (1), (2) y (3)	684869526,129	24	28536230,255	7,645	0,000
Error	1545331555,200	414	3732684,916		
Total	29278055261,000	495			
Total corregido	23799148331,931	494			
Coeficiente de correlación: 0,923					

Tabla C-8: Tabla ANOVA para búsquedas restringidas y los tres modelos bajo carga baja

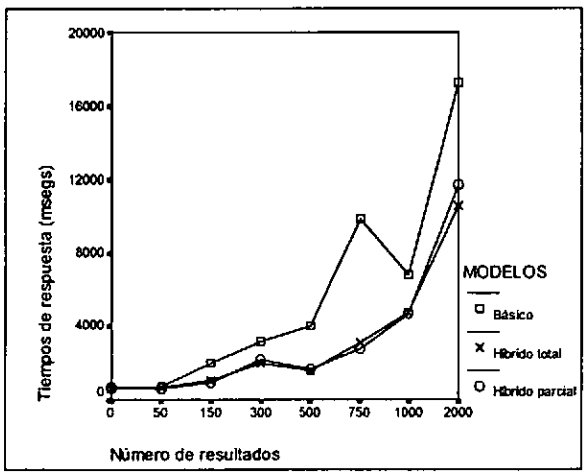

Figura C-10: Tiempos de respuesta estimados en función del número de resultados para los tres modelos (carga baja)

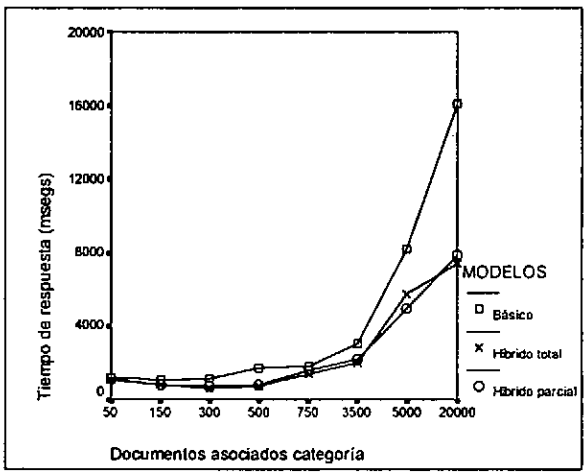

Figura C-11: Tiempos de respuesta estimados en función del número de documentos asociados a la categoría (carga baja)

Análisis de la varianza de los modelos híbridos con información total y con información parcial.

TABLA ANOVA					
	Suma de cuadrados	G.L.	Media cuadrática	F	Signficación
Modelo corregido	8441282638,824	53	159269483,751	35,215	0,000
Intersección	2068774029,195	1	2068774029,195	457,419	0,000
Número resultados (1)	2880523147,104	7	411503306,729	90,986	0,000
Documentos asociados categoría (2)	2366767106,436	7	338109586,634	74,758	0,000
Modelo (3)	140549,460	1	140549,460	0,031	0,860
Interacción (1) y (2)	3359365722,877	12	279947143,573	61,898	0,000
Interacción (1) y (3)	12940322,350	7	1848617,479	0,409	0,897
Interacción (2) y (3)	9660888,707	7	1380126,958	0,305	0,951
Interacción (1), (2) y (3)	14793301,137	12	1232775,095	0,273	0,993
Error	1248268542,900	276	4522712,112	35,215	
Total	11980441277,000	330			
Total corregido	9689551181,724	329			
Coeficiente de correlación: 0,846					

Tabla C-9: Tabla ANOVA para búsquedas restringidas y los modelos híbridos bajo carga baja

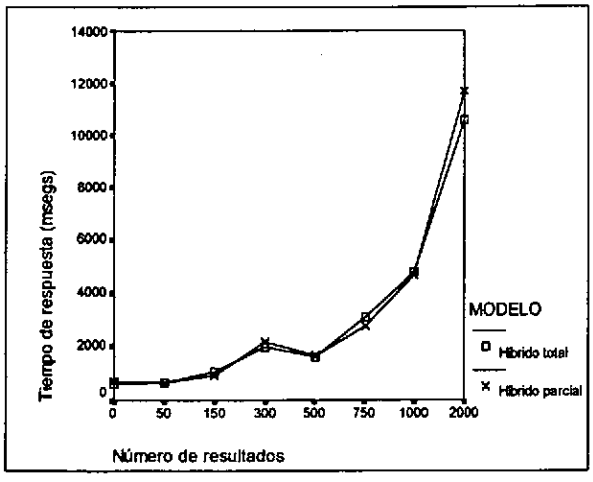

Figura C-12: Tiempos de respuesta estimados en función del número de resultados para los modelos propuestos (carga baja)

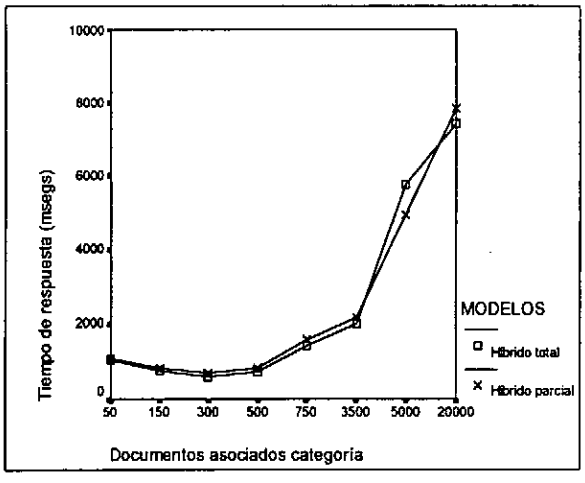

Figura C-13: Tiempos de respuesta estimados en función del número de documentos asociados a la categoría para los modelos propuestos (carga baja)

C.2.3. Carga media

Análisis de la varianza considerando los modelos básicos, híbrido con información total e híbrido con información parcial.

TABLA ANOVA					
	Suma de cuadrados	G.L.	Media cuadrática	F	Signficación
Modelo corregido	41872890957,145	80	523411136,964	72,492	0,000
Intersección	8023493137,332	1	8023493137,332	1111,244	0,000
Número resultados (1)	9877157198,050	7	1411022456,864	195,425	0,000
Documentos asociados categoría (2)	11510891215,172	7	1644413030,739	227,749	0,000
Modelo (3)	716943675,201	2	358471837,601	49,648	0,000
Interacción (1) y (2)	12145710710,569	12	1012142559,214	140,181	0,000
Interacción (1) y (3)	2372745229,012	14	169481802,072	23,473	0,000
Interacción (2) y (3)	3079455656,564	14	219961118,326	30,464	0,000
Interacción (1), (2) y (3)	3122475084,798	24	130103128,533	18,019	0,000
Error	2989195584,200	414	7220279,189		
Total	54211349506,000	495			
Total corregido	44862086541,345	494			
Coeficiente de correlación: 0,920					

Tabla C-10: Tabla ANOVA para búsquedas restringidas y los tres modelos bajo carga media

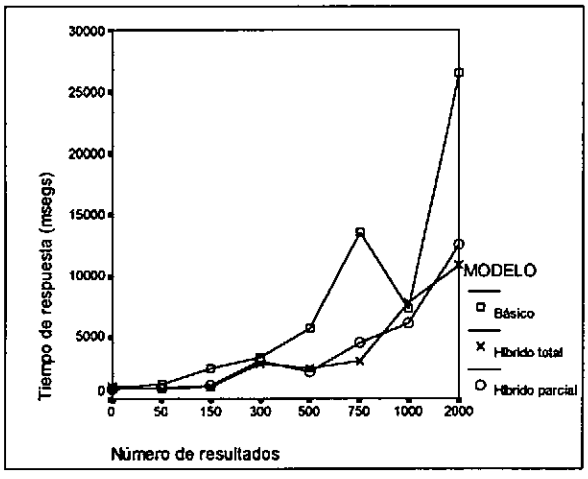

Figura C-14: Tiempos de respuesta estimados en función del número de resultados para los tres modelos (carga media)

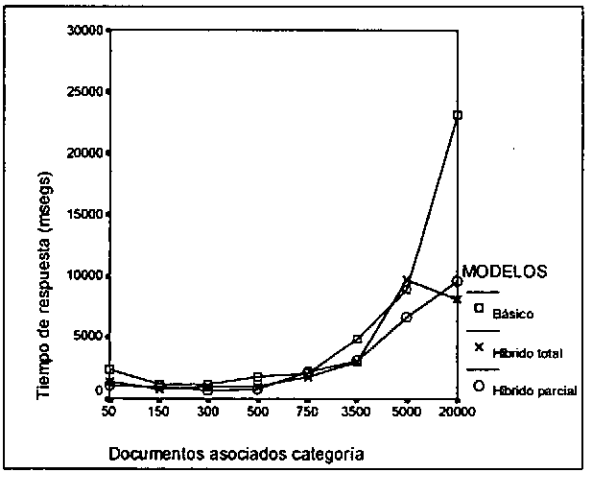

Figura C-15: Tiempos de respuesta estimados en función del número de documentos asociados a la categoría (carga media)

Análisis de la varianza de los modelos híbridos con información total y con información parcial.

TABLA ANOVA					
	Suma de cuadrados	G.L.	Media cuadrática	F	Signficación
Modelo corregido	10659925713,652	53	201130673,842	69,813	0,000
Intersección	3341242609,044	1	3341242609,044	1159,749	0,000
Número resultados (1)	3285942905,246	7	469420415,035	162,936	0,000
Documentos asociados categoría (2)	3391100209,884	7	484442887,126	168,151	0,000
Modelo (3)	1516609,989	1	1516609,989	,526	0,469
Interacción (1) y (2)	3931654939,762	12	327637911,647	113,724	0,000
Interacción (1) y (3)	54793057,350	7	7827579,621	2,717	0,010
Interacción (2) y (3)	83384158,838	7	11912022,691	4,135	0,000
Interacción (1), (2) y (3)	101670466,676	12	8472538,890	2,941	0,001
Error	795157194,800	276	2881004,329		
Total	15089361415,000	330			
Total corregido	11455082908,452	329			
Coeficiente de correlación: 0,917					

Tabla C-11: Tabla ANOVA para búsquedas restringidas y los modelos híbridos bajo carga media

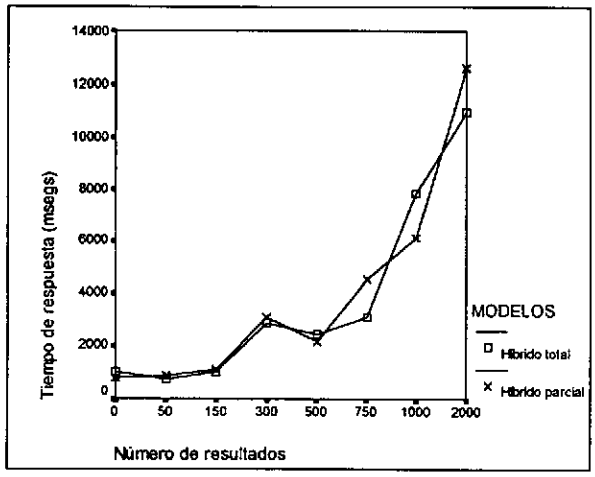

Figura C-16: Tiempos de respuesta estimados en función del número de resultados para los modelos propuestos (carga media)

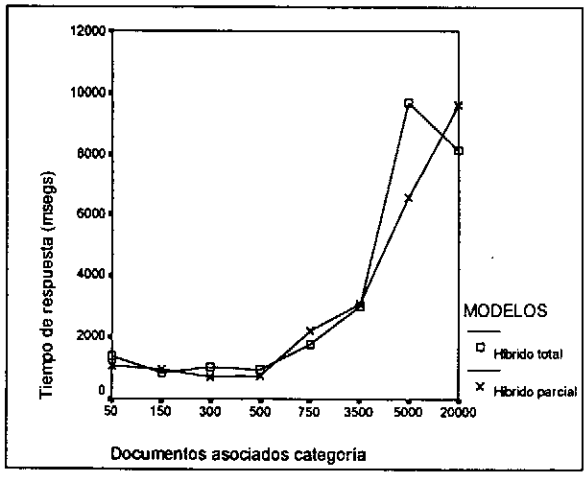

Figura C-17: Tiempos de respuesta estimados en función del número de documentos asociados a la categoría para los modelos propuestos (carga media)

C.2.4. Carga alta

Análisis de la varianza considerando los modelos básicos, híbrido con información total e híbrido con información parcial.

TABLA ANOVA					
	Suma de cuadrados	G.L.	Media cuadrática	F	Signficación
Modelo corregido	54211179030,183	80	677639737,877	37,929	0,000
Intersección	13585141260,844	1	13585141260,844	760,387	0,000
Número resultados (1)	14258680804,565	7	2036954400,652	114,012	0,000
Documentos asociados categoría (2)	13981255645,249	7	2330209274,208	130,426	0,000
Modelo (3)	873505504,835	2	436752752,418	24,446	0,000
Interacción (1) y (2)	19154844804,244	12	1473449600,326	82,472	0,000
Interacción (1) y (3)	2184801454,257	14	156057246,733	8,735	0,000
Interacción (2) y (3)	1769004280,110	14	147417023,343	8,251	0,000
Interacción (1), (2) y (3)	2579844234,711	24	99224778,258	5,554	0,000
Error	7396557693,300	414	17866081,385		
Total	75399826443,000	495			
Total corregido	61607736723,483	494			
Coeficiente de correlación: 0,857					

Tabla C-12: Tabla ANOVA para búsquedas restringidas y los tres modelos bajo carga alta

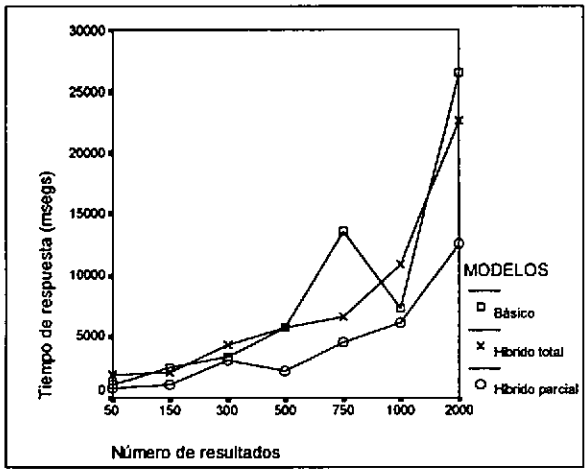

Figura C-18: Tiempos de respuesta estimados en función del número de resultados para los tres modelos (carga alta)

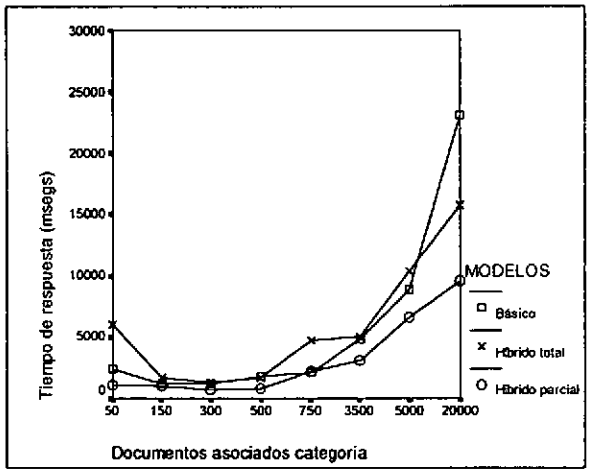

Figura C-19: Tiempos de respuesta estimados en función del número de documentos asociados a la categoría (carga alta)

Análisis de la varianza de los modelos híbridos con información total y con información parcial.

TABLA ANOVA					
	Suma de cuadrados	G.L.	Media cuadrática	F	Signficación
Modelo corregido	23731487728,391	53	447763919,404	23,754	0,000
Intersección	7359210349,050	1	7359210349,050	390,415	0,000
Número resultados (1)	5366862759,513	7	766694679,930	40,674	0,000
Documentos asociados categoría (2)	6488190916,612	7	1081365152,769	57,368	0,000
Modelo (3)	605931072,296	1	605931072,296	32,145	0,000
Interacción (1) y (2)	9300859784,738	12	715450752,672	37,956	0,000
Interacción (1) y (3)	251299227,455	7	35899889,636	1,905	0,069
Interacción (2) y (3)	467378994,962	7	77896499,160	4,133	0,001
Interacción (1), (2) y (3)	1131888017,724	12	87068309,056	4,619	0,000
Error	5202519303,900	276	18849707,623		
Total	36277838352,000	330			
Total corregido	28934007032,291	329			
Coeficiente de correlación: 0,786					

Tabla C-13: Tabla ANOVA para búsquedas restringidas y los modelos híbridos bajo carga alta

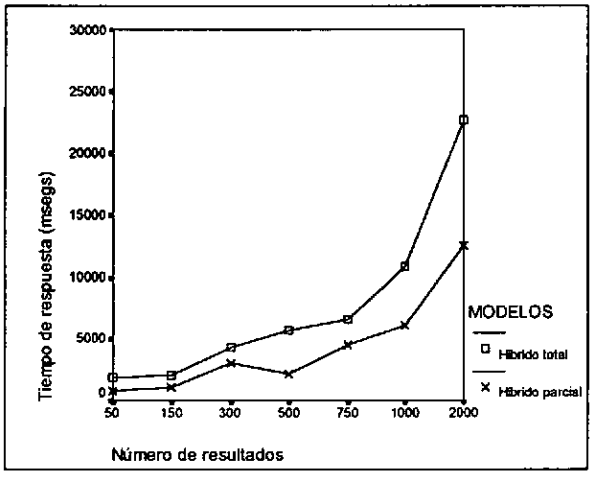

Figura C-20: Tiempos de respuesta estimados en función del número de resultados para los modelos propuestos (carga alta)

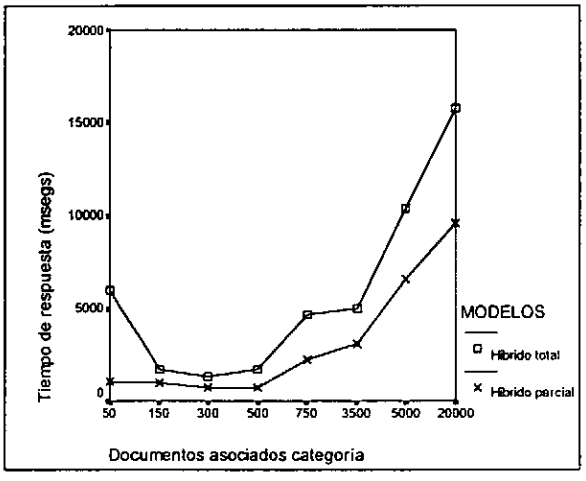

Figura C-21: Tiempos de respuesta estimados en función del número de documentos asociados a la categoría para los modelos propuestos (carga alta)

C.2.5. Carga saturada

Análisis de la varianza considerando los modelos básicos, híbrido con información total e híbrido con información parcial.

TABLA ANOVA					
	Suma de cuadrados	G.L.	Media cuadrática	F	Signficación
Modelo corregido	134783877813,700	80	1684798472,671	8,143	0,000
Intersección	56461837340,518	1	56461837340,518	272,908	0,000
Número resultados (1)	36229706631,184	7	5175672375,883	25,017	0,000
Documentos asociados categoría (2)	42583602120,392	7	6083371731,485	29,404	0,000
Modelo (3)	867397071,695	2	433698535,848	2,096	0,124
Interacción (1) y (2)	39489899115,243	12	3290824926,270	15,906	0,000
Interacción (1) y (3)	3680973127,652	14	262926651,975	1,271	0,222
Interacción (2) y (3)	6520440003,844	14	465745714,560	2,251	0,006
Interacción (1), (2) y (3)	8293206115,466	24	345550254,811	1,670	0,026
Error	85652376157,100	414	206889797,481		
Total	282987461725,000	495			
Total corregido	220436253970,800	494			
Coeficiente de correlación: 0,536					

Tabla C-14: Tabla ANOVA para búsquedas restringidas y los tres modelos bajo carga saturada

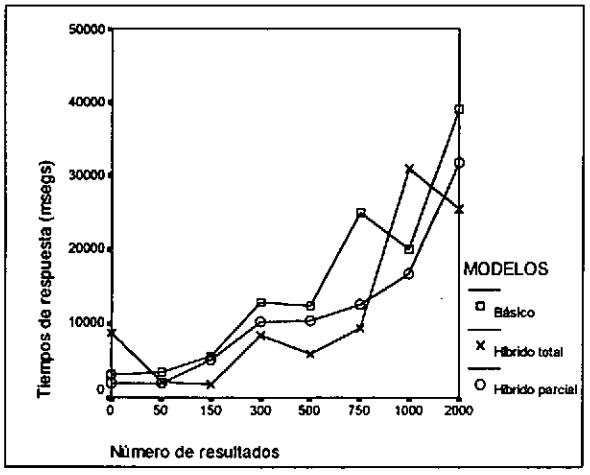

Figura C-22: Tiempos de respuesta estimados en función del número de resultados para los tres modelos (carga saturada)

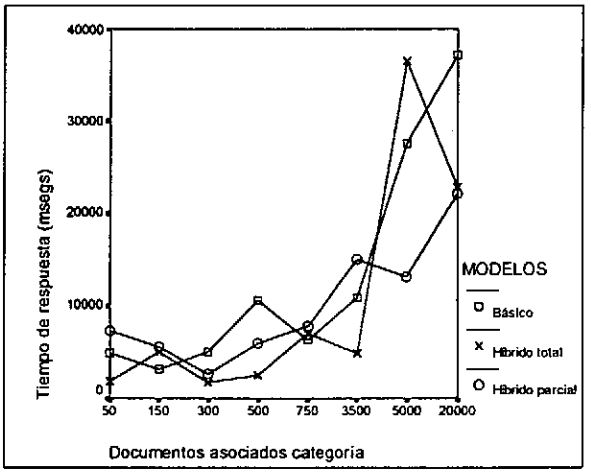

Figura C-23: Tiempos de respuesta estimados en función del número de documentos asociados a la categoría (carga saturada)

Análisis de la varianza de los modelos híbridos con información total y con información parcial.

TABLA ANOVA					
	Suma de cuadrados	G.L.	Media cuadrática	F	Signficación
Modelo corregido	69215416569,524	53	1305951256,029	5,787	0,000
Intersección	31540990447,085	1	31540990447,085	139,770	0,000
Número resultados (1)	21013440401,536	7	3001920057,362	13,303	0,000
Documentos asociados categoría (2)	18982328602,962	7	2711761228,995	12,017	0,000
Modelo (3)	17552053,149	1	17552053,149	0,078	0,781
Interacción (1) y (2)	22206688219,205	12	1850557351,600	8,201	0,000
Interacción (1) y (3)	1685736471,421	7	240819495,917	1,067	0,385
Interacción (2) y (3)	3108164599,659	7	444023514,237	1,968	0,060
Interacción (1), (2) y (3)	4269079926,788	12	355756660,566	1,576	0,098
Error	62283230809,700	276	225663879,745		
Total	165006909500,000	330			
Total corregido	131498647379,224	329			
Coeficiente de correlación: 0,435					

Tabla C-15: Tabla ANOVA para búsquedas restringidas y los modelos híbridos bajo carga saturada

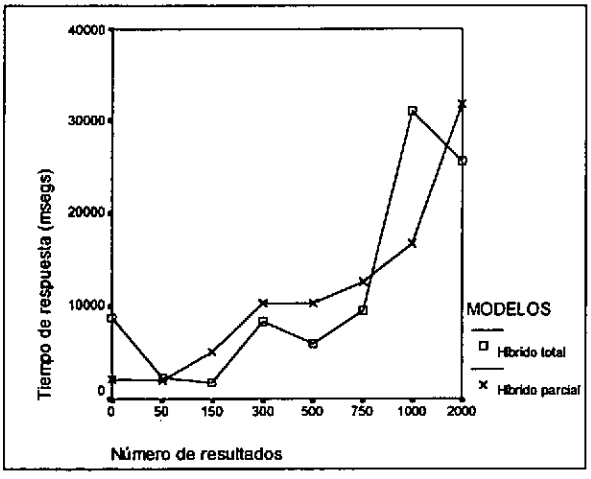

Figura C-24: Tiempos de respuesta estimados en función del número de resultados para los modelos propuestos (carga saturada)

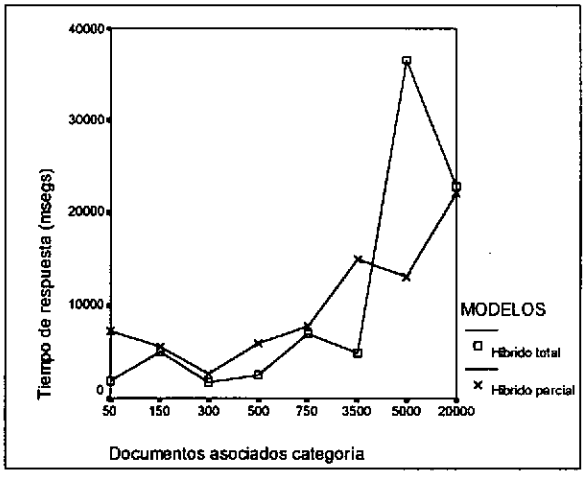

Figura C-25: Tiempos de respuesta estimados en función del número de documentos asociados a la categoría para los modelos propuestos (carga saturada)

C.2.6. Análisis de los efectos de la profundidad de las categorías

En esta sección se exponen los test ANOVA realizados con el objetivo de determinar si la profundidad de la categoría a la que se restringe la búsqueda presenta algún tipo de repercusión sobre el rendimiento de los tres modelos.

En cada análisis de la varianza se examinan tres factores: número de resultados, profundidad y modelo, frente a la variable respuesta que está constituida por el tiempo de respuesta de la búsqueda. Cada análisis ha sido realizado en las cinco situaciones de carga típicas que se detallan a continuación.

C.2.6.1. Carga nula

TABLA ANOVA					
	Suma de cuadrados	G.L.	Media cuadrática	F	Signficación
Modelo corregido	12168132932,329	44	276548475,735	2655,153	0,000
Intersección	4062413667,271	1	4062413667,271	39003,397	0,000
Número resultados (1)	677505218,676	2	338752609,338	3252,377	0,000
Profundidad categoría (2)	7183283490,773	4	1795820872,693	17241,749	0,000
Modelo (3)	502457734,116	2	251228867,058	2412,058	0,000
Interacción (1) y (2)	2352716064,347	8	294089508,043	2823,565	0,000
Interacción (1) y (3)	112854274,258	4	28213568,564	270,880	0,000
Interacción (2) y (3)	1004751290,907	8	125593911,363	1205,832	0,000
Interacción (1), (2) y (3)	334564859,253	16	20910303,703	200,761	0,000
Error	18747968,400	180	104155,380		
Total	16249294568,000	225			
Total corregido	12186880900,729	224			
Coeficiente de correlación: 0,998					

Tabla C-16: Tabla ANOVA para búsquedas restringidas, analizando la profundidad, bajo carga nula

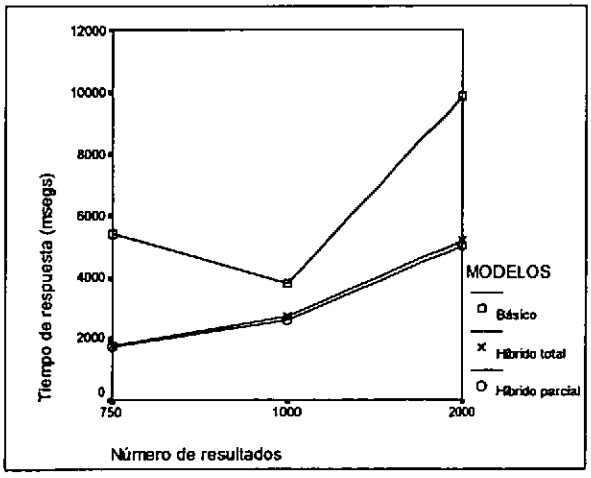

Figura C-26: Tiempos de respuesta estimados en función del número de resultados (carga nula)

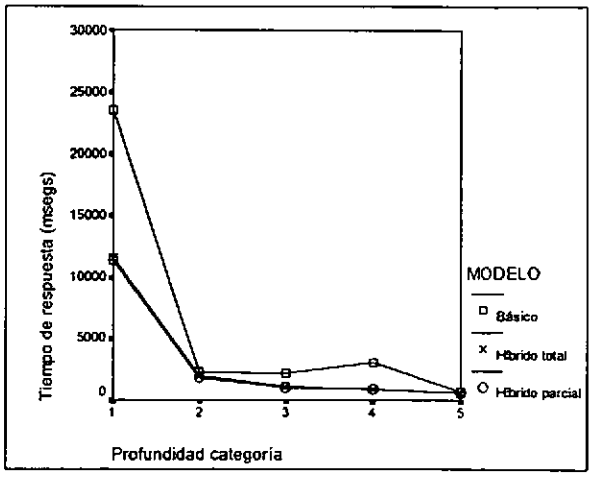

Figura C-27: Tiempos de respuesta estimados en función de la profundidad de la categoría (carga nula)

C.2.6.3. Carga media

TABLA ANOVA					
	Suma de cuadrados	G.L.	Media cuadrática	F	Signficación
Modelo corregido	28860473855,040	44	655919860,342	66,722	0,000
Intersección	10619384940,160	1	10619384940,160	1080,241	0,000
Número resultados (1)	1645362338,907	2	822681169,453	83,686	0,000
Profundidad categoría (2)	17995388020,329	4	4498847005,082	457,638	0,000
Modelo (3)	1256335413,440	2	628167706,720	63,899	0,000
Interacción (1) y (2)	5006855183,271	8	625856897,909	63,664	0,000
Interacción (1) y (3)	188121436,133	4	47030359,033	4,784	0,000
Interacción (2) y (3)	2335544783,138	8	291943097,892	29,697	0,000
Interacción (1), (2) y (3)	432866679,822	16	27054167,489	2,752	0,000
Error	1769503312,800	180	9830573,960		
Total	41249362108,000	225			
Total corregido	30629977167,840	224			
Coeficiente de correlación: 0,928					

Tabla C-18: Tabla ANOVA para búsquedas restringidas, analizando la profundidad, bajo carga media

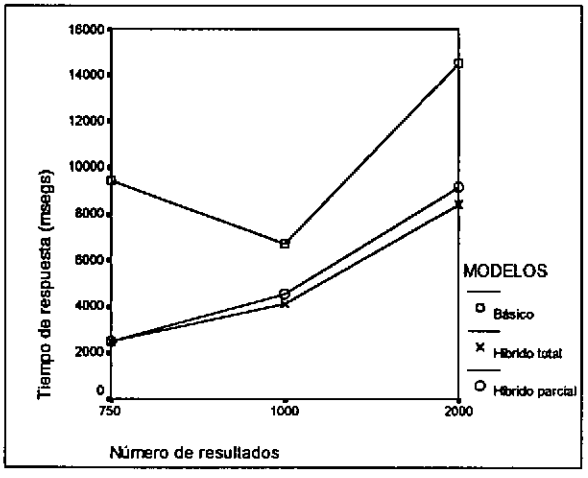

Figura C-30: Tiempos de respuesta estimados en función del número de resultados (carga media)

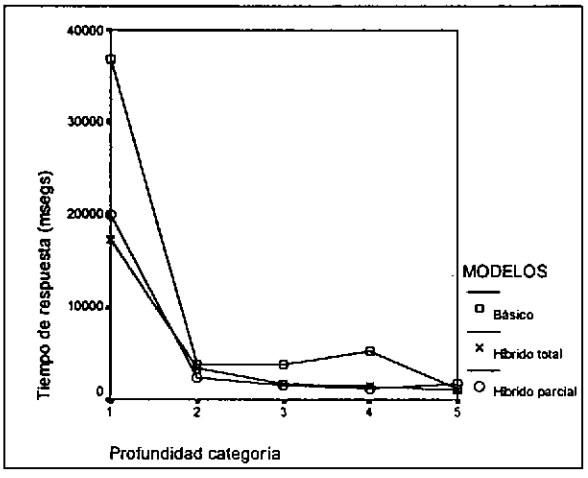

Figura C-31: Tiempos de respuesta estimados en función de la profundidad de la categoría (carga media)

C.2.6.4. Carga alta

TABLA ANOVA					
	Suma de cuadrados	G.L.	Media cuadrática	F	Signficación
Modelo corregido	93298864407,022	44	2120428736,523	14,582	0,000
Intersección	31432689653,777	1	31432689653,777	216,163	0,000
Número resultados (1)	6180831948,809	2	3090415974,404	21,253	0,000
Profundidad categoría (2)	51958102746,266	4	12989525686,567	89,329	0,000
Modelo (3)	2480911434,462	2	1240455717,231	8,531	0,000
Interacción (1) y (2)	21436968867,413	8	2679621108,427	18,428	0,000
Interacción (1) y (3)	1232174017,351	4	308043504,338	2,118	0,080
Interacción (2) y (3)	6494694851,493	8	811836856,437	5,583	0,000
Interacción (1), (2) y (3)	3515180541,227	16	219698783,827	1,511	0,100
Error	26174119485,200	180	145411774,918		
Total	150905673546,000	225			
Total corregido	119472983892,222	224			
Coeficiente de correlación: 0,727					

Tabla C-19: Tabla ANOVA para búsquedas restringidas, analizando la profundidad, bajo carga alta

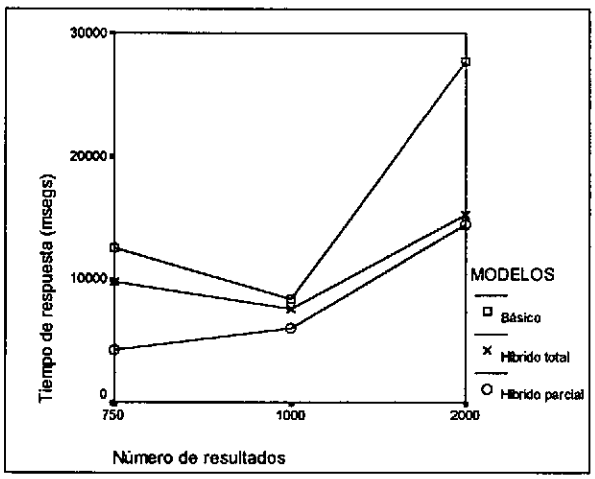

Figura C-32: Tiempos de respuesta estimados en función del número de resultados (carga alta)

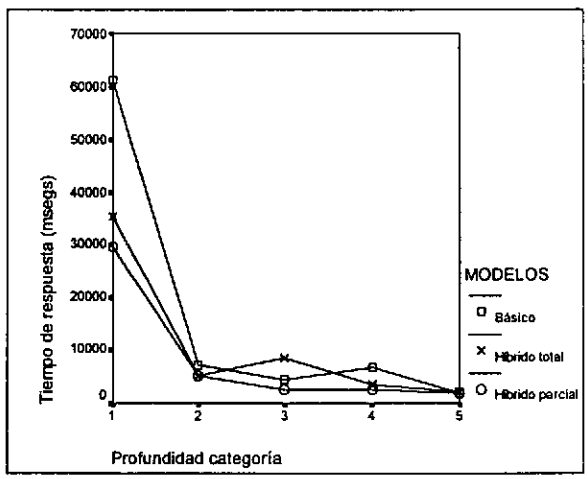

Figura C-33: Tiempos de respuesta estimados en función de la profundidad de la categoría (carga alta)

C.2.6.5. Carga saturada

TABLA ANOVA					
	Suma de cuadrados	G.L.	Media cuadrática	F	Signficación
Modelo corregido	119611188863,529	44	2718436110,535	6,492	0,000
Intersección	79500840086,468	1	79500840086,468	189,872	0,000
Número resultados (1)	2354485318,569	2	1177242659,284	2,812	0,063
Profundidad categoría (2)	73283596926,640	4	18320899231,660	43,756	0,000
Modelo (3)	12723525217,982	2	6361762608,991	15,194	0,000
Interacción (1) y (2)	11710918861,120	8	1463864857,640	3,496	0,001
Interacción (1) y (3)	4765879603,298	4	1191469900,824	2,846	0,025
Interacción (2) y (3)	12430120153,440	8	1553765019,180	3,711	0,000
Interacción (1), (2) y (3)	2342662782,480	16	146416423,905	0,350	0,991
Error	75367186398,000	180	418706591,100		
Total	274479215348,000	225			
Total corregido	194978375261,529	224			
Coeficiente de correlación: 0,519					

Tabla C-20: Tabla ANOVA para búsquedas restringidas, analizando la profundidad, bajo carga saturada

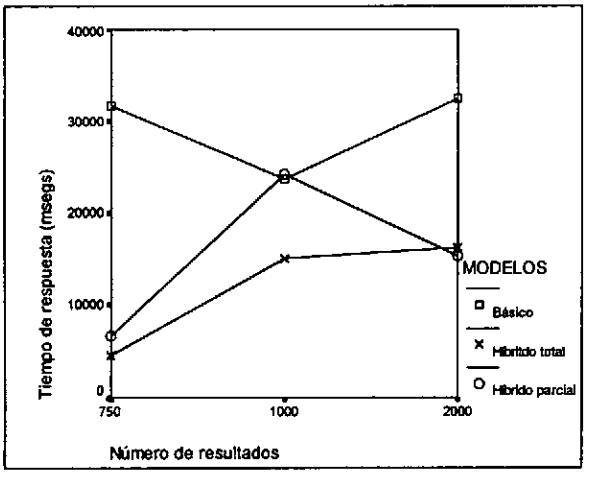

Figura C-34: Tiempos de respuesta estimados en función del número de resultados (carga saturada)

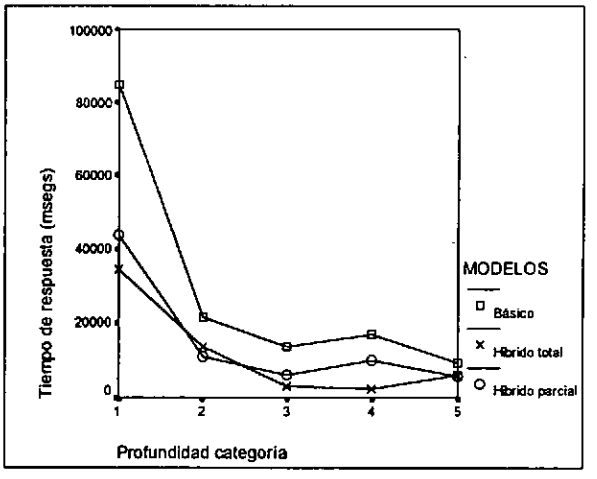

Figura C-35: Tiempos de respuesta estimados en función de la profundidad de la categoría (carga saturada)