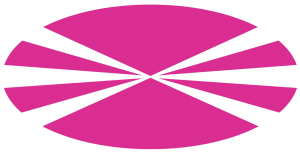


UNIVERSIDADE DA CORUÑA
DEPARTAMENTO DE COMPUTACIÓN

TÉCNICAS DE INDEXACIÓN Y RECUPERACIÓN DE DOCUMENTOS UTILIZANDO REFERENCIAS GEOGRÁFICAS Y TEXTUALES

Tesis Doctoral
A Coruña, Septiembre 2009

Doctorando: Diego Seco Naveiras
Directores: Dr. Miguel Ángel Rodríguez Luaces
Dr. José Ramón Ríos Viqueira

Tesis doctoral dirigida por:

Miguel Ángel Rodríguez Luaces
Departamento de Computación
Facultade de Informática
Universidade da Coruña
15071 A Coruña (España)
Tel: +34 981 167000 ext. 1254
Fax: +34 981 167160
luaces@udc.es

José Ramón Ríos Viqueira
Departamento de Electrónica y Computación
Instituto de Investigacións Tecnolóxicas
Universidade de Santiago de Compostela
15782 Santiago de Compostela (España)
Tel: +34 981 563100 ext. 13966, 16000
Fax: +34 981 528012
joserios@usc.es

A Lucía

Resumen

Internet y la *World Wide Web* se han convertido en un enorme repositorio de información consultado diariamente por millones de usuarios. Además, otros repositorios de información, como las bases de datos documentales o las bibliotecas digitales, también han aumentado su popularidad considerablemente. Esto ha provocado que la recuperación de información se haya convertido en una de las áreas de investigación más importantes dentro de la informática.

Aunque estos repositorios contienen información de distinta naturaleza, la información más habitual es de tipo textual. A menudo, en el texto de un documento se pueden encontrar referencias geográficas que permiten asignar a ese documento una zona del espacio en la cual es relevante. Los usuarios de los sistemas que enumerábamos demandan cada vez más servicios que les permitan situar la información recuperada en un mapa. Además, también está aumentando el interés en consultas que permitan recuperar documentos relevantes no sólo para un tema determinado sino también para una zona determinada. El desarrollo de arquitecturas de sistemas, estructuras de indexación y otros componentes que permitan satisfacer estas necesidades es el objetivo principal de una nueva área de investigación denominada recuperación de información geográfica (GIR).

En esta tesis abordamos varios temas de interés en el área. En primer lugar, las estructuras de indexación que permiten recuperar documentos empleando tanto su ámbito textual como su ámbito espacial no tienen en cuenta la naturaleza jerárquica del espacio geográfico ni las relaciones topológicas entre los objetos espaciales que indexan. Por tanto, nuestro primer objetivo es desarrollar una estructura que solucione los problemas debidos a estas limitaciones. Esta estructura constituye la base de la arquitectura para sistemas GIR que proponemos como segundo objetivo de la tesis. Estudiamos las limitaciones de las arquitecturas de los sistemas GIR propuestas hasta la fecha y proponemos una arquitectura genérica, modular y extensible. Además desarrollamos un prototipo de sistema basado en dicha arquitectura. Finalmente, como tercer objetivo de esta tesis proponemos una estructura para indexar objetos geográficos optimizada para las características de la información que se maneja habitualmente en sistemas GIR.

Agradecimientos

Hace ya más de cuatro años que Nieves me presentó en laboratorio a algunos de los que hoy sois mis compañeros. Nunca le agradeceré bastante el haberme acogido tan bien, el haberme aconsejado y guiado hasta aquí. Pero sobre todo tengo que agradecerle el haberme presentado a Miguel. Más que mi director para mi eres un padre en mi vida profesional. Gracias por esta tesis. Pero sobre todo, gracias por atenderme siempre, por leer mis inquietudes escritas en media hoja mal cortada y hasta por haber sido mi enfermero. También quisiera agradecerle a Viqueira el contribuir con sus conocimientos a mejorar esta tesis. Ya hace bastante del primer día que tuve la suerte de tener una reunión de trabajo contigo y me quedé impresionado con tus conocimientos.

También quiero dar las gracias a mis compañeros del laboratorio y Enxenio, los que estáis y los que estuvieron. Porque todos escribisteis un trocito de esta tesis. Porque cuando llegué al laboratorio era un chico tímido que no sabía lo que era la investigación y ahora me apetece trabajar en este mundo, me apetece levantarme cada día y compartirlo con vosotros. Algunos os convertisteis en algo más, en algo que hará que os lleve muy dentro, os eche de menos y os recuerde siempre allá donde esté.

No me puedo olvidar de mi familia y sobre todo de mis padres. Porque la investigación es un mundo complicado y muchas veces tenéis que aguantar mi peor cara. Cuando me preguntan por qué escogí este camino cuando hay otros más fáciles siempre respondo que vosotros me lo hacéis fácil, tengo la suerte de teneros ahí y de saber que nunca me fallareis. Sé que debería decíroslo más, pero el que me apoyéis aun así hace que merezcáis más este agradecimiento.

Dicen que tener un amigo es muy complicado y yo tengo tantos que no tengo espacio para nombraros a todos. Me alegra mucho que tengáis claro que mi casa es vuestra casa, ahora quiero que sepáis que ésta también es vuestra tesis.

Por último, gracias Lucía. Porque todos lloramos de tristeza, muchos de risa... pero llorar de felicidad sólo lo conseguimos unos pocos afortunados. Un día te enumeré axiomas... me quedé muy corto. Esta tesis fue una prueba a tu paciencia y te lo agradezco. Pero dejarlo todo por mi y acompañarme al fin del mundo es una prueba de amor que no creo que superase nadie más. Gracias.

Índice general

Índice de figuras

Capítulo 1

Introducción

Esta tesis presenta los resultados del trabajo que hemos realizado en el ámbito de un nuevo campo de investigación denominado recuperación de información geográfica. Como se mostrará a lo largo de este documento, nuestras principales aportaciones son una estructura de indexación que tiene en cuenta tanto el ámbito textual como el ámbito espacial de los documentos, una arquitectura completa para sistemas de recuperación de información geográfica que utiliza dicha estructura como pilar fundamental y una estructura de indexación espacial diseñada teniendo en cuenta las características específicas de la información geográfica gestionada en este tipo de sistemas.

1.1. Contextualización y motivación

La necesidad de gestionar información ha sido uno de los factores clave en la consolidación de la informática como una ingeniería imprescindible para el desarrollo de la sociedad. A lo largo de los años se han propuesto gran cantidad de arquitecturas de sistemas, estructuras de indexación y otros componentes con el objetivo fundamental de permitir un acceso eficiente a información almacenada en enormes bases de documentos. La *Recuperación de Información* (IR) [BYRN99] es el área de investigación que centra sus esfuerzos en este objetivo. Se trata de un campo de investigación ya clásico, iniciado con los trabajos de Salton en los años 60 [Sal63], que recientemente ha experimentado un desarrollo espectacular motivado por el crecimiento de Internet y la necesidad de realizar búsquedas en la web. Una característica muy importante de la IR es que se ocupa de los problemas de recuperar información por su contenido y no por sus metadatos por lo que existen técnicas de IR para recuperar información de diversos tipos: textos, imágenes, archivos de sonido y vídeo, etc. A menudo, cuando la información es de tipo textual se incluyen

referencias geográficas dentro del texto (por ejemplo, en las noticias de prensa se suele hacer mención del lugar donde sucedió la noticia). El tener en cuenta estas referencias geográficas proporciona un valor añadido a los sistemas de recuperación de información clásicos.

La importancia de las referencias geográficas reside en las características especiales que tienen debido a su naturaleza espacial. En la investigación en *Sistemas de Información Geográfica* (GIS) [Wor04] se le ha dedicado mucho esfuerzo al estudio de estas características y al desarrollo de sistemas capaces de aprovecharlas. Este campo ha recibido mucha atención en los últimos años debido, en gran parte, a que las recientes mejoras en el hardware han hecho posible que el desarrollo de este tipo de sistemas sea abordable por muchas organizaciones. Además, dos organismos internacionales (ISO [ISO02] y el *Open Geospatial Consortium* [Ope03]) están realizando un importante esfuerzo colaborativo para definir estándares y especificaciones para el desarrollo de sistemas interoperables. A nivel europeo, la directiva INSPIRE [Dir07] (INfrastructure for Spatial InfoRmation in Europe) también ha supuesto un gran avance en el ámbito de los GIS corporativos y destaca la importancia futura de la información geográfica. Gracias a todas estas iniciativas muchas organizaciones públicas están trabajando en el desarrollo de infraestructuras de datos espaciales [Glo07] que les permitan compartir su información espacial.

Estas dos áreas de investigación han avanzado de forma independiente a lo largo de los años. Por un lado, las estructuras de indexación y técnicas propuestas desde el campo de la IR no tienen en cuenta la naturaleza espacial de las referencias geográficas citadas en los textos. Por otro lado, las estructuras de indexación espacial no son directamente aplicables en sistemas de recuperación de información. Sin embargo, los usuarios demandan cada vez más servicios que les permitan situar la información en su contexto espacial e incluso acceder a esa información mediante consultas que tengan en cuenta las características especiales de la información espacial. Esta demanda ha producido que investigadores de cada área empiecen a prestar atención a la otra dando como resultado un nuevo campo de investigación denominado *Recuperación de Información Geográfica* (GIR). El objetivo de este nuevo campo es proponer arquitecturas de sistemas, estructuras de indexación y otros componentes que permitan desarrollar sistemas mediante los cuales los usuarios puedan *recuperar documentos relevantes tanto temática como geográficamente en respuesta a consultas de la forma <tema, localización>*. La consulta "*tesis doctorales sobre sistemas de información geográfica publicadas en España*" es un ejemplo del tipo de consultas que se estudian en este nuevo campo. El lector familiarizado con los sistemas de recuperación de información clásicos sabrá que la relevancia de los documentos en los motores de búsqueda textual se basa en la frecuencia de aparición de las palabras buscadas en los textos y, por tanto, si en un documento no aparece explícitamente la palabra *España* su relevancia se verá disminuida con respecto a esta consulta. Esto sucede aunque aparezca la palabra *Madrid* (o cualquier comunidad autónoma, provincia o ciudad de España) ya que los sistemas de IR tradicionales no están preparados para tener en

cuenta las características especiales de la información espacial (por ejemplo, la relación *contenido en* entre Madrid y España).

El campo de investigación en GIR es todavía muy joven. SPIRIT (*Spatially-Aware Information Retrieval on the Internet*) se puede considerar el primer proyecto importante en el área y las publicaciones derivadas de él son un muy buen punto de partida para introducirse en el tema: [JPR+02, JAF03, JAFV04, FJA05]. Los trabajos posteriores proponen mejoras de los distintos componentes del sistema, fundamentalmente de su estructura de indexación. Las estructuras propuestas hasta la fecha se pueden categorizar en dos grupos: *híbridas*, si combinan en una única estructura el índice textual y el índice espacial, y de *doble índice*, si los mantienen por separado. En [VJJS05] los autores concluyen que manteniendo separados ambos índices se consigue un menor coste de almacenamiento aunque, por contra, puede implicar mayores tiempos de respuesta. La diferencia más importante de las propuestas que se han ido realizando es la estructura de indexación espacial que emplean. Por ejemplo, en el proyecto SPIRIT se emplea un *grid*, en [ZXW+05] se emplea un R-tree y en [CSM06] se comparan las dos anteriores con el empleo de curvas de llenado del espacio. Sin embargo, un problema común de todas las estructuras propuestas es que ninguna de ellas tiene en cuenta la naturaleza jerárquica del espacio geográfico y las relaciones topológicas [EH92, CF96] entre los objetos espaciales que indexan. Por tanto, el primer objetivo de esta tesis es definir una estructura de indexación que tenga en cuenta estas particularidades debidas a la naturaleza espacial de las referencias geográficas y a las características especiales que de ella se derivan.

Aunque el núcleo de todo sistema GIR es la indexación, hay otras tareas que no debemos olvidar. Una de las más complicadas es la obtención de referencias geográficas en textos y su traducción a un modelo geográfico del mundo (por ejemplo, sus coordenadas de latitud y longitud). Los trabajos derivados de Web-a-where [AHSS04], STEWARD [LSSS07] y el producto comercial METACARTA [RBB03] tratan diferentes aspectos de este problema y presentan sistemas completos enfocados desde este punto de vista. Tanto estos sistemas como las arquitecturas en las que se basan están muy orientadas hacia el problema más familiar para los autores (desambiguación del lenguaje natural, indexación textual, indexación espacial, etc.) y no son adecuadas como marco general para el desarrollo de sistemas GIR. Consideramos que una arquitectura genérica, completa, modular y extensible es un requisito fundamental para el avance de esta nueva línea de investigación. Por tanto, el segundo objetivo de esta tesis es definir una arquitectura completa para sistemas GIR que tenga en cuenta las aproximaciones existentes en el estado del arte del área. Además, se pretende desarrollar un prototipo de un sistema GIR completo basado en esta arquitectura. El objetivo de este prototipo es que sirva como marco de pruebas de la investigación realizada en el área, permitiendo comparar la eficacia y la eficiencia de las distintas soluciones planteadas por los investigadores para la totalidad de las tareas que componen el sistema.

Por último, en la bibliografía se han documentado una gran variedad de estructuras

de indexación espacial. En [GG98] y en [Sam06] se puede encontrar un buen resumen de ellas. La principal clasificación de estas estructuras las divide en *métodos de acceso a puntos* (PAMs, del inglés *Point Access Methods*), si sólo permiten la indexación de puntos, y *métodos de acceso espacial* (SAMs, del inglés *Spatial Access Methods*), si permiten la indexación de cualquier tipo de objeto geográfico. Sin embargo, ninguna de estas estructuras está optimizada para las características particulares de la información geográfica con la que se trabaja en sistemas GIR. Una primera característica deseable para una estructura que se vaya a emplear en este tipo de sistemas es estar optimizada para indexar puntos (por lo que se encontraría dentro del grupo de los PAMs). Esto se debe a que una de las operaciones más habituales consiste en representar los documentos en los lugares que mencionan si esos lugares se ajustan a la consulta del usuario. Otra característica deseable es que la estructura debe ser compacta y muy rápida por lo que debe trabajar preferiblemente en memoria principal. Una última característica es que este tipo de sistemas está pensado para trabajar con colecciones semi-estáticas de documentos por lo que la estructura se puede optimizar para indexar colecciones de información geográfica conocida *a priori*. Por tanto, el último objetivo de esta tesis es definir una estructura de indexación espacial de acuerdo a estas características deseables para los sistemas GIR.

1.2. Objetivos

Como ya hemos expuesto, esta tesis abarca tres objetivos principales que se recopilan y detallan brevemente a continuación:

- **Diseñar una estructura de indexación que tenga en cuenta las características textuales y espaciales de los documentos**.

  La estructura diseñada debe solucionar los problemas de las soluciones propuestas en la bibliografía. Estos problemas se resumen de manera general en que no tienen en cuenta ciertas características especiales que son debidas a la naturaleza espacial de la información con la que trabajan. En concreto, las soluciones propuestas no tienen en cuenta la naturaleza jerárquica del espacio geográfico ni las relaciones topológicas entre los objetos espaciales que indexan.

  La estructura que permite describir de forma adecuada las características específicas del espacio geográfico es la *ontología*, la cual se define como una *especificación explícita y formal de una conceptualización compartida* [Gru93b, Gru93a]. La estructura de indexación para sistemas GIR que proponemos combina un índice textual, un índice espacial y una ontología.

- **Diseñar una arquitectura para sistemas de recuperación de información geográfica**.

La arquitectura diseñada debe ofrecer un soporte global para el desarrollo de sistemas GIR, es decir, la arquitectura debe abarcar todas las tareas necesarias para la construcción de una estructura de indexación sobre una colección de documentos así como todas aquellas necesarias para que los usuarios puedan realizar consultas sobre dicha estructura. Debido a que la investigación en el área es todavía muy reciente, esta arquitectura debe ser lo más modular y extensible posible para permitir la integración y comparación de nuevos resultados de investigación en las diferentes etapas propias de este tipo de sistemas.

Además, se construirá un prototipo de un sistema completo basado en esta arquitectura. Este prototipo estará basado en software libre para que todos los miembros de la comunidad puedan emplearlo para probar y comparar sus resultados de investigación. Por tanto, todas las librerías desarrolladas así como todas aquellas ya implementadas de las que se haga uso deben tener licencias de uso libre para propósitos de investigación.

- **Diseñar una estructura de indexación espacial optimizada para las características de la información geográfica gestionada habitualmente en sistemas GIR**.

  La tendencia en el mundo GIS sobre la investigación en estructuras de indexación espacial avanza hacia el diseño de estructuras cada vez más generales (orientadas a polígonos, para almacenamiento en memoria secundaria, etc.). Estas estructuras no tienen en cuenta en absoluto las características propias de la información geográfica que se maneja en sistemas GIR.

  La estructura de indexación espacial propuesta debe estar orientada a la información geográfica que se maneja habitualmente en sistemas GIR para explotar sus características. En primer lugar, debe tener en cuenta que las colecciones son relativamente estáticas en el tiempo y, por tanto, es posible conocer *a priori* la distribución de las referencias geográficas en el espacio y optimizar la estructura en función de esa distribución. Además, las referencias geográficas se traducen normalmente en puntos por lo que debe ser muy eficiente con este tipo de datos espaciales. También debe ser muy compacta y eficiente, se deben emplear técnicas propias del campo de investigación en compresión para permitir que la estructura opere en memoria principal con colecciones de tamaños reales.

1.3. Alcance e interés del trabajo de esta tesis

Como hemos comentado anteriormente, el campo de investigación en GIR es muy reciente. Los primeros artículos sobre el proyecto SPIRIT, nombrado anteriormente como primer proyecto importante dentro del área, se presentaron en el año 2002. Sin embargo, a pesar de su juventud, este campo está atrayendo el interés de

muchos investigadores. Como prueba de este interés podemos citar el *Workshop on Geographic Information Retrieval* que se celebra anualmente desde el año 2004 en congresos tan prestigiosos como ACM SIGIR [SIG07] o ACM CIKM [CIK08]. Los informes elaborados como resumen de estos *workshops* señalan, entre otras líneas de investigación fundamentales en recuperación de información geográfica, las siguientes:

- Arquitecturas para motores de búsqueda geográficos.
- Indexación espacial de documentos e imágenes.
- Extracción del contexto espacial de documentos y otros conjuntos de datos.
- Diseño, construcción, mantenimiento y métodos de acceso para ontologías espaciales, *gazetteers* y tesauros geográficos.

Durante los últimos años se han propuesto nuevas arquitecturas de sistemas, estructuras de indexación y otros componentes que, si bien han supuesto considerables avances en el estado del arte, dejan todavía muchos temas abiertos en todas estas líneas de investigación. Antes de presentar cada contribución describiremos las aproximaciones existentes en el estado del arte, sus ventajas e inconvenientes.

Los tres objetivos que hemos presentado en la sección anterior constituyen también las tres aportaciones principales de este trabajo de tesis. En primer lugar, describimos de forma detallada la estructura de indexación que forma el núcleo del sistema de recuperación de información geográfica. Dicha estructura combina un índice textual, un índice espacial y una estructura ontológica para permitir resolver consultas textuales puras, espaciales puras, y consultas con una componente textual y una componente espacial. Su principal aportación sobre otras propuestas es que, gracias a la ontología, representa de manera explícita y formal las relaciones existentes entre los objetos geográficos que está indexando. En segundo lugar, definimos una arquitectura para sistemas completos de recuperación de información geográfica. Esta arquitectura es genérica y está definida desde un punto de vista global, abarcando todas las tareas necesarias para la construcción del repositorio de documentos y su posterior explotación mediante consultas realizadas por los usuarios. Además, completamos esta aportación con el desarrollo de un prototipo de sistema siguiendo esta arquitectura y lo ofrecemos a la comunidad científica para que pueda servir como marco para la comparación de los resultados de investigación de los diversos grupos. Por último, presentamos un nuevo índice espacial cuya principal aportación es estar optimizado para tener en cuenta las características de la información geográfica que se maneja habitualmente en los sistemas GIR.

1.4. Organización de la tesis

La metodología seguida para el desarrollo del trabajo de esta tesis contempla principalmente las etapas de estudio del estado del arte, desarrollo de nuevas propuestas, validación experimental e integración en soluciones existentes. Siguiendo esta metodología y teniendo en cuenta los tres objetivos principales descritos en la sección 1.2 la organización del resto de la memoria se organiza en otros cinco capítulos.

En el capítulo 2 presentamos un análisis del estado del arte en recuperación de información geográfica. Realizamos una aproximación al tema desde el punto de vista de la recuperación de información y de los sistemas de información geográfica, describiendo modelos y estructuras de indexación clásicas en cada una de las áreas. Esta aproximación introducirá al lector de forma más clara en el campo de la recuperación de información geográfica. En cuanto a este campo, presentamos las diferentes aproximaciones estudiadas tanto para el diseño de arquitecturas de sistemas GIR completos como para el diseño de estructuras de indexación. Además, detallamos las principales ventajas e inconvenientes de cada propuesta.

En el capítulo 3 describimos una estructura de indexación para sistemas GIR que resuelve los inconvenientes de las estructuras estudiadas en el estado del arte. En este capítulo, además de realizar la descripción detallada de la estructura, estudiamos en profundidad sus ventajas e inconvenientes, su rendimiento y los tipos de consulta que permite responder.

La estructura de indexación propuesta se tiene que enmarcar en una arquitectura completa para sistemas GIR. En el capítulo 4 describimos nuestra propuesta de arquitectura para el desarrollo de sistemas GIR. Además, en este capítulo presentamos una descripción detallada de los componentes de la arquitectura y de la interacción necesaria entre los mismos para realizar las principales operaciones soportadas por el sistema.

La información geográfica que se gestiona habitualmente en sistemas GIR tiene unas características muy determinadas. Sin embargo, las estructuras de indexación espacial propuestas en el campo de los GIS no están optimizadas para tenerlas en cuenta. Por este motivo, en el capítulo 5 analizamos esas características y proponemos una nueva estructura de indexación espacial optimizada para este tipo de sistemas. Además de la estructura, en este capítulo presentamos los resultados de la comparación con otras estructuras de indexación espacial, qué ventajas tiene con respecto a ellas, qué inconvenientes y el rendimiento en diferentes escenarios.

Finalmente, en el capítulo 6 presentamos las conclusiones obtenidas en la realización de esta tesis y las líneas de trabajo futuro que quedan abiertas.

Como material adicional, en el apéndice A listamos las publicaciones y otros resultados de investigación del doctorando derivadas de la investigación realizada en esta tesis, las cuales han sido para nosotros un referente externo de evaluación.

Capítulo 2

Estado del arte

En este capítulo revisamos el estado del arte de la recuperación de información geográfica y de los dos campos que dan origen al mismo: la recuperación de información y los sistemas de información geográfica. En primer lugar, en la sección 2.1 estudiamos el campo de la recuperación de información. A continuación, en la sección 2.2 estudiamos el campo de los sistemas de información geográfica. Estas dos secciones están destinadas a familiarizar al lector con los conceptos básicos de cada campo e introducir de forma más clara el estudio del estado del arte en el campo de la recuperación de información geográfica. Este estudio lo realizamos en la sección 2.3. Por último, terminamos el capítulo con las conclusiones extraídas del estudio del estado del arte en la sección 2.4.

2.1. Recuperación de información

Aunque el campo de investigación en recuperación de información [BYRN99, MRS08] ha estado activo durante las últimas décadas, la creciente importancia de Internet y de la *World Wide Web* ha hecho de él uno de los campos de investigación más importantes hoy en día. Se han propuesto muchas estructuras de indexación, técnicas de compresión y algoritmos de recuperación diferentes en los últimos años. Pero lo más importante es que estas propuestas se han empleado ampliamente en la implementación de bases de datos documentales, bibliotecas digitales y motores de búsqueda en web.

En esta sección estudiamos el estado del arte de este campo de investigación. En primer lugar, en la sección 2.1.1 presentamos los conceptos básicos del campo. A continuación, la sección 2.1.2 describe los modelos o métodos con los que se aborda la tarea de recuperar información relevante para el usuario. En las secciones 2.1.3 y 2.1.4 estudiamos estructuras de indexación clásicas y nuevas propuestas que han ido

apareciendo y que se basan en la aplicación de técnicas de compresión para reducir el tamaño de los índices. En la sección 2.1.5 presentamos una tarea que se realiza habitualmente en este tipo de sistemas, la expansión de consultas (*query expansion*). Además, en la sección 2.1.6 estudiamos cómo se evalúa si los sistemas de recuperación de información funcionan adecuadamente. Finalmente, la sección 2.1.7 resume las ideas más importantes presentadas acerca de la recuperación de información.

2.1.1. Conceptos básicos

Uno de los principales problemas que existe a la hora de hablar de recuperación de información es el desconocimiento de la gente no relacionada con el campo acerca de qué abarca este concepto. Se han publicado muchos libros de texto [BYRN99, MRS08], cursos [All07] y otros recursos que abordan este tema y proporcionan excelentes definiciones aclarando qué es exactamente la recuperación de información. Manning et. al [MRS08] realizan una definición muy intuitiva de este concepto: *la recuperación de información (IR) se ocupa de encontrar material (normalmente documentos) de una naturaleza no estructurada (normalmente texto) que se encuentra en grandes colecciones (normalmente almacenada de forma digital) y satisface una necesidad de información*. Esta definición menciona explícitamente una de las principales confusiones al hablar del tema: la naturaleza no estructurada del material a buscar. Esto deja fuera del ámbito de la IR temas como la búsqueda en bases de datos, donde la información se encuentra estructurada en un conjunto de campos, las búsquedas son exactas y los sistemas deben satisfacer ciertos requisitos de control de concurrencia, recuperación, etc. Además, la definición es lo suficientemente general como para no dejar fuera otros temas que sí se encuentran en el campo de la IR que, en contra de un pensamiento bastante generalizado, no es solamente recuperación de documentos. La organización automática (*clustering*), la respuesta de preguntas (*question answering*) y la elaboración de resúmenes (*summarization*) son algunos de los temas más conocidos que se encuentran dentro de este campo. Sin duda, una de las principales fuentes de confusión es la popularidad que han alcanzado los motores de búsqueda en web. Hoy en día, la mayor parte de la gente, con conocimientos de informática o no, han oído hablar y han utilizado alguno de estos motores como Google, Yahoo, Lycos, etc. Esta popularidad provoca inevitablemente que los no expertos en el tema limiten el concepto IR al ámbito de la búsqueda en web.

Una vez aclarado qué es y qué abarca la recuperación de información, el siguiente paso consiste en entender cómo se realiza. La idea fundamental que se encuentra bajo la mayoría de los sistemas de este tipo es la *similitud*. Si nos fijamos en la definición de Manning, *recuperar material que satisface una necesidad de información*; la necesidad de información suele estar expresada con una consulta y, por tanto, recuperar material que satisfaga esa necesidad se puede traducir en encontrar material que sea lo más parecido posible a los términos clave indicados en la consulta. La situación más habitual, y también la más relacionada con el alcance de este trabajo, consiste en la

recuperación de documentos. Por este motivo, a partir de ahora emplearemos el término documento, en lugar de otros más genéricos aunque, sin lugar a dudas, más correctos en cuanto al alcance de la IR. Es decir, para nuestros intereses vamos a simplificar el objetivo de la IR en encontrar documentos que se parezcan lo máximo posible a una consulta formulada mediante una serie de términos clave.

En la figura 2.1 presentamos un esquema simplificado para aclarar el proceso de recuperación de información de acuerdo con la especificación que acabamos de realizar. En la parte superior de la figura se representan las *necesidades de información* de los usuarios y los *documentos* con los que se alimenta el sistema. Estos conceptos no son lo suficientemente concretos como para ser manejados por sistemas de información. Por este motivo, el esquema indica que deben ser *representados* de manera más concreta en *consultas* y *objetos indexados* respectivamente. Mediante la *comparación* de las consultas con los objetos indexados se pueden obtener los *objetos recuperados* que se le presentarán al usuario. El esquema contempla también una tarea de *mejora de los resultados* por medio de la cual se puede especificar más la necesidad de información o la consulta para obtener resultados que se ajusten mejor a las necesidades del usuario. Esta tarea la puede llevar a cabo tanto el usuario como el propio sistema de manera automática.

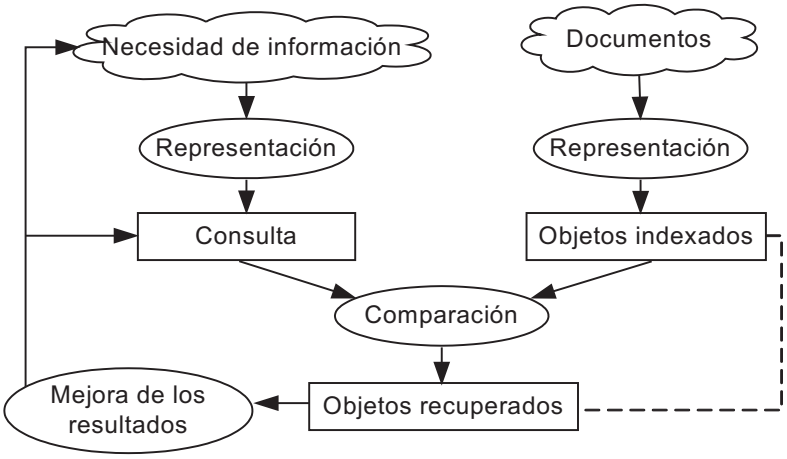


Figura 2.1: Esquema del proceso de recuperación de información.

Este esquema, aunque simplificado, representa las tareas más importantes que intervienen en la recuperación de información. Estas tareas, representadas en la figura con formas elípticas, son las que diferencian los distintos métodos o modelos de recuperación de información. Es decir, la manera de representar los documentos y las consultas, y la forma de medir la similitud entre ambos son algunas de las características distintivas de los modelos de recuperación de información. En la sección

2.1.2 veremos algunos ejemplos de estos modelos como el booleano, el probabilístico y el vectorial. Estos modelos necesitan unas estructuras físicas que den soporte al proceso de acceso a la información de manera eficiente. En la sección 2.1.3 describimos algunas de las estructuras ya clásicas en el área y en la sección 2.1.4 describimos nuevas técnicas que han ido apareciendo en los últimos años basadas sobre todo en el empleo de la compresión para disminuir el tamaño de las estructuras pero también para mejorar su eficiencia. Además, en la sección 2.1.5 describimos la expansión de consultas, una técnica muy habitual que se encuentra enmarcada dentro de la tarea de *mejora de los resultados* del sistema. Finalmente, aunque es una tarea que no se encuentra representada en el esquema de la figura, no podemos cerrar el estudio del estado del arte en la IR sin hablar de la evaluación de este tipo de sistemas. Esta tarea, destinada fundamentalmente a determinar si los sistemas de IR funcionan correctamente y de si un cambio determinado mejora su eficacia y/o eficiencia, la describimos en la sección 2.1.6. Aunque el campo es mucho más amplio creemos que estas son las tareas más importantes en lo que se refiere al alcance de esta tesis.

2.1.2. Modelos de recuperación de información

El término modelo tiene un significado bien conocido dentro del campo de la recuperación de información. Este significado no es más que una adaptación directa de su significado general (*esquema teórico de un sistema o de una realidad compleja que se elabora para facilitar su comprensión y el estudio de su comportamiento*) a este dominio concreto. De forma simplificada, podemos considerar un modelo como un método para representar tanto documentos como consultas en sistemas de IR y comparar la similitud de esas representaciones. Para ello, los modelos de recuperación tienen que proporcionar de manera implícita o explícita una definición de relevancia. Además, pueden describir el proceso computacional, el proceso humano y las variables que intervienen en el proceso global.

Los tres modelos que vamos a analizar en esta sección (el modelo booleano, el modelo probabilístico y el modelo vectorial) representan tanto los documentos como las consultas mediante un conjunto de términos clave (*keywords*) que se emplean como términos de indexación. La principal diferencia entre ellos reside en la forma de representar dichos términos y en la forma de relacionarlos.

Otra característica que diferencia los distintos modelos de recuperación de información consiste en cómo realizan la comprobación de si un documento se ajusta a una consulta. Si el documento se tiene que corresponder exactamente con la consulta se trata de un modelo de ajuste exacto (*exact match*), mientras que si el documento se puede corresponder sólo de manera parcial con la consulta se trata de un modelo de mejor ajuste (*best match*). En los modelos del primer grupo el resultado de una consulta es un conjunto (no existe orden entre los elementos) donde se encuentran los elementos que satisfacen la búsqueda. Por el contrario, en los modelos del segundo

grupo el resultado es una lista (un *ranking*) donde se encuentran los documentos en orden según lo bien que se ajustan a la consulta. Entre las ventajas de los modelos del primer grupo están el ser muy eficientes, previsibles, fáciles de explicar y que funcionan bien cuando el usuario es capaz de expresar exactamente lo que necesita (aunque esta es también su principal desventaja ya que es muy complicado que eso suceda). El modelo booleano es un ejemplo de este tipo. Los modelos de mejor ajuste, como el probabilístico o el vectorial, son más sencillos de emplear y, por tanto, significativamente más eficaces; además, existen implementaciones optimizadas que proporcionan una eficiencia similar.

Modelo booleano

El modelo booleano es uno de los modelos de recuperación de información más sencillos. Está basado en la teoría de conjuntos y el álgebra de Boole. Tanto los documentos como las consultas se representan empleando términos clave (*keywords*) pero, además, las consultas se formulan como expresiones lógicas que combinan mediante operadores del álgebra de Boole (por ejemplo, AND, OR, NOT) los términos clave de búsqueda. La idea en la que está basado este modelo es que un término clave puede estar presente o ausente en un documento y, por tanto, sólo serán relevantes aquellos documentos que contengan los términos clave que se indica en la consulta. Siguiendo estrictamente esta definición del modelo, el resultado de una consulta no contendría documentos que pueden ser relevantes a pesar de no encajar a la perfección con la consulta. Este grave inconveniente provocó que la mayoría de las implementaciones no sigan exactamente la definición sino que, basándose en esa idea, devuelven documentos que no se ajustan perfectamente a la consulta, aunque si en alguna medida, y ordenan el resultado en un *ranking* según ese nivel de ajuste.

Consideremos un ejemplo sencillo, supongamos que el usuario realiza una consulta formulada como q = *{tesis AND (gis OR ir) AND NOT gir}* (es decir, desea encontrar tesis que traten el tema de gis o de ir pero no el tema de gir). El enfoque básico del modelo booleano recuperará sólo aquellos documentos donde aparezcan los términos clave *tesis*, *gis* o *ir*, y no aparezca el término *gir*. Sin embargo, un documento donde no aparezca el término *tesis* puede ser relevante en alguna medida para el usuario (por ejemplo, puede ser un proyecto de máster).

La principal ventaja de este modelo es su sencillez y su eficiencia. Por contra, su principal inconveniente es la dificultad para formular exactamente las necesidades de información de los usuarios empleando el álgebra de Boole y términos clave. Otro inconveniente es que sólo obtiene los documentos que se ajustan perfectamente a la consulta y no proporciona *rankings* con los resultados. Aunque este inconveniente se ha tratado de resolver en muchas implementaciones, la eficacia del modelo es inferior a la de otras alternativas.

Modelo probabilístico

El modelo probabilístico [RJ76] define la recuperación como un proceso de clasificación, de tal forma que para cada consulta existen dos clases: la clase de los documentos relevantes y la clase de los documentos no relevantes. Por tanto, dado un documento D y una consulta determinada se puede calcular con qué probabilidad pertenece el documento a cada una de esas clases. Si la probabilidad de que pertenezca a la clase de los documentos relevantes es mayor que la probabilidad de que pertenezca a la clase de los no relevantes (es decir, $P(R|D) > P(NR|D)$) el documento será relevante para la consulta. Del mismo modo, se puede elaborar un *ranking* de relevancia según el ratio $P(D|R) \div P(D|NR)$. Las diferentes maneras en las que se pueden estimar estas probabilidades dan lugar a los diferentes modelos probabilísticos. Este modelo asume que la relevancia de un documento para una consulta es independiente del resto de documentos de la colección. Además, asume que hay un conjunto de documentos que el usuario prefiere como resultado de la consulta y que, si se elabora un *ranking* en orden decreciente de probabilidad de que los documentos sean relevantes, se obtiene la mayor eficacia posible (es decir, se minimiza la probabilidad de error).

Este modelo necesita un método para calcular las probabilidades iniciales. Aunque existen numerosas alternativas para realizar este cálculo, la más habitual posiblemente sea emplear las frecuencias de los términos clave en cada documento y la frecuencia en el total de los documentos. Además, el modelo es iterativo, tras una primera aproximación donde se obtiene un subconjunto inicial se emplean los documentos en dicho subconjunto para refinar la búsqueda, recalculando las probabilidades. Este proceso se repite hasta obtener las probabilidades definitivas.

Modelo vectorial

La idea fundamental en la que se basa el modelo vectorial [SL68] es considerar que tanto la importancia de un término clave con respecto a un documento como los documentos y las consultas se pueden representar como un vector en un espacio de alta dimensionalidad. Por tanto, para evaluar la similitud entre un documento y una consulta; es decir, para obtener la relevancia del documento con respecto a la consulta, simplemente hay que realizar una comparación de los vectores que los representan. Uno de los métodos más habituales para comparar el grado de similitud es calcular el coseno del ángulo que forman ambos vectores. Cuanto más se parezcan los vectores, más próximo a cero grados será el ángulo que forman y, en consecuencia, más se aproximará a uno el coseno de ese ángulo. Esta forma de comparar los vectores se conoce como *fórmula del coseno*. En la figura 2.2 presentamos un ejemplo en el que se muestran los vectores representativos de dos documentos (D_1 y D_2) y de la consulta (q). Intuitivamente, el documento D_2 es más relevante para la consulta q que el documento D_1. La fórmula del coseno no hace más que trasladar esta idea intuitiva a la formalización de los vectores. El ángulo que forman los vectores de D_2 y q es menor

que el que forman los vectores de D_1 y q, en consecuencia, el coseno del primer ángulo es mayor que el del segundo.

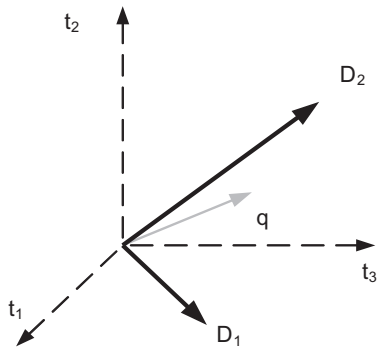


Figura 2.2: Comparación de similitud en el modelo vectorial.

La cuestión que queda pendiente es cómo se construyen los vectores para representar tanto documentos como consultas. En ambos casos, la construcción se realiza a partir de los vectores de los términos clave que los componen (por ejemplo, sumando dichos vectores). Estos vectores de términos clave tienen unos coeficientes (longitudes y pesos) que representan entre otros factores la presencia del término y su importancia. Por ejemplo, en la figura 2.2 el vector del documento D_1 se construye a partir de la suma de los vectores que representan la importancia de los términos clave t_1, t_2 y t_3 en ese documento. La construcción de los vectores que representan a D_2 y a q se realiza de manera análoga. El modelo no define cómo se tienen que establecer los coeficientes que representan la importancia de cada término en el documento aunque, sin duda, el método $tf \cdot idf$ (o alguna de sus variantes) es el más empleado. Este método se basa en que los pesos están determinados por la frecuencia de aparición del término clave en el documento (el parámetro *tf*) y la frecuencia inversa de aparición del término en el conjunto total de documentos (el parámetro *idf*). Intuitivamente, el factor *tf* captura la idea de que un término clave es más importante en un documento donde se menciona muchas veces que en otro donde se menciona menos. El factor *idf*, en cambio, captura la idea de que un término clave que aparece en muchos documentos tiene menor poder a la hora de discriminar documentos relevantes que otro que aparece en menos.

En el capítulo 4, donde presentamos nuestro prototipo, veremos que para la indexación textual empleamos Lucene [Apa08a]. Lucene emplea una combinación del modelo vectorial y del modelo booleano. Por este motivo, vamos a presentar algunas de sus ecuaciones a modo de caso de éxito del modelo vectorial. La ecuación 2.1 muestra la fórmula empleada por Lucene para la asignación de pesos. No es nuestro objetivo explicar todos los términos que forman parte de la ecuación, simplemente indicar que

es una variante del método general donde el factor *idf* se calcula en escala logarítmica.

$$w_{t,d} = \frac{tf_{d,t} \cdot log(N/idf_t + 1)}{\sqrt{\#elementos\ en\ d\ en\ el\ mismo\ campo\ que\ t}} \tag{2.1}$$

Finalmente, la ecuación 2.2 muestra una simplificación de la fórmula con la que se obtiene la relevancia de un documento *d* para una consulta *q* (conocida como la fórmula de *scoring* de Lucene). En esta fórmula, podemos observar de nuevo la importancia de los factores *tf* e *idf*. Además, intervienen *otros factores* entre los que se encuentran la influencia o estímulo del campo donde se encuentra el término (factor establecido manualmente durante la etapa de creación del índice) y un factor de normalización de la longitud del campo dependiente del número de términos en el mismo (también establecido durante la creación del índice).

$$relevancia_{d,q} = \sum_{t\ en\ q} tf(t\ en\ d) \cdot idf(t) \cdot otrosFactores \tag{2.2}$$

A pesar de su simplicidad, habitualmente el modelo vectorial logra unos resultados superiores a los de las demás alternativas y, por este motivo, es el modelo más empleado hoy en día. Entre los casos de éxito más conocidos, además del ya mencionado Lucene, se encuentra el sistema SMART [SL65] creado por uno de los precursores de la IR, Gerard Salton, y su equipo en la Universidad de Cornell. También, es la base de la mayoría de los motores de búsqueda en web.

2.1.3. Estructuras de indexación textual clásicas

Antes de empezar a hablar de estructuras de indexación en sistemas de IR es necesario introducir los diferentes tipos de consultas que se pueden resolver en este tipo de sistemas. Podemos distinguir tres tipos principales de consultas: las consultas basadas en términos clave, las consultas de reconocimiento de patrones en texto y las consultas sobre la estructura de los textos. El primer grupo de consultas constituye el núcleo de la IR debido fundamentalmente a que en dicho grupo se encuentran las consultas que se realizan típicamente en la web. Dentro de este grupo de consultas se encuentran aquellas que se realizan empleando términos clave de forma individual, frases formadas por términos clave, y aquellas que emplean operadores lógicos para manipular conjuntos de términos y documentos. Generalmente, las consultas de reconocimiento de patrones son más complejas y se utilizan como complemento a las consultas del primer grupo para proporcionar a los sistemas capacidades de recuperación más avanzadas. Por último, las consultas sobre la estructura de los textos son más específicas y dependientes del modelo de texto particular. Estas consultas tienen en cuenta la estructura de los documentos y no su contenido permitiendo, por ejemplo, aprovechar las ventajas de la estructura del formato HTML. En esta sección

nos centraremos en las estructuras de indexación que dan soporte al primer tipo de consultas ya que los restantes grupos no se encuentran en el alcance de este trabajo de tesis.

El índice invertido, también conocido como fichero invertido, es sin duda la estructura de indexación textual más conocida. En [ZM06] se puede encontrar una buena descripción de esta estructura y de los avances relacionados con ella que se han realizado en los últimos años. El índice invertido es muy fácil de mantener y permite resolver de manera eficiente consultas basadas en términos clave, sobre todo cuando se buscan los términos clave de manera individual.

En la figura 2.3 mostramos un ejemplo de construcción de un índice invertido sobre dos textos. Es un ejemplo simplificado ya que, a la hora de construir este tipo de estructuras, los sistemas reales emplean una serie de técnicas de preprocesado de los textos como puede ser el borrado de *palabras sin significado* (es decir, términos muy empleados y con poca utilidad para resolver consultas como *en*, *de*, *y*, etc.) que producirían que algunas de las palabras que mostramos no formasen parte del índice. El índice invertido es una estructura orientada a palabra compuesta principalmente por dos elementos: el *vocabulario* y la lista de *ocurrencias*. El vocabulario está formado por el conjunto de todas las palabras o términos clave que se citan en los documentos a indexar. Por otra parte, las ocurrencias son las listas que se almacenan para cada término clave indicando en qué documentos e incluso en qué posiciones dentro de ellos aparece cada término. Las posiciones pueden ser palabras (como en el ejemplo de la figura) o caracteres. La decisión de emplear palabras o caracteres depende fundamentalmente de las consultas que tenga que resolver habitualmente la estructura, ya que emplear palabras simplifica las consultas de frases y de proximidad mientras que emplear caracteres facilita un acceso directo al texto.

Para resolver consultas empleando esta estructura el algoritmo de búsqueda básico se compone de tres pasos. En primer lugar, se realiza una búsqueda en el vocabulario con el objetivo de encontrar los términos clave que forman la consulta. En segundo lugar, se obtiene la lista de ocurrencias asociada a cada término clave que se localizó en el vocabulario en el paso anterior. Finalmente, se manipulan las listas de ocurrencias obtenidas para resolver la consulta. Dicha manipulación será diferente según el tipo de consulta del que se trate. Por ejemplo, si en la consulta se buscan términos clave de manera individual la manipulación de la lista de ocurrencias consistirá en combinarlas para obtener una única lista ordenada con los resultados.

A lo largo de los años, se han ido proponiendo variantes sobre esta estructura básica para mejorar la eficiencia de determinados tipos de consultas y para disminuir el espacio necesario para su almacenamiento. En [BYRN99] puede encontrarse una buena descripción de alguna de estas variantes y de sus características más importantes. Además, aunque el índice invertido es la técnica más eficiente para resolver búsquedas de términos clave de manera individual, cuando se realizan otros tipos de consultas menos habituales, como pueden ser las búsquedas de frases formadas por términos

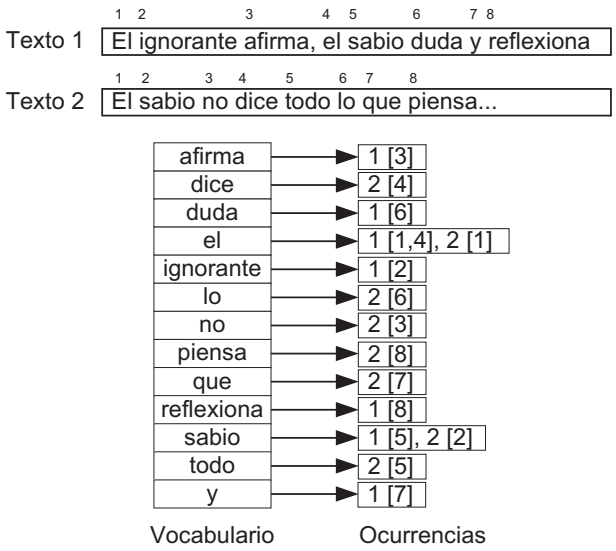


Figura 2.3: Ejemplo de construcción de un índice invertido.

clave, existen otras estructuras como los *arrays de sufijos* [MM90] que son más rápidas. Sin embargo, los arrays de sufijos tienen el inconveniente de ser más complicados de construir y de mantener.

Aunque podríamos describir muchas otras estructuras de indexación en este apartado, consideramos que la descripción del índice invertido es suficiente para el alcance de esta tesis. Además, en la siguiente sección vamos a estudiar nuevas aproximaciones que han aparecido recientemente basándose en la idea de emplear la compresión para disminuir el tamaño tanto de los índices como de los documentos a indexar. Este estudio completa el estado del arte de las estructuras de indexación en IR.

2.1.4. Estructuras de indexación basadas en técnicas de compresión

Investigaciones recientes han demostrado que las técnicas de compresión de textos son útiles no sólo para ahorrar espacio en disco sino también, y más importante si cabe, para mejorar los tiempos de procesamiento, de transmisión y de transferencia a disco a la hora de realizar búsquedas. Las técnicas de compresión diseñadas especialmente para textos en lenguaje natural permiten buscar en el texto comprimido hasta ocho veces más rápido que en el texto original [TM97, MNZBY98], además de disminuir

considerablemente el espacio de almacenamiento necesario. Se pueden obtener *ratios* de compresión de entre el 25 % y el 35 %; es decir, el texto comprimido ocupa entre el 25 % y el 35 % del texto original.

Para el lector no familiarizado con el tema, la idea base de la compresión es la sustitución de los caracteres o de las palabras de los textos por códigos, normalmente de longitud variable, que suelen ser más cortos a medida que aumenta la frecuencia de aparición del caracter o de la palabra y más largos a medida que disminuye la misma. Los mejores ratios de compresión se obtienen con modelos basados en la codificación de palabras [Mof89] en lugar de caracteres. Siguiendo la Ley de Zipf [Zip49], las palabras presentan una distribución de frecuencias mucho más sesgada que la de los caracteres. Esto hace que los textos, representados como una secuencia de palabras, sean altamente compresibles alcanzando ratios de compresión del 25 % con la codificación óptima de Huffman [Huf52].

Además de Huffman, existen muchas otras técnicas de codificación que, si bien no alcanzan ratios de compresión tan buenos (aproximadamente el 35 %), son auto-sincronizados. Esta propiedad es muy importante ya que permite distinguir los límites de los códigos empleados en la codificación desde cualquier posición del texto comprimido y, por tanto, permite el acceso aleatorio a cualquier posición del texto. Esto no es una propiedad trivial ya que, como acabamos de mencionar, los códigos empleados para comprimir suelen ser de longitud variable. Algunos ejemplos de estas técnicas son los códigos Tagged Huffman [MNZBY00], End-Tagged Dense Codes y (s,c)-Dense Codes [BFNP07]. Una propuesta aparecida recientemente [BFLN08] consigue reorganizar el texto comprimido y, empleando muy poco espacio adicional, mejorar considerablemente las capacidades de búsqueda dentro del texto sin que esto influya drásticamente en los tiempos de compresión y descompresión. Esta propuesta es especialmente relevante para esta tesis ya que la reorganización se realiza empleando una estructura conocida como *wavelet tree* (dicha estructura es también la base del método de acceso a puntos que proponemos como uno de los objetivos de este trabajo). Con esta aproximación los autores consiguen que los códigos estén sincronizados permitiendo búsquedas rápidas y accesos a posiciones aleatorias del texto. Además, la reorganización del texto comprimido lo convierte en una representación implícitamente indexada, con lo que permite búsquedas de palabras en tiempo independiente de la longitud del texto.

El *wavelet tree* [GGV03] es una estructura compacta diseñada originalmente para indexar los caracteres de un texto. Esta estructura se ha empleado recientemente en otras áreas para almacenar e indexar distintos tipos de información de forma compacta. Además de para reorganizar el contenido de textos comprimidos [BFLN08], en [MN08] se emplea para la indexación de imágenes. La herramienta básica empleada en el *wavelet tree* es la operación *rank* sobre vectores de bits: la consulta $rank(B,i) = rank_1(B,i)$ devuelve el número de bits establecidos a uno en el prefijo $B[1,i]$ de un vector de bits $B[1,n]$. De manera simétrica, $rank_0(B,i) = i - rank_1(B,i)$. La operación dual a $rank_1$ es $select_1(B,j)$, que permite obtener la posición del j-ésimo bit establecido

a uno en B. La definición de $select_0$ es análoga. Por ejemplo, dado un bitmap $B = 1000110$, $rank_1(B,5) = 2$ y $select_0(B,4) = 7$. Tanto la operación de *rank* como la de *select* se han estudiado mucho en los últimos años y existen implementaciones que permiten su ejecución en tiempo constante y con un consumo de espacio bastante pequeño. Además, la implementación de estas operaciones de forma eficiente continúa siendo un tema de interés hoy en día [MN07, NM07].

2.1.5. Expansión de consultas

La expansión de consultas (*query expansion*) se enmarca en la tarea de *mejora de los resultados* que mostrábamos en el esquema simplificado de la IR en la sección 2.1.1. Esta tarea se centra en la necesidad de mejora de las consultas realizadas por los usuarios para lidiar con problemas derivados de la sinonimia. Este término se refiere al empleo de diferentes palabras para referirse al mismo concepto y tiene un impacto muy grande en el correcto funcionamiento de los sistemas de recuperación de información. La idea básica para mejorar los resultados reside en refinar la búsqueda realizada originalmente por el usuario con más información que se sabe que está relacionada con los términos buscados. Según como se obtenga dicha información para refinar las búsquedas se puede hablar de distintos métodos de mejora de los resultados. La principal clasificación de estos métodos los divide en métodos globales, donde se encuentra la expansión de consultas, y métodos locales, donde se encuentra la retroalimentación. El primer grupo de métodos se caracteriza por reformular los términos de una consulta de manera independiente de la consulta original y de los resultados obtenidos. Por el contrario, el segundo grupo de métodos se caracteriza por reformular la consulta teniendo en cuenta el conjunto de documentos obtenido inicialmente.

Un ejemplo típico de retroalimentación se produce cuando los usuarios realizan una consulta y seleccionan del conjunto de resultados aquellos que consideran más relevantes para volver a realizar la consulta. Por otra parte, un ejemplo típico de expansión de consultas consiste en emplear un tesauro para extender la consulta automáticamente con términos que pueden estar relacionados con los conceptos buscados. Un tesauro se puede definir como un diccionario que contiene listados de palabras o términos empleados para representar conceptos. Además de emplear tesauros, también se ha propuesto el empleo de ontologías para realizar la expansión de consultas. Aunque no es el objetivo de esta sección explicar las aproximaciones para expansión de consultas basadas en ontologías, aprovechamos para introducir este concepto ya que la estructura de indexación para sistemas de recuperación de información geográfica que presentamos en el capítulo 3 combina una estructura basada en ontologías con un índice textual y un índice espacial.

Ontologías

El fin último de las ontologías es permitir que las máquinas puedan intercambiar información de forma efectiva y eficiente. Para ello proporcionan formalismos y estructuran la información permitiendo un cierto grado de razonamiento automático. Gruber [Gru93a] creó una de las definiciones más citadas del concepto de ontología en el ámbito de la informática: *una especificación explícita y formal sobre una conceptualización compartida*. La explicación que realiza Gruber de los distintos conceptos que menciona en esta definición es la siguiente:

- *Conceptualización* se refiere a un modelo abstracto de algún fenómeno del mundo del cual se identifican los conceptos que son relevantes.
- *Explícita* hace referencia a la necesidad de especificar de forma consciente los distintos conceptos que conforman la ontología.
- *Formal* indica que la especificación se debe representar por medio de un lenguaje de representación formalizado.
- *Compartida* refleja que una ontología debe contener el conocimiento aceptado, como mínimo, por el grupo de personas que han de usarla.

A la hora de modelar ontologías también es Gruber [Gru93b] uno de los autores más citados al identificar los cinco componentes básicos que se detallan a continuación:

- *Conceptos*. Son las ideas básicas que se intentan formalizar. Los conceptos pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc. En una ontología se suelen organizar en taxonomías a las que se les pueden aplicar los mecanismos de herencia.
- *Relaciones*. Representan las interacciones y los enlaces entre los conceptos del dominio. Las relaciones más habituales son binarias aunque también se pueden representar relaciones con una aridad superior.
- *Funciones*. Son casos especiales de relaciones donde se identifican elementos mediante el cálculo de una función que considera varios elementos de la ontología.
- *Instancias*. Se usan para representar elementos determinados en una ontología.
- *Axiomas*. Los axiomas formales sirven para modelar sentencias que son siempre ciertas. Normalmente, se usan para representar conocimiento que no puede ser formalmente definido por los componentes descritos anteriormente consiguiendo así una mayor capacidad expresiva del dominio. Además, también se usan para verificar la consistencia de la propia ontología.

Las ontologías tienen muchas aplicaciones y están vinculadas con muchas áreas. En primer lugar, se pueden emplear como *vocabularios de referencia*. En este sentido las ontologías proporcionan un marco de términos comúnmente aceptado por todos los usuarios de la comunidad. En segundo lugar, se emplean como *taxonomías*. En ellas los términos y los conceptos que forman un vocabulario están representados mediante jerarquías de relaciones *is-a* (es-un). El tercer uso está relacionado con la definición de *esquemas*. Hay una relación muy estrecha entre esquemas y ontologías. La principal diferencia reside en que un esquema se suele crear con el objetivo de desarrollar una aplicación del mismo mientras que una ontología no se desarrolla pensando en la aplicación sino simplemente en la necesidad de formalizar el dominio de interés y dar servicio a la comunidad que trabaja en ese dominio. Finalmente, las ontologías están estrechamente vinculadas con la lógica. Sowa [Sow99] describe las relaciones que existen entre las ontologías y la lógica de la siguiente forma: la lógica proporciona la estructura formal y las reglas de inferencia mientras que las ontologías definen los tipos de cosas que existen en un dominio de aplicación.

También es importante mencionar los lenguajes ontológicos. Las ontologías se pueden definir de manera informal empleando el lenguaje natural, de manera semi-informal empleando una forma estructurada y restringida del lenguaje natural, de manera semi-formal empleando un lenguaje artificial formalmente definido, o de manera rigurosamente formal empleando un lenguaje artificial perfectamente definido. Actualmente, existen muchos lenguajes ontológicos (por ejemplo, Ontolingua, RDF, SHOE, etc.) pero, sin duda, un de los más conocidos y especialmente relevante para este trabajo de tesis, ya que es el que empleamos para formalizar nuestra ontología, es OWL (*Web Ontology Language*) [Wor08]. Este *lenguaje de ontologías para la web* diseñado por el W3C ha sido pensado para ser compatible con la arquitectura de la *World Wide Web* en general y de la Web Semántica en particular. Proporciona un lenguaje que otorga a las ontologías las características de ser distribuidas a través de varios sistemas, escalables a las necesidades de la web, compatibles con los estándares web de accesibilidad e internacionalización, abiertas y extensibles. OWL se puede emplear para representar de forma explícita el significado de los términos pertenecientes a un vocabulario y para definir las relaciones que existen entre ellos. Este lenguaje se divide en tres sublenguajes: OWL-Lite, OWL-DL y OWL-Full, cada uno de los cuales proporciona un conjunto definido sobre el que trabajar, siendo el más sencillo OWL-Lite y el más completo OWL-Full. Cada uno de ellos varía en la expresividad que permite así como también en el nivel de inferencia de conocimiento que se puede realizar.

Finalmente, cerramos el estudio acerca de las ontologías citando alguna de las herramientas más importantes de cara a la manipulación de ontologías del conocimiento: Ontolingua, WebOnto, OntoEdit, Racer, Jena y Protégé. Esta última ha sido la seleccionada para la creación de nuestra ontología del espacio geográfico. Protégé [HLEY05] es una herramienta de edición que permite al diseñador construir una ontología del dominio y diseñar formularios personalizados para la adquisición

del conocimiento. Asimismo, la herramienta es una plataforma que se puede ampliar con *widgets* gráficos para tablas, diagramas u otro tipo de componentes para acceder a aplicaciones embebidas. Protégé incorpora una librería que otras aplicaciones pueden emplear para acceder y visualizar su base de conocimientos.

2.1.6. Evaluación en recuperación de información

La evaluación de los sistemas desarrollados dentro del campo ha sido uno de los temas a los que más atención se le ha prestado desde los orígenes de la IR. Dentro de este tema se abordan cuestiones relacionadas con las medidas acerca de lo buenos que son los sistemas desarrollados (si cumplen su función de manera eficaz y eficiente), si las mejoras aplicadas sobre un sistema realmente mejoran sus resultados, etc.

Prueba de este interés podemos citar la conferencia TREC (del inglés *Text REtrieval Conference*) [Nat07b] que se celebra anualmente desde el año 1.992. El objetivo fundamental de esta conferencia es ofrecer un soporte sólido a la investigación de la comunidad de IR proporcionando la infraestructura necesaria para la evaluación de los sistemas desarrollados en el área. La conferencia se organiza en varias líneas que se corresponden con diferentes grupos de sistemas particulares dentro de la IR. Por ejemplo, algunas de las líneas que forman parte de la conferencia actualmente son las de recuperación *ad-hoc*, recuperación con colecciones y consultas en varios idiomas, respuesta de preguntas, etc. La organización de todas las líneas es muy similar ya que la conferencia sigue una filosofía competitiva en la que las mejores aproximaciones suelen ser la base de las propuestas de las siguientes ediciones de la conferencia. Cuando se acepta una nueva línea por parte de la comunidad se publica su especificación, más adelante se publican las colecciones de documentos que formarán la base de la línea en una determinada edición, los investigadores interesados en participar envían los resultados de sus sistemas con esas colecciones y, finalmente, se publican los resultados globales obtenidos por todos los participantes.

Además de los informes publicados anualmente en esta conferencia, los investigadores del área han definido a lo largo de los años de qué temas se ocupa exactamente la evaluación de los sistemas de IR. Algunos de los aspectos que se pueden evaluar en un sistema de IR son la velocidad a la hora de obtener resultados, la presentación de los documentos o la habilidad para encontrar documentos relevantes. Sin embargo, con el paso del tiempo la evaluación de la eficacia de la recuperación se ha ido imponiendo como el aspecto más importante a evaluar en estos sistemas. Además, se han propuesto diferentes funciones para medir esta eficacia. Aunque existen otras medidas que podríamos citar en este apartado como la función E y la F sin duda, la precisión y la recuperación son las medidas más conocidas. En [BYRN99] y [MRS08] se puede encontrar una buena descripción de estas medidas.

Todas estas medidas se suelen basar en el concepto de relevancia. Como no es fácil proporcionar una definición de relevancia completamente satisfactoria la comunidad de

IR optó por permitir que cada modelo proporcione su propia definición de relevancia. Por este motivo, en la sección 2.1.2, al hablar de los modelos de IR ya veíamos que una de las cosas que debían definir de manera explícita dichos modelos era el concepto de relevancia. Intuitivamente, decidir si un documento es relevante o no depende de factores como quién juzga esa relevancia, qué criterio sigue para hacerlo (los seres humanos no suelen ser muy consistentes en su criterio), etc. Por estos motivos, con colecciones reales de documentos es imposible conocer el conjunto de documentos relevantes para cada consulta y los modelos definen este concepto en términos como la similitud entre los documentos y las consultas, la probabilidad de que un documento pertenezca a la clase de los documentos relevantes, etc. Una vez que todos los modelos realizan su propia definición de relevancia es posible medir su eficacia con funciones como la precisión y la recuperación. En primer lugar, la precisión calcula la proporción de documentos relevantes que se han recuperado dentro del conjunto total de los documentos que se han recuperado. La ecuación 2.3 define la fórmula de la precisión empleando la notación de conjuntos del álgebra de Boole.

$$Precisión = \frac{|relevantes \cap recuperados|}{|recuperados|} \tag{2.3}$$

En segundo lugar, la recuperación calcula la proporción de documentos relevantes que se han recuperado dentro del conjunto total de los documentos relevantes que existen en la colección. La ecuación 2.4 define la fórmula de la recuperación empleando la notación de conjuntos del álgebra de Boole.

$$Recuperación = \frac{|relevantes \cap recuperados|}{|relevantes|} \tag{2.4}$$

Intuitivamente la precisión mide que el sistema no recupere muchos documentos que en realidad no son relevantes y la recuperación mide que el sistema recupere una buena proporción de los documentos relevantes. Un buen sistema de IR debe presentar buenas medidas tanto de precisión como de recuperación ya que una cobra sentido en función de la otra. Por ejemplo, se podría alcanzar una recuperación del 100 % recuperando todos los documentos ya que en el conjunto resultado están incluidos todos los documentos relevantes. Sin embargo, la precisión en ese caso sería muy baja ya que también se están recuperando muchos documentos no relevantes.

2.1.7. Resumen

La recuperación de información es uno de los campos de investigación más activos hoy en día debido principalmente a su estrecha vinculación con Internet y la *World Wide Web*. En este apartado hemos realizado una revisión del estado del arte de este campo pero, debido a su amplitud, nos hemos centrado fundamentalmente en los

modelos, estructuras, técnicas de evaluación, y otros componentes relacionados con la indexación de documentos y el acceso a los mismos mediante consultas de diversos tipo.

En primer lugar, definimos el alcance de este campo y presentamos los conceptos básicos más importantes en la sección 2.1.1. A continuación, presentamos los modelos o métodos más representativos para abordar el proceso de la recuperación de información en la sección 2.1.2. Estos modelos definen entre otras cosas cómo se realiza la representación de los documentos a indexar, de las consultas y el proceso computacional. Además, deben proporcionar una definición del concepto de relevancia (cada modelo debe definir qué es un documento relevante). Los tres modelos que describimos son el booleano, el probabilístico y el vectorial. Todos ellos necesitan una estructura de indexación que proporcione un acceso eficiente a los documentos. En la sección 2.1.3 nos centramos en estructuras clásicas, el índice invertido fundamentalmente, y en la sección 2.1.4 estudiamos nuevas técnicas que han ido apareciendo basándose en la compresión. Además, prestamos especial interés a una propuesta reciente para la reorganización de textos comprimidos que les aporta capacidades de indexación. Nuestro interés se debe a que la propuesta se basada en el *wavelet tree*, una estructura compacta que es la base de nuestra estructura de indexación espacial que proponemos como uno de los objetivos de esta tesis. En la sección 2.1.5 presentamos una tarea muy habitual para mejorar los resultados dentro los sistemas de IR, la expansión de consultas. Además, aprovechamos para presentar el concepto de ontología ya que es una estructura ampliamente empleada para este propósito y que también empleamos en este trabajo de tesis, aunque con otra finalidad. La sección 2.1.6 presenta la evaluación de los sistemas de IR, uno de los temas de investigación clave dentro del área que se centra en el estudio de si los sistemas desarrollados funcionan bien, de si los cambios aplicados a un sistema mejoran sus resultados o no, etc.

2.2. Sistemas de información geográfica

A pesar de no estar tan consolidado como el campo de la recuperación de información, otro campo que ha recibido mucha atención en los últimos años es el de los sistemas de información geográfica [Wor04]. Las mejoras recientes en el hardware han hecho posible que la implementación de este tipo de sistemas sea abordable por muchas organizaciones. Además, se ha llevado a cabo un importante esfuerzo colaborativo por parte de dos organismos internacionales (ISO [ISO02] y el *Open Geospatial Consortium* [Ope03]) para definir estándares y especificaciones para la interoperabilidad de los sistemas. Este esfuerzo ha hecho posible que muchas organizaciones públicas estén trabajando en la construcción de infraestructuras de datos espaciales [Glo07] que les permitan compartir su información geográfica.

En esta sección estudiamos el estado del arte de este campo de investigación. En primer lugar, en la sección 2.2.1 presentamos los conceptos básicos del campo.

A continuación, la sección 2.2.2 describe la tendencia actual en el campo hacia la definición de servicios estándar que faciliten el desarrollo de sistemas interoperables. En la sección 2.2.3 presentamos las arquitecturas que se han propuesto en los últimos años para la gestión de la información espacial. Además, en la sección 2.2.4 estudiamos las estructuras de indexación espacial más conocidas y las más aplicables en la recuperación de información geográfica. Por último, la sección 2.2.5 resume las ideas más importantes presentadas acerca de los sistemas de información geográfica.

2.2.1. Conceptos básicos

En la literatura podemos encontrar muchas definiciones diferentes del término GIS y cada una de ellas considera la funcionalidad de estos sistemas desde diferentes perspectivas. En todas ellas podemos identificar tres aspectos diferentes:

- Un GIS es una *colección de herramientas informáticas para realizar análisis y simulaciones geográficas* [LT92, BM98, RSV01].
- Un GIS está basado en un conjunto de *estructuras de datos y algoritmos para recuperar y manipular información geográfica* [Wor04, RSV01].
- Un GIS es una utilidad que *ayuda a tomar decisiones en tareas relacionadas con la geografía* [LGMR01, HA03].

Una definición que combina estos tres aspectos podría describir un GIS como un *sistema informático para modelar, capturar, almacenar, manipular, consultar, recuperar, analizar y visualizar información de manera eficiente, donde parte de esa información es de naturaleza geográfica*.

Los investigadores y profesionales del campo siempre han asumido que existen diferencias considerables entre los sistemas de información de propósito general y los sistemas de información geográfica. Estas diferencias se deben fundamentalmente a las características especiales de la información geográfica. Entre estas características especiales podemos destacar las siguientes:

- *Grandes volúmenes de datos y estructura intrínseca compleja*. Debido a esto adquiere especial importancia la definición de procedimientos eficientes de manipulación y transmisión de la información. Además, también determina una importancia destacada de los metadatos encargados de describir la información geográfica gestionada en los GIS.
- *Número ilimitado de tipos posibles de análisis por parte de los usuarios finales*. Por tanto, los sistemas deben proporcionar una funcionalidad primitiva lo más amplia posible y ofrecer la posibilidad de extensión por parte de los desarrolladores para la funcionalidad avanzada específica.

- *Necesidad de técnicas de análisis y transformación especiales*. Algunos ejemplos de estas técnicas son los algoritmos de búsqueda de rutas, análisis del terreno o la interpolación de datos geográficos. Estas técnicas requieren que los sistemas cuenten con representaciones especiales de los datos geográficos, con estructuras eficientes para el acceso a ellas, con algoritmos específicos de consulta y con interfaces de usuario adaptadas para permitir la interacción con ellas (por ejemplo, mediante herramientas de dibujo de objetos geográficos que puedan ser empleados como parámetros en consultas).

- *Necesidad de opciones de visualización avanzadas*. A diferencia de los sistemas de información tradicionales, donde se manejan fundamentalmente tipos de datos numéricos y textuales, la información geográfica es más compleja y necesita ser presentada al usuario con el nivel de detalle apropiado a diferentes escalas. Por ejemplo, el límite de una ciudad se puede representar como un polígono cuando se visualiza de cerca o como un punto cuando el nivel de detalle requerido es menor.

- *Estrecha relación con la componente tiempo*. Aunque este punto sigue siendo a día de hoy el menos integrado en sistemas comerciales, desde hace años se reconoce que hay una relación muy estrecha entre espacio y tiempo. Esta relación es muy importante para el desarrollo de GIS que soporten información geográfica dinámica.

A modo de resumen, la información geográfica tiene unas características especiales que determinan una serie de diferencias entre un GIS y un sistema de información tradicional. Estas diferencias se reflejan en todos los niveles del sistema desde el almacenamiento de datos, donde se necesitan nuevos tipos de datos y estructuras de acceso a ellos, hasta la presentación de información al usuario, donde se necesitan opciones de visualización mucho más avanzadas que en los sistemas tradicionales. En las próximas secciones veremos las características de los GIS que consideramos más relevantes para el alcance de esta tesis. Comenzaremos revisando en la sección 2.2.2 los estándares internacionales que se han ido definiendo en el área y cuyo objetivo último es definir una arquitectura completa para GIS. Estos estándares son importantes para nosotros porque, debido a la estrecha relación con el campo de la recuperación de información geográfica, podemos emplear tanto la arquitectura como alguno de los estándares definidos como base de nuestra propuesta. Los estándares propuestos por los organismos internacionales están todavía lejos de completar la arquitectura, por este motivo, en la sección 2.2.3 estudiamos otras propuestas de arquitecturas para la gestión de la información geográfica. Por último, en la sección 2.2.4 repasamos las estructuras que permiten un acceso eficiente a la información geográfica.

2.2.2. Estándares internacionales

La definición de estándares, si bien es muy importante en muchas áreas sobre todo en el ámbito de las ingenierías, cobra una especial relevancia en el campo de los sistemas de información geográfica debido a su aplicación en muchas otras áreas (por ejemplo, arquitectura, ingeniería civil, gestión medioambiental, etc.). Sin estos estándares cada desarrollador enfocaría los sistemas desde su propio punto de vista muy condicionado por el campo de aplicación del GIS específico. Esta situación, que tuvo lugar hace años, fue la desencadenante de la creación en 1.994 del *Open Geospatial Consortium* (OGC) [Ope07b] con el objetivo de fomentar la interoperabilidad entre las herramientas GIS que se desarrollan.

Además del OGC, la organización ISO (del inglés *International Organization for Standardization*) por medio del ISO/TC 211 (*ISO Technical Committee 211*) [ISO02] también ha participado activamente en el proceso de estandarización. Aunque originalmente estas dos organizaciones trabajaban de manera independiente, hoy en día trabajan de manera conjunta y con propósitos bien definidos y diferenciados. Mientras que el ISO/TC 211 trabaja en estándares estáticos, abstractos y de alto nivel, el OGC se centra más en estándares orientados a la industria y la tecnología. Este esfuerzo colaborativo ha hecho posible que muchas organizaciones públicas estén participando en el desarrollo de infraestructuras de datos espaciales [Glo07] para compartir su información espacial [Dir07].

Para los objetivos de esta tesis la estandarización es importante por dos motivos. En primer lugar, porque de ella surgió una arquitectura para GIS en la que nos basamos a la hora de definir la arquitectura para sistemas de recuperación de información geográfica que proponemos como segunda aportación de esta tesis. En segundo lugar, porque hay muchos puntos en común entre ambos campos, lo que nos permite utilizar en nuestra arquitectura servicios estandarizados en el campo de los GIS.

La arquitectura para sistemas de información geográfica es el objetivo último del OGC y del ISO/TC 211. Estas organizaciones han ido definiendo todos los componentes de la arquitectura mediante la especificación de estándares. Estos estándares los podemos clasificar en tres grupos de acuerdo a su naturaleza: modelos, lenguajes y servicios. En la primera categoría se encuentran los estándares que definen modelos para caracterizar y codificar la información geográfica y la información de soporte relacionada. En la segunda categoría las especificaciones definen lenguajes para codificar la semántica, sintaxis y esquemas de los recursos geográficos. Finalmente, en la tercera categoría se encuentran las definiciones de servicios. Es decir, de procesos de negocio perfectamente encapsulados y con una interfaz de acceso bien definida.

Todos estos estándares se pueden agrupar en una serie de capas que se corresponden con la separación en capas típica de los sistemas de información en general. En la figura 2.4 mostramos un esquema simplificado de la arquitectura lógica de un GIS según el OGC. En esta figura, se pueden identificar las tres capas lógicas. En la parte

inferior, la *capa de gestión de la información* encargada del almacenamiento físico de los datos y del mantenimiento de los mismos, incluyendo conjuntos de datos, esquemas conceptuales y metadatos. En el centro de la figura se encuentra la *capa de procesamiento*, encargada de procesos computacionales que involucran cantidades de datos substanciales. En la parte superior se encuentra la *capa de interacción con el usuario*, responsable de la interacción física con el usuario. Además, la arquitectura se completa con unos componentes trasversales que se pueden agrupar en *servicios de flujo de trabajo y tareas*, encargados de procesos que involucran varias tareas que deben ser realizadas por distintos componentes, y *servicios de comunicación*, encargados de conectar los distintos servicios y capas entre sí. En la siguiente sección estudiaremos un poco más en profundidad la situación actual de las arquitecturas para la gestión de la información espacial.

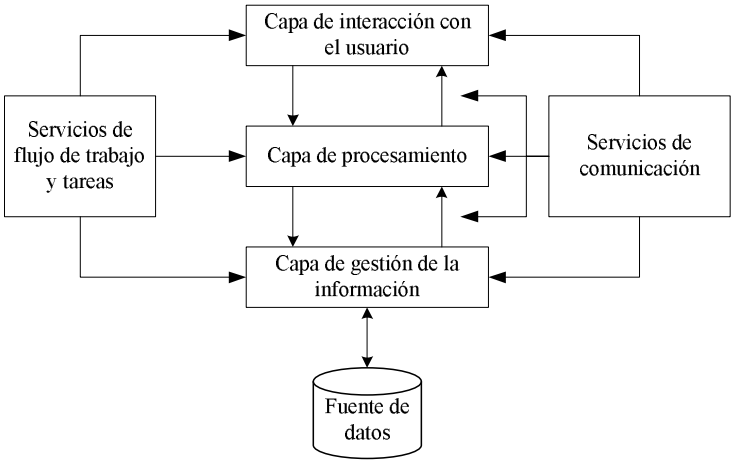


Figura 2.4: Arquitectura lógica de un GIS según el OGC.

Como ya hemos mencionado la estandarización también resulta interesante para nuestro trabajo por la posibilidad de emplear servicios estandarizados en nuestra arquitectura para sistemas de recuperación de información geográfica. Algunos de los estándares más importantes y consolidados que podemos emplear en nuestra arquitectura son aquellos empleados para el almacenamiento y acceso a la información geográfica. Dentro de este grupo se encuentra el estándar SFS [Ope06] (del inglés *Simple Features Standard*). En concreto, la especificación que citamos proporciona un procedimiento bien definido para que las aplicaciones almacenen y accedan a sus datos en bases de datos relacionales u objeto-relacionales. También hemos de tener en cuenta los estándares para la representación en mapas de información geográfica. Sin duda, el ejemplo más conocido es el WMS [Ope02] (del inglés *Web Map Service*). Este estándar permite generar imágenes (PNG, JPEG, GIF, etc.) que representan los mapas

definidos a partir de la información geográfica disponible en las fuentes de datos a las que tiene acceso el servicio. Finalmente, podemos emplear alguno de los estándares para la representación e intercambio de valores de datos geográficos. GML [Ope07a] (del inglés *Geographic Markup Language*) es posiblemente el estándar más empleado para este propósito. Se define como un sublenguaje de XML, descrito mediante un esquema (*XML Schema*), que permite modelar, transportar y almacenar información geográfica sobre todo con propósitos de intercambio entre los diversos servicios que componen el GIS.

2.2.3. Arquitectura genérica para GIS

Como veíamos en la sección anterior, el objetivo último de los organismos de estandarización es la definición de una arquitectura global para sistemas de información geográfica mediante la estandarización de los servicios que componen dicha arquitectura. Sin embargo, este objetivo último está bastante lejos de haberse alcanzado. En este tema se centraba la tesis doctoral de uno de los directores del presente trabajo, Miguel R. Luaces [Lua04], cuya principal aportación es una arquitectura genérica para sistemas de información geográfica. En esta sección, vamos a describir brevemente esta arquitectura ya que es totalmente compatible con las propuestas que se han ido realizando tanto desde el ISO/TC 211 como desde el OGC.

En la figura 2.5 mostramos la propuesta de arquitectura para GIS realizada en la tesis que acabamos de citar. Al igual que las propuestas desde el OGC y el ISO/TC 211, y siguiendo el ejemplo de las arquitecturas de los sistemas de información de propósito general, la arquitectura contempla tres capas que agrupan funcionalidad relacionada: la *capa de datos*, la *capa de la lógica de la aplicación* y la *capa de presentación*. Cada una de estas capas no está formada por un módulo de software monolítico sino que está formada por un conjunto de componentes software independientes que dividen su funcionalidad. A continuación, vamos a describir brevemente la responsabilidad y los componentes de cada capa:

- *Capa de datos*. Esta capa proporciona funcionalidad de gestión de datos de manera independiente de la tecnología software. Debido a que puede haber diferentes tipos de fuentes de datos, la arquitectura interna de la capa debe estar organizada siguiendo un patrón de mediadores-adaptadores. Esta organización marca la división en tres capas independientes. La capa de *fuentes de datos* proporciona un adaptador para cada tipo de fuente de datos. Esos adaptadores permiten que la capa *mediador* pueda proporcionar un modelo conceptual único para toda la información geográfica, incluyendo tipos de datos, lenguaje de consulta, metadatos y catálogo de información. Finalmente, la capa de *servicios de datos* proporciona una colección de perfiles del modelo conceptual para aplicaciones específicas ocultando parte de la complejidad subyacente.

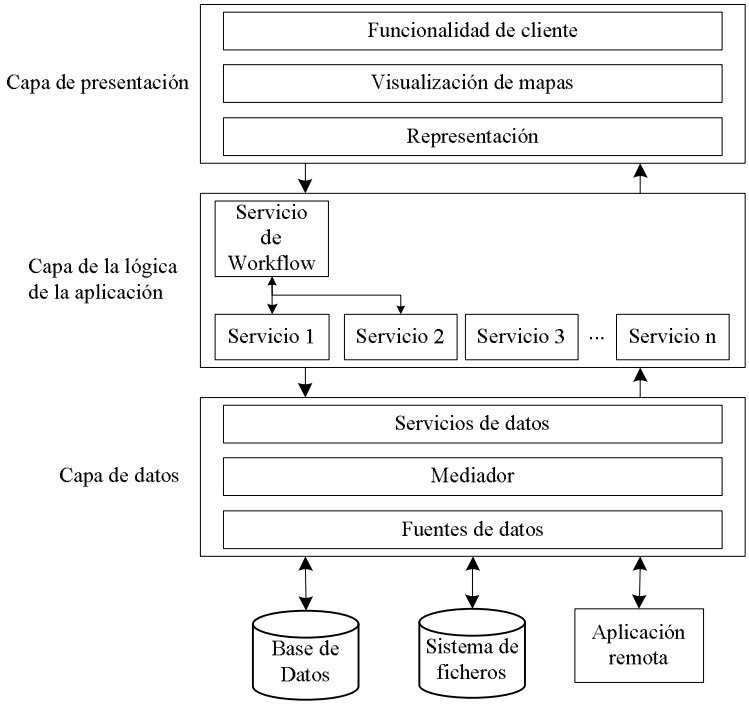


Figura 2.5: Una arquitectura genérica para GIS.

- *Capa de la lógica de la aplicación*. En esta capa se implementa la funcionalidad propia del sistema de tal forma que, de cara a la capa que se asienta en el siguiente nivel, proporciona una interfaz bien definida donde se ofrecen todas las operaciones de alto nivel propias del sistema. A diferencia de la capa de datos, esta capa no se encuentra dividida en otras capas más pequeñas sino que propone que la funcionalidad debe ser implementada por un conjunto de *servicios* independientes. Cada uno de estos servicios debe realizar una tarea sencilla y bien definida. Además, debe proporcionar una interfaz clara donde se defina el conjunto de operaciones que implementa y una descripción de los resultados. Para realizar tareas más complejas, la capa contempla la definición de un *servicio de workflow* que permita construir nuevos servicios encadenando una colección de otros más sencillos.

- *Capa de presentación*. Implementa la interfaz de usuario del sistema que permite la visualización de datos, la manipulación de datos y la entrada de datos. Esta capa se divide en otras tres más sencillas. La capa de *representación*, encargada de convertir la información geográfica en representaciones cartográficas que puedan ser renderizadas en un dispositivo de visualización; la capa de *visualización de mapas* que se encarga de renderizar las representaciones cartográficas en el dispositivo de visualización; y la capa de *funcionalidad de cliente* que permite la implementación de funcionalidades para la interfaz de usuario.

Esta arquitectura, además de ser compatible con las especificaciones del OGC y del ISO TC/211, tiene otras características que hacen que se deba tener en cuenta, además de para el desarrollo de GIS, para la definición de arquitecturas en áreas tan similares como la recuperación de información geográfica. En primer lugar, es *conceptual*, en el sentido de que se puede poner en práctica empleando muchas implementaciones diferentes. Es *flexible* a cambios en los requisitos funcionales gracias a la separación en capas. La modularidad hace que los servicios o componentes definidos sean altamente *reusables*. Además, es *extensible* ya que se puede añadir fácilmente nueva funcionalidad incluyendo nuevos módulos. Es *escalable* debido a la alta modularidad de la arquitectura en tres capas que permite la distribución de los componentes de la aplicación en varios servidores. Finalmente, es *tolerante a fallos* ya que la arquitectura facilita la implementación de redundancia en cada capa.

2.2.4. Estructuras de indexación espacial

Como veíamos en la sección 2.2.1, la información geográfica tiene una serie de características especiales, entre las que se encuentra la necesidad de gestión de grandes volúmenes de datos, que aportan unos requisitos específicos a los GIS con respecto a los sistemas de información tradicionales. Estas características han determinado que

uno de los temas de investigación más importantes en el área haya sido el desarrollo de estructuras de indexación que permitan un acceso eficiente a las voluminosas colecciones de datos gestionadas en este tipo de sistemas. A lo largo de los años, se han propuesto muchas estructuras diferentes para lograr este objetivo. Dichas estructuras se pueden clasificar [GG98] de manera general en métodos de acceso a puntos (PAMs, del inglés *Point Access Methods*) y métodos de acceso espacial (SAMs, del inglés *Spatial Access Methods*). Los métodos que se encuentran en el primer grupo se caracterizan por mejorar el acceso a colecciones de datos formadas por puntos espaciales. En cambio, los métodos que se encuentran en el segundo grupo son más generales ya que trabajan con colecciones de todo tipo de objetos geográficos (por ejemplo, puntos, líneas, polígonos, etc.).

En los últimos años, se han propuesto muchos métodos de acceso espacial y muchos métodos de acceso a puntos diferentes. En [GG98] y en [Sam06] se puede encontrar un buen resumen de las estructuras más relevantes en cada categoría. El objetivo de los métodos en ambos grupos es mejorar la eficiencia a la hora de acceder a subconjuntos de datos que se ajustan a una consulta determinada. Existen muchos tipos diferentes de consultas que se pueden resolver mediante estructuras de indexación espacial (por ejemplo, consultas de los k vecinos más cercanos, de proximidad, de adyacencia, etc.). Sin duda, uno de los tipos de consulta más empleados en GIS reales y que deben resolver las estructuras de ambos grupos son las consultas de tipo región o ventana. Este tipo de consulta define una región rectangular en el espacio (también conocida como ventana de consulta) y obtiene como resultado todos los objetos geográficos indexados en la estructura que se solapan con dicha región.

El R-tree [Gut84] es uno de los métodos de la categoría SAM más populares y se puede considerar un ejemplo paradigmático. Esta estructura se basa en un árbol balanceado derivado del B-tree [BM72] que divide el espacio en rectángulos de cobertura mínima (MBRs, del inglés *Minimum Bounding Rectangles*) agrupados jerárquicamente y que pueden solaparse entre ellos o no. El número de nodos hijo de cada nodo interno varía entre un mínimo y un máximo. El árbol se mantiene balanceado dividiendo aquellos nodos que tienen un número de descendientes por encima del umbral de carga máxima y combinando aquellos otros que tienen un número de descendientes por debajo del umbral de carga mínima. Cada nodo hoja tiene asociado un MBR que delimita el área del espacio que cubre ese nodo. Además, los nodos internos también almacenan un MBR que delimita el área que cubren todos sus descendientes. La descomposición del espacio que proporciona el R-tree es adaptativa (es decir, dependiente de la distribución de los objetos geográficos indexados) y puede presentar solapes (es decir, los nodos del árbol pueden representar regiones no disjuntas). En la figura 2.6 mostramos un ejemplo de construcción de este tipo de estructura. En la parte izquierda de la figura mostramos los objetos distribuidos en el espacio y en la parte derecha la estructura resultante. Evidentemente, esta estructura no sólo permite indexar rectángulos sino que lo que estamos representando son los MBRs de los objetos reales.

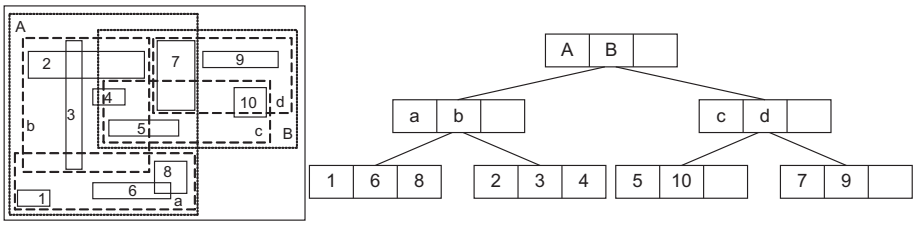


Figura 2.6: Ejemplo de construcción de un R-tree.

Sobre la base del R-tree propuesto por Guttman en 1.984 [Gut84] se han ido proponiendo muchas variantes para mejorar su eficiencia tanto en el caso general (por ejemplo, el R+-tree [SRF87] y el R*-tree [BKSS90]) como en problemas más específicos (por ejemplo, las versiones de carga masiva diseñadas para colecciones de objetos estáticas).

En primer lugar, las variantes más conocidas para mejorar la eficiencia del R-tree en el caso general son el R+-tree y el R*-tree. La diferencia de estas estructuras, tanto con la variante original como entre ellas, reside en cómo se realiza el proceso de construcción del árbol para tratar de disminuir la cantidad de solapes, ya que los solapes penalizan la eficiencia de las consultas. En concreto, el R+-tree elimina el solape en los nodos internos insertando los objetos en múltiples nodos hoja si es necesario. Por su parte, el R*-tree se basa en la idea de que la variante original es muy susceptible del orden de inserción de los objetos y trata de minimizar los solapes mediante una estrategia de *borrado y re-inserción* para buscar a cada objeto el lugar más apropiado en el árbol.

En segundo lugar, los algoritmos de carga masiva (*bulk loading*) para colecciones estáticas más conocidos son *Nearest-X* (NX) [RL85], *Hilbert Sort* (HS) [KF93] y *Sort-Tile-Recursive* (STR) [LLE97]. La idea de estos algoritmos consiste en preprocesar los MBRs de los objetos a indexar construyendo grupos de M rectángulos próximos que se almacenarán en el mismo nodo (ya que M es el factor de ramificación del nodo). El árbol se construye de abajo hacia arriba repitiendo recursivamente el algoritmo de creación de grupos. Las diferentes versiones se diferencian en cómo se construyen esos grupos. Por ejemplo, el STR R-tree, que es el algoritmo de empaquetado más eficiente, ordena los MBRs en el eje X, los agrupa en $S = \sqrt{\lceil N/M \rceil}$ franjas verticales, ordena cada franja en el eje Y, y construye cada nodo tomando los MBRs en grupos de M.

Por otra parte, el K-d-tree [Ben75] es uno de los métodos de la categoría PAM que más se ha empleado, debido fundamentalmente a su sencillez y eficiencia. Cuando esta estructura se emplea para indexar colecciones de puntos se denomina más comúnmente *Point k-d-tree*. En general, el K-d-tree, y los métodos derivados de él, se basan en árboles de búsqueda binarios que representan una subdivisión recursiva del dominio

basada en el valor de un único atributo en cada nivel del árbol. En la figura 2.7 mostramos un ejemplo de construcción de un K-d-tree sobre una colección de puntos distribuidos en el espacio. De nuevo, en la parte izquierda de la figura representamos los objetos distribuidos en el espacio y como se va realizando la partición del mismo y en la parte derecha representamos la estructura resultante.

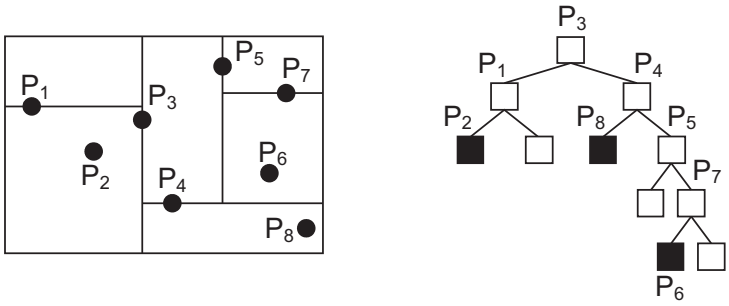


Figura 2.7: Ejemplo de construcción de un K-d-tree.

Existen numerosas variantes de esta estructura que se suelen identificar según cómo realizan la partición del espacio. En concreto, las dos características más importantes de dicha partición son: i) si las líneas de partición del espacio deben pasar por puntos reales del conjunto a indexar y ii) si se sigue algún esquema estricto de alternancia entre los ejes de partición (por ejemplo, si se va alternando cíclicamente entre las dimensiones). Como veremos en el capítulo 5, en los experimentos que realizamos para comparar nuestra estructura de indexación espacial frente a otras alternativas empleamos una aproximación estática propuesta en [Ben75] que asume que se conoce de antemano la colección completa de puntos a indexar. En esta variante, las líneas de partición deben pasar obligatoriamente por los puntos del conjunto de datos y los ejes deben alternarse cíclicamente en un orden constante.

Ambas estructuras, al igual que la mayoría de las propuestas de estructuras de indexación espacial existentes, están optimizadas para minimizar el tiempo necesario para resolver distintos tipos de consultas, fundamentalmente consultas de tipo región. Sin embargo, estas estructuras ignoran casi por completo el espacio necesario para almacenar la estructura. En nuestra opinión, una buena relación entre el espacio necesario para almacenar la estructura y su eficiencia a la hora de realizar búsquedas es más interesante que únicamente la optimización de la eficiencia de las búsquedas. Para realizar esta afirmación nos basamos en que el tiempo de acceso a disco es varios ordenes de magnitud superior al tiempo de acceso a memoria principal. Por tanto, si conseguimos reducir el tamaño de la estructura de indexación, e incluso de los datos, lo suficiente como para que pueda operar en memoria principal con conjuntos reales de datos, la eficiencia de la estructura propuesta será mucho mayor que la de cualquier

alternativa que tenga que trabajar en disco.

2.2.5. Resumen

Los sistemas de información geográfica constituyen un campo de investigación muy atractivo debido a su utilidad en áreas tan diversas como la arquitectura, la ingeniería civil o la gestión medioambiental. En esta sección, hemos revisado los conceptos más importantes del estado del arte del campo. En primer lugar, en la sección 2.2.1 revisamos los conceptos básicos de los GIS centrándonos en las características de la información geográfica que diferencian a estos sistemas de los sistemas de información tradicionales. En segundo lugar, en la sección 2.2.2 repasamos las propuestas de dos organismos internacionales, OGC e ISO/TC 211, para definir estándares para lograr la interoperabilidad de los GIS. El objetivo final de estos organismos es definir una arquitectura completa para el desarrollo de GIS mediante la definición de estándares. Tanto la arquitectura, a nivel general, como los estándares, a nivel particular, son importantes para nuestros objetivos porque existen muchos puntos en común y es posible tanto la reutilización de algunas de las ideas en las que se basa la arquitectura como de alguno de los servicios. El objetivo final de estos organismos de definir una arquitectura completa está todavía bastante lejos de alcanzarse. Por este motivo, en la sección 2.2.3 estudiamos una arquitectura genérica para la gestión de información espacial propuesta en el 2.004 en la tesis doctoral de uno de los directores del presente trabajo. Esta arquitectura nos sirve como base de nuestra propuesta de arquitectura para sistemas de recuperación de información geográfica. Finalmente, en la sección 2.2.4 estudiamos algunas de las estructuras de indexación espacial más destacadas.

2.3. Recuperación de información geográfica

Recientemente ha surgido un nuevo campo de investigación en la confluencia de la recuperación de información con los sistemas de información geográfica denominado recuperación de información geográfica. Este campo trata de cubrir una demanda cada vez más habitual por parte de los usuarios que es la posibilidad de acceder a la información mediante consultas donde se tenga en cuenta tanto el ámbito textual como el ámbito espacial de la misma. Esta necesidad no está ligada tan sólo a la web sino que también aparece en bibliotecas digitales, sistemas de gestión documental y en cualquier sistema de información en general que gestione información textual.

A pesar de ser un campo de investigación muy reciente, ha atraído la atención de investigadores procedentes tanto del área de los GIS como de la IR y, en los últimos años, han aparecido bastantes propuestas en las diferentes líneas de trabajo que lo componen. En esta sección estudiamos la situación actual del estado del arte

de este campo. En primer lugar, en la sección 2.3.1 presentamos algunos conceptos básicos para familiarizar al lector con el tema. A continuación, la sección 2.3.2 presenta las consultas de usuario típicas que se pueden resolver en este tipo de sistemas. En la sección 2.3.3 presentamos las propuestas que han ido apareciendo para extraer el ámbito espacial de los documentos. Además, en la sección 2.3.4 estudiamos las estructuras de indexación que permiten tener en cuenta, además del ámbito textual de los documentos, su ámbito espacial. La sección 2.3.5 describe las propuestas para ordenar en un *ranking* de relevancia los resultados obtenidos en estos sistemas. Por último, la sección 2.1.7 resume las ideas más importantes presentadas acerca de los sistemas de recuperación de información geográfica.

2.3.1. Conceptos básicos

Al igual que hicimos en el estado del arte de la IR y de los GIS, en este primer apartado vamos a definir qué es exactamente la recuperación de información geográfica, qué temas se encuentran entre sus objetivos y cuales no. Además, ya que este nuevo campo ha surgido de la confluencia de la IR y de los GIS vamos a identificar puntos en común con esos campos y también algunas diferencias.

En primer lugar, partiendo de las definiciones de IR y de GIS realizadas en las secciones 2.1.1 y 2.2.1 respectivamente, podemos proponer la siguiente definición: *la recuperación de información geográfica (GIR) se ocupa de encontrar material (normalmente documentos) de una naturaleza no estructurada (normalmente texto) que se encuentra en grandes colecciones (normalmente almacenada de forma digital) y satisface una necesidad de información donde parte de esa información es de naturaleza geográfica*. Al igual que sucede en la IR, el caso más habitual consiste en la recuperación de documentos. Por este motivo, realizando la misma simplificación, podemos decir que el objetivo de la recuperación de información geográfica es desarrollar sistemas informáticos que permitan a los usuarios *recuperar documentos relevantes tanto temática como geográficamente en respuesta a consultas de la forma* <*tema, localización*>. Para lograr este objetivo, entre los temas de interés del área se encuentran la definición de arquitecturas de sistemas, estructuras de indexación y otros componentes que permitan modelar, capturar, almacenar, manipular, consultar, recuperar, analizar y visualizar información de manera eficiente. Además, estas tareas presentan una dificultad añadida sobre la que ya tienen en el área de la IR debido a que parte de esa información es de naturaleza geográfica, con las características especiales y requisitos que eso conlleva, y que analizábamos en la sección 2.2.

A pesar de la naturaleza geográfica de parte de la información, común entre los GIS y los sistemas GIR, existen dos diferencias fundamentales en los requisitos de ambos tipos de sistemas que se deben tener en cuenta. En primer lugar, la componente espacial de las consultas en sistemas GIR es mucho más sencilla que las consultas que se pueden realizar en GIS y que involucran relaciones espaciales complejas. Por

ejemplo, una consulta típica en un sistema GIR puede ser *monumentos en A Coruña* y, por tanto, la única relación espacial que hay que comprobar es si el ámbito espacial del documento se encuentra dentro de la geometría asociada con A Coruña o no. Por otro lado, una consulta sencilla en GIS puede ser *monumentos en ayuntamientos adyacentes al lugar donde se encuentra un determinado hotel*. En este caso, habría que encontrar el ayuntamiento donde se encuentra el hotel y luego buscar los monumentos que se encuentran en los ayuntamientos adyacentes a él. La segunda diferencia se encuentra en las consultas típicas de ambos sistemas. Mientras que en los GIS las consultas son parecidas a las de las bases de datos tradicionales, donde de manera categórica los objetos pertenecen al resultado o no, en los sistemas GIR son parecidas a las de los sistemas de IR, donde los objetos pueden pertenecer al resultado con una cierta probabilidad. Siguiendo con las consultas del ejemplo, en el caso de los sistemas GIR, monumentos situados en Ferrol (un ayuntamiento próximo a A Coruña aunque no adyacente) pueden ser relevantes para el usuario (aunque con menor importancia que los situados en A Coruña), mientras que en el caso de los GIS sólo los monumentos en los ayuntamientos adyacentes forman parte del resultado de la consulta.

De manera similar a lo que ocurre con la IR, donde muchas veces se restringe su alcance a la recuperación de documentos en la web, una confusión habitual a la hora de definir el alcance de la recuperación de información geográfica es restringirlo a lo que hacen los motores de *búsqueda local* como Google Maps [Goo07], Yahoo Mapas [Yah07] o Microsoft Live Search [Mic07]. Aunque podemos considerar estos sistemas dentro del ámbito de la recuperación de información geográfica, ya que cumplen la definición al recuperar información relevante tanto temática como geográficamente, están posiblemente más próximos al campo de los GIS. Su funcionalidad principal se basa en el almacenamiento estructurado (por ejemplo, en tablas de bases de datos) de negocios, organizaciones, puntos de interés y elementos pertenecientes a otras categorías prefijadas. Para cada uno de los elementos se almacena también su situación geográfica y esto permite que los usuarios puedan realizar búsquedas que las tengan en cuenta (por ejemplo, *hoteles en A Coruña*). Para completar los resultados, pueden buscar la parte textual de las consultas en sus motores de búsqueda en web tradicionales.

Además de definir el alcance de la recuperación de información geográfica, este apartado también tiene el objetivo de describir las tareas más importantes que componen el proceso. El esquema básico del mismo no dista mucho del presentado en la sección 2.1.1 para el proceso de recuperación de información. Sin embargo, algunas de las tareas deben ser ampliadas, e incluso completadas con otras, para tener en cuenta el ámbito espacial de los documentos. En primer lugar, se amplía la necesidad de información del usuario para tener en cuenta la componente espacial y, por lo tanto, se modifica la forma de representarla en consultas. Por este motivo, en la sección 2.3.2 estudiamos las consultas relevantes para este tipo de sistemas. En segundo lugar, para poder resolver estos nuevos tipos de consultas, antes de indexar los documentos se debe extraer su ámbito espacial. Este proceso, denominado habitualmente geo-

referenciación de los documentos, se puede dividir en dos más sencillos: localizar posibles nombres de lugar en los textos de los documentos y, posteriormente, traducir esos nombres de lugar a coordenadas geográficas. En la sección 2.3.3 describimos las propuestas que se han realizado para abordar esta tarea. La geo-referenciación nos permite incluir los documentos en estructuras de indexación que, además de indexar los términos clave de los documentos, indexan sus coordenadas o su referente geográfico. Describimos las propuestas relacionadas con estas estructuras en la sección 2.3.4. Finalmente, la tarea de obtener los objetos recuperados que satisfacen las consultas y ordenarlos según su relevancia en un *ranking* también debe ser extendida para tener en cuenta la relevancia espacial. En la sección 2.3.5 estudiamos alternativas para calcular esta relevancia espacial y combinarla con la textual.

2.3.2. Consultas de interés

Como ya hemos mencionado, las consultas habituales en los sistemas GIR tienen una componente textual y una componente espacial. Cada una de estas componentes está basada en las consultas propias de la IR y de los GIS respectivamente. Sin embargo, como veíamos en el apartado anterior, existen algunas diferencias significativas. En esta sección vamos a revisar las consultas típicas tanto en sistemas de IR como en GIS y analizar qué tipos de consultas o qué características de esas consultas son aplicables en sistemas GIR.

En la sección 2.1.3 enumerábamos los tres tipos de consulta más habituales en sistemas de IR: consultas basadas en términos clave, consultas de reconocimiento de patrones en texto y consultas sobre la estructura de los textos. Veíamos también que las consultas basadas en términos clave constituyen el núcleo de la IR. En este grupo se encuentran las consultas típicas que se realizan en los motores de búsqueda y se formulan mediante palabras, frases y operadores lógicos que permiten combinarlas. Este tipo de consultas también son las más interesantes para la recuperación de información geográfica, y por tanto para el alcance de esta tesis. Dentro de este grupo se encuentran las consultas que se realizan empleando términos clave, bien de forma individual o bien en un contexto (por ejemplo, términos agrupados en frases), las consultas que emplean operadores lógicos y las consultas en lenguaje natural. Todas estas consultas tienen en común que la necesidad de información del usuario se expresa empleando un conjunto de términos clave.

Por otra parte, en la sección 2.2.4 enumerábamos algunos tipos de consulta habituales en GIS: consultas de los k vecinos más cercanos, de proximidad, de adyacencia, de región, etc. En [GG98] se puede encontrar un buen resumen de estos tipos de consulta. Las consultas espaciales con más aplicabilidad en sistemas GIR son las consultas de tipo región o ventana, en las que el usuario define una ventana de consulta y obtiene como resultado todos los objetos geográficos que se solapan con ella. Esto se debe a que, como veíamos en la sección anterior, la complejidad de las

consultas espaciales de los usuarios de sistemas GIR con respecto a los usuario de GIS es mucho menor.

Por tanto, podemos distinguir cuatro tipos de consultas interesantes en sistemas de recuperación de información geográfica. En primer lugar, son interesantes las consultas textuales puras que proceden directamente del campo de la IR (por ejemplo, documentos donde aparezcan las palabras *hotel* y *mar*). Los documentos en sistemas GIR se encuentran indexados tanto temática como geográficamente por lo que es posible acceder a ellos empleando únicamente la componente textual. De manera análoga, son interesantes las consultas espaciales puras que proceden directamente del campo de los GIS (por ejemplo, documentos que se refieran a un área geográfica determinada como puede ser una ventana de consulta). El tercer tipo de consultas representa las consultas que combinan los dos tipos anteriores. Es decir, consultas con una componente textual expresada empleando términos clave y una componente espacial expresada mediante una consulta espacial (por ejemplo, documentos con la palabra *hotel* que se encuentren geo-referenciados en una ventana de consulta). Finalmente, podemos distinguir un cuarto tipo de consultas en las cuales el usuario emplea únicamente términos clave para expresar su necesidad de información, como en las consultas textuales puras, pero algunos de esos términos clave hacen referencia a un lugar o entidad geográfica (por ejemplo, documentos con la palabra *hotel* referidos a *España*). En el capítulo 3, donde describimos nuestra estructura de indexación, volveremos a tratar en profundidad estos tipos de consulta para ver cómo se pueden resolver con nuestra estructura.

2.3.3. Geo-referenciación de documentos

La geo-referenciación de un documento es el proceso mediante el cual se le asigna un referente espacial a ese documento. Por referente espacial nos referimos a su ámbito geográfico que se puede expresar empleando las coordenadas geográficas de cada lugar citado en el texto, un *bounding box* que agrupe esas coordenadas o cualquier otra forma de definir la zona del espacio geográfico en la cual es relevante ese documento. Por ejemplo, si un documento cita las ciudades de *Londres* y *Liverpool*, para geo-referenciar ese documento podríamos usar las coordenadas de esas dos ciudades, un *bounding box* que contenga a ambas o incluso la geometría completa de *Inglaterra*.

El principal problema que están abordando los investigadores que trabajan en este tema es la ambigüedad de las referencias geográficas. Un estudio presentado en [GM05] estima que más del 67 % de los nombres de lugar citados en textos son ambiguos. Además, en [TSM03], los autores distinguen dos tipos de ambigüedad. En primer lugar, un nombre de lugar puede ser ambiguo. Por ejemplo, Londres es la capital del Reino Unido pero también es una ciudad de Ontario en Canadá. Este tipo de ambigüedad se conoce también como *polisemia*. En segundo lugar, se pueden emplear muchos nombres de lugar para la misma localización geográfica. Por ejemplo,

la ciudad de Los Ángeles también puede ser referida como LA. Este segundo tipo de ambigüedad se denomina *sinonimia*. Podemos considerar un tercer tipo de ambigüedad que se debe al empleo del mismo nombre tanto para referirse a un lugar como para referirse a una organización, compañía o persona. Por ejemplo, Santiago se puede emplear para referirse a una persona o a una ciudad. De estos tres tipos de ambigüedad, la polisemia es posiblemente el más difícil de resolver ya que requiere emplear información geográfica contenida en el contexto.

Podemos descomponer el proceso de geo-referenciación en dos etapas: la localización de los nombres de lugar que se citan en el texto y la traducción de esos nombres de lugar a su referente geográfico. El referente geográfico permite incluir los documentos en estructuras de indexación espacial. A continuación, estudiaremos en más detalle estas dos etapas y las soluciones propuestas para abordarlas.

Localización de los nombres de lugar

A diferencia de la información gestionada en GIS, que tiene una estructura determinada y, por tanto, es posible conocer con certeza en qué campos se encuentran almacenadas las referencias geográficas y el tipo de las mismas (es decir, si son ciudades, provincias, códigos postales, etc.), la información en sistemas GIR no está estructurada. En este tipo de sistemas, la información se encuentra en el texto de los documentos y, para localizar posibles referencias geográficas, hay que analizar esos textos.

Las técnicas más habituales para realizar el análisis de documentos con el objetivo de encontrar referencias geográficas provienen del área del análisis lingüístico. En concreto, el etiquetado de los componentes del texto (*Part-Of-Speech tagging*) [Bri92] y las técnicas de reconocimiento de entidades con nombre (*Named-Entity Recognition*) [CR97, PKLS05] son las técnicas más empleadas para este propósito. El etiquetado de los componentes del texto consiste en etiquetar los *tokens* o elementos indivisibles del texto secuencialmente con etiquetas sintácticas, como *verbo* o *gerundio*. Por otra parte, el reconocimiento de entidades con nombre se ocupa de la búsqueda en el texto de referencias que pertenecen a un conjunto predefinido de categorías, como *nombres de persona*, *nombres de organización* o *nombres de lugar*. La combinación de ambas técnicas presenta unos resultados bastante buenos para la obtención de candidatos a nombre de lugar.

Sin embargo, a pesar de los buenos resultados para la obtención de candidatos a nombre de lugar, a la hora de traducir esos candidatos a verdaderas referencias geográficas es donde aparecen los problemas relacionados con la ambigüedad. Podríamos decir que éste es un problema a medio camino entre las dos etapas que componen la geo-referenciación. En primer lugar, hay que determinar si los candidatos son verdaderas referencias geográficas o no. Para este propósito se pueden emplear los *gazetteers* que son diccionarios geográficos que contienen listas de nombres de lugar e

información relevante acerca de esos lugares (por ejemplo, su localización geográfica). Aunque hace unos años era un problema disponer de *gazetteers* completos [PCV+00], a día de hoy existen muchos recursos de este tipo disponibles y, además, de manera gratuita. Un problema mucho más complicado es la desambiguación del nombre de lugar una vez que se ha comprobado que es una referencia geográfica al estar incluido en un *gazetteer*. Cuando los seres humanos leemos textos empleamos pistas contenidas en todo el documento para realizar esta desambiguación. Por ejemplo, si en un documento aparece la palabra Santiago y también A Coruña asumiremos que el texto se refiere a Santiago de Compostela, mientras que si aparece Atacama asumiremos que se refiere a Santiago de Chile. Realizar este proceso de forma automática es una tarea bastante complicada. En la tesis doctoral de B. Martins [dGM08] se describen unos principios básicos para ayudar a distinguir el verdadero sentido de una referencia geográfica:

- *Un referente por texto*. Una referencia geográfica ambigua que aparezca varias veces en un texto es muy probable que se refiera siempre al mismo lugar.
- *Referentes relacionados en el texto*. Se pueden emplear otras referencias mencionadas en el texto para desambiguar un nombre de lugar. El ejemplo que mencionábamos para distinguir entre un documento referido a Santiago de Compostela de otro referido a Santiago de Chile se basa en este principio.
- *Referente más probable*. Si no se pueden emplear otras referencias contenidas en el texto, se le puede asignar el referente más probable al nombre de lugar. Por ejemplo, si Londres aparece sin más referencias geográficas en un texto es mucho más probable que se refiera a la capital del Reino Unido que a la ciudad de Ontario.

En el proyecto SPIRIT [JPR+02] se emplea una ontología del espacio geográfico en lugar de un *gazetteer*. La búsqueda en la ontología, además de comprobar si un nombre de lugar candidato es una verdadera referencia geográfica, proporciona la desambiguación en base a la profundidad en la que se encuentran los posibles referentes. Esta desambiguación sigue el principio del referente más probable ya que en los nodos menos profundos de la ontología se encuentran lugares más importantes que en los nodos más profundos (por ejemplo, las provincias están menos profundas que las ciudades). Además del proyecto SPIRIT, los tres trabajos más relevantes en cuanto a localización de nombres de lugar y su desambiguación son Web-a-where [AHSS04], STEWARD [LSSS07] y el producto comercial MetaCarta [RBB03]. Web-a-where emplea *contenedores espaciales* para identificar lugares en documentos; es decir, emplea el principio de referentes relacionados en base a relaciones topológicas definidas en un *gazetteer*. MetaCarta emplea técnicas de procesamiento del lenguaje natural y STEWARD emplea una aproximación híbrida ya que realiza la desambiguación teniendo en cuenta para cada localización la población

de las distintas opciones y la distancia entre los posibles referentes asociados con cada nombre de lugar localizado en el texto.

Traducción de los nombres de lugar

Además de la desambiguación de los nombres de lugar que, como explicábamos en el apartado anterior, se encuentra a medio camino entre las dos etapas, en esta segunda etapa se abordan las tareas que permiten traducir los nombres de lugar desambiguados a referentes geográficos que se utilizan para indexar espacialmente los documentos.

La primera característica que diferencia las propuestas existentes en la bibliografía es el tipo de objeto geográfico que se emplea para representar los referentes de los documentos. Algunos tipos de objetos geográficos usados en la bibliografía para este propósito son los puntos del espacio correspondientes a cada lugar mencionado en el texto, el *bounding box* que cubre el espacio que ocupan todos esos puntos o el centroide de dicho *bounding box*. La segunda característica es si el referente de cada documento consiste en un único objeto geográfico o si pueden ser varios. Los documentos se representan mejor cuando el referente puede estar formado por varios objetos geográficos ya que en un mismo documento se pueden mencionar muchos lugares diferentes que pueden estar muy distantes en el espacio. Por ejemplo, un documento acerca de la economía mundial puede citar las principales bolsas que se encuentran en todo el mundo. Sin embargo, por simplicidad muchas de las aproximaciones propuestas en la bibliografía asumen que el referente está formado por un único objeto geográfico.

Uno de los proyectos pioneros, anterior incluso al proyecto SPIRIT, en cuanto a la geo-referenciación de documentos contenidos en bibliotecas digitales es GIPSY (*Georeferenced Information Processing SYstem*) [Lar95]. En este proyecto, las referencias geográficas encontradas en los textos se convierten en representaciones geométricas (por ejemplo, en polígonos) y a todas esas representaciones se les asigna un valor de ponderación en función de propiedades derivadas del contenido de los documentos (por ejemplo, de la frecuencia de aparición del término). Luego se agregan los polígonos construyendo representaciones topográficas en tres dimensiones teniendo en cuenta los factores de ponderación y se establece un umbral para determinar la elevación mínima que determina que el área es relevante.

En cuanto al proyecto SPIRIT, por cada referencia geográfica mencionada en el texto de un documento se almacena su *bounding box*; es decir, el referente de cada documento está formado por varios *bounding boxes*. En este proyecto no se abordaba el tema de la desambiguación de esas referencias geográficas. Este esquema también se emplea en otros trabajos como [ZXW+05].

En [SC01], los autores proponen una alternativa para asignar a cada documento un referente geográfico formado por varios puntos. Para todos los posibles lugares que se pueden asociar con un documento se almacenan sus coordenadas ponderadas empleando la frecuencia de aparición del nombre de lugar en el documento. Se calcula

el centroide del polígono que forman todos esos puntos y la desviación típica del mismo, teniendo en cuenta los pesos que ponderan cada punto. Todos los puntos que distan más de dos veces la desviación típica se descartan y los restantes forman el referente geográfico del documento.

Finalmente, en el proyecto Web-a-where [AHSS04] ya hemos mencionado que para la desambiguación se emplean contenedores espaciales basados en referentes relacionados en base a relaciones topológicas definidas en un *gazetteer*. Estas relaciones son sobre todo de tipo *parte de* (por ejemplo, Galicia es *parte de* España). Con los términos desambiguados se realiza una agregación de lugares relacionados creando una taxonomía. Los niveles de esa taxonomía se ordenan según su relevancia y los que están por encima de un determinado umbral se emplean como referente del documento.

2.3.4. Estructuras de indexación

Para poder satisfacer las necesidades de información de los usuarios de sistemas GIR, las estructuras de indexación desarrolladas para este tipo de sistemas deben permitir la resolución de consultas que tengan tanto una componente textual como una componente espacial. En las secciones 2.1.3 y 2.1.4 analizábamos algunas de las estructuras de indexación más relevantes que se han propuesto para la recuperación de información textual. Por otra parte, en la sección 2.2.4 analizábamos algunas de las estructuras de indexación espacial clásicas. En esta sección vamos a estudiar qué propuestas se han realizado para combinar estructuras de ambos tipos, permitiendo así resolver las consultas de interés en sistemas GIR.

En primer lugar, las estructuras de indexación propuestas en el proyecto SPIRIT [VJJS05] se basan en la combinación de una estructura de *grid* [NHS81] con un índice invertido (el índice textual clásico). La estructura *grid* es uno de los índices espaciales más sencillos. Esta estructura divide el espacio en celdas y cada punto se asocia a la celda en la que está contenido. Los autores de este trabajo realizaron pruebas combinando el *grid* y el índice invertido en una única estructura y también manteniéndolos por separado. Su conclusión más importante es que manteniendo ambos índices separados se consigue un menor coste de almacenamiento aunque, por contra, puede implicar mayores tiempos de respuesta. Además, tener separados los índices tiene ventajas en cuanto a la modularidad, facilidad de implementación y facilidad de mantenimiento. Sus resultados también muestran que los métodos propuestos son capaces de competir en términos de velocidad y coste de almacenamiento con estructuras de indexación textual clásicas. Aunque la estructura propuesta es muy sencilla, y ya se han propuesto otras que la superan tanto en velocidad como en coste de almacenamiento, este trabajo es muy relevante ya que ha establecido una de las características distintivas de todas las propuestas posteriores. Dicha característica establece la distinción entre estructuras híbridas, que combinan los

índices textual y espacial en una única estructura, y estructuras de doble índice, que los mantienen por separado.

Trabajos más recientes, como [MSA05, CSM06], describen las dos estrategias base de la indexación en sistemas GIR teniendo en cuenta las propuestas del proyecto SPIRIT. Estas dos propuestas se nombran como *Text-First* y *Geo-First*. A nivel general, ambos algoritmos asumen la existencia de un índice espacial y de un índice textual, y emplean la misma estrategia para resolver las consultas: primero se emplea un índice para filtrar los documentos (el índice textual en el caso de *Text-First* y el índice espacial en el caso de *Geo-First*); el conjunto de documentos resultante se ordena por sus identificadores y posteriormente se filtra usando el otro índice (el índice espacial en el caso de *Text-First* y el índice textual en el caso de *Geo-First*). Estos nombres se pueden emplear también para estructuras híbridas. De tal modo que si la estructura emplea primero el índice textual es de tipo *Text-First* y si la estructura emplea primero el índice espacial es de tipo *Geo-First*. En la figura 2.8 mostramos las tres propuestas básicas de la indexación en recuperación de información geográfica. Las tres estructuras emplean un índice textual (sombreado en la figura) y un índice espacial (sin sombrear). La estructura de la izquierda es de doble índice, ya que mantiene por separado un índice textual y un índice espacial, mientras que las otras dos son estructuras híbridas. La estructura del centro pertenece a la categoría de *Geo-First*, ya que se accede primero al índice espacial, y la estructura de la derecha a la de *Text-First*, ya que se accede primero al índice textual.

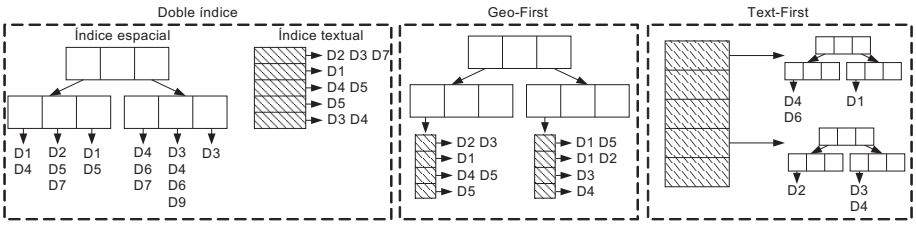


Figura 2.8: Estructuras de indexación básicas en sistemas GIR.

En [ZXW+05] los autores proponen emplear un índice invertido y un R-tree (en lugar de la estructura *grid*), y realizan pruebas combinándolos de las tres formas que acabamos de describir. En sus experimentos concluyen que mantener por separado los índices es menos eficiente (esta misma conclusión había sido obtenida en [VJJS05]) y que sus estructuras son más eficientes que las que emplean la estructura de *grid*.

En [CSM06], los autores comparan tres estructuras que combinan un índice invertido con la estructura *grid*, con el R-tree y con curvas de llenado del espacio [Mor66, BKK99]. Las curvas de lleando del espacio se basan en el almacenamiento de los objetos espaciales en un orden determinado por la forma de la curva de llenado.

La conclusión de los autores es que la estructura que emplea estas curvas de llenado del espacio mejora el rendimiento de las otras aproximaciones.

Finalmente, en el proyecto STEWARD [LSSS07], los autores proponen emplear una estructura que mantenga por separado un índice invertido y un *Quad-tree* [NS86]. El Quad-tree es una estructura similar al *grid* (ambas son dirigidas por el espacio y no por los objetos a indexar), que va dividiendo el espacio en cuadrantes hasta que los objetos se pueden almacenar en una página de disco. Además, en este trabajo se propone el empleo de un optimizador de consultas que decida si emplear primero el índice textual o el índice espacial en función de la previsión de cuál va a devolver menos resultados. Para poder emplear este optimizador de consultas el sistema debe almacenar estadísticas que permitan realizar la estimación del número de documentos resultante de una búsqueda de términos clave particular o de una ventana de consulta espacial determinada.

2.3.5. Ranking de relevancia

Como mencionábamos en la sección 2.3.1 y ampliábamos en la sección 2.3.2, las consultas de interés en sistemas GIR son similares a las consultas clásicas de sistemas de recuperación de información en el sentido de que los documentos pueden pertenecer al resultado con una determinada relevancia. Es decir, a diferencia de lo que sucede en el área de los GIS o de las bases de datos, las consultas no devuelven un conjunto de resultados donde se encuentran todos los que satisfacen la consulta sino que devuelven una lista ordenada en función de la relevancia o de la importancia que puede tener cada documento para el usuario. Esta lista ordenada es lo que se denomina *ranking de relevancia*.

En el estado del arte del campo de la recuperación de información (ver sección 2.1) describimos cómo tienen que definir el concepto de relevancia los distintos modelos de IR y cómo pueden elaborar los *rankings* de relevancia teniendo en cuenta esa definición. Incluso describimos la fórmula de *scoring* o de relevancia de una librería muy utilizada en el área como es Lucene. Sin embargo, como ya hemos mencionado, en el campo de los GIS no existe este concepto de relevancia. Por tanto, el primer objetivo que se tuvo que abordar en el área de la recuperación de información geográfica para poder elaborar *rankings* de relevancia de los resultados fue la definición del concepto de relevancia espacial. El segundo objetivo, una vez definida la relevancia espacial, consistió en definir cómo se combina esa relevancia espacial con la relevancia textual. En este apartado, estudiamos los trabajos más relevantes sobre estos dos temas.

En cuanto al primer objetivo, en [GR04] los autores definen medidas cuantitativas para el concepto de relevancia espacial empleando *bounding boxes* y teniendo en cuenta factores como la distancia, el grado de solapamiento o el tamaño relativo del área del *bounding box*. En [JAT01] se estudia cómo combinar la distancia en una ontología con la distancia Euclídea entre los centroides para crear una medida de similitud

geográfica. Además, dentro del proyecto Tumba, en [MSA05] los autores definen algunos principios interesantes para calcular la relevancia espacial cuando el sistema cuenta con una ontología espacial. Por ejemplo, emplear la distancia topológica basada en relaciones como la adyacencia, conectividad e inclusión jerárquica, tener en cuenta el número de parientes no comunes entre dos lugares, el número mínimo de relaciones directas que los separan, etc. Dentro del mismo proyecto, en [AS06] se proponen unas fórmulas que permiten calcular la relevancia geográfica en base a esos principios. Unas fórmulas más sencillas, que son aplicables en índices espaciales aunque no se disponga de ontología, se presentan en [ZXW+05]. Estas fórmulas calculan la relevancia espacial en consultas que pueden definir diferentes tipos de relaciones entre los referentes geográficos de los documentos indexados y las ventanas de consulta (por ejemplo, contenido, solape o vecindad).

En cuanto al segundo objetivo, la suma ponderada es posiblemente el método más empleado, y uno de los más sencillos, para combinar la relevancia textual y la relevancia espacial [MSA05, AS06, ZXW+05]. Aunque los autores de [MSA05] revisan otros métodos para combinar las relevancias (por ejemplo, de manera lineal, empleando el producto o empleando el máximo) concluyen que la suma ponderada es el método más eficaz. Además, en [YC07] los autores toman como base para combinar las relevancias textual y espacial la suma ponderada, y proponen un método para calcular los pesos de ambas relevancias teniendo en cuenta la *especificidad* de cada ámbito de la consulta. Si el ámbito textual es más específico que el ámbito espacial (que por tanto es más general), entonces el peso que pondera la relevancia textual será mayor que el que pondera la relevancia espacial y viceversa.

2.3.6. Resumen

La recuperación de información geográfica es un área de investigación muy reciente que ha aparecido en la intersección de dos áreas más consolidadas como son la recuperación de información y los sistemas de información geográfica. Su objetivo principal es que los usuarios puedan realizar consultas sobre colecciones de documentos teniendo en cuenta tanto su ámbito textual como su ámbito espacial. En esta sección hemos estudiado el estado del arte de este campo, en el cual se enmarca el presente trabajo de tesis.

En primer lugar, en la sección 2.3.1 hemos revisado los conceptos básicos de la recuperación de información geográfica definiendo qué temas abarca y describiendo el proceso típico de sistemas GIR para ver qué tareas componen dicho proceso. Empezando a analizar este proceso desde su objetivo, el primer tema de interés que nos encontramos son las necesidades de información de los usuarios de sistemas GIR y las consultas de interés para los mismos. En la sección 2.3.2 estudiamos estas consultas y su relación con las consultas típicas en sistemas de IR y GIS. Para permitir resolver estas consultas todo el proceso de sistemas GIR debe estar preparado para tener en

cuenta tanto el ámbito textual como el ámbito espacial de los documentos. El proceso lo podemos descomponer en el flujo de trabajo para la construcción de la estructura de indexación que da soporte a las consultas de los usuarios y en la resolución de esas consultas empleando dicha estructura. Para construir una estructura de indexación para este tipo de sistemas, en primer lugar, hay que analizar los documentos para asignarles un referente espacial (es decir, el área o áreas del espacio en las cuales ese documento es relevante). Esta tarea, conocida como geo-referenciación de documentos, la describimos en la sección 2.3.3. Una vez que se obtiene el referente geográfico de cada documento, se pueden indexar esos documentos en estructuras que tengan en cuenta dichos referentes geográficos además del texto de los documentos. En la sección 2.3.4 estudiamos las propuestas que se ha realizado para estructuras de indexación en sistemas GIR. Finalmente, en la otra parte del proceso, encargada de la resolución de consultas, existe una línea de investigación importante dedicada a la elaboración de *rankings* de relevancia de los resultados proporcionados por las estructuras de indexación. Estas propuestas se ocupan tanto de definir la relevancia espacial como de estudiar cómo se puede combinar esa relevancia espacial con la relevancia textual propia de sistemas de IR. En la sección 2.3.5 analizamos cómo abordan ambos asuntos las propuestas existentes en la bibliografía.

2.4. Conclusiones y discusión

En este capítulo hemos revisado el estado del arte de la recuperación de información geográfica. La aproximación seguida introduce al lector en el campo desde las dos áreas de las ha surgido: la recuperación de información y los sistemas de información geográfica. A pesar de que este campo de investigación es muy reciente, en los últimos años se han propuesto varias arquitecturas de sistemas, estructuras de indexación y otros componentes que permiten la recuperación de documentos teniendo en cuenta tanto el ámbito textual como el espacial de los mismos. Sin embargo, en esta tesis identificamos varios problemas en las alternativas propuestas y realizamos nuestras propias aportaciones para solucionar estos problemas.

En primer lugar, las estructuras de indexación diseñadas para tener en cuenta tanto el ámbito textual como el ámbito espacial de los documentos no contemplan ciertas características especiales debidas a la naturaleza geográfica de parte de la información que están indexando. En la sección 2.3.4 analizamos estas propuestas de estructuras para sistemas GIR. Las dos características más importantes ignoradas por las soluciones propuestas son la naturaleza jerárquica del espacio geográfico y las relaciones topológicas entre los objetos espaciales que indexan. En el capítulo 3 presentamos una estructura de indexación que combina un índice textual, un índice espacial y una ontología del espacio geográfico para tener en cuenta estas características.

En segundo lugar, las arquitecturas propuestas para sistemas de recuperación de información geográfica no son completas ya que se han diseñado orientadas al problema más familiar para los autores (por ejemplo, a la geo-referenciación y desambiguación de documentos, a la indexación de los mismos o a la presentación de resultados a los usuarios). La segunda aportación de nuestro trabajo de tesis consiste en el diseño de una arquitectura completa y genérica para sistemas GIR que cubre esta necesidad. Esta arquitectura, que presentamos en el capítulo 4, se basa en las arquitecturas propuestas para GIS tanto desde los organismos internacionales ISO y OGC (ver sección 2.2.2), como en la tesis doctoral de uno de los directores de este trabajo (ver sección 2.2.3).

Finalmente, identificamos ciertas características propias de la información geográfica que se maneja habitualmente en sistemas GIR y que no tienen en cuenta las estructuras de indexación espacial que estudiamos en la sección 2.2.4. Por ejemplo, las colecciones de documentos tanto en sistemas GIR como en sistemas de de IR son semi-estáticas, por lo que se puede optimizar la estructura de indexación espacial asumiendo que se conoce *a priori* la distribución de la información geográfica. Además, los referentes que se almacenan normalmente para los documentos son puntos que representan la posición de los lugares citados en el texto (ver sección 2.3.3). En el capítulo 5 presentamos una estructura de indexación espacial optimizada para tener en cuenta estas características.

Capítulo 3

Descripción de la estructura de indexación para recuperación de información geográfica

En nuestra opinión existen dos requisitos fundamentales que se deben cumplir en el diseño de una estructura de indexación en el ámbito de los sistemas de recuperación de información geográfica. En primer lugar, la estructura debe proporcionar una descripción apropiada de la naturaleza jerárquica del espacio geográfico para permitir la asociación de información con los distintos elementos singulares que forman la jerarquía. En segundo lugar, la estructura debe resolver de forma eficiente consultas textuales puras, consultas espaciales puras y consultas que especifiquen tanto información espacial como información textual.

En este capítulo describimos la estructura de indexación para recuperación de información geográfica que proponemos para satisfacer estos requisitos. En primer lugar, en la sección 3.1 formalizamos la ontología del espacio geográfico en la que se basa nuestra estructura de indexación, que describimos en la sección 3.2. A continuación, en la sección 3.3 definimos los tipos de consulta de interés en un sistema de recuperación de información geográfica y presentamos los algoritmos necesarios para resolver dichas consultas en nuestra estructura. La sección 3.4 analiza los experimentos realizados para comparar el rendimiento de nuestra estructura frente a otras alternativas. Finalmente, presentamos un resumen de las características más importantes de nuestra estructura en la sección 3.5.

3.1. Ontología espacial

La principal carencia de las estructuras de indexación documentadas en la bibliografía, y por tanto el primer requisito para nuestra estructura, es la descripción apropiada de la naturaleza jerárquica del espacio geográfico. Para cumplir con este requisito, basamos el diseño de nuestra estructura de indexación en una ontología [Gru93b, Gru93a] del espacio geográfico que describe los objetos existentes en nuestro dominio y las relaciones entre ellos.

Como veíamos en el estudio del estado del arte (ver sección 2.1.5), una ontología nos proporciona un vocabulario de clases y relaciones para describir un ámbito determinado, en este caso el espacio geográfico. Nuestro objetivo no es definir una ontología que describa todo el espacio geográfico (ya que sería una tarea muy costosa en tiempo y esfuerzo) sino sólo para el ámbito determinado en el que se vaya a utilizar el sistema. Es decir, la ontología en sí misma no es un fin sino un medio para lograr el objetivo que es la estructura de indexación basada en ella. Para que esta aportación sea realmente útil, es importante que toda la arquitectura, incluyendo la ontología y la estructura basada en ella, esté diseñada para ser fácilmente adaptable a cualquier dominio de aplicación. Por ejemplo, aunque en el prototipo definamos una ontología y una estructura de indexación para un dominio general donde se considere que los niveles que se deben tener en cuenta son los continentes, los países, las regiones y el resto se consideren en una categoría lugares poblados; todo el sistema debe estar preparado para ser adaptado a un dominio más específico como puede ser el ámbito de una provincia, con sus ayuntamientos, lugares, barrios, etc.

A la hora de definir una ontología existen varios lenguajes ontológicos que proporcionan distintos niveles de formalismo y facilidades de razonamiento. El lenguaje OWL [Wor08], estandarizado por el W3C, permite definir ontologías con varios niveles de detalle. Dicho lenguaje se puede categorizar en tres especies o sub-lenguajes: OWL-Lite, OWL-DL y OWL-Full. Nuestra ontología espacial, que está accesible en web en la URL http://lbd.udc.es/ontologies/spatialrelations, la hemos definido empleando la especie OWL-DL. Las clases OWL se pueden interpretar como conjuntos que contienen individuos (también conocidos como instancias). A su vez, estos individuos se pueden considerar como instancias de clases. Nuestra ontología describe ocho clases de interés: *SpatialThing*, *GeographicalThing*, *GeographicalRegion*, *GeopoliticalEntity*, *PopulatedPlace*, *Region*, *Country* y *Continent*. Además, existen relaciones jerárquicas entre *SpatialThing*, *GeographicalThing*, *GeographicalRegion* y *GeopoliticalEntity* ya que:

- *GeopoliticalEntity* es subclase de *GeographicalRegion*
- *GeographicalRegion* es subclase de *GeographicalThing* y
- *GeographicalThing* es subclase de *SpatialThing*.

Es decir, estas cuatro clases están organizadas en una jerarquía de especialización superclase – subclase, también conocida como *taxonomía*. Las subclases especializan (están *subsumidas por*) sus superclases. *GeopoliticalEntity* tiene cuatro subclases: *PopulatedPlace*, *Country*, *Continent* y *Region*, y todos los individuos son miembros de esas subclases. Estas cuatro subclases tienen necesariamente una condición de aserción relativa a sus relaciones con cada una de las otras. Están conectadas por la propiedad *spatiallyContainedBy* que describe la existencia de una relacion de contenido espacial entre ellas. Por ejemplo, todos los individuos de la clase *PoulatedPlace* están espacialmente contenidos (*spatiallyContainedBy*) en individuos de la clase *Region* (esto se describe en OWL como *PopulatedPlace spatiallyContainedBy only (AllValuesFrom) Region*). La figura 3.1 muestra un ejemplo de estas relaciones. En la figura las clases de la ontología se representan como círculos, los individuos como rectángulos y las relaciones como líneas etiquetadas.

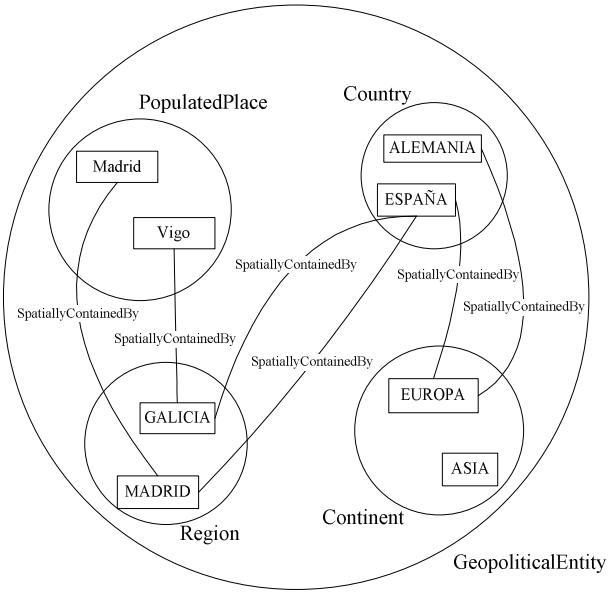


Figura 3.1: Instancias de la ontología.

3.2. Descripción de la estructura

La formalización de la ontología nos permite la definición de una estructura de indexación basada en ella. Por tanto, la estructura que proponemos es un árbol con cuatro niveles, cada uno de ellos correspondiente a una de las subclases de

GeopoliticalEntity. El nivel superior del árbol contiene un nodo por cada una de las instancias de la clase *Continent*. A su vez, cada nodo en ese nivel referencia las instancias de la clase *Country* que están conectadas por medio de la relación de contenido espacial (*spatiallyContainedBy*). Los niveles correspondientes a *Region* y a *PopulatedPlace* los construimos empleando la misma estrategia. Es decir, la estructura del árbol sigue la taxonomía de la ontología. La figura 3.2 muestra la estructura de indexación espacial construida con las instancias mostradas en la figura 3.1. En el árbol de la figura mostramos algunos nodos representativos de cada nivel. Además, los triángulos sombreados que cuelgan de ellos representan sub-árboles de los que no detallamos su contenido.

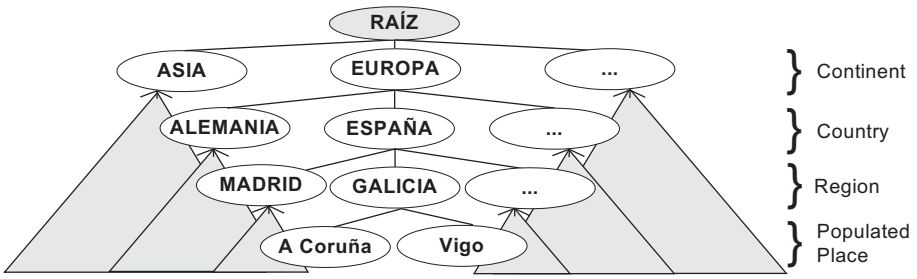


Figura 3.2: Árbol basado en la ontología.

Cada nodo contiene la siguiente información, además de la lista de nodos hijo que están contenidos espacialmente por él (*spatiallyContainedBy*): (i) la palabra clave (un nombre de lugar), (ii) el MBR o *bounding box* de la geometría que representa ese lugar y (iii) una lista con los identificadores de los documentos que incluyen referencias geográficas a ese lugar. Además, cada nodo contiene un R-tree que mejora el acceso desde un nodo a los nodos hijo que satisfacen una consulta. Esto mejora considerablemente la eficiencia de nuestra estructura.

Al ser una estructura en forma de árbol y almacenar los MBRs en cada nodo, podemos usar la estructura para resolver consultas espaciales empleando el mismo algoritmo que se utiliza con las estructuras de indexación espacial clásicas. Dicho algoritmo consiste en descender a través de la estructura descartando aquellas ramas del árbol que no intersecan con la ventana de consulta. Los nodos del árbol que continúan tras el descenso constituyen el resultado de la consulta.

La ventaja principal de esta estructura de indexación espacial sobre otras alternativas es que los nodos intermedios de la estructura tienen significado en el espacio geográfico y pueden tener información adicional asociada. Por ejemplo, podemos asociar una lista de documentos que referencien a un país determinado y

usar esa lista de documentos para resolver consultas que combinen una parte textual y una parte espacial. Además, dado que existe una relación superclase – subclase entre niveles, los niveles inferiores pueden heredar las propiedades asociadas con sus niveles superiores. En particular, los documentos asociados con un nodo de la estructura también se refieren a todos los nodos en el subárbol que parte de él. Como veremos en la sección 3.3.3, estas relaciones permiten que nuestra estructura realice expansión de consultas (es decir, le permite recuperar documentos donde no aparezcan los términos buscados pero sí alguno relacionado) de forma directa. Otra ventaja es que la estructura es general en el sentido de que la ontología del espacio geográfico se puede adaptar para cada aplicación en particular. Por ejemplo, si una aplicación en concreto usa un área restringida del espacio geográfico donde las clases *Continent* y *Country* no son necesarias pero, en cambio, son necesarias las clases *Province*, *Municipality*, *City* y *Suburb*, podemos definir una ontología del espacio diferente y basar la estructura de indexación en ella ya que la relación *spatiallyContainedBy* continúa siendo válida entre las clases. Finalmente, podemos definir relaciones espaciales adicionales en la ontología, como por ejemplo la de adyacencia (*spatiallyAdjacent*), y mantener esas relaciones en la estructura de indexación para mejorar las capacidades de consulta del sistema.

Por tanto, la estructura espacial que acabamos de describir satisface el primer requisito que indicábamos en la introducción de este capítulo; es decir, proporciona una descripción apropiada de la naturaleza jerárquica del espacio geográfico permitiendo la asociación de información con los distintos elementos singulares que forman dicha jerarquía. Para satisfacer el segundo requisito (es decir, para resolver de forma eficiente consultas textuales puras, consultas espaciales puras y consultas que combinen información espacial con información textual) empleamos dos componentes auxiliares en la estructura: un índice textual y una tabla *hash* con nombres de lugar. El índice textual es un índice invertido clásico que asocia a cada palabra una lista de documentos que la contienen. Finalmente, la tabla *hash* con nombres de lugar almacena para cada topónimo su posición en la estructura de indexación y se usa para optimizar la resolución de un tipo particular de consultas muy habitual (ver *consultas textuales con nombres de lugar* en la sección 3.3.1). La figura 3.3 muestra un ejemplo de la estructura de indexación completa con el índice invertido, la tabla *hash* con nombres de lugar y la estructura de indexación espacial.

3.3. Resolución de consultas

3.3.1. Tipos de consulta soportados

Una de las características más importantes de toda estructura de indexación es el tipo de consultas que permite resolver. En el capítulo 2 analizábamos los tipos de consultas relevantes en sistemas de recuperación de información y sistemas

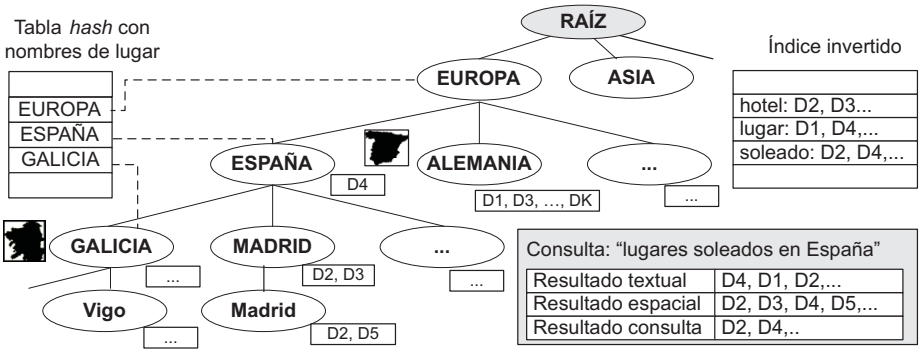


Figura 3.3: Ejemplo de la estructura de indexación.

de información geográfica clásicos y, también, en sistemas de recuperación de información geográfica. En nuestra opinión los tipos de consulta más relevantes en un sistema de recuperación de información geográfica son los siguientes:

- *Consultas textuales puras*. Estas son consultas del tipo "*recuperar todos los documentos donde aparezcan las palabras* hotel *y* mar". Son las consultas típicas del campo de la recuperación de información que describimos en la sección 2.1.3.

- *Consultas espaciales puras*. Un ejemplo de este tipo de consultas es "*recuperar todos los documentos que se refieran a la siguiente área geográfica*". El área geográfica en la consulta puede ser un punto, una ventana de consulta o un objeto complejo como un polígono. Estas consultas, procedentes del campo de los sistemas de información geográfica (ver sección 2.2.4), adquieren un nuevo enfoque en los sistemas GIR al definir el concepto de relevancia espacial. En estos sistemas, los objetos en el resultado pertenecen al mismo con un valor de relevancia (es decir, no se hace simplemente la distinción en objetos que pertenecen al resultado y objetos que no lo hacen).

- *Consultas textuales sobre un área geográfica*. En este caso se proporciona un área geográfica de interés junto con el conjunto de palabras. Un ejemplo de este tipo de consultas es "*recuperar todos los documentos con la palabra* hotel *que se refieren a la siguiente área geográfica*". Al igual que en las consultas espaciales puras el área geográfica de la consulta puede ser un punto, una ventana de consulta o un objeto complejo.

- *Consultas textuales con nombres de lugar*. En este tipo de consultas algunos de los términos son nombres de lugar. Por ejemplo, "*recuperar todos los*

documentos con la palabra hotel *referidos a España*". Estas consultas, que se realizan frecuentemente en sistemas de recuperación de información clásicos, alcanzan una nueva dimensión es sistemas GIR al poder aprovechar las características de la parte espacial de la consulta.

3.3.2. Algoritmos de resolución de consultas

Las consultas textuales puras se pueden resolver de forma trivial porque un índice textual forma parte de la estructura de indexación. Como veíamos en la sección 2.1.3, la estructura de indexación textual clásica es el índice invertido. Esta estructura almacena para cada palabra en el texto (organizado como un *vocabulario*) una lista de punteros a los documentos donde aparecen. Todas estas listas se denominan *ocurrencias*. Cuando el usuario realiza una consulta el algoritmo de resolución obtiene el resultado mediante la intersección de todas las listas de *ocurrencias* de las palabras en la consulta.

De manera similar, las consultas espaciales puras se pueden resolver porque la estructura de indexación se construye como un índice espacial y, por tanto, se puede emplear el mismo algoritmo empleado con índices espaciales. Se desciende en la estructura teniendo en cuenta sólo aquellos nodos cuyos MBRs intersecan con el área geográfica de la consulta. Esta operación devuelve un conjunto de documentos candidatos que tiene que ser refinado empleando la referencia geográfica real para decidir si cada documento forma parte del resultado o no.

La principal ventaja de la estructura de indexación que proponemos sobre una estructura textual clásica o espacial clásica es que se puede emplear para resolver el tercer y el cuarto tipo de consultas que no se pueden solucionar de manera sencilla con ellas. En ambos tipos de consulta, que podríamos agrupar usando el término *consultas híbridas*, el índice textual se emplea para recuperar la lista de documentos que contienen las palabras buscadas y la estructura de indexación espacial se emplea para obtener la lista de documentos que hacen referencia al área geográfica. Por tanto, la intersección de ambas listas es el resultado de la consulta.

En el caso de consultas del tercer tipo (por ejemplo, "*lugares soleados en España*", ver figura 3.3), nuestro sistema usa la tabla *hash* con nombres de lugar para acceder de forma eficiente al nodo de la estructura que representa al lugar *España*. Por tanto, la estructura propuesta nos permite en este caso ahorrar mucho tiempo en el descenso del árbol. A partir de ese nodo, el algoritmo desciende en la estructura visitando todos los nodos y añadiendo al resultado los documentos asociados con ellos.

Por último, en el caso de consultas del cuarto tipo, la parte espacial de las mismas se resuelve empleando el mismo algoritmo que en el segundo tipo y, por tanto, que en los índices espaciales clásicos. Como nuestra estructura de indexación espacial puede no ser un árbol balanceado, al depender del dominio real de aplicación, podría presentar problemas de rendimiento al resolver las consultas. Sin embargo, en la

sección 3.4 presentamos los resultados de los experimentos que demuestran que, en general, nuestra propuesta presenta una eficiencia satisfactoria.

3.3.3. Expansión de los términos de consulta

Una ventaja sobre los índices textuales y espaciales clásicos es que nuestra estructura de indexación puede realizar fácilmente expansión de los términos de consulta (ver sección 2.1.5) sobre referencias geográficas porque está construida sobre una ontología del espacio geográfico. Como ya hemos comentado, la expansión de los términos de consulta no es algo novedoso ni en el campo de la recuperación de información en general ni en el de la recuperación de información geográfica en particular. Tampoco lo es el utilizar ontologías para este propósito. La principal aportación que realizamos es que nuestra estructura lo realiza de manera implícita, sin añadir más complejidad a los algoritmos de búsqueda.

Consideremos la siguiente consulta "*recuperar todos los documentos que se refieran a* España". El servicio de evaluación de consultas descubrirá que *España* es una referencia geográfica. El índice de nombres de lugar se empleará para localizar rápidamente el nodo interno que representa el objeto geográfico correspondiente a *España*. Entonces, todos los documentos asociados con este nodo forman parte del resultado de la consulta. Sin embargo, todos los hijos de este nodo son objetos geográficos que están contenidos en *España* (por ejemplo, la ciudad de *Madrid*). De este modo, todos los documentos referenciados por el subárbol forman también parte del resultado de la consulta. La consecuencia es que la estructura de indexación se ha empleado para expandir la consulta porque el resultado contiene no sólo aquellos documentos que incluyen el término *España*, sino también aquellos documentos que incluyen el nombre de un objeto geográfico contenido en *España* (por ejemplo, todas las ciudades y regiones de *España*).

3.4. Experimentos

A lo largo de este capítulo hemos demostrado que nuestra estructura de indexación tiene una ventaja cualitativa sobre sistemas que combinan un índice textual con una estructura de indexación espacial clásica. Dicha ventaja reside en que nuestra estructura permite realizar expansión de los términos de consulta (*query expansion*) de forma directa ya que está basada en una ontología del espacio geográfico. Por tanto, nuestra estructura amplía las capacidades de consulta del sistema con una funcionalidad que no se puede implementar de forma sencilla con una estructura de indexación espacial clásica. Sin embargo, a diferencia de éstas, nuestra estructura de indexación no es balanceada y, por tanto, la eficiencia en la resolución de consultas puede ser peor. En esta sección vamos a describir los experimentos que hemos realizado para comparar

Tabla 3.1: Descripción de la colección de documentos del TREC.

	FT-91	**FT-94**
Documentos	5.368	71.489
Documentos con candidatos	4.652	60.823
Documentos geo-referenciados	4.182	54.899
Entradas de texto	64.711	251.057
Entradas espaciales	21.282	64.843

nuestra estructura con otras alternativas basadas en índices espaciales clásicos. El objetivo final de estos experimentos es demostrar que el desbalanceo de nuestra estructura no penaliza la eficiencia y, por tanto, no supone un problema grave.

3.4.1. Descripción de las colecciones de prueba

En nuestros experimentos empleamos dos colecciones de documentos procedentes del *Financial Times*. La primera de ellas contiene 5.368 noticias del año 1.991 (TREC FT-91) y la segunda 71.489 noticias del año 1.994 (TREC FT-94). La Tabla 3.1 resume las características más importantes de estas colecciones. En la primera fila se muestra la comparación en cuanto a número de elementos donde se puede observar que la colección FT-94 tiene trece veces más documentos que la colección FT-91. Utilizar dos colecciones con tanta diferencia en el número de documentos nos permite comprobar la escalabilidad de las diferentes aproximaciones con respecto al número de documentos.

Para entender las restantes filas de la tabla debemos recordar las tareas que intervienen en el proceso de indexación de documentos en sistemas GIR (ver sección 2.1). A modo de resumen, para indexar espacialmente un documento, en primer lugar, se deben obtener las posibles referencias geográficas mencionadas en él. La segunda fila de la tabla muestra el número de documentos en cada colección en los cuales se identificó al menos un término candidato a ser referencia geográfica. Sin embargo, muchos de estos candidatos no son verdaderos nombres de lugar (debido a que las técnicas empleadas para localizarlos tienen una precisión alta pero que no alcanza el 100 %) y, por tanto, no seremos capaces de geo-referenciarlos. En la tercera fila de la tabla mostramos el número de documentos con una o más referencias geográficas correctamente asignadas. Finalmente, las *entradas de texto* y las *entradas espaciales* se corresponden con el número de entradas y de nodos en el índice textual y en el índice espacial respectivamente.

3.4.2. Descripción de los experimentos

Como ya hemos mencionado al principio de esta sección, el objetivo de estos experimentos es comparar nuestra estructura de indexación, construida tal y como se describe en este capítulo, con otras estructuras que emplean una estructura de indexación espacial clásica.

La primera estructura contra la que nos comparamos emplea un índice textual y un R-tree, en el que cada par {identificador de documento, lugar} se almacena en un elemento (por ejemplo, {d2, Madrid (40'26, 3'41)} significa que el documento con identificador *d2* menciona la ciudad de *Madrid*). Si otro documento, por ejemplo d5, también menciona la ciudad de Madrid será necesario un nuevo elemento {d5, Madrid (40'26, 3'41)}. La segunda estructura, que denominamos *Improved R-tree*, también emplea un índice textual y un R-tree pero, en este caso, los lugares se almacenan una única vez. Esta mejora sobre la estructura anterior reduce considerablemente el tamaño de la estructura y mejora su rendimiento.

Para la realización de los experimentos, además de las colecciones de documentos y de las estructuras de indexación, necesitamos un conjunto de ventanas de consulta. Para construirlas, implementamos un algoritmo de generación de ventanas de consulta espaciales de tamaño parametrizable. Dicho algoritmo está basado en los experimentos diseñados para evaluar el rendimiento del R*-tree tal y como presentaron sus autores en [BKSS90]. Las ventanas que generamos están uniformemente distribuidas en el espacio. Además, su proporción se determina de tal forma que el ratio entre la *extensión en x* y la *extensión en y* varía de manera uniforme entre 0,25 y 2,25. La figura 3.4 muestra varias ventanas de consulta generadas empleando este algoritmo.

3.4.3. Resultados de los experimentos

En nuestros experimentos comparamos el rendimiento de las tres estructuras de indexación descritas para resolver consultas agrupadas según su área. Formamos cuatro grupos que representan respectivamente el 0,001 %, 0,01 %, 0,1 % y 1 % del espacio de entrada. Para cada área, generamos 100.000 ventanas de consulta aleatorias y promediamos el tiempo de ejecución de cada consulta. La Tabla 3.2 muestra los resultados de este experimento. La primera fila de la tabla muestra los resultados obtenidos por nuestra estructura (en milisegundos), la segunda muestra los resultados obtenidos por la estructura que emplea un R-tree y la tercera muestra los resultados obtenidos por la estructura que emplea un R-tree mejorado.

A la vista de los resultados, la estructura de indexación que emplea un R-tree, modificada para tener en cuenta las características específicas del problema (es decir, la estructura denominada *Improved R-tree*), mejora el rendimiento de la ejecución de consultas con respecto a nuestra estructura. Como ya hemos mencionado, este resultado es esperado ya que nuestra estructura de indexación no es balanceada. Sin embargo,

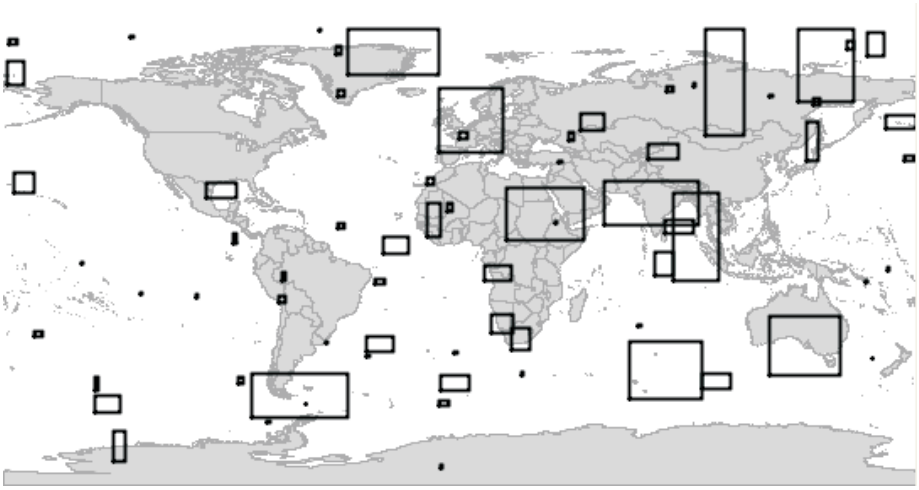


Figura 3.4: Ejemplo de ventanas de consulta generadas aleatoriamente.

los resultados muestran que las diferencias no son muy significativas y, por tanto, creemos que el rendimiento de nuestra estructura es aceptable. Además, tanto nuestra solución como la denominada *Improved R-tree* presentan resultados similares también en términos de escalabilidad. En base a estos resultados podemos concluir que nuestra estructura debe ser considerada como una alternativa al uso de estructuras que emplean índices espaciales puros ya que recordemos que parte con una ventaja cualitativa con respecto a ellas.

Otra conclusión importante a tener en cuenta se refiere al rendimiento y la escalabilidad de la solución que emplea un R-tree sin ningún tipo de modificación.

Tabla 3.2: Onto-index vs. R-tree.

	FT-91				**FT-94**			
Área (%)	0,001	0,01	0,1	1	0,001	0,01	0,1	1
Onto-index	0,013	0,017	0,052	0,360	0,016	0,035	0,228	2,938
R-tree	0,010	0,016	0,057	0,370	0,041	0,113	0,583	4,704
Improved	0,008	0,007	0,023	0,154	0,008	0,016	0,095	1,069

Tabla 3.3: Densidad de documentos baja vs. densidad de documentos alta.

	Densidad alta				**Densidad baja**			
Área (%)	0,001	0,01	0,1	1	0,001	0,01	0,1	1
Onto-index	0,03	0,11	1,05	9,84	0,02	0,03	0,09	0,4
R-tree	0,07	0,22	1,64	12,85	0,02	0,03	0,07	0,2

Esta solución presenta un rendimiento aceptable con la colección FT-91 pero su escalabilidad es considerablemente peor que la de las otras alternativas. La explicación reside en el número de nodos de esta estructura. La solución que usa un R-tree sin modificar necesita indexar un elemento por cada par {identificador de documento, lugar}. Esto hace que el número de nodos sea considerablemente más alto que en las otras alternativas donde cada lugar se indexa una única vez y tiene asociada una lista con todos los documentos que lo citan. Un resultado inesperado es que el rendimiento de la estructura con el R-tree sin modificar mejora el de nuestra estructura cuando la colección y las ventanas de consulta son pequeñas. Para descubrir la razón de este comportamiento realizamos experimentos adicionales considerando la distribución de los documentos como un factor que podría influir en el rendimiento. La Tabla 3.3 divide los resultados globales obtenidos con la colección FT-91 en dos zonas con diferente densidad de documentos (zonas con una densidad de documentos alta y zonas con una densidad de documentos baja). Cuando la densidad de documentos es baja, el número de nodos en ambas estructuras es similar. Sin embargo, cuando la densidad de documentos es alta la diferencia en el número de nodos es bastante más significativa. Por tanto, cuando la ventana de consulta es pequeña la probabilidad de que dicha ventana caiga en una zona con alta densidad de documentos es pequeña y esto hace que el rendimiento del R-tree sea mejor. Por el contrario, cuando la ventana de consulta es más grande dicha probabilidad se hace mayor y esto produce que el rendimiento del R-tree sea peor.

3.5. Resumen

En este capítulo hemos presentado una estructura de indexación para recuperación de información geográfica que combina un índice textual, un índice espacial y una ontología del espacio geográfico. Antes de describir dicha estructura en la sección 3.2, mostramos la ontología del espacio geográfico que nos permite formalizar el dominio en la sección 3.1. Describimos esta ontología empleando OWL-DL, un sub-lenguaje de OWL, y permitimos el acceso a ella vía web. Además, en la sección 3.3 hemos descrito cómo se pueden resolver los tipos de consultas clásicas, y hemos

presentado nuevos tipos de consulta y los algoritmos necesarios para resolver dichas consultas. Finalmente, en la sección 3.4 hemos analizado los experimentos realizados para demostrar que nuestra estructura no sólo tiene una ventaja cualitativa sobre otras estructuras que emplean un índice espacial clásico, sino que también presenta unos resultados de rendimiento aceptables comparada con ellas.

Capítulo 4

Descripción de la arquitectura propuesta

En este capítulo describimos la arquitectura genérica que proponemos para el desarrollo de sistemas de recuperación de información geográfica y los componentes por los que está formada. En primer lugar, en la sección 4.1 damos una visión general de la arquitectura y de los distintos subsistemas que la componen. Como veremos, la arquitectura se divide en tres capas que agrupan respectivamente los subsistemas que intervienen en el flujo de trabajo para la construcción de la estructura de indexación (sección 4.2), los servicios de procesado (sección 4.3) y las interfaces de usuario (sección 4.4). En la sección 4.5 presentamos el funcionamiento de la arquitectura mediante la descripción de la interacción de los distintos componentes en escenarios de uso típicos. Por último, en la sección 4.6 resumimos las conclusiones más importantes acerca de nuestra propuesta de arquitectura.

4.1. Arquitectura general

Una de las principales conclusiones que pudimos extraer del estudio del estado del arte (capítulo 2) es la identificación de las funcionalidades básicas necesarias para el desarrollo de un sistema de recuperación de información geográfica completo. El objetivo último de este tipo de sistemas es resolver consultas que combinan una componente textual y una componente espacial. Por tanto, una arquitectura completa para sistemas GIR debe estar centrada en torno a una estructura de indexación que permita resolver dichas consultas (como la que proponemos en el capítulo anterior). Además, debe proporcionar la infraestructura necesaria para la construcción y mantenimiento de la estructura a partir de una colección de documentos. En esta

infraestructura se enmarcan la interfaz de administración, los componentes para la identificación de referencias geográficas en documentos y para la geo-referenciación de las mismas, y los servicios de procesado empleados por cualquiera de los anteriores. Finalmente, la arquitectura también debe proporcionar toda la infraestructura necesaria para la explotación de la estructura de indexación mediante la resolución de consultas. En este grupo de componentes se encuentran la interfaz de consulta, y los servicios de procesado empleados para la representación cartográfica de las soluciones y para la gestión de la estructura.

La figura 4.1 ilustra nuestra propuesta para la arquitectura de un sistema de recuperación de información geográfica. La arquitectura se divide en tres capas: el flujo de trabajo para la construcción del índice, los servicios de procesado y las interfaces de usuario. La parte inferior de la figura muestra los componentes que intervienen en el flujo de trabajo para la construcción del índice. En la sección 4.2 describimos este flujo de trabajo. En la figura se pueden ver los cuatro componentes que forman esta capa: el módulo de abstracción de documentos (descrito en la sección 4.2.1), el proceso de indexación textual (sección 4.2.2), el proceso de indexación espacial (sección 4.2.3) y la propia estructura de indexación (sección 4.2.4). En el capítulo anterior describimos conceptualmente la estructura de indexación, presentamos sus características, ventajas e inconvenientes, mientras que en este capítulo describimos cómo se integra dicha estructura en nuestra arquitectura y qué implicaciones tiene en cuanto a su implementación.

En la parte central de la figura se encuentran los componentes que forman la capa de servicios de procesado. En la mitad izquierda se muestra el servicio de ontología del espacio geográfico. La principal utilidad de este servicio se encuentra en la parte espacial de la construcción de la estructura de indexación y, por tanto, lo describimos en la sección 4.2.3. En la mitad derecha se muestran dos servicios empleados en la resolución de consultas. El situado más a la derecha es el servicio de evaluación de consultas (descrito en la sección 4.3.1), el cual se encarga de recibir las consultas y de resolverlas empleando para ello la estructura de indexación. El otro servicio es un *Web Map Service* (WMS) [Ope02] (descrito en la sección 4.3.2) que empleamos para la creación de representaciones cartográficas de los resultados de las consultas. Sobre estos servicios se asienta el módulo de recuperación de información geográfica que es el encargado de coordinar las tareas realizadas por cada uno de los servicios anteriores en respuesta a las peticiones de los usuarios.

Finalmente, en la capa superior de la arquitectura se encuentran las dos interfaces de usuario que permiten la interacción con el sistema: la interfaz de usuario de administración, destinada a realizar el mantenimiento de la estructura de indexación y del sistema en general, y la interfaz de usuario de consulta, empleada por los usuarios para realizar consultas al sistema e interactuar con los resultados. Describimos ambas interfaces de usuario en la sección 4.4.

En esta propuesta podemos ver la influencia de las arquitecturas para sistemas

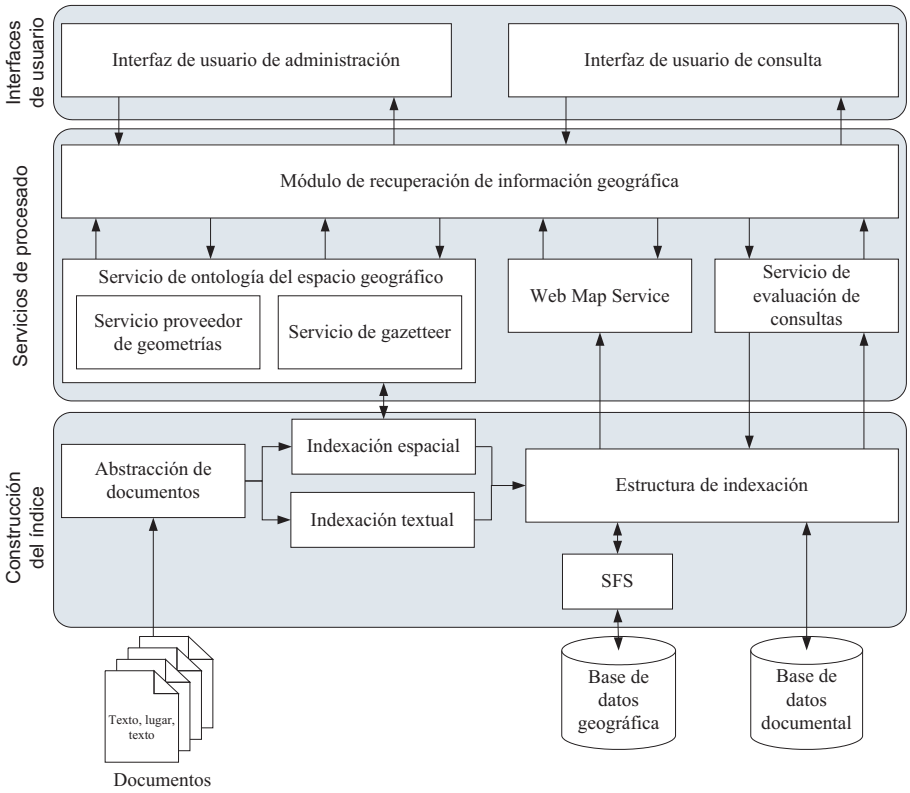


Figura 4.1: Arquitectura general para sistemas GIR.

de información geográfica que hemos presentado en el estudio del estado del arte (ver secciones 2.2.2 y 2.2.3). En primer lugar, la arquitectura está dividida en tres capas donde la inferior se encarga de la gestión de datos, la intermedia se encarga del procesado de la información y la superior se encarga de la presentación de información al usuario. Además, la capa de procesado define una arquitectura similar a la propuesta en la sección 2.2.3. Se basa en servicios independientes con funcionalidades bien definidas que gracias a un módulo de control (el módulo de recuperación de información geográfica) pueden trabajar en conjunto para solucionar tareas complejas. Por último, podemos ver que la arquitectura tiene en cuenta los estándares propuestos desde el área de los GIS. Por ejemplo, se emplea un WMS [Ope02] para la generación de mapas y el estándar SFS [Ope06] para el almacenamiento de información geográfica en base de datos.

4.2. Flujo de trabajo para la construcción del índice

La capa inferior de la arquitectura tiene como objetivo la creación de la estructura de indexación que constituye la base del resto del sistema. Por tanto, esta capa se puede definir de forma simplificada como un proceso que recibe como entrada una colección de documentos y produce como resultado una estructura de indexación. En realidad, este proceso es un flujo de trabajo en el que intervienen varios subsistemas que describimos en las próximas secciones. El resultado final del proceso, además de la estructura de indexación, contiene información almacenada en las bases de datos documental y geográfica que permite mejorar la calidad de los resultados presentados a los usuarios. Para entender mejor este flujo de trabajo, en la sección 4.5 describimos la interacción de los componentes de la arquitectura en varios escenarios de uso típicos como la creación de la estructura de indexación.

4.2.1. Abstracción de documentos

En la parte inferior izquierda de la figura de la arquitectura se encuentra el módulo de abstracción de documentos. Es importante recordar que uno de los requisitos que imponemos a la arquitectura es que debe ser genérica. Por tanto, debe soportar la indexación de distintos tipos de documentos. Estos documentos pueden ser diferentes no sólo en cuanto a ser almacenados empleando formatos de archivo diferentes (texto plano, XML, HTML, etc.), sino también en cuanto a sus esquemas de contenido. Un esquema de contenido podría tener un conjunto de atributos de interés para el sistema (por ejemplo, identificador del documento, autor y texto del documento), mientras que otro esquema podría tener un conjunto de atributos diferente (por ejemplo, identificador del documento, resumen, texto, autor y fuente).

Para ofrecer soporte a la indexación de distintos tipos de documentos hemos definido una abstracción para el concepto *documento* que lo representa como un conjunto de *campos*. Cada uno de estos campos contiene un valor extraído del texto del documento. Además, cada campo puede ser almacenado, indexado o ambas cosas. Si un campo es almacenado, su contenido se guarda en la estructura de indexación y puede ser recuperado mediante consultas. Si es indexado, su contenido se emplea para la construcción de la estructura de indexación. Esta definición de un documento como un conjunto de campos es similar a la empleada por el motor de búsqueda textual Lucene [Apa08a]. La principal diferencia reside en que en nuestra abstracción un campo se puede indexar en el índice textual, en el índice espacial o en ambos índices mientras que en Lucene, al ser un motor de búsqueda textual clásico, no tiene sentido el concepto de indexación espacial.

Para soportar diferentes tipos de documentos y diferentes formatos de archivo, el sistema expone la abstracción de documentos como una interfaz de programación que se puede extender mediante implementaciones particulares para diferentes configuraciones de formatos de archivo y esquemas de documento. Para soportar una nueva configuración, el desarrollador tan sólo tiene que implementar la interfaz *DocumentCreator* que define las operaciones que deben ser implementadas para crear documentos. En la figura 4.2 se muestra un diagrama de clases con la interfaz, un ejemplo de clase de implementación y la factoría encargada de instanciar el creador de documentos adecuado a la colección.

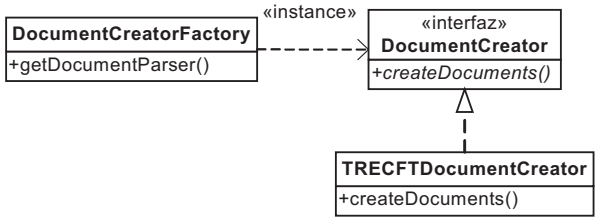


Figura 4.2: Módulo de abstracción de documentos.

A modo de ejemplo, en la validación experimental del sistema hemos indexado documentos pertenecientes a la colección del Financial Times [Nat08]. Esta colección de documentos está etiquetada empleando SGML (*Standard Generalized Markup Language*). Cada documento tiene una etiqueta <DOCNO> que contiene la cadena de caracteres de identificación TREC y una etiqueta <TEXT> que incluye el contenido principal del documento. La figura 4.3 muestra un ejemplo parcial de un documento de esta colección.

Para dar soporte a esta colección de documentos, hemos definido el analizador adecuado para dichos documentos (*TRECFTDocumentCreator*) que implementa la

```
<DOC>
      <DOCNO>FT941-6371</DOCNO>
      <TEXT>
            Senior European company executives are being invited to 'vote'
            for Europe's Most Respected Companies . . .
      </TEXT>
</DOC>
```

Figura 4.3: Ejemplo de documento del Financial Times (TREC).

interfaz *DocumentCreator* y que construye documentos con dos campos. El primer campo, para el contenido etiquetado como DOCNO, es almacenado pero no indexado. El segundo campo, para el contenido de la etiqueta TEXT, no se almacena pero se indexa tanto en el índice textual como en el espacial.

4.2.2. Indexación textual

En el capítulo anterior hemos presentado la estructura de indexación que forma el núcleo de nuestro prototipo. Dicha estructura de indexación contiene tanto un índice textual como un índice espacial (es una estructura de doble índice). Sin embargo, como veíamos en la sección 2.3.4, existen otras propuestas de estructuras de indexación híbridas que combinan en una sola estructura la indexación textual y espacial. Ya hemos mencionado también que uno de los requisitos que le exigimos a nuestra arquitectura es que sea general y, por tanto, que no esté ligada a nuestra estructura de indexación. Para cumplir con este requisito, hemos puesto especial interés en que todas las etapas involucradas en la construcción del índice estén diseñadas por contrato; es decir, sin depender de una estructura determinada. Una de las consecuencias inmediatas de este diseño es que tanto la parte textual como la parte espacial de la estructura deben cumplir las interfaces donde definimos las operaciones que deben soportar (la creación, actualización, borrado y los distintos tipos de consulta). Esto hace que sea muy sencillo cambiar la implementación de cualquiera de las partes. Pero no sólo eso, sino que la estructura de indexación completa también está determinada por una interfaz y, por tanto, si se desea integrar en la arquitectura una estructura híbrida también es posible. Las etapas de indexación textual e indexación espacial se definen de la forma más genérica posible de tal manera que únicamente se encargan del procesado de los documentos para obtener la información necesaria para construir un índice textual y un índice espacial respectivamente.

En nuestro prototipo empleamos Lucene [Apa08a] para implementar el índice textual. Lucene es una librería escrita completamente en Java que permite el desarrollo de motores de búsqueda textuales de alto rendimiento. Es una librería de código

abierto que forma parte del proyecto Apache. Lucene emplea una representación objetual de los documentos indexables. Un *documento* de Lucene contiene varios *campos*. Un campo es un par (nombre, valor) e información sobre si es almacenado y/o indexado. Los valores de los campos se establecen empleando analizadores. Estos analizadores implementan varias técnicas que se emplean habitualmente en sistemas de IR para reducir el número de palabras indexadas y mejorar el rendimiento del índice (por ejemplo, borrado de *stopwords*, *stemmers*, etc.). El analizador más sofisticado construido en el núcleo de Lucene es el *StandardAnalyzer*. Este analizador contiene reglas para direcciones de correo electrónico, acrónimos, nombres de máquina y números en punto flotante. Además, convierte el valor a letras minúsculas y borra *stopwords*.

En la etapa de indexación textual del flujo de trabajo el sistema construye un índice de Lucene. Cada una de las abstracciones de documentos construidas en la etapa anterior se inserta en el índice textual. El identificador de documento se almacena pero no se indexa en el índice textual y cada campo marcado como indexable en el índice textual o en ambos índices se indexa *tokenizado* (dividido en los componentes léxicos indexables) en el índice de Lucene.

4.2.3. Indexación espacial

Tanto si la estructura de indexación mantiene por separado un índice textual y un índice espacial (es decir, es una estructura de doble índice) como si los combina en una estructura híbrida, es necesaria una etapa de indexación espacial para obtener toda la información necesaria en la construcción de la parte espacial de la estructura. Esta etapa se puede llevar a cabo de forma solapada con el análisis textual de los documentos (borrado de *stopwords*, lematización de los términos indexables, etc.) ya que, aunque se realizan sobre la misma colección de documentos, son procesos independientes.

La indexación espacial es la tarea más compleja de todo el flujo de trabajo y se puede dividir en dos etapas principales. En la primera etapa el sistema analiza los campos de los documentos marcados como indexables espacialmente y extrae los candidatos a nombre de lugar del texto. En la segunda etapa se procesan todos esos candidatos a nombre de lugar para determinar cuales se corresponden con verdaderos lugares y, en ese caso, geo-referenciarlos. En las siguientes secciones describimos en más detalle cada una de estas dos etapas.

Obtención de candidatos a nombre de lugar

La etapa de obtención de candidatos a nombre de lugar consiste en procesar todos los campos de los documentos marcados como indexables espacialmente en la etapa de abstracción de documentos (ver sección 4.2.1) buscando posibles referencias a lugares. Como explicábamos en la sección 2.3.3, existen dos técnicas dentro del área

de análisis lingüístico que se han empleado con bastante éxito para este propósito: etiquetado *Part-Of-Speech* y *Named-Entity-Recognition*. La primera de ellas consiste en etiquetar los *tokens* del texto secuencialmente con etiquetas sintácticas, como *verbo* o *gerundio*. La segunda se ocupa de la búsqueda de menciones en el texto de elementos que pertenecen a un conjunto predefinido de categorías, como *nombres de persona*, *nombres de organización* o *nombres de lugar*. La combinación de ambas técnicas presenta unos resultados bastante buenos para la obtención de candidatos a nombre de lugar.

Nuestro módulo de obtención de candidatos a nombre de lugar emplea la herramienta de lenguaje natural LingPipe [Ai08] para la búsqueda de los candidatos. LingPipe proporciona un conjunto de librerías, implementadas en el lenguaje de programación Java, para realizar análisis del lenguaje humano. Entre esas librerías se encuentran algunas para realizar el etiquetado *Part-Of-Speech* y *Named-Entity-Recognition*. Además, esta librería se puede emplear de forma libre para propósitos de investigación. LingPipe involucra el entrenamiento supervisado de un modelo estadístico para reconocer entidades. Los datos de entrenamiento deben estar etiquetados con todas las entidades de interés y sus correspondientes tipos. En la validación del sistema con la colección del Financial Times empleamos LingPipe entrenado con el corpus MUC6 (http://www.ldc.upenn.edu) etiquetado con nombres de lugar, de persona y de organización. Por tanto, el módulo realiza un post-procesado para filtrar las entidades resultantes seleccionando sólo los nombres de lugar y descartando los de persona y los de organización.

Geo-referenciacion de los candidatos

Después de descubrir una colección de nombres de lugar candidatos, el sistema debe distinguir los falsos candidatos y geo-referenciar los verdaderos. Recordemos que en este contexto geo-referenciar un nombre de lugar supone no sólo obtener sus coordenadas en un sistema de coordenadas particular, sino también obtener toda la información necesaria para incluir el lugar en un índice espacial. En el caso de la estructura que presentamos en el capítulo anterior, para geo-referenciar correctamente un topónimo debemos obtener todos los antecesores del topónimo en la jerarquía que define la ontología del espacio geográfico. Para este propósito hemos desarrollado un servicio de ontología del espacio geográfico construido empleando un *gazetteer* y un *proveedor de geometrías*. Aunque este servicio se encuentra en la capa de servicios de procesado (ver sección 4.3), lo explicamos aquí debido a que se emplea principalmente en esta etapa del flujo de trabajo.

Un *gazetteer* es un diccionario geográfico que contiene, además de nombres de lugar, nombres alternativos, medidas de población y otra información relacionada con los lugares. En la implementación de nuestro prototipo empleamos *Geonames* [Geo07] que nos proporciona una base de datos con información geográfica bajo licencia

creative commons. Esta base de datos contiene más de dos millones de lugares de todo el mundo, con sus coordenadas en latitud/longitud en WGS84 (*World Geodetic System 1984*). Además, todos los lugares tienen una categoría asignada por lo que es posible clasificarlos siguiendo una taxonomía (por ejemplo, continentes, países, regiones, etc.).

Sin embargo, Geonames (y los *gazetteers* en general) no proporcionan otras geometrías para los lugares distintas de un simple punto representativo. Muchas veces, esto supone un problema ya que, por ejemplo, para la parte espacial de nuestra estructura necesitamos la geometría real del lugar (por ejemplo, el límite de los países o de las regiones). Para solucionar este inconveniente, hemos definido un servicio proveedor de geometrías que complementa la información suministrada por el *gazetteer*. Como base de este servicio hemos empleado la cartografía de *Vector Map* (VMap) [Nat07a]. VMap es una versión actualizada y mejorada de la cartografía proporcionada por la *National Imagery and Mapping Agency's Digital Chart of the World*. Aunque esta información cartográfica se encuentra en un formato propietario, existen herramientas libres que permiten su conversión a formatos abiertos como *shapefile*. Un ejemplo de estas herramientas es FWTools [FWT07]. Con esta información cartográfica hemos creado una base de datos espacial empleando el SGBD PostgreSQL [Pos07] y su extensión para manejo de información espacial PostGIS [Ref07]. Esto nos permite gestionar y realizar mejoras sobre esta información empleando muchas herramientas avanzadas disponibles en el campo de los sistemas de información geográfica. Aunque la implementación del prototipo emplea Geonames y VMap, el sistema ha sido diseñado de manera que esos componentes son fácilmente intercambiables. Todos los accesos a ellos se realizan mediante interfaces genéricas que pueden ser fácilmente implementadas por otros componentes.

El objetivo del servicio de geo-referenciación de nombres de lugar es obtener para cada candidato a nombre de lugar toda la información necesaria para incluirlo en la estructura de indexación, en caso de corresponderse con un verdadero topónimo. Para ello emplea tanto el *gazetteer* como el servicio proveedor de geometrías. En este punto del flujo de trabajo, la arquitectura debe estar preparada para tratar con algunos problemas de los que depende en gran parte la calidad de los resultados del sistema, en términos de precisión y recuperación (ver sección 2.1.6). En la sección 2.3.3 estudiábamos estos problemas provocados por la ambigüedad de los nombres de lugar. En primer lugar, un nombre de lugar puede ser ambiguo (*polisemia*). Por ejemplo, *Londres* es la capital del *Reino Unido* pero también es una ciudad en *Ontario*, *Canadá*. En segundo lugar, los nombres de lugar para una misma localización geográfica no suelen ser únicos (por ejemplo, *Los Ángeles* y *LA*). La principal implicación práctica de estos problemas es que el servicio encargado de realizar esta tarea debe estar preparado para devolver más de una descripción para cada lugar con sus correspondientes valores de importancia que permitan compararlas entre sí.

Generalmente, los *gazetteers* proporcionan información que permite caracterizar cada lugar, como el tipo de entidad de población de que se trata, su población, si es capital o no, etc. En nuestro prototipo combinamos estos datos, que nos proporciona

Geonames, para almacenar en cada nodo un valor de importancia que nos permite comparar ese nodo con todos los que están referidos por el mismo nombre de lugar. Por tanto, el atributo importancia no representa un valor absoluto sino que permite comparar lugares referidos por el mismo nombre de lugar de tal modo que los nodos asociados con los lugares más importantes almacenarán un 1 en dicho atributo, los nodos asociados con los lugares siguientes en orden de importancia almacenarán un 2 y así sucesivamente. Por ejemplo, si el nombre de lugar es *England* la importancia de *England* (*United Kingdom*) es 1 (ya que tiene el tipo de entidad de mayor rango), la importancia de *England* (*Arkansas*) es 2 (ya que tiene una población de tamaño considerable) y la importancia de *England* (*Oppland Fylke*) es 3 (ya que es una ciudad muy pequeña). En la sección 4.3.1 veremos cómo empleamos este valor para calcular la relevancia espacial de las consultas.

Para la tarea de geo-referenciación de nombres de lugar hemos diseñado e implementado una estructura jerárquica siguiendo el patrón de diseño *Cadena de Responsabilidad* [GHJV96]. La figura 4.4 muestra un diagrama de clases de este componente. La jerarquía está compuesta por cuatro niveles (*Continent*, *Country*, *Region* y *PopulatedPlace*); uno por cada nivel de la ontología del espacio geográfico usada en el sistema. Cada nivel de la jerarquía define una conexión con el *gazetteer* y con el proveedor de geometrías (*Geometry Supplier*) para recuperar los datos necesarios para completar la tarea. El diseño está especialmente cuidado en cuanto a su facilidad de extensión ya que, al depender directamente de la ontología del espacio geográfico, este componente tiene que ser adaptado cuidadosamente al dominio de aplicación del sistema. Siguiendo con el ejemplo que expusimos al describir la estructura, si una aplicación en concreto usa un área restringida del espacio geográfico donde las clases *Continent* y *Country* no son necesarias pero, en cambio, son necesarias las clases *Province*, *Municipality*, *City* y *Suburb*, esta jerarquía tiene que adaptarse a la nueva configuración de la ontología del espacio geográfico. Por tanto, es muy importante que sea fácil implementar nuevos componentes y configurar el sistema con los que debe emplear.

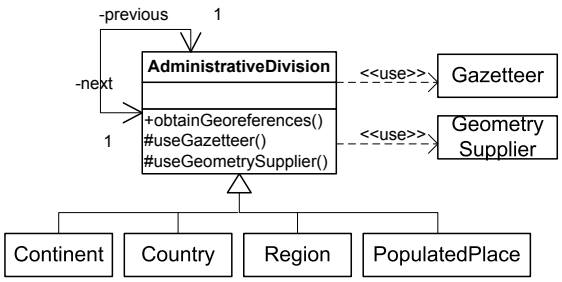


Figura 4.4: Módulo de geo-referenciación.

Además del diseño, también es importante entender el algoritmo de georeferenciación desarrollado sobre esta jerarquía. El algoritmo que hemos diseñado requiere dos pasadas, la primera de las cuales se basa en un recorrido descendente en la jerarquía y la segunda en un recorrido ascendente. En el recorrido descendente, en cada nivel de la jerarquía se emplea su conexión con el *gazetteer* para obtener todos los posibles lugares que se pueden asociar con el candidato a nombre de lugar consultado. Si el *gazetteer* no devuelve ningún lugar entonces ese candidato se descarta. El recorrido ascendente se inicia con cada lugar encontrado en el *gazetteer* en el recorrido anterior. El objetivo de esta segunda pasada es completar toda la información necesaria para poder incluir el lugar encontrado en la estructura de indexación. Para ello, se emplea el *gazetteer* para acceder a los nodos de nivel superior y el proveedor de geometrías para completar la información proporcionada por el *gazetteer*. Por ejemplo, si el candidato a nombre de lugar consultado es *Londres*. En la primera pasada, el sistema obtendrá al menos dos lugares que responden a ese nombre (uno en Canadá y otro en el Reino Unido). En el recorrido ascendente se completará toda la información necesaria de esos dos lugares hasta formar las descripciones completas de *Europa, Reino Unido, Inglaterra, Londres* y *Norte América, Canadá, Ontario, Londres*.

4.2.4. Estructura de indexación

El resultado final de todo el flujo de trabajo que hemos descrito a lo largo de esta sección es la inclusión de los documentos en una estructura de indexación. La figura 4.5 muestra un diagrama de clases de nuestra estructura de indexación. El diseño e implementación de dicha estructura están basados en las ideas presentadas en el capítulo anterior. Por tanto, el componente principal de la estructura de indexación es un árbol compuesto por nodos (*IndexStructureNode*) que representan nombres de lugar. Estos nodos están conectados por medio de relaciones de contenido espacial (*spatiallyContainedBy*). En cada uno almacenamos el nombre de lugar, el MBR de la geometría que representa dicho lugar, un valor que permite comparar la importancia de todos los nodos referidos por el mismo nombre de lugar (en la sección anterior explicamos cómo se calcula este valor) y la lista con los identificadores de los documentos que incluyen referencias geográficas a ese lugar. Finalmente, cada nodo emplea un R-tree para mejorar el rendimiento a la hora de acceder a los nodos hijos. Como ya vimos en la sección 2.2.4, el R-tree es uno de los índices espaciales clásicos más empleados para mejorar la eficiencia en sistemas de información geográfica.

El diseño se completa con dos estructuras. En primer lugar, un índice textual con todas las palabras en los documentos que se emplea para resolver consultas textuales (este índice ya se describió en la sección 4.2.2). En segundo lugar, para mejorar la eficiencia de ciertos tipos de consulta muy habituales, empleamos una tabla *hash* con nombres de lugar que almacena para cada nombre de lugar su posición en la estructura de indexación. Esto proporciona un acceso directo a un nodo determinado mediante

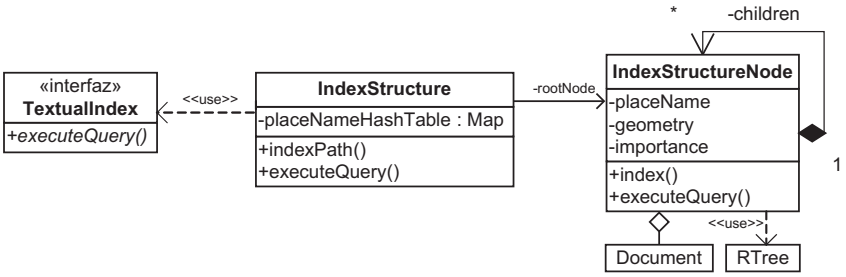


Figura 4.5: Diagrama de clases de la estructura de indexación.

una palabra clave obtenida con el servicio de *gazetteer* si la palabra procesada es un nombre de lugar.

Mantener separados los índices para el ámbito textual y para el espacial tiene muchas ventajas que ya hemos descrito en el capítulo anterior. En primer lugar, permite procesar todas las consultas textuales de manera eficiente empleando el índice textual y todas las consultas espaciales empleando el índice espacial. Además, el sistema puede soportar consultas que combinen aspectos textuales y espaciales. También, se pueden gestionar de manera independiente actualizaciones en cada índice lo que facilita que se puedan añadir y borrar datos. Finalmente, se pueden aplicar optimizaciones a cada estructura de indexación de manera individual. Por el contrario, la estructura presenta dos inconvenientes principales. En primer lugar, el árbol que soporta la estructura es posiblemente no balanceado penalizando la eficiencia del sistema. En segundo lugar, los sistemas ontológicos tienen una estructura fija y, por tanto, nuestra estructura es estática y debe ser construida *ad-hoc*.

4.3. Servicios de procesado

4.3.1. Servicio de evaluación de consultas

El servicio de evaluación de consultas es el componente de la arquitectura que se encarga de responder las consultas de los usuarios empleando para ello la estructura de indexación. Por tanto, los tipos de consulta que oferta este servicio al usuario dependen de la estructura de indexación que se integre en el sistema. En el capítulo anterior hemos presentado los diferentes tipos de consulta que se pueden resolver empleando nuestra estructura de indexación y también hemos descrito los algoritmos necesarios para responder dichas consultas. Sin embargo, para que el resultado le sea útil a los usuarios, el servicio de evaluación de consultas debe elaborar un *ranking*

de relevancia de los documentos obtenidos empleando la estructura de indexación. Es decir, debe ordenar los resultados de forma que la posición indique lo bien que se ajusta el documento a la consulta realizada. En esta sección describimos las ecuaciones que se emplean para calcular la relevancia de los resultados para cada tipo de consultas: *consultas textuales puras*, *consultas espaciales puras* y *consultas híbridas*.

Consultas textuales puras

Las *consultas textuales puras* se responden empleando el índice textual que forma parte de nuestra estructura de indexación. Ésta es una de las ventajas de mantener por separado la indexación textual y la espacial. Como en la implementación de nuestro prototipo empleamos Lucene, la relevancia textual se calcula según la fórmula de *scoring* de esta librería. Dicha fórmula emplea una combinación del *Vector Space Model* (VSM) y del modelo booleano de IR [BYRN99]. Una característica muy importante de la fórmula es la garantía de que todos los valores se encuentran en el rango [0, 1]. En la sección 2.1.2 presentamos las ecuaciones más importantes que emplea Lucene. Además, en [GH05, Apa08b] se puede encontrar más información acerca de ellas.

Consultas espaciales puras

Del mismo modo que las consultas textuales puras se responden empleando el índice textual que forma parte de la estructura, las *consultas espaciales puras* se responden empleando la componente espacial de nuestra estructura de indexación. Dado que un documento en el conjunto de resultados puede incluir referencias a uno o más nombres de lugar relevantes para la consulta, tenemos que calcular la relevancia del documento d para la consulta q debida a cada lugar l. Denotamos esta relevancia como $toponymRelevance_{q,d,l}$. Además de garantizar que esta relevancia es mayor que cero sólo para aquellos lugares referidos en el documento relevantes para la consulta, también garantizamos que mantiene un valor inferior a uno para facilitar su integración con la relevancia textual cuando las consultas son híbridas. Finalmente, obtenemos la relevancia espacial del documento d con respecto a la consulta q como la relevancia máxima debida a cualquiera de los lugares relevantes (ecuación 4.1).

$$spatialRelevance_{q,d} = max\{toponymRelevance_{q,d,l}\} \tag{4.1}$$

A la hora de calcular la relevancia espacial debida a un lugar (denotada como $toponymRelevance_{q,d,l}$) debemos tener en cuenta que el usuario puede indicar el ámbito espacial de una consulta empleando un *nombre de lugar*, *un nodo de la ontología* o una *ventana de interés*. La relevancia en cada uno de estos ámbitos tiene una semántica diferente y, por tanto, también se debe calcular de forma diferente. Cuando

se emplea un nombre de lugar para indicar el ámbito espacial, el sistema realiza la traducción de ese nombre a los nodos en la estructura que están asociados con él. Después, el algoritmo desciende en los sub-árboles, iniciados en cada uno de esos nodos, obteniendo la lista de documentos asociados con los nodos visitados en dicho descenso. Es decir, el resultado de la consulta contiene todos los documentos asociados con los nodos relacionados con el nombre de lugar consultado y todos los documentos asociados con nodos en sus respectivos sub-árboles. Dado que puede haber más de un nodo en la estructura de indexación espacial para un mismo nombre de lugar (debido a la polisemia), los nodos incluyen un atributo *importance* que representa la importancia del lugar asociado con el nodo en comparación con los lugares asociados con otros nodos que están referidos por el mismo nombre de lugar. En la sección 4.2.3 explicamos como se establece el valor de este atributo. Para cada nombre de lugar, este atributo tiene siempre el valor uno para el nodo más importante y valores mayores para el resto de nodos. La ecuación 4.2 combina este atributo con la distancia (*distance*) desde la raíz del subárbol al nodo donde está asociado el documento para calcular la relevancia espacial de cada documento d debida al nombre de lugar l.

$$toponymRelevance_{q,d,l} = \frac{0,5^{distance}}{importance} \tag{4.2}$$

El algoritmo para resolver las consultas cuando el ámbito espacial se indica directamente mediante un nodo de la ontología es una simplificación del anterior ya que sólo es necesario descender en un sub-árbol. Por tanto, la ecuación 4.3 es muy similar a la anterior pero con una modificación para reflejar que el nodo raíz del sub-árbol se indica manualmente, no existe ambigüedad y, por tanto, tiene la máxima relevancia.

$$toponymRelevance_{q,d,l} = \begin{cases} 1 & \text{si l se especificó en la consulta} \\ \frac{0,5^{distance}}{importance} & \text{en otro caso} \end{cases} \tag{4.3}$$

En la figura 4.6 mostramos un esbozo de la estructura de indexación que es útil para entender la diferencia entre ambos tipos de consulta y sus fórmulas para el cálculo de la relevancia. Cada nodo en la figura lo anotamos con su importancia entre paréntesis. Como ejemplo del primer tipo de consultas, supongamos que el usuario realiza una consulta especificando el nombre de lugar *England*. En este caso, la relevancia de un documento debida a *England* (una importante ciudad de *Arkansas*) será mayor que la relevancia debida a *England* (una pequeña ciudad de *Oppland Fylke*) pero menor que la relevancia debida a *England* (una parte de *United Kingdom*). En concreto, los valores de relevancia en este caso son 0,5 para *England (Arkansas)*, 0,33 para *England (Oppland)* y 1,0 para *England (United Kingdom)*. Además, la relevancia de los documentos asociados con ciudades importantes de *England (UK)*, como *London*

o *Liverpool*, debida a esas ciudades es 0,5. Este valor es lo suficientemente alto como para tenerlo en cuenta y, por tanto, es probable que los documentos asociados con estas ciudades salgan en posiciones altas del *ranking*. Supongamos ahora, como ejemplo del segundo tipo de consultas, que el usuario seleccionó directamente el nodo *England (Arkansas)*. En este caso, la relevancia de los documentos asociados con ese nodo es 1,0 ya que se indica explícitamente el interés en documentos con referencias geográficas a esa lugar (es decir, no existe ambigüedad en la consulta). Los documentos asociados con los nodos de *England (UK)*, *England (Oppland)* y con sus descendientes tendrán relevancia 0 y, por tanto, no saldrán en el *ranking* de resultados debido a que el algoritmo no visitará dichos nodos.

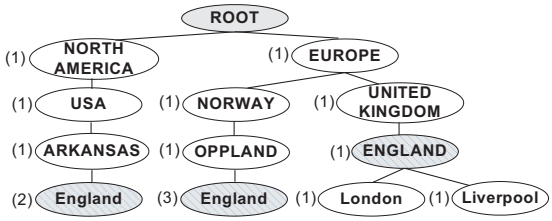


Figura 4.6: Consultas especificadas empleando un nombre de lugar vs consultas especificadas seleccionando un nodo.

Por último, cuando el ámbito espacial se indica empleando una ventana de consulta el algoritmo empleado para resolver las consultas es el clásico de las estructuras de indexación espacial. Es decir, se desciende en el árbol desde la raíz teniendo en cuenta solamente aquellos nodos cuya geometría asociada interseca con la ventana de consulta. La ecuación 4.4 establece cómo calculamos la relevancia espacial en este caso. A nivel conceptual, calculamos esta relevancia basándonos en otras más sencillas como son aquella debida a la distancia al centro de la ventana ($dc_{q,l}$), y al área de solape entre el nodo y la ventana ($oa_{q,l}$). Las variables w_{dc} y w_{oa} son factores de pesado para ponderar la importancia de cada una de las relevancias anteriores en la relevancia final. Además, seguimos empleando la importancia intrínseca del nodo (*importance*) como factor corrector para que los nodos más importantes referidos por un nombre de lugar tengan también más importancia en la relevancia espacial.

$$toponymRelevance_{q,d,l} = \frac{w_{dc} * dc_{q,l} + w_{oa} * oa_{q,l}}{importance} \tag{4.4}$$

La ecuación 4.5 define cómo calculamos la relevancia debida a la distancia al centro de la ventana. $centerDistance_{q,l}$ representa la distancia Euclídea entre el lugar l y la ventana de consulta q. De manera similar, $cornerDistance_q$ es un factor de ponderación que representa la máxima distancia al centro de la ventana, esto nos permite normalizar

el resultado en el intervalo [0, 1]. La idea subyacente a la ecuación reside en que cuando un usuario dibuja una ventana de consulta está focalizando su interés en la zona central de la ventana. Por tanto, lugares más próximos al centro de la ventana deben tener mayor relevancia que lugares más alejados.

$$dc_{q,l} = 1 - \frac{centerDistance_{q,l}}{cornerDistance_q} \tag{4.5}$$

La relevancia debida al área de solape con la ventana de consulta la calculamos de acuerdo a la ecuación 4.6. Esta ecuación tiene en cuenta que la geometría asociada a un nodo puede ser un punto (por ejemplo, en un nodo hoja) y, por tanto, el concepto de solape entre la consulta y el nodo no tiene sentido. En este caso, la ecuación determina la relevancia únicamente en función del tamaño de la ventana y es inversamente proporcional a su área. La fórmula $\frac{1}{[area(q)+1]^{0,15}}$ es la ecuación de la curva de la figura 4.7. Los puntos de interés marcados en la gráfica se corresponden con el tamaño medio de las regiones, de los países y de los continentes. Si el usuario dibuja una ventana del tamaño de una región parece lógico pensar que está totalmente interesado en los lugares poblados de esa región. Sin embargo, a medida que aumenta el tamaño de la ventana ese interés se irá reduciendo.

$$oa_{q,l} = \begin{cases} \frac{1}{[area(q)+1]^{0,15}} & \text{si la geometría de l es un punto} \\ max\{0, \frac{area(l \cap q)}{area(q)} - \frac{area(l \otimes q)}{area(l)}\} & \text{en otro caso} \end{cases} \tag{4.6}$$

La figura 4.8 muestra un ejemplo de ventana de consulta en el centro de Italia para aclarar las ecuaciones anteriores. El ejemplo se basa en dos regiones, *Umbria* y *Abruzzi*, de las cuales dibujamos en línea discontinua sus MBRs, y un lugar poblado, *Rome*, del cual dibujamos su punto representativo. Tanto estos MBRs como la ventana de consulta, q, se emplean para calcular el área de sus respectivas entidades (es decir, *area(Umbria)*, *area(Abruzzi)*, *area(q)*). La región de *Umbria* la utilizamos para ilustrar la relevancia debida al área de solape (ecuación 4.6). Esta relevancia se calcula empleando el área de la intersección de la región con la consulta ($area(Umbria \cap q)$) y el área en la parte de la región que no interseca con la consulta ($area(Umbria \otimes q)$). Además, la figura también muestra tres distancias empleadas para calcular la relevancia debida a la distancia al centro de la ventana (ecuación 4.5). El factor de ponderación *cornerDistance* lo representamos con una línea sólida, y las distancias de *Rome* y *Abruzzi* al centro de la ventana de consulta las representamos con líneas punteadas.

Observando la figura podemos deducir que la relevancia debida a la distancia al centro de la ventana es mayor para *Rome* que para *Umbria* y *Abruzzi*. Los valores aproximados son $dc_{q,Rome} = 0,7$, $dc_{q,Abruzzi} = 0,5$ y $dc_{q,Umbria} = 0,45$. La relevancia debida al área de solape con la ventana es muy similar para *Abruzzi* y *Umbria*

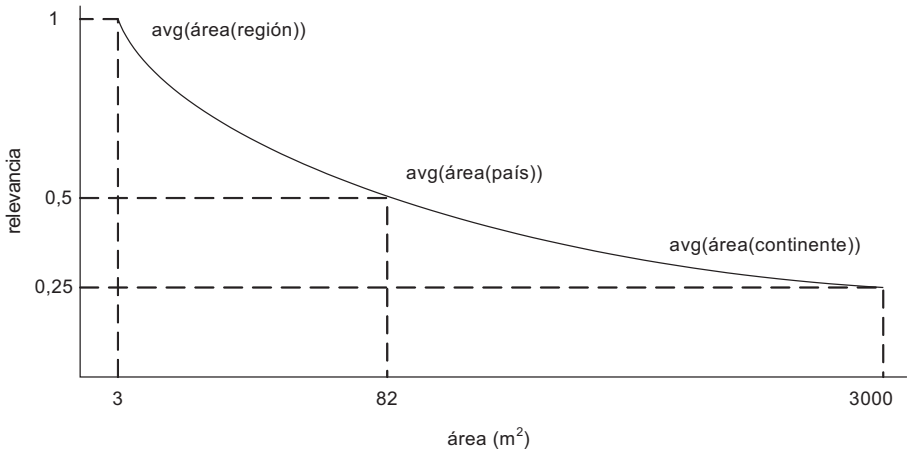


Figura 4.7: Relevancia espacial debida únicamente al tamaño de la ventana.

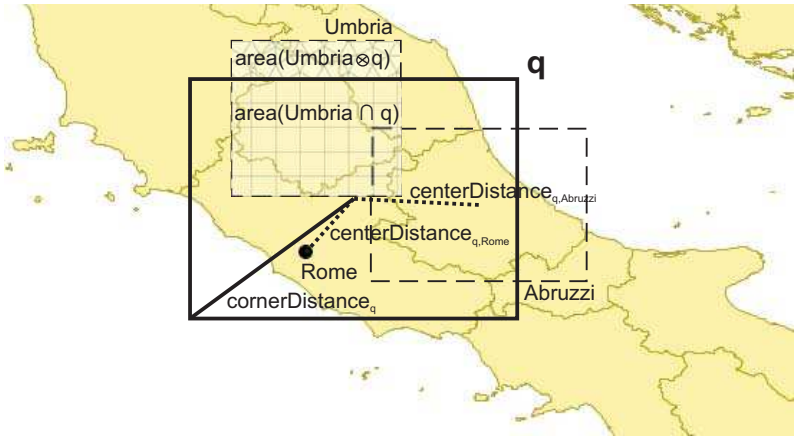


Figura 4.8: Consultas especificadas empleando una ventana de consulta.

ya que, aunque el área de la intersección de la consulta con la región de *Abruzzi* es un poco superior que en el caso de la región de *Ubria* ($area(Abruzzi \cap q) > area(Umbria \cap q)$), también lo es el área de la parte la región que queda fuera de la consulta ($area(Abruzzi \otimes q) > area(Umbria \otimes q)$). Los valores aproximados que obtenemos son $oa_{q,Abruzzi} = oa_{q,Abruzzi} = 0,85$. En cuanto a *Rome*, al ser un punto la relevancia debida al área de solape depende únicamente del tamaño de la ventana. En este caso, la ventana tiene un tamaño sensiblemente superior al tamaño medio de una región y, por tanto, podemos aproximar esta relevancia como $oa_{q,Abruzzi} = 0,75$. Asumiendo que los tres lugares se corresponden con nodos de relevancia 1 y tomando los factores de ponderación ($w_{dc} = w_{oa} = 0,5$), las relevancias finales que obtenemos son 0,73 para *Rome*, 0,68 para *Abruzzi* y 0,65 para *Umbria*.

Consultas híbridas

Las *consultas híbridas* se resuelven empleando los índices textual y espacial del mismo modo que acabamos de describir. Por tanto, el problema de crear el *ranking* de relevancia en este tipo de consultas se puede simplificar a cómo combinar la relevancia textual con la relevancia espacial (ambas calculadas con las fórmulas que acabamos de describir). Para hacer este cálculo empleamos la suma ponderada de ambas relevancias tal y como se muestra en la ecuación 4.7.

$$relevance_{q,d} = w_t * textualRelevance_{q,d} + w_s * spatialRelevance_{q,d} \qquad (4.7)$$

La suma ponderada es uno de los métodos más simples para combinar las relevancias y posiblemente el más empleado [MSA05, AS06, ZXW+05]. Además, es la base de otros métodos más complicados publicados recientemente [YC07]. En nuestra propuesta asumimos $w_t = 1 - w_s$ y calculamos w_t para normalizar la diferencia de escala entre el *ranking* textual y el espacial. Aunque acotamos tanto la relevancia textual como la espacial al intervalo [0, 1], los valores no son directamente comparables. Puede suceder, por ejemplo, que los valores de relevancia textual obtenidos sean siempre inferiores a 0,6 mientras que en los valores de relevancia espacial haya algunos próximos a 1. En este caso combinarlos asumiendo $w_t = w_s = 0,5$ daría mucha más importancia a la relevancia textual que a la espacial. Para corregir esta diferencia de escala emplearíamos un w_t mayor que 0,5.

4.3.2. Web map service

Aunque resolver correctamente las consultas es el objetivo principal de todo sistema de recuperación de información, presentar de manera amigable para el usuario los resultados también es una cualidad muy deseable. Nuestra arquitectura contempla la inclusión de un *Web Map Service* (WMS) [Ope02]. Este componente tiene

dos objetivos principales. El primer objetivo es proporcionar los mapas interactivos para la interfaz de usuario de consulta. Esto permite que el usuario pueda navegar por la cartografía que proporciona el sistema buscando zonas de interés para su consulta. El segundo objetivo es la creación de representaciones cartográficas de los resultados de las consultas. A diferencia de los sistemas de recuperación de información tradicionales, en este tipo de sistemas los documentos son relevantes para una consulta porque están geo-referenciados en un lugar determinado. Por tanto, realizar la representación cartográfica de los lugares donde están geo-referenciados los documentos relevantes aporta un valor añadido a estos sistemas.

El servicio WMS es un estándar definido por el OGC [Ope03] para proporcionar mapas con datos geo-referenciados. Este servicio define un protocolo sencillo sobre HTTP para la realización de peticiones y la obtención de los mapas. De forma simplificada, en las peticiones se especifica la capa, o capas, en las que se encuentra la información geográfica y el área de interés; y la respuesta contiene una imagen, o un conjunto de imágenes, que se pueden visualizar en cualquier navegador web. Este servicio es posiblemente uno de los más utilizados en el campo de los sistemas de información geográfica. Por este motivo, existen una gran cantidad de implementaciones, tanto de pago como en software libre, que proporcionan este servicio. Además, como veremos en la descripción de la interfaz de usuario de consulta (sección 4.4), también existen muchos clientes que permiten incluir mapas en un entorno web de forma sencilla. Estos mapas se generan mediante consultas a servicios WMS.

En el desarrollo de nuestro prototipo empleamos Geoserver [Geo08]. Geoserver es uno de los WMS de referencia dentro del mundo del software libre. Está escrito en Java y tiene una comunidad de desarrolladores muy activa. Como información geográfica de base para el servicio empleamos VMap [Nat07a]. Esta información geográfica es la misma que empleamos en el servicio de ontología del espacio geográfico (ver sección 4.2.3). La separación en capas y servicios nos aporta una independencia física de los datos que permite que esta información sea compartida. Además, como ya hemos mencionado es posible la mejora de los datos de base empleando muchas herramientas GIS avanzadas. El tener la base de datos separada y compartida permite que las mejoras se reflejan automáticamente en todo el sistema.

4.4. Interfaces de usuario

El sistema tiene dos interfaces de usuario diferentes: una interfaz de usuario de administración y una interfaz de usuario de consulta. La primera de ellas la hemos desarrollado como una aplicación *stand-alone* y se emplea para gestionar la colección de documentos. Las funcionalidades principales de esta interfaz son: crear índices, añadir documentos a índices, cargar y almacenar índices, y otras opciones relacionadas

con la gestión de índices y del sistema en general. La pantalla principal de esta interfaz muestra información útil acerca del índice cargado entre la que se encuentra el número de documentos indexados, los campos de cada uno de esos documentos, el número de nombres de lugar en el índice, etc.

La figura 4.9 muestra una captura de pantalla de la interfaz de usuario de consulta. Esta interfaz la hemos desarrollado como una aplicación web empleando el API de Open Layers [OSG08]. Este API proporciona gran cantidad de utilidades para manipular mapas y añadir contenido a esos mapas.

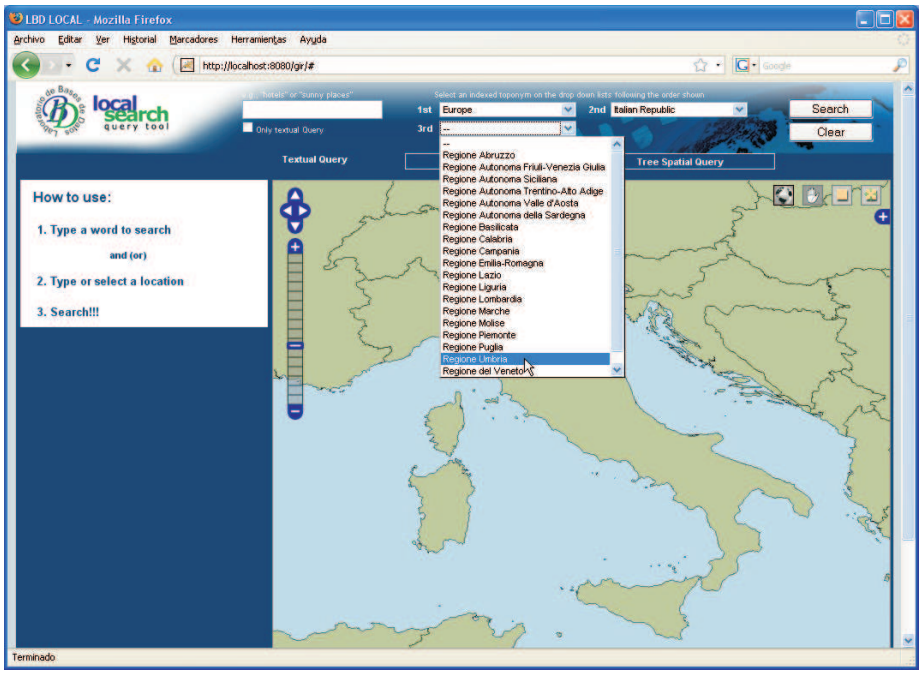


Figura 4.9: Interfaz de usuario de consulta (I).

En el capítulo anterior hemos descrito los tipos de consultas que se pueden resolver con este sistema. Estas consultas tienen dos ámbitos diferentes: un ámbito textual y un ámbito espacial. El especificar uno u otro, o ambos, da lugar a los diferentes tipos de consulta. Por tanto, la interfaz de usuario de consulta debe permitir al usuario indicar ambos ámbitos. En concreto, el ámbito espacial se puede introducir de tres maneras mutuamente excluyentes:

- *Tecleando el nombre de lugar*. En este caso, el usuario introduce el nombre de

lugar en un campo de texto. Este es el método más ineficiente ya que el sistema tiene que obtener todas las geo-referencias asociadas con el nombre de lugar tecleado y éste es un proceso costoso en tiempo.

- *Seleccionando el nombre de lugar en un árbol*. En este caso, el usuario selecciona sucesivamente un continente, un país contenido en ese continente, una región en ese país y un lugar poblado en esa región. Si el usuario quiere especificar un nombre de lugar de nivel más alto que lugar poblado no tiene que rellenar todos los niveles. La operación es muy sencilla e intuitiva gracias a que la interfaz está implementada con un componente desarrollado a medida empleando la tecnología AJAX [Gar05] que permite recuperar en un segundo plano los nombres de lugar del siguiente nivel. Cuando el usuario selecciona un lugar en el componente, el mapa de la parte derecha enfoca automáticamente el lugar seleccionado. En la figura 4.9 mostramos un ejemplo donde el usuario navegó hasta el nivel de Italia y está visualizando las regiones en las que puede centrar el interés de su búsqueda.

- *Dibujando el contexto espacial de interés en el mapa*. El usuario puede navegar empleando el mapa de la parte derecha para seleccionar el contexto espacial de interés. Una vez que lo ha localizado, puede dibujar una ventana de consulta sobre ese contexto. El sistema usará esa ventana de consulta si el usuario no tecleó un nombre de lugar ni seleccionó un lugar en el árbol. En la figura 4.9 mostramos un ejemplo donde el usuario dibujó una ventana de consulta centrada en la zona de Roma en Italia.

En esta figura también se puede observar como se presentan los resultados de la consulta. La consulta realizada tiene una parte textual (*artist painter*) y una parte espacial (indicada mediante la ventana de consulta centrada en la zona de Roma). Los resultados se muestran en la parte izquierda de la pantalla ordenados según su relevancia. Como el *ranking* de resultados puede contener muchos documentos, no se muestran todos en una pantalla sino que se encuentran paginados y se puede navegar por las distintas páginas. Además, en esta navegación nunca se pierde el contexto espacial (el mapa, la ventana de consulta, etc.). La interfaz permite visualizar el contenido de cada documento pinchando en la lupa que se encuentra al lado de cada uno y también permite situarlo en su ámbito espacial (pinchando en el identificador de cada documento se muestran los nombres de lugar que cita y se puede situar cada uno en el mapa).

4.5. Dinámica del sistema

Una vez finalizada la descripción de la estructura estática del sistema, consideramos importante describir también la estructura dinámica del mismo. Esta estructura se

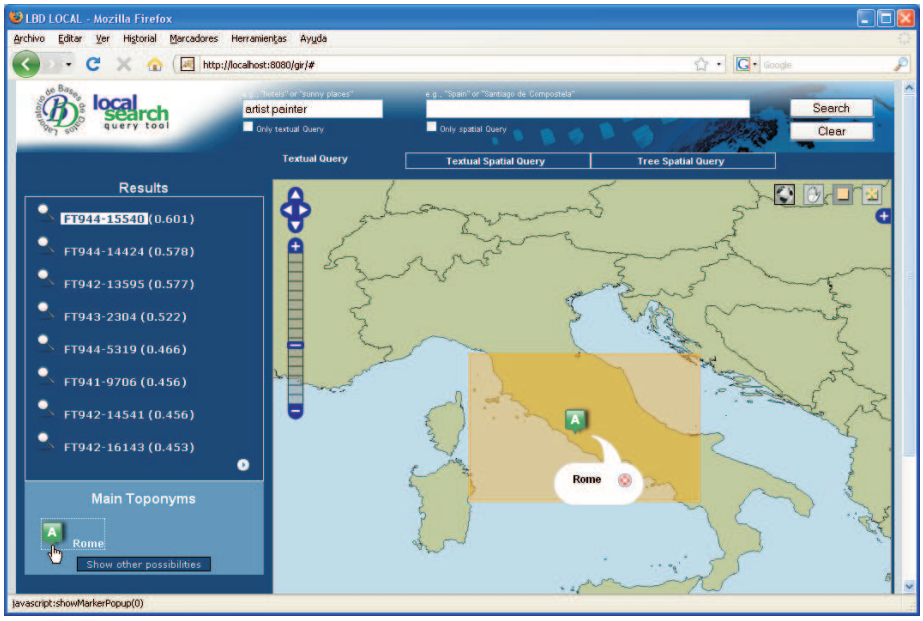


Figura 4.10: Interfaz de usuario de consulta (II).

refleja en forma de interacciones entre los distintos componentes para la realización de tareas en escenario de uso típicos. Entender el flujo de información y el paso de mensajes entre los subsistemas da una visión mucho más general de la arquitectura y permite comprenderla en su totalidad. Como escenarios de uso típicos hemos seleccionado la creación de la estructura de indexación por parte del administrador del sistema y la realización de consultas por parte de los usuarios. En las siguientes secciones describimos estos escenarios empleando para ello diagramas de secuencia.

4.5.1. Creación de la estructura de indexación

El primer escenario de uso típico es la creación de una estructura de indexación a partir de una colección de documentos. En este escenario, además de crear la estructura de indexación, el sistema almacena en base de datos toda la información necesaria para la consulta de los documentos indexados. En la figura 4.11 mostramos un diagrama de secuencia para aclarar la interacción en este escenario de uso.

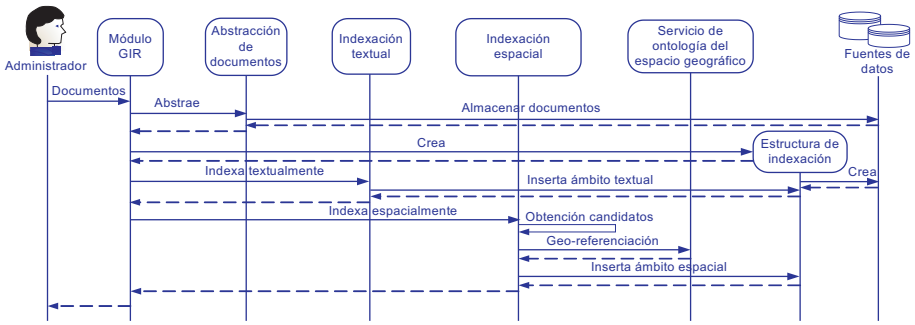


Figura 4.11: Secuencia de la creación de una estructura.

1. En este escenario el usuario administrador interactúa con el sistema mediante la interfaz de usuario de administración. El módulo de recuperación de información geográfica es el encargado de coordinar la ejecución de todos los componentes que intervienen en el flujo de trabajo.
2. La primera tarea de dicho flujo de trabajo es la abstracción de documentos. En esta tarea se preparan los documentos para su procesado en el resto del flujo de trabajo del sistema y se almacenan en base de datos para que puedan ser consultados con posterioridad.
3. La siguiente tareas consiste en la inserción en la estructura de indexación de la información contenida en el texto de los documentos. Por tanto, el módulo

de recuperación de información debe crear previamente dicha estructura de indexación para que se pueda llevar a cabo esta tarea.

4. Entre la indexación textual y la espacial no existe una dependencia de secuencialidad. En la indexación textual se analizan todos los campos de los documentos marcados como indexables textualmente y se crea la componente textual de la estructura de indexación a partir de esa información.

5. Además, en la indexación espacial se analizan todos los campos de los documentos marcados como indexables espacialmente. Dicho análisis consiste en un primer proceso en el que se buscan posibles nombres lugar, lo que denominamos obtención de candidatos. El segundo proceso consiste en geo-referenciar todos esos candidatos. El resultado de este segundo proceso es la información geográfica necesaria para crear la componente espacial de la estructura de indexación.

Si la estructura de indexación fuese híbrida, y no de doble índice como en nuestro prototipo, las tareas de análisis texual y análisis espacial se podrían realizar en paralelo igual que acabamos de describir. Sin embargo, la creación de la estructura se debería realizar en una única tarea.

4.5.2. Realización de una consulta

El escenario de uso más habitual es la realización de consultas por parte de los usuarios. Como hemos hecho constar ya varias veces, la resolución de las consultas depende tanto de cómo sea la estructura de indexación como del tipo de consulta. Sin embargo, para realizar un diagrama de secuencia a alto nivel, que sea válido a nivel de la arquitectura e independiente de la consulta y de la estructura, podemos suponer una consulta genérica con una parte textual y una parte espacial. La figura 4.12 esquematiza el diagrama de secuencia para este escenario de uso que describimos a continuación.

1. El usuario interactúa con el sistema mediante la interfaz de usuario de consulta. Como ya hemos visto, esta interfaz le permite indicar las palabras clave a buscar en la consulta, pero también le permite navegar por el mapa para visualizar el área de interés para la consulta. Esta navegación la representamos mediante la petición de *mapa* que, aunque aparece una única vez, se realiza para cada movimiento en el mapa, para cada ajuste del nivel de zoom, etc. Las peticiones al WMS provocan que este componente tenga que recuperar la información geográfica necesaria de las fuentes de datos.

2. Una vez que el usuario tiene seleccionada el área de interés en el mapa puede realizar una consulta. Esta consulta la recibe el módulo de recuperación de información geográfica que, al igual que en el escenario anterior, es el encargado

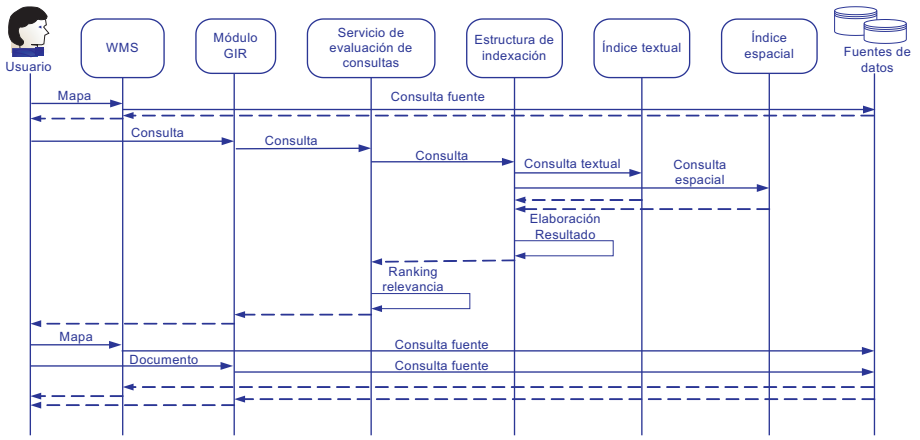


Figura 4.12: Secuencia de una consulta.

de coordinar todos los servicios que ofrece el sistema y, por tanto, conoce en qué componente debe delegar la petición.

3. El componente encargado de resolver las consultas es el servicio de evaluación de consultas. Como ya hemos visto en la sección 4.3.1, este servicio delega en la estructura de indexación la resolución de las consultas.

4. La estructura de indexación divide la consulta en su componente textual, que la resuelve el índice textual, y en su componente espacial, que la resuelve la parte espacial de la estructura. Una vez obtenidos ambos resultados los combina en una única lista de resultados.

5. Esta es la lista que le devuelve como resultado al servicio de evaluación de consultas, que es el encargado de la elaboración del *ranking* de resultados; es decir, de poner los elementos de la lista en orden según su relevancia que representa la importancia del documento para el usuario según la consulta que realizó.

6. Finalmente, el *ranking* de resultados le llega al módulo de recuperación de información geográfica que se los envía al usuario para que pueda analizarlos e interactuar con ellos.

7. El usuario puede consultar cualquiera de los documentos incluidos en el *ranking* de resultados mediante peticiones al módulo de recuperación de información geográfica. Dicho módulo se encarga de recuperar el documento solicitado de la fuente de datos donde se encuentra almacenado.

8. Además, el usuario puede representar los documentos geo-referenciados en el mapa del que dispone en la interfaz de usuario mediante peticiones al WMS. Como vemos en la figura, estos dos últimos tipos de peticiones se realizan de forma asíncrona. Esto es posible debido a que empleamos la tecnología AJAX en la interfaz de usuario. Además, esta tecnología hace que la interfaz sea más amigable y cómoda para el usuario.

4.6. Resumen

En este capítulo hemos presentado nuestra propuesta de arquitectura para sistemas de recuperación de información geográfica y los componentes por los que está formada. En la sección 4.1 mostramos una visión general de nuestra propuesta. En ella la arquitectura está dividida en tres capas que agrupan componentes de funcionalidad relacionada. En primer lugar, en la capa de construcción del índice (ver sección 4.2) se incluyen los componentes involucrados en el flujo de trabajo que permite indexar documentos en la estructura que constituye la base del sistema y almacenarlos en las fuentes de datos para que puedan ser consultados por los usuarios. En segundo lugar, la capa de servicios de procesado (ver sección 4.3) agrupa una serie de servicios independientes que se pueden emplear desde otros componentes de la arquitectura. Además, define un módulo de control que encapsula el acceso a estos servicios independientes y permite encadenarlos para realizar tareas más complicadas. La última capa es la de interfaces de usuario (ver sección 4.4) donde se definen las dos interfaces de usuario que permiten la interacción con el sistema tanto para su administración como para la realización de consultas. Finalmente, la sección 4.5 completa la descripción de la arquitectura detallando el funcionamiento del sistema en escenarios de uso típicos: creación de la estructura de indexación y ejecución de una consulta de usuario.

Capítulo 5

SpatialWT: indexación del espacio basada en *Wavelet Trees*

El desarrollo de estructuras de indexación que permitan recuperar objetos espaciales de manera eficiente ha sido uno de los principales temas de interés en el área de los sistemas de información geográfica en las últimas décadas. La mayoría de estas estructuras han sido diseñadas pensando en las características específicas de la memoria secundaria. Sin embargo, en los últimos años el precio de la memoria principal se ha reducido considerablemente y, por tanto, hoy en día es posible almacenar índices espaciales completos en memoria principal sin necesidad de acceder a disco. Además, la aparición de los sistemas de recuperación de información geográfica ha puesto de manifiesto las características especiales de la información geográfica que manejan este tipo de sistemas. Dichas características definen nuevos requisitos que se deben tener en cuenta a la hora de diseñar estructuras de indexación espacial que se vayan a emplear en sistemas GIR.

En este capítulo presentamos una estructura de indexación espacial para memoria principal que tiene en cuenta las características particulares de la información geográfica gestionada habitualmente en los sistemas GIR. Esta estructura se basa en el *wavelet tree*, una estructura compacta para representar secuencias que ya hemos introducido en el capítulo 2. En primer lugar, ampliamos la motivación de nuestra propuesta de estructura de indexación espacial en la sección 5.1. A continuación, describimos dicha estructura en la sección 5.2. La sección 5.3 presenta los experimentos que hemos realizado para comparar nuestra propuesta frente a otras técnicas de indexación espacial clásicas que hemos introducido en el estudio del estado del arte de los sistemas de información geográfica (ver sección 2.2.4). Finalmente, cerramos el capítulo con un resumen de las ideas más importantes presentadas en él en la sección 5.4.

5.1. Motivación

La popularidad de los sistemas de información geográfica (GIS) [Wor04] ha aumentado considerablemente en los últimos años. Una de las principales causas de este aumento de popularidad son las mejoras recientes en el hardware que han permitido que el desarrollo de este tipo de sistemas sea abordable por muchas organizaciones. Además, esta tecnología se emplea cada vez más en un mayor número de áreas distintas, como la generación de cartografía, la planificación urbanística, el *marketing* o la arqueología.

Una de las características más destacables de los sistemas de información geográfica es la necesidad de gestionar enormes cantidades de información. Esto ha motivado que uno de los temas de investigación más importantes en el área sea el diseño de estructuras de indexación que permitan un acceso eficiente a la información. Como veíamos en la sección 2.2.4, las estructuras que se han ido proponiendo durante los últimos años para lograr este objetivo se pueden clasificar de manera general en métodos de acceso a puntos (PAMs, del inglés *Point Access Methods*) y métodos de acceso espacial (SAMs, del inglés *Spatial Access Methods*). Los métodos del primer grupo se caracterizan por mejorar el acceso a colecciones de datos formadas por puntos espaciales mientras que los métodos del segundo grupo permiten indexar colecciones de todo tipo de objetos geográficos (puntos, líneas, polígonos, etc.).

La mayoría de los métodos propuestos, tanto en una categoría como en la otra, han sido diseñados para su empleo en memoria secundaria. Esto se debe fundamentalmente a razones históricas. Hace años, las memorias eran pequeñas y caras, y esto convertía en inviable el pensar en el diseño de estructuras de indexación espacial que se pudiesen emplear en memoria principal. Sin embargo, en los últimos años, el precio de las memorias se ha reducido considerablemente, aumentando su tamaño también de manera notable. Por tanto, hoy en día es posible diseñar índices espaciales completos que puedan operar en memoria principal. Para ello, a la hora de diseñar nuevas estructuras de indexación no se debe pensar en optimizar tan sólo la eficiencia de las consultas sino también el espacio necesario para el almacenamiento de la estructura.

Por otra parte, en la motivación de la estructura que proponemos también influyen las características especiales de la información geográfica que gestionan habitualmente los sistemas GIR. En primer lugar, en este tipo de sistemas, al igual que en los sistemas de IR, la información es semi-estática [BYRN99]. Esto permite pensar en el diseño de estructuras de indexación espacial que puedan aprovechar el conocer *a priori* la distribución en el espacio de los objetos que se van a indexar. Además, muchas veces en este tipo de sistemas es suficiente con indexar las coordenadas de los lugares mencionados en los documentos. Esto aumenta el interés en el desarrollo de nuevos métodos de acceso a puntos. Finalmente, el proceso de resolución de consultas en este tipo de sistemas es complejo ya que las consultas tienen aspectos textuales y espaciales que se deben combinar para obtener los resultados. Por tanto, la eficiencia a la hora

de resolver consultas es también un requisito fundamental de estas estructuras. Las estructuras compactas en memoria principal tienen ventaja para satisfacer este requisito ya que el acceso a disco es varios órdenes de magnitud más costoso que el acceso a memoria principal.

En este capítulo presentamos un nuevo método de acceso a puntos optimizado para tener en cuenta las características que acabamos de mencionar y que almacena tanto el índice como la colección de puntos en una estructura compacta. Esta estructura presenta una buena relación entre el espacio necesario para su almacenamiento y la eficiencia de las búsquedas, lo cual la hace adecuada para su empleo en memoria principal.

En los últimos años, la idea de almacenar de forma conjunta los datos y la estructura de indexación que permite un acceso eficiente a los mismos se ha empleado con éxito en varios campos de investigación. Estas estructuras, que permiten almacenar los datos y el propio índice de manera conjunta, se denominan auto-índices. Por ejemplo, en [BFLN08] se presenta una aproximación para indexar documentos empleando *wavelet trees*. Originalmente, el *wavelet tree* [GGV03] se diseñó como un auto-índice organizado en forma de árbol binario para representar e indexar secuencias. En nuestro trabajo analizamos esta estructura y la adaptamos a las características especiales de la información geográfica gestionada habitualmente en sistemas GIR.

5.2. Descripción de la estructura

5.2.1. Construcción de la estructura

Podemos formalizar el problema que tratamos de abordar de la siguiente manera. Supongamos que nuestra colección de objetos está formada por N puntos, $P = P_1 \dots P_N$, cada uno definido por dos coordenadas (latitud y longitud) que identifican su posición en el espacio con respecto a un sistema de referencia. Podemos asumir que estos puntos están distribuidos en una matriz de $N \times N$ donde sólo hay un punto en cada fila y columna. Esto no supone una restricción muy importante ya que, si dos puntos tienen la misma coordenada, podemos ordenarlos arbitrariamente y asignarles filas o columnas consecutivas en la matriz. En la figura 5.1 mostramos el proceso de creación de la matriz representativa de los puntos de entrada. Es importante observar que lo único que mantiene la matriz representativa es el orden, no hay distancias ni proporciones. Esto es lo que nos permite crear una matriz que represente cualquier conjunto de puntos aunque algunos de ellos tengan coordenadas repetidas.

El *wavelet tree* es una estructura compacta que nos permite almacenar en poco espacio la matriz que acabamos de construir y con unas propiedades interesantes para realizar búsquedas en ella. Considerando una matriz de $N \times N$, podemos construir un *wavelet tree* con $\lceil \log_2 N \rceil$ niveles y N bits por nivel. Este *wavelet tree* nos permite

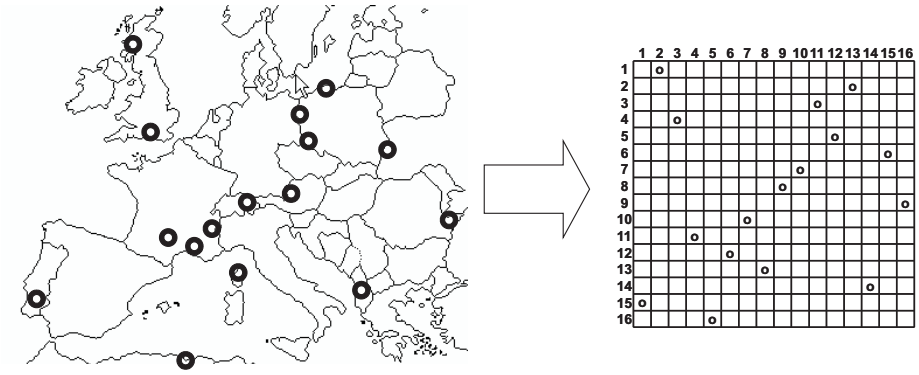


Figura 5.1: Creación de la matriz representativa de los datos.

almacenar la permutación del orden de los puntos en una dimensión (por ejemplo, longitud) al orden de los puntos en la otra dimensión (por ejemplo, latitud). Sean $X = P_{X_1} \dots P_{X_N}$ e $Y = P_{Y_1} \dots P_{Y_N}$ dichas permutaciones donde los puntos están ordenados por sus longitudes y latitudes respectivamente. Por ejemplo, en la figura 5.1 podemos nombrar los puntos de izquierda a derecha (es decir, P_i es el i-ésimo punto empezando a contar por la izquierda). Por tanto, la permutación de las longitudes la podemos escribir como $X = P_1 \dots P_{16}$. En este caso, la otra permutación la escribimos como $Y = P_2 P_{13} P_{11} \dots P_1 P_5$. Estas permutaciones nos indican, por ejemplo, que el punto P_1 es el primero en el orden de las longitudes y el penúltimo en el orden de las latitudes (es decir, el situado más a la izquierda y el penúltimo más abajo).

La raíz del *wavelet tree* que nos permite almacenar estas permutaciones es un bitmap $B = b_1 \dots b_N$ de la misma longitud que el conjunto de puntos (es decir, de N posiciones). Cada posición i del bitmap representa al punto en la i-ésima posición de la primera permutación (es decir, longitud). Siguiendo con el ejemplo, la posición 1 representa el punto $P_{X_1} = P_1$, la 2 representa el punto $P_{X_2} = P_2$, etc. El valor que se almacena en cada posición puede ser $b_i = 0$ si $P_{X_i} \in P_{Y_1} \dots P_{Y_{N/2}}$ o $b_i = 1$ si $P_{X_i} \in P_{Y_{N/2+1}} \dots P_{Y_N}$. Es decir, si el punto se encuentra en la primera mitad de la segunda permutación se almacena un 0 y en caso contrario se almacena un 1. En esta estructura cada nodo indexa la mitad de símbolos que su nodo padre. La secuencia de puntos marcados con un 1 en el vector se procesan en el hijo derecho del nodo, mientras que aquellos marcados con 0 se procesan en el hijo izquierdo del nodo. Este proceso se repite recursivamente en cada nodo hasta que se alcanzan los nodos hoja donde la secuencia de símbolos indexados se corresponde con la permutación en la segunda dimensión (es decir, latitud). En la figura 5.2 mostramos el *wavelet tree* construido con el ejemplo de la figura 5.1. Para aportar claridad al ejemplo mostramos para cada

elemento del bitmap a qué posición de la segunda permutación se corresponde (estas posiciones las tachamos para indicar que realmente no almacenamos esos valores).

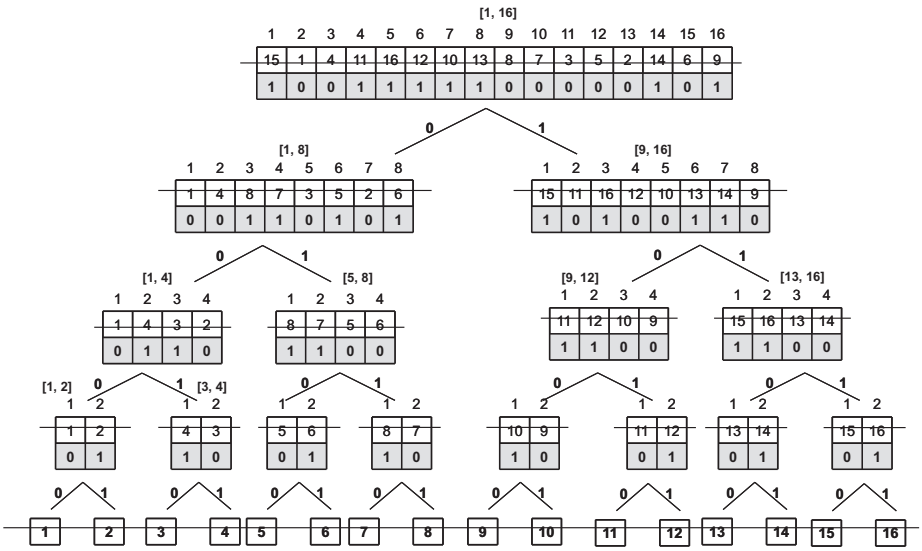


Figura 5.2: Creación del wavelet tree (sólo se almacenan los datos sombreados).

Para entender el proceso de construcción del *wavelet tree* de la figura vamos a tomar como ejemplo el punto P_8. Como hemos nombrado los puntos de izquierda a derecha, este punto se corresponde con la octava posición en el orden de las longitudes (es decir, se encuentra en la octava columna de la matriz) y, como se puede observar en la figura 5.1, se corresponde con la decimotercera posición en el orden de las latitudes (ya que se encuentra en la decimotercera fila de la matriz). Por tanto, en el primer nivel del *wavelet tree*, donde tenemos un único nodo que cubre todas las filas (desde la 1 a la 16), el punto P_8 se encuentra en la posición 8 ya que los puntos en este nodo siguen el orden de las longitudes. Si nos fijamos en la posición 8 del nodo raíz vemos que la fila correspondiente para ese nodo (el valor tachado) es 13 (es decir, la decimotercera fila o el valor decimotercero en el orden de las latitudes). Además, el valor almacenado en la posición 8 del bitmap es un 1 ya que el punto P_8 se procesa en el hijo derecho del nodo. Esto se debe a que el segundo nivel del árbol tiene dos nodos: el izquierdo, que cubre el rango de filas [1, 8], y el derecho, que cubre el rango [9, 16] donde se encuentra contenido el valor 13. En concreto, el punto P_8 se encuentra en la posición 6 del hijo derecho del nodo raíz debido a que en el primer nivel hay otros 5 puntos situados antes que él que se almacenan también en el mismo nodo (en las posiciones 1, 4, 5, 6 y 7 del bitmap también se almacena el valor 1 que indica que esos puntos se

almacenan en el hijo derecho). Es decir, el *wavelet tree* mantiene el orden de los puntos entre niveles. Repitiendo el proceso, este nodo tiene dos hijos de los cuales el izquierdo cubre el rango de filas [9, 12] y el derecho cubre el rango de filas [13, 16]. El punto P_8 se almacena en el hijo derecho y, por tanto, en la posición 6 del bitmap del hijo derecho del nodo raíz se almacena un 1. El árbol completo se construye repitiendo este proceso hasta alcanzar los nodos hoja para todos los puntos de la colección.

5.2.2. Resolución de consultas

Como hemos explicado en la sección 2.1.4, el *wavelet tree* emplea dos operaciones básicas para poder descender y ascender en el árbol. Estas dos operaciones, denominadas respectivamente *rank* y *select*, se pueden realizar en tiempo constante empleando unas estructuras auxiliares que ocupan un un 37,5 % de espacio adicional sobre el tamaño del *wavelet tree*. En realidad este es el espacio requerido en la implementación de la propuesta de Jacobson [Jac89] pero existen otras implementaciones [GGMN05, OS07] de estas estructuras auxiliares que ocupan menos espacio manteniendo una buena eficiencia e incluso que se pueden parametrizar para que ocupen más o menos espacio en función de la eficiencia requerida. La operación $rank_1(B,i)$ devuelve el número de bits establecidos a uno en el prefijo $B[1,i]$. Por ejemplo, si B es el bitmap del nodo raíz de la figura 5.2, $rank_1(B,8) = 6$. Este resultado lo podemos emplear para conocer que existen 5 puntos antes que P_8 que almacenan un 1 en el bitmap y, por tanto, que están en el hijo derecho del nodo raíz antes que P_8. La operación dual a *rank* es *select*. $select_1(B,j)$ permite obtener la posición del j-ésimo bit establecido a uno en B. Por ejemplo, siendo B de nuevo el bitmap del nodo raíz $select_1(B,6) = 8$. Este resultado lo podemos emplear para conocer que el sexto punto que se procesa en el nodo derecho del segundo nivel se encuentra en la posición 8 del nodo raíz (es decir, podemos conocer que se trata del punto P_8).

Por tanto, la estructura que acabamos de describir nos permite obtener de forma sencilla en qué posición de la segunda dimensión se encuentra un punto del cual se sabe su posición en la primera dimensión, simplemente descendiendo en el árbol. Para acceder desde una posición determinada de un nodo del árbol al siguiente nivel se emplea la operación *rank* y el valor almacenado en esa posición. El bit b_i del bitmap de un nodo indica si el punto correspondiente se indexa en el hijo izquierdo ($b_i = 0$) o en el hijo derecho ($b_i = 1$). Además, $rank_{b_i}(B,i)$ nos indica la posición en la que se almacena dicho punto en el nodo hijo. Este proceso se repite hasta que se alcanza un nodo hoja donde la posición señala el orden en la otra dimensión. El algoritmo 1 muestra el pseudocódigo de este proceso. Por ejemplo, para conocer en qué fila se encuentra el punto de la columna 6 se mira el valor que hay en la posición 6 del nodo raíz y se accede a la posición $rank_1(B,6) = 4$ del nodo derecho (se calcula $rank_1$ porque en la posición 6 hay un 1). Este proceso se repite en todos los niveles del árbol hasta alcanzar el nodo hoja donde se obtiene la posición 12 que, como se puede comprobar en la matriz de la figura 5.1, es la fila correspondiente a la columna 6.

Algoritmo 1 Obtención de la fila correspondiente a una columna.

Entrada: c, una columna
Salida: f, la fila del punto correspondiente

```
pos ← c
nodoActual ← nodoRaiz
mientras nodoActual no es hoja hacer
    valor ← wt[nodoActual][pos]
    pos ← rank_valor(nodoActual, pos)
    si valor = 0 entonces
        nodoActual ← nodoActual.hijoIzquierdo
    si no
        nodoActual ← nodoActual.hijoDerecho
    fin si
fin mientras
f ← pos
devolver f
```

De manera similar, podemos conocer en qué posición de la primera permutación se encuentra un punto del cual se sabe su posición en la otra dimensión simplemente ascendiendo en el árbol. Para acceder al nivel anterior en el árbol desde un nodo se emplea la operación *select* y el valor que etiqueta la rama del árbol (es decir, el valor que da acceso a ese nodo). Como nuestra estructura es un árbol binario perfecto, es muy sencillo conocer en cada nivel del árbol si el nodo actual es un hijo derecho o si es un hijo izquierdo. Por este motivo, no tenemos que almacenar las etiquetas de las ramas. Mostramos el pseudocódigo de este procedimiento en el algoritmo 2. En el ejemplo, si queremos conocer en qué columna se encuentra el punto de la fila 13 se comprueba si el nodo que contiene la posición 13 en el último nivel del árbol es un hijo derecho o si es izquierdo (por claridad en la explicación asumimos la existencia del último nivel ficticio donde cada nodo almacena un único punto). Como es un hijo izquierdo se accede a la posición $select_0(B,1) = 1$ del nodo padre (el 1 que empleamos en el cálculo se debe a que 13 se almacena en la primera, y única, posición del nodo correspondiente). En el siguiente nivel se calcula $select_0(B,1) = 3$ y luego $select_1(B,3) = 6$ (1 por ser un hijo derecho y 3 por ser el valor calculado en el paso anterior) y así hasta el último nivel donde se obtiene la posición $select_1(B,6) = 8$ que nos dice que el punto se encuentra en la columna 8. En la implementación real, donde no existe el último nivel ficticio, es fácil empezar el algoritmo desde el nivel anterior empleando el valor almacenado en la posición que correspondería en caso de que los puntos en ese nodo estuviesen ordenados y comprobando que, en tal caso, las posiciones impares almacenan un 0 y las pares un 1.

Sin embargo, para poder resolver consultas espaciales de tipo región empleando este índice (es decir, para obtener los puntos contenidos en una ventana o región

Algoritmo 2 Obtención de la columna correspondiente a una fila.

Entrada: f, una fila
Salida: c, la columna del punto correspondiente

```
pos ← 1
nodoActual ← nodosHoja[f]
mientras nodoActual no es raiz hacer
  nodoPadre ← nodoActual.padre
  si nodoActual es hijoIzquierdo entonces
    pos ← select_0(nodoPadre, pos)
  si no
    pos ← select_1(nodoPadre, pos)
  fin si
  nodoActual ← nodoPadre
fin mientras
c ← pos
devolver c
```

rectangular del espacio) necesitamos tres estructuras auxiliares: dos vectores con las coordenadas ordenadas en cada dimensión y los identificadores de los puntos almacenados en el mismo orden que uno de esos dos vectores. Esto se debe a que el *wavelet tree* sólo almacena la relación entre las permutaciones, sin tener para nada en cuenta las relaciones en el mundo real (por ejemplo, las coordenadas de los puntos o las distancias entre ellos). Sin embargo, las ventanas de consulta se definen empleando dos puntos (por ejemplo, el de la esquina inferior izquierda y el de la esquina superior derecha de la ventana) que se especifican empleando coordenadas espaciales. Por tanto, los vectores con las coordenadas se emplean para traducir la ventana de consulta a dos rangos que indican respectivamente las columnas y las filas donde se encuentran los puntos que pueden formar parte de la solución de la consulta.

Una vez realizada la traducción de la consulta, el rango de columnas (longitudes) indica el rango de posiciones válidas en el nodo raíz del *wavelet tree*. El descenso en el árbol se realiza tal y como acabamos de explicar para una posición aunque se optimiza considerablemente teniendo en cuenta dos principios básicos del *wavelet tree*. En primer lugar, que los puntos consecutivos en un nodo se mantienen consecutivos en los nodos hijo. Por tanto, en el descenso, por cada conjunto de posiciones consecutivas, sólo se necesitan cuatro operaciones de *rank* (dos por cada uno de los dos nodos hijo: una para la primera posición del conjunto de puntos consecutivos y otra para la última). En segundo lugar, que el rango de filas (latitudes) se puede emplear para podar el descenso en el árbol. Cada nodo en el *wavelet tree* contiene puntos cuya fila (es decir, su orden en el vector de latitudes ordenadas) se encuentra en un rango determinado. Si ese rango no interseca con el rango de filas que indica la consulta, el algoritmo no tiene que continuar por esa rama. En el algoritmo 3 mostramos el pseudocódigo de este

proceso de descenso en el árbol.

Algoritmo 3 Descenso en el árbol de un rango de columnas.

Entrada: minC, maxC, minF, maxF; límites de los rangos de columnas y filas
Salida: filas, filas de los puntos que satisfacen la consulta

```
nodoActual ← nodoRaiz
minRango ← minC
maxRango ← maxC
si nodoActual es hoja entonces
  añadir minC a filas
si no
  si nodoActual.hijoIzquierdo.rango ∩ [minF, maxF] entonces
    llamada recursiva con:
    nodoActual ← nodoActual.hijoIzquierdo
    minRango ← rank_0(nodoActual, minRango − 1) + 1
    maxRango ← rank_0(nodoActual, maxRango)
  fin si
  si nodoActual.hijoDerecho.rango ∩ [minF, maxF] entonces
    llamada recursiva con:
    nodoActual ← nodoActual.hijoDerecho
    minRango ← rank_1(nodoActual, minRango − 1) + 1
    maxRango ← rank_1(nodoActual, maxRango)
  fin si
fin si
devolver filas
```

En la figura 5.3 mostramos el *wavelet tree* del ejemplo que venimos empleado en este capítulo con los vectores de coordenadas ordenados y los identificadores (IDs(X) e IDs(Y)). En cuanto a estos últimos, sólo necesitamos uno de los dos vectores. La elección de uno u otro condiciona el algoritmo de resolución de consultas, y tiene sus ventajas e inconvenientes como veremos a continuación. La figura representa la resolución de un ejemplo de consulta de tipo región donde la ventana se puede definir como $q =< (27'53, 15'75), (30'71, 19) >$. Esta consulta se traduce al rango de columnas de interés [6, 10] y al rango de filas de interés [9, 14]. Si nos fijamos en el vector de longitudes ordenadas (es decir, en el vector *X*), la primera longitud que satisface el ser mayor o igual que 27,53 es 28,01, que se encuentra en la posición 6 (límite inferior del rango). De igual modo, la última longitud que satisface el ser menor o igual que 30,71 es 29,05 que se encuentra en la posición 10 (límite superior del rango). Estos valores se pueden obtener de manera eficiente empleando búsquedas binarias. La traducción al rango de filas de interés se realiza de manera análoga. El algoritmo de resolución de consultas comienza con el descenso en el árbol. Como ya hemos mencionado, sólo es necesario calcular los *ranks* del principio y del final

de las posiciones consecutivas. De este modo, el proceso se iniciaría calculando $rank_0(B,6-1)+1=3$, $rank_0(B,10)=4$, $rank_1(B,6-1)+1=4$ y $rank_1(B,10)=6$. En realidad sólo dos de ellos ya que $rank_0$ y $rank_1$ de la misma posición son dependientes y uno se puede obtener en función del otro ($rank_0(B,i)+rank_1(B,i)=i$). Por tanto, en el segundo nivel los rangos de interés son el [3, 4] en el hijo izquierdo y el [4, 6] en el derecho. Sin embargo, el hijo izquierdo sabemos que no puede contener soluciones ya que cubre el rango de filas [1, 8] que no interseca con las filas de interés para la consulta [9, 14]. Por este motivo el algoritmo no desciende por esa rama. Este proceso se repite en todo el descenso hasta alcanzar los nodos hoja.

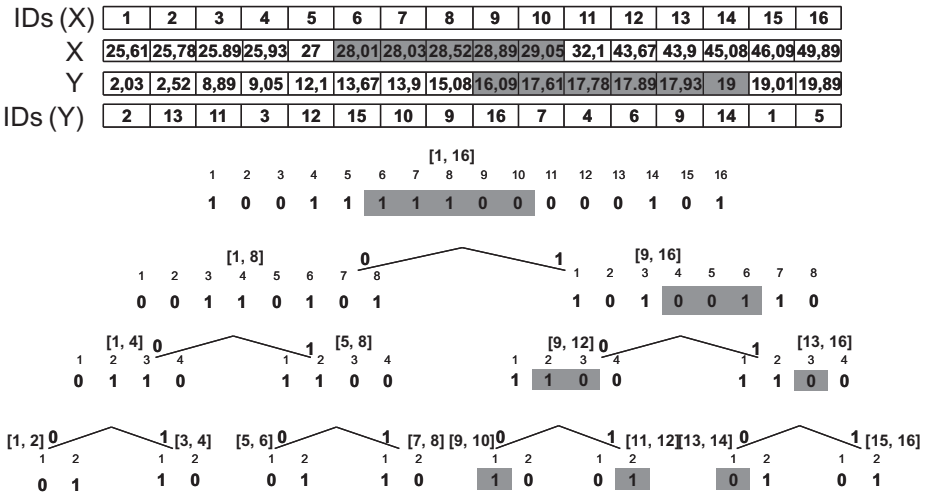


Figura 5.3: Resolución de consultas empleando el wavelet tree.

Una vez que se alcanzan los nodos hoja el algoritmo es diferente según el orden en el que se almacenen los identificadores. Si se almacenan en el mismo orden que los nodos hoja (es decir, en el orden de las latitudes), al alcanzar el último nivel del árbol ya sabemos los identificadores que forman parte del resultado de la consulta. Esta versión de la estructura es más sencilla algorítmicamente y como veremos en la sección 5.3 también es más eficiente, a pesar de que tiene la desventaja de tener que alcanzar siempre los nodos hoja para obtener el resultado. Por el contrario, si la estructura almacena los identificadores en el mismo orden que el nodo raíz (es decir, en el orden de las longitudes) el algoritmo es más complicado ya que una vez se comprueba que es una latitud válida hay que volver a ascender para obtener el identificador. Esta implementación tiene la ventaja de que dicha comprobación se puede cumplir en niveles altos del árbol y, por tanto, se puede detener el descenso y comenzar el ascenso. Sin embargo, los experimentos realizados demuestran que la

eficiencia de esta versión es inferior.

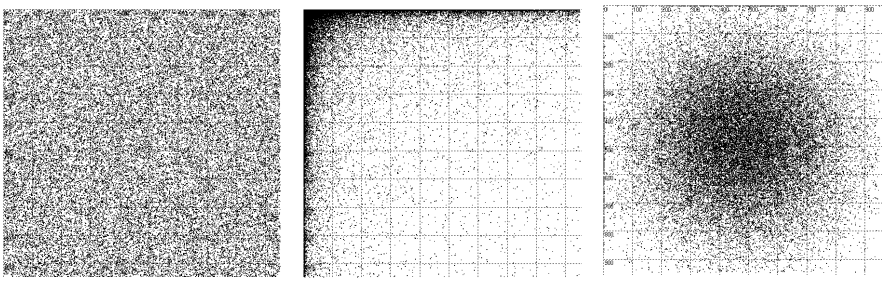
5.3. Experimentos

5.3.1. Descripción de los experimentos

En esta sección presentamos los experimentos que hemos realizado para comparar la eficiencia de nuestra estructura frente a otros índices espaciales clásicos. Esta comparación tiene dos partes. En primer lugar, comparamos los requisitos de espacio de las estructuras y, en segundo lugar, su eficiencia a la hora de resolver consultas espaciales. Como veremos, los resultados demuestran que nuestra estructura presenta una muy buena relación entre el espacio necesario para almacenarla y la eficiencia a la hora de resolver las consultas.

En los resultados que vamos a presentar a continuación comparamos cinco estructuras espaciales. Las dos primeras se corresponden con las variantes de nuestra estructura de indexación que hemos presentado en la sección 5.2. En la primera de ellas, denominada DPW-tree (del inglés, *down point wavelet tree)*, los identificadores de los puntos se almacenan siguiendo la permutación que representan los nodos hoja y, por tanto, sólo es necesario descender en el árbol para obtener los identificadores de los puntos. En la segunda, denominada UPW-tree (del inglés, *up point wavelet tree*), los identificadores se almacenan siguiendo la permutación representada en el nodo raíz y, por tanto, una vez que se comprueba en el descenso que un punto pertenece al resultado hay que volver a ascender en el árbol para obtener su identificador. Las estructuras tercera y cuarta son dos variantes del R-tree clásico adaptadas para trabajar en memoria principal. Aunque el R-tree se encuentra en la categoría de métodos de acceso espacial en general y no está optimizado para la indexación de puntos, es la estructura más empleada en los sistemas de información geográfica que se desarrollan hoy en día y, por tanto, es importante ver cuánto se podrían beneficiar dichos sistemas empleando nuestra estructura. Las dos variantes del R-tree que empleamos son el R*-tree, que es una versión dinámica de la estructura diseñada para minimizar el número de solapamientos mediante borrados e reinserciones, y el STR R-tree, que es una versión estática de la estructura y, por tanto, puede realizar una partición del espacio mucho más favorable. En [Had09] pueden encontrarse más detalles acerca de las implementaciones de estas dos estructuras que empleamos en nuestros experimentos. Por último, la quinta estructura es un K-d-tree que representa los métodos de acceso a puntos. La variante de K-d-tree que hemos seleccionado es posiblemente la más eficiente ya que está optimizada para situaciones en las que se conoce *a priori* el conjunto de puntos a indexar. Los detalles de implementación de la versión que empleamos se describen en [Tag09]. Aunque es una implementación para Matlab en el mismo sitio de descargas está disponible el código fuente en C++ que es que empleamos en nuestros experimentos.

Las colecciones de prueba sobre las que vamos a construir los índices espaciales están formadas por datos de dos tipos: sintéticos y reales. En primer lugar, empleamos tres colecciones de datos generados de manera sintética. En la primera de ellas los puntos se encuentran distribuidos uniformemente en el espacio y realizamos pruebas con cuatro tamaños de colección con 2^{19}, 2^{20}, 2^{23} y 2^{24} puntos. Las otras dos colecciones sintéticas contienen un millón de puntos generados empleando una distribución Zipf (tamaño del mundo = 1000×1000 y $\rho = 1$) en un caso y una distribución Gauss (tamaño del mundo = 1000×1000, media = 500 y desviación típica = 200) en el otro. En la figura 5.4 mostramos ejemplos de estas tres distribuciones.

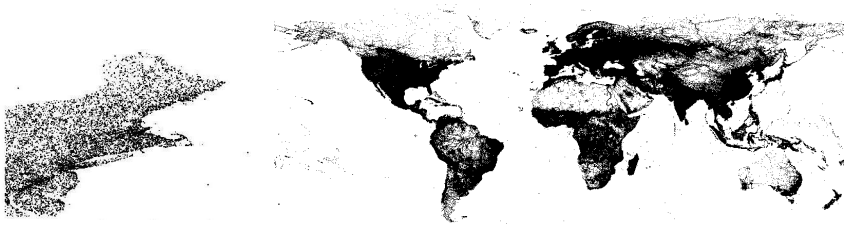


(a) Distribución uniforme (b) Distribución de Zipf (c) Distribución de Gauss

Figura 5.4: Colecciones de puntos sintéticas.

En segundo lugar, también realizamos pruebas con dos colecciones de datos reales. La primera de ellas contiene 123.593 puntos correspondientes a direcciones postales de Nueva York, Filadelfia y Boston. Esta colección, denominada *NE dataset*, se encuentra disponible en el portal dedicado al R-tree [The08]. La segunda colección real contiene 2.693.569 puntos correspondientes a lugares poblados de todo el mundo disponibles en el *gazetteer* Geonames [Geo07]. En la figura 5.5 mostramos cómo se encuentran distribuidos los datos en estas dos colecciones.

Para la realización de los experimentos, además de las estructuras de indexación y de las colecciones de puntos, necesitamos un conjunto de ventanas de consulta (es decir, de regiones donde buscar puntos que se encuentren en ellas). Para construir estas ventanas empleamos el mismo algoritmo que en los experimentos que hemos presentado en la sección 3.4. Como ya hemos explicado, este algoritmo basado en los experimentos del R*-tree, nos permite generar ventanas uniformemente distribuidas en el espacio y cuya proporción se determina de tal forma que el ratio entre la *extensión en x* y la *extensión en y* varía de manera uniforme entre 0,25 y 2,25.



(a) NE dataset (b) Geonames

Figura 5.5: Colecciones de puntos reales.

5.3.2. Comparación de espacio

El espacio necesario para almacenar las estructuras de indexación espacial ha sido un factor muy poco valorado a la hora de diseñarlas ya que la mayoría de estas estructuras estaban pensadas para trabajar en disco. Recientemente, con el crecimiento de las memorias principales, algunas de las estructuras clásicas se adaptaron para trabajar en memoria principal pero sin preocuparse demasiado del tamaño que ocupan. Por este motivo, creemos que el desarrollo de estructuras que presenten una buena relación entre el espacio que ocupan y la eficiencia a la hora de resolver consultas es un tema de especial interés hoy en día en el área de los GIS.

La estructura que proponemos, en sus dos variantes, necesita almacenar las coordenadas de los N puntos (dos vectores de N elementos que son números en punto flotante de 8 bytes), los identificadores (un vector de N elementos que son números enteros de 4 bytes) y el *wavelet tree*. El *wavelet tree* es una estructura muy compacta que sólo necesita $N \times \lceil \log_2 N \rceil$ bits (se necesitan N bits por nivel, un bit por punto y hay $\lceil \log_2 N \rceil$ niveles). Además, para poder realizar las operaciones de *rank* y *select* en tiempo constante se necesitan unas estructuras auxiliares que ocupan un 37,5 % de espacio adicional sobre el tamaño del *wavelet tree* [GGMN05]. Es decir, la estructura completa ocupa $20 \times N + (N \times \lceil \log_2 N \rceil \times 1,375)/8$ bytes.

En cuanto al espacio necesario para la construcción de un R-tree sobre una colección de N puntos, lo podemos estimar suponiendo un factor de ramificación medio (M). Además, para ser justos en la comparación calculamos el espacio del R-tree suponiendo que la estructura almacena en los nodos hoja únicamente los identificadores de los puntos (es decir, realizamos una distinción entre nodos hoja y nodos no hoja, donde se almacenan MBRs), y que la estructura almacena los nodos de manera contigua en memoria sin pérdida de espacio. Es decir, asumimos que los nodos están completamente llenos aunque en realidad en las versiones dinámicas del R-tree, como el R*-tree, se estima la carga media de un nodo en torno al 70 % (en versiones estáticas, como el STR R-tree, la carga media sí está próxima al 100 %). Por tanto, el

Tabla 5.1: Comparación del espacio requerido por los índices espaciales.

Estructura	Espacio total	Espacio por punto
PW-tree	$20 \times N + (N \times \lceil \log_2 N \rceil \times 1,375)/8$	23.69
R-tree	$20 \times N + 36 \times N/(M-1)$	21.24
K-d-tree	$20 \times N + 16 \times (2^h - 1 + (N \bmod 2^{\lfloor \log_2 N \rfloor}))$	36.00

espacio necesario para almacenar las hojas es $4 \times N$ bytes, y sumándole la tabla con las coordenadas de los puntos el espacio asciende a $20 \times N$ bytes. Cada hoja cuesta un MBR y un puntero en su nodo padre, lo cual requiere 36 bytes. Como en total en el árbol hay $N/(M-1)$ nodos, el espacio total necesario para el almacenamiento del R-tree es $20 \times N + 36 \times N/(M-1)$ bytes. En nuestros experimentos el factor de ramificación con el que presentan mejores resultados las variantes del R-tree es 30.

Finalmente, un K-d-tree para indexar N puntos tiene una altura $h = \lceil \log_2 N \rceil$ y $2^h - 1 + (N \bmod 2^{\lfloor \log_2 N \rfloor})$ nodos, donde cada nodo necesita 16 bytes (un número en punto flotante y dos punteros). Al igual que en el caso del R-tree, hay que tener en cuenta los $20 \times N$ bytes de la tabla de puntos.

En la tabla 5.1 mostramos la comparativa del espacio requerido por las diferentes estructuras de indexación espacial. Como nuestras dos alternativas ocupan exactamente el mismo espacio mostramos sus datos en una única fila bajo el nombre de PW-tree. Del mismo modo, agrupamos en una única fila los datos de las dos versiones del R-tree. Las columnas de resultados muestran el espacio teórico total de cada estructura y el espacio necesario por cada punto (este espacio lo indicamos en $bytes/punto$). En el caso del R-tree realizamos este cálculo asumiendo un factor de ramficiación $M = 30$. La principal conclusión a la vista de los resultados es que nuestra estructura necesita mucho menos espacio que el K-d-tree y que es comparable en cuanto a espacio con una versión del R-tree diseñada específicamente para almacenar puntos.

5.3.3. Comparación de tiempo

Para realizar la comparación de tiempos tenemos en cuenta dos parámetros que pueden influir en las pruebas: la selectividad de las consultas y el tamaño de las colecciones. La selectividad de las consultas viene determinada por el tamaño de las ventanas empleadas. En nuestros experimentos creamos ventanas de cuatro tamaños diferentes que representan el 0,01 %, el 0,1 %, el 1 % y el 10 % del área total del espacio donde se encuentran representados los puntos. En la figura 5.7 mostramos varias gráficas donde se puede observar la influencia de estos factores en el tiempo necesario para resolver las consultas. Para mejorar la presentación de los resultados en

estas gráficas representamos el eje X en escala logarítmica.

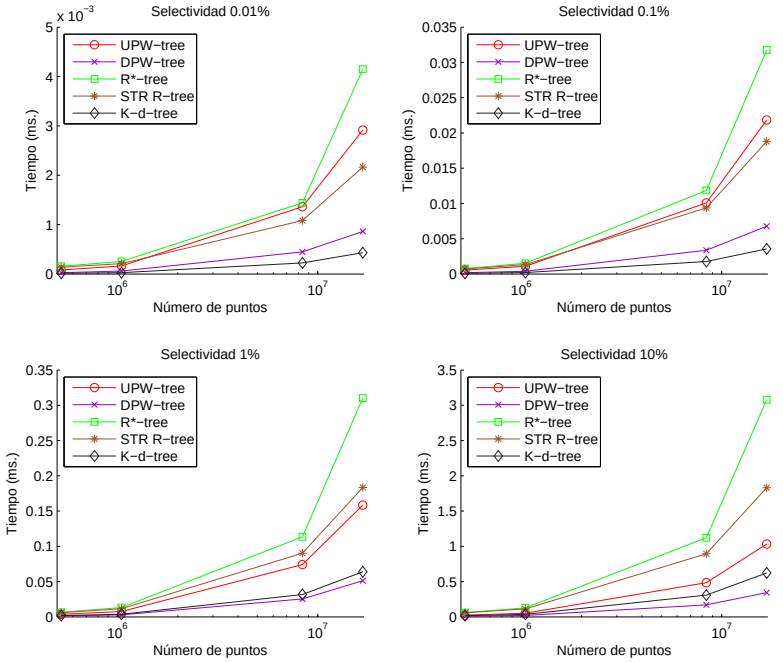


Figura 5.6: Comparación de tiempos.

Además de estos experimentos que nos permiten observar la escalabilidad de las distintas propuestas, realizamos otros para comprobar el comportamiento de las estructuras implementadas frente a distintas distribuciones de datos tanto sintéticas como reales. En la figura 5.7 presentamos los resultados de estos tests.

La conclusión más importante que se puede extraer a la vista de los resultados es que nuestra estructura es competitiva en cuanto a la eficiencia de resolución de consultas, llegando incluso a ganarle al K-d-tree en consultas poco selectivas. El K-d-tree se puede asegurar que es la estructura más eficiente pero, como veíamos en el apartado anterior, esa eficiencia se basa en una necesidad de espacio de almacenamiento mucho mayor. Como esperábamos, las variantes del R-tree no son competitivas en tiempo al ser métodos más generales. Las incluimos en estos resultados porque el R-tree es el ejemplo paradigmático de estructura de indexación espacial y, además, se debe tener muy en cuenta ya que sigue siendo la estructura más empleada

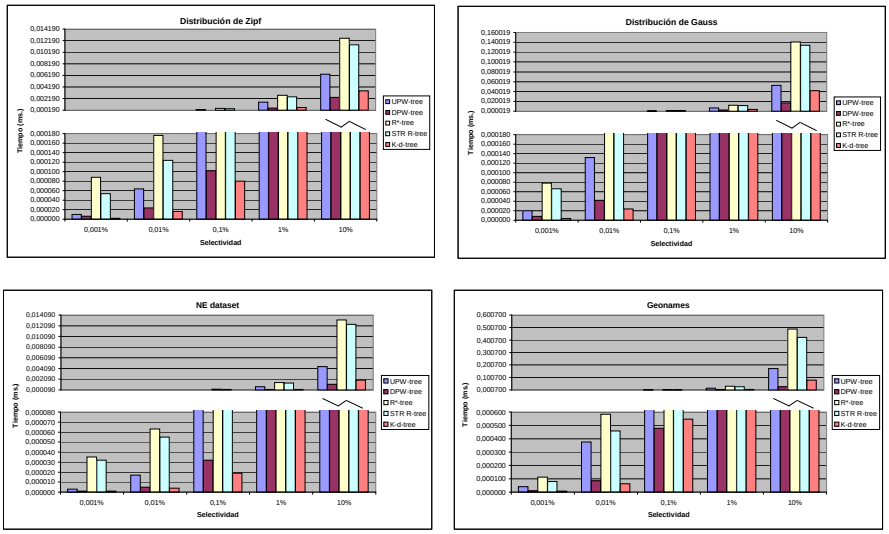


Figura 5.7: Comparación de tiempos (otras colecciones).

hoy en día. En cuanto a las dos versiones de nuestra estructura, el DPW-tree (la versión que sólo desciende en la estructura) es más eficiente que el UPW-tree (la versión que necesita ascender en el árbol). Como explicábamos en la sección 5.2.2, el UPW-tree tiene la ventaja de no tener que alcanzar siempre los nodos hoja sino que puede iniciar el ascenso en el árbol en cuanto comprueba que se trata de un punto que satisface la consulta. Sin embargo, a la vista de los resultados esta ventaja no es suficiente para mejorar la eficiencia del DPW-tree, que siempre tiene que alcanzar los nodos hoja (aunque no necesita volver a ascender en el árbol).

5.4. Resumen

En este capítulo hemos presentado un nuevo índice espacial que pertenece a la categoría de métodos de acceso a puntos. Este índice espacial está basado en el *wavelet tree*, una estructura compacta ampliamente utilizada en áreas como la recuperación de información para la creación de auto-índices. La principal ventaja de esta estructura frente a otros índices espaciales es que presenta una buena relación entre el espacio necesario para almacenarla y la eficiencia de las búsquedas. El poco espacio que necesita le permite operar en memoria principal aún con colecciones realmente grandes de puntos.

En primer lugar, en la sección 5.1 presentamos la motivación de esta estructura. Dicha motivación se encuentra sobre todo en el aumento del tamaño de la memoria principal que ha permitido el desarrollo de estructuras de indexación que no necesiten acceder a disco y en las características específicas de la información geográfica que se gestiona en sistemas GIR. A continuación, en la sección 5.2 describimos la estructura de indexación espacial que proponemos. Esta sección incluye una descripción detallada tanto de los componentes de la estructura como de los algoritmos que hemos desarrollado en base a ella para resolver consultas espaciales de tipo ventana. Presentamos dos versiones de nuestra estructura: DPW-tree y UPW-tree. La base de ambas es un *wavelet tree* que almacena la permutación de los puntos ordenados en una dimensión a los puntos ordenados en la otra dimensión. Además, se necesitan dos vectores con las coordenadas ordenadas en cada dimensión y un vector con los identificadores de los puntos. La principal diferencia entre las dos versiones reside en el orden en que se almacenan esos identificadores, que puede ser el de cualquiera de los dos vectores de coordenadas y que condiciona los algoritmos de resolución de consultas. Finalmente, en la sección 5.3 presentamos los resultados de los experimentos que hemos realizado para comparar nuestra propuesta frente a otros métodos de acceso espacial y métodos de acceso a puntos. La principal conclusión es que la eficiencia de nuestra estructura para resolver consultas es comparable a la del K-d-tree (uno de los métodos de acceso a puntos más eficientes) y mucho mejor que la de cualquier versión, dinámica o estática, del R-tree (el método de acceso espacial más empleado), y con un requerimiento de espacio que le permite operar en memoria principal con colecciones reales de puntos.

Capítulo 6

Conclusiones y trabajo futuro

En este capítulo final resumimos las ideas, aportaciones y resultados más importantes obtenidos en este trabajo de tesis. En la sección 6.1 presentamos las conclusiones, indicando las aportaciones principales del trabajo realizado, y en la sección 6.2 describimos las líneas de trabajo futuro para continuar con la investigación iniciada con esta tesis.

6.1. Conclusiones y aportaciones principales

Este trabajo de tesis presenta los resultados que hemos obtenido en el campo de investigación de la recuperación de información geográfica. Este campo, que ha surgido hace pocos años, tiene su origen en dos más consolidados como son la recuperación de información y los sistemas de información geográfica. Su objetivo principal, y también el objetivo principal de este trabajo de tesis, es la indexación de información teniendo en cuenta tanto su ámbito textual como su ámbito geográfico. Este objetivo general lo concretamos en tres sub-objetivos más específicos que constituyen las tres aportaciones principales de este trabajo: una estructura de indexación que tenga en cuenta tanto el ámbito textual como el espacial de los documentos, una arquitectura completa para sistemas GIR y un método de acceso a puntos optimizado para tener en cuenta las características propias de este tipo de sistemas.

En primer lugar, la estructura de indexación que hemos propuesto para sistemas GIR combina un índice textual, un índice espacial y una ontología del espacio geográfico. Basándose en estos tres elementos, nuestra estructura permite resolver consultas clásicas, como pueden ser las consultas textuales y espaciales puras, pero también nuevos tipos de consultas que combinan aspectos textuales y espaciales. Los dos ejemplos típicos de estos nuevos tipos de consultas son las consultas textuales sobre

un área geográfica (por ejemplo, restringidas a una ventana de consulta) y las consultas textuales referidas a un nombre de lugar determinado. Además, nuestra estructura de indexación presenta una ventaja cualitativa muy importante sobre otras alternativas para sistemas GIR que se han propuesto recientemente. Esta ventaja es que realiza expansión de los términos de consulta de manera implícita. Es decir, al estar basada en una ontología que representa de manera apropiada el espacio geográfico referido por los documentos indexados (manteniendo las relaciones, la organización jerárquica, etc.), en el resultado de las consultas se obtienen documentos que no tienen que citar explícitamente los lugares referidos en la consulta sino que pueden citar otros que se sabe que están relacionados con ellos. Finalmente, nuestra estructura presenta unos resultados comparables en términos de eficiencia a los de otras alternativas propuestas para el mismo objetivo que, sin embargo, no pueden realizar expansión de consultas de manera implícita y necesitan técnicas auxiliares que añaden más complejidad al sistema.

En segundo lugar, presentamos una arquitectura completa para sistemas GIR. Esta arquitectura soporta todo el flujo de trabajo para la inclusión de nuevos documentos en la estructura de indexación que constituye el núcleo de los sistemas desarrollados en base a ella. Además, define también todos los componentes necesarios para que la información indexada pueda ser consultada de manera cómoda por los usuarios, y presentada de manera clara y organizada. Entre las características más importantes de la arquitectura se encuentran el ser genérica, modular, flexible y escalable. Estas características que planteamos como requisitos básicos desde el primer análisis de la arquitectura tienen su motivación en que la recuperación de información geográfica es un campo muy reciente y activo, lo que hace que estén apareciendo continuamente nuevas soluciones a las distintas tareas (representadas en la arquitectura en forma de módulos) que intervienen en este tipo de sistemas. Por tanto, la arquitectura debe permitir de la manera más cómoda posible probar e integrar las propuestas que van apareciendo. A modo de ejemplo, las primeras tareas que debe realizar el sistema para incluir nuevos documentos consisten en dar de alta esos documentos en base de datos, analizarlos, extraer su texto y las referencias geográficas mencionadas en ellos. Esos documentos para nuestra arquitectura son una entidad abstracta, lo que la hace más genérica y le permite trabajar con distintos formatos de archivos y esquemas de contenido. Además, encontrar las referencias geográficas mencionadas en los textos es una tarea compleja y, de momento, todas las propuestas están lejos de obtener una eficacia cercana al procesamiento manual. Por este motivo, la arquitectura permite incorporar y probar fácilmente las propuestas de componentes de localización de referencias geográficas que se van proponiendo.

Creemos que esta aportación no estaría completo sin el desarrollo de un prototipo de sistema completo basado en esta arquitectura. Este prototipo lo hemos desarrollado bajo el paradigma del software libre, empleando componentes disponibles siempre que fue posible, y desarrollando y ofreciendo a la comunidad nuevas alternativas cuando no lo fue o cuando los componentes existentes no cumplían los requisitos exigidos.

Finalmente, nuestra última aportación es un nuevo método de acceso a puntos; es decir, una estructura de indexación espacial que permite trabajar con colecciones de puntos. Esta estructura está basada en el *wavelet tree* una estructura muy compacta que nos permite indexar colecciones de puntos en memoria principal. Los resultados de los experimentos que hemos realizado demuestran que nuestra estructura presenta una relación muy buena entre la eficiencia de las búsquedas y el espacio necesario para almacenarla. Además, esta estructura la hemos diseñado teniendo en cuenta las características propias de la información geográfica propia de sistemas GIR.

El prototipo de la arquitectura nos ha permitido valorar el grado de consecución de los objetivos y el cumplimiento de los requisitos impuestos a la arquitectura. Además, las evaluaciones de los revisores de los numerosos congresos nacionales e internacionales en los que hemos publicado los resultados obtenidos nos han servido también de validación externa del trabajo realizado.

6.2. Líneas de trabajo futuro

Del mismo modo que el apartado anterior, vamos a organizar las líneas de trabajo futuro iniciadas con este trabajo siguiendo los tres objetivos concretos del mismo. En primer lugar, tenemos pensado realizar mejoras de la estructura de indexación propuesta para la recuperación de información geográfica. Una de las posibles mejoras es la inclusión de nuevas relaciones, como puede ser la de adyacencia. Este tipo de relaciones se pueden representar fácilmente en la ontología del espacio geográfico y la estructura de indexación fue diseñada con el requisito de que este tipo de cambios se puedan incorporar de manera sencilla. Además, tenemos planeado explorar el uso de otras ontologías para determinar cómo afecta esta selección al índice resultante. Como hemos visto en el capítulo 3, una de las características de nuestra estructura es que, debido a que las ontologías son estáticas, la estructura se debe adaptar al dominio concreto en el que se vaya a emplear. Por tanto, es interesante comprobar qué sucede cuando en lugar de trabajar con documentos de noticias de todo el mundo trabajamos con documentos localizados en un área más restringida como puede ser un país, una comunidad autónoma o una provincia. En cada uno de estos casos la ontología y la estructura deberían restringirse a esos ámbitos para obtener mejores resultados. Por último, la estructura que proponemos es de doble índice y puede resultar interesante diseñar una estructura híbrida con las mismas características para comparar en qué medida afecta esta decisión a la eficiencia del sistema.

En segundo lugar, como venimos resaltando a lo largo de este trabajo, los componentes que forman la arquitectura (o el prototipo desarrollado en base a la misma) son frutos de trabajos de investigación reciente y, por tanto, constituyen líneas abiertas de investigación. Los componentes relacionados con la detección y traducción de nombres de lugar mencionados en documentos a referencias geográficas que puedan

ser empleadas en índices espaciales constituyen posiblemente la línea de trabajo futuro más atractiva. Los problemas derivados de la ambigüedad de los nombres de lugar hacen que las técnicas de resolución de topónimos no alcancen todavía unos resultados comparables a la geo-referenciación manual. Por tanto, el desarrollo de componentes de desambiguación de topónimos que tengan en cuenta el contexto (es decir, otros topónimos que se mencionan en el mismo documento) está atrayendo el interés de muchos investigadores del área.

Por último, el método de acceso a puntos que proponemos constituye posiblemente la línea de trabajo futuro más prometedora. Estamos trabajando en variaciones de la estructura que permitan la inserción y el borrado de puntos una vez que se ha construido la estructura. También, tenemos planeado desarrollar algoritmos que nos permitan resolver otros tipos de consultas espaciales interesantes (como la consulta de los k vecinos más cercanos). Además, la idea de emplear el *wavelet tree* para indexar puntos distribuidos en el espacio es fácilmente generalizable y estamos trabajando con bastante éxito en la indexación de cualquier tipo de objeto geográfico y no sólo de puntos. Al igual que la mayoría de los métodos de acceso espacial empleamos el MBR como simplificación de los objetos geográficos. Nuestra propuesta para indexar estos MBR emplea dos *wavelet trees* que permiten almacenar las permutaciones de los puntos de apertura de los MBR a los de cierre. Otra posible alternativa es que los dos *wavelet trees* almacenen las permutaciones de una dimensión a la otra para los puntos de apertura y cierre. Los experimentos preliminares muestran que la eficiencia de nuestras alternativas es comparable a la del R-tree con unos requisitos de espacio mucho menores.

Bibliografía

[AHSS04] E. Amitay, N. Har'El, R. Sivan, y A. Soffer. Web-a-where: geotagging web content. En *SIGIR'04: Proc. of the 27th ACM SIGIR Conference*, pp. 273–280, New York, USA, 2004.

[Ai08] Alias-i. LingPipe, Natural Language Tool. Fecha de consulta: Marzo de 2008. Disponible en: http://www.alias-i.com/lingpipe/.

[All07] J. Allan. Graduate level course on Information Retrieval. Fecha de consulta: Enero de 2007. Disponible en: http://ciir.cs.umass.edu/cmpsci646/.

[Apa08a] Apache. Lucene. Sitio web. Fecha de consulta: Marzo de 2008. Disponible en: http://lucene.apache.org.

[Apa08b] Apache Lucene. Scoring. Fecha de consulta: Junio de 2008. Disponible en: http://lucene.apache.org/java/2_2_0/scoring.html.

[AS06] L. Andrade y M.J. Silva. Relevance Ranking for Geographic IR. En *GIR'06: Proc. of the workshop on Geographic Information Retrieval*, 2006.

[Ben75] J.L. Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, 1975.

[BFLN08] N.R. Brisaboa, A. Fariña, S. Ladra, y G. Navarro. Reorganizing compressed text. En *SIGIR'08: Proc. of the 31th ACM SIGIR Conference*, pp. 139–146, Singapore, 2008.

[BFNP07] N. Brisaboa, A. Fariña, G. Navarro, y J. Paramá. Lightweight natural language text compression. *Information Retrieval*, 10(1):1 – 33, 2007.

[BKK99] C. Böhm, G. Klump, y H.P. Kriegel. XZ-Ordering: A Space-Filling Curve for Objects with Spatial Extension. En *Proc. of the SSD Conference*, pp. 75–90, 1999.

[BKSS90] N. Beckmann, H.P. Kriegel, R. Schneider, y B. Seeger. The R*-tree: an efficient and robust access method for points and rectangles. *SIGMOD Rec.*, 19(2):322–331, 1990.

[BM72] R. Bayer y E.M. McCreight. Organization and Maintenance of Large Ordered Indices. *Acta Inf.*, 1:173–189, 1972.

[BM98] P. Burrough y R. McDonnell. *Principles of Geographical Information Systems*. Oxford University Press, 1998. ISBN: 0-19-823365-5.

[Bri92] E. Brill. A simple rule-based part of speech tagger. En *ANLP'92: Proc. of the 3rd Conference on Applied Natural Language Processing*, pp. 152–155, 1992.

[BYRN99] R. Baeza-Yates y B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[CF96] E. Clementini y P.D. Felice. A Model for Representing Topological Relationships Between Complex Geometric Features in Spatial Databases. *Information Sciences*, 90:121–136, 1996.

[CIK08] CIKM. Conference on Knowledge Management. Sitio web. Fecha de consulta: Junio de 2008. Disponible en: http://www.cikm2008.org/.

[CR97] N. Chinchor y P. Robinson. MUC-7 named entity task defnition. En *Proc. of the MUC-7, the 7th Message Understanding Conference*, 1997.

[CSM06] Y.Y. Chen, T. Suel, y A. Markowetz. Efficient query processing in geographic web search engines. En *SIGMOD'06: Proc. of the ACM SIGMOD Conference*, pp. 277–288, 2006.

[dGM08] B.E. da Graça Martins. *Geographically Aware Web Text Mining*. PhD thesis, Universidade de Lisboa, 2008.

[Dir07] Directiva INSPIRE. Sitio web. Fecha de consulta: Mayo de 2007. Disponible en: http://www.ec-gis.org/inspire/.

[EH92] M.J. Egenhofer y J.R. Herring. Categorizing Binary Topological Relations Between Regions, Lines, and Points in Geographic Databases. Technical report, Department of Surveying Engineering, University of Maine, 1992.

[FJA05] G. Fu, C.B. Jones, y A.I. Abdelmoty. Ontology-Based Spatial Query Expansion in Information Retrieval. En *ODBASE'05: Proc. of the On the Move to Meaningful Internet Systems*, volume 3761 of *LNCS*, pp. 1466 – 1482, 2005.

[FWT07] FWTools. Open Source GIS Binary Kit for Windows and Linux. Fecha de consulta: Abril 2007. Disponible en: http://fwtools.maptools.org.

[Gar05] J. Garrett. Ajax: A New Approach to Web Applications. Fecha de consulta: Abril de 2007. Disponible en: http://www.adaptivepath.com/publications/essays/archives/000385.php.

[Geo07] Geonames. Gazetteer. Fecha de consulta: Marzo de 2007. Disponible en: http://www.geonames.org.

[Geo08] Geoserver Community. GeoServer WMS. Fecha de consulta: Enero de 2008. Disponible en: http://geoserver.org.

[GG98] V. Gaede y O. Günther. Multidimensional access methods. *ACM Comput. Surv.*, 30(2):170–231, 1998.

[GGMN05] R. González, S. Grabowski, V. Mäkinen, y G. Navarro. Practical Implementation of Rank and Select Queries. En *Proc. of the 4th WEA (Poster)*, pp. 27–38, 2005.

[GGV03] R. Grossi, A. Gupta, y J. Vitter. High-Order Entropy-Compressed Text Indexes. En *SODA'03: Proc. of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 841–850, 2003.

[GH05] O. Gospodnetić y E. Hatcher. *Lucene IN ACTION*. Manning, 2005. ISBN: 1932394281.

[GHJV96] E. Gamma, R. Helm, R. Johnson, y J. Vlissides. *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley, 1996.

[Glo07] Global Spatial Data Infrastructure Association. Sitio web. Fecha de consulta: Mayo de 2007. Disponible en: http://www.gsdi.org/.

[GM05] E. Garbin y I. Mani. Disambiguating toponyms in news. En *HLT-EMNLP'05: Proc. of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 363–370, 2005.

[Goo07] Google. Maps España. Fecha de consulta: Marzo de 2007. Disponible en: http://maps.google.es/.

[GR04] F. Godoy y A. Rodríguez. Defining and comparing content measures of topological relations. *GeoInformatica*, pp. 347–371, 2004.

[Gru93a] T.R. Gruber. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. En *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Deventer, The Netherlands, 1993.

[Gru93b] T.R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199 – 220, 1993.

[Gut84] A. Guttman. R-Trees: A Dynamic Index Structure for Spatial Searching. En *SIGMOD'84: Proc. of the ACM SIGMOD Conference*, pp. 47–57, Boston, Massachusetts, 1984.

[HA03] J.E. Harmon y S.J. Anderson. *The Design and Implementation of Geographic Information Systems*. John Wiley & Sons, 2003. ISBN: 0-471-20488-9.

[Had09] M. Hadjieleftheriou. Spatial Index Library. Fecha de consulta: Enero de 2009. Disponible en: http://research.att.com/ marioh/spatialindex/.

[HLEY05] M. Hogeboom, F. Lin, L. Esmahi, y C. Yan. Constructing kwoledge bases for e-learning using protégé 2000 and web services. En *AINA'05: Proc. of the 19th International Conference on Advanced Information Networking and Applications*, pp. 215–220, 2005.

[Huf52] D.A. Huffman. A method for the construction of minimum-redundancy codes. En *Proc. of the Inst. Radio Eng.*, volume 40, pp. 1098–1101, 1952.

[ISO02] ISO/IEC. Geographic Information – Reference Model. International Standard 19101, ISO/IEC, 2002.

[Jac89] G. Jacobson. Space-efficient static trees and graphs. *Symposium on Foundations of Computer Science*, 0:549–554, 1989.

[JAF03] C.B. Jones, A.I. Abdelmoty, y G. Fu. Maintaining ontologies for geographical information retrieval on the web. En *ODBASE'03: Proc. of the On The Move to Meaningful Internet Systems*, volume 2888 of *LNCS*, 2003.

[JAFV04] C.B. Jones, A.I. Abdelmoty, G. Fu, y S. Vaid. The SPIRIT Spatial Search Engine: Architecture, Ontologies and Spatial Indexing. En *Proc. of the 3rd Int. Conf. on Geogr. Inform. Science*, volume 3234 of *LNCS*, pp. 125 – 139, 2004.

[JAT01] C. Jones, H. Alani, y D. Tudhope. Geographical information retrieval with ontologies of place. En *COSIT'01: Proc. of the 3rd International Conference on Spatial Information Theory*, pp. 322–335, 2001.

[JPR+02] C.B. Jones, R. Purves, A. Ruas, M. Sanderson, M. Sester, M.J. van Kreveld, y R. Weibel. Spatial information retrieval and geographical ontologies an overview of the SPIRIT project. En *SIGIR'02: Proc. of the 25th ACM SIGIR Conference*, pp. 387 – 388, 2002.

[KF93] I. Kamel y C. Faloutsos. On Packing R-trees. En *CIKM'93: Proc. of the 2nd ACM CIKM Conference*, pp. 490–499, 1993.

[Lar95] R. Larson. Geographic information retrieval and spatial browsing. *Geographic Information Systems and Libraries: Patrons, Maps, and Spatial Information*, pp. 81–123, 1995.

[LGMR01] P. Longley, M. Goodchild, D. Maguire, y D. Rhind. *Geographic Information Systems and Science*. John Wiley & Sons, 2001. ISBN: 0-471-49521-2.

[LLE97] S. Leutenegger, M. Lopez, y J. Edgington. STR: A Simple and Efficient Algorithm for R-Tree Packing. En *ICDE'97: Proc. of the 13th International Conference on Data Engineering*, pp. 497–506, 1997.

[LSSS07] M.D. Lieberman, H. Samet, J. Sankaranarayanan, y J. Sperling. STEWARD: Architecture of a Spatio-Textual Search Engine. En *ACMGIS'07: Proc. of the 15th ACM Int. Symp. on Advances in GIS*, pp. 186 – 193, 2007.

[LT92] R. Laurini y D. Thompson. *Fundamentals of spatial informations systems*. The APIC Series 37 - Academic Press, 1992. ISBN: 0-12-438380-7.

[Lua04] M.R. Luaces. *A Generic Architecture for Geographic Information Systems*. PhD thesis, Universidade da Coruña, 2004.

[Mic07] Microsoft. Live Search Maps. Fecha de consulta: Marzo de 2007. Disponible en: http://maps.live.com/.

[MM90] U. Manber y G. Myers. Suffix arrays: a new method for on-line string searches. En *SODA'90: Proc. of the Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 319–327, Philadelphia, USA, 1990.

[MN07] V. Mäkinen y G. Navarro. Rank and select revisited and extended. *Theor. Comput. Sci.*, 387(3):332 – 347, 2007.

[MN08] V. Mäkinen y G. Navarro. On Self-Indexing Images - Image Compression with Added Value. En *DCC'08: Proc. of the Data Compression Conference*, pp. 422–431, Washington, DC, USA, 2008.

[MNZBY98] E. Moura, G. Navarro, N. Ziviani, y R. Baeza-Yates. Fast searching on compressed text allowing errors. En *SIGIR'98: Proc. of the 21th ACM SIGIR Conference*, pp. 298–306, 1998.

[MNZBY00] E. Moura, G. Navarro, N. Ziviani, y R. Baeza-Yates. Fast and flexible word searching on compressed text. *ACM Transactions on Information Systems*, 18(2):113 – 139, 2000.

[Mof89] A. Moffat. Word-based text compression. *Software Practice & Experience*, 19(2):185 – 198, 1989.

[Mor66] G.M. Morton. A computer Oriented Geodetic Data Base and a New Technique in File Sequencing. Technical report, IBM Ltd., 1966.

[MRS08] C.D. Manning, P. Raghavan, y H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[MSA05] B. Martins, M.J. Silva, y L. Andrade. Indexing and ranking in Geo-IR systems. En *GIR'05: Proc. of the workshop on Geographic Information Retrieval*, pp. 31–34, New York,USA, 2005.

[Nat07a] National Imagery and Mapping Agency (NIMA). Vector Map Level 0. Fecha de consulta: Marzo de 2007. Disponible en: http://www.mapability.com.

[Nat07b] National Institute of Standards and Technology (NIST). Text REtrieval Conference (TREC). Fecha de consulta: Mayo de 2007. Disponible en: http://trec.nist.gov/.

[Nat08] National Institute of Standards and Technology (NIST). TREC Special Database 22, TREC Document Database: Disk 4. Fecha de consulta: Marzo de 2008. Disponible en: http://www.nist.gov/srd/nistsd22.htm.

[NHS81] J. Nievergelt, H. Hinterberger, y K.C. Sevcik. The Grid File: An Adaptable, Symmetric Multi-Key File Structure. En *Proc. of the ECI Conference*, pp. 236–251, 1981.

[NM07] G. Navarro y V. Mäkinen. Compressed full-text indexes. *ACM Comput. Surv.*, 39(1), 2007.

[NS86] R.C. Nelson y H. Samet. A consistent hierarchical representation for vector data. En *Proc. of the SIGGRAPH Conference*, pp. 197–206, 1986.

[Ope02] Open GIS Consortium, Inc. OpenGIS Web Map Service Implementation Specification. OpenGIS Project Document 01-068r3, Open GIS Consortium, Inc., 2002.

[Ope03] Open GIS Consortium, Inc. OpenGIS Reference Model. OpenGIS Project Document 03-040, Open GIS Consortium, Inc., 2003.

[Ope06] Open GIS Consortium, Inc. OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 2: SQL option. OpenGIS Project Document 06-104r3, Open GIS Consortium, Inc., 2006.

[Ope07a] Open GIS Consortium, Inc. OpenGIS Geography Markup Language (GML) Encoding Standard. OpenGIS Project Document 07-036, Open GIS Consortium, Inc., 2007.

[Ope07b] Open GIS Consortium, Inc. Sitio web. Fecha de consulta: Febrero de 2007. Disponible en: http://www.opengeospatial.org/.

[OS07] D. Okanohara y K. Sadakane. Practical Entropy-Compressed Rank/Select Dictionary. En *Proc. of the 9th ALENEX*, 2007.

[OSG08] OSGeo. Open Layers API. Fecha de consulta: Mayo de 2008. Disponible en: http://openlayers.org.

[PCV+00] G. Petasis, A. Cucchiarelli, P. Velardi, G. Paliouras, V. Karkaletsis, y C. Spyropoulos. Automatic adaptation of proper noun dictionaries through cooperation of machine learning and probabilistic methods. En *SIGIR'00: Proc. of the ACM SIGIR Conference*, pp. 128–135, 2000.

[PKLS05] J. Pustejovsky, R. Knippen, J. Littman, y R. Saurí. Temporal and event information in natural language text. *Computers and the Humanities*, 39:123–164, 2005.

[Pos07] PostgreSQL. Sitio web. Fecha de consulta: Enero de 2007. Disponible en: http://www.postgresql.org.

[RBB03] E. Rauch, M. Bukatin, y K. Baker. A confidence-based framework for disambiguating geographic terms. En *HLT-NAACL'03: Proc. of the workshop on Analysis of Geogr. References*, pp. 50–54, Morristown, USA, 2003.

[Ref07] Refractions Research. PostGIS. Sitio web. Fecha de consulta: Enero de 2007. Disponible en: http://postgis.refractions.net.

[RJ76] S.E. Robertson y S.K. Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.

[RL85] N. Roussopoulos y D. Leifker. Direct Spatial Search on Pictorial Databases Using Packed R-trees. En *SIGMOD'85: Proc. of the ACM SIGMOD Conference*, 1985.

[RSV01] P. Rigaux, M. Scholl, y A. Voisard. *Spatial Databases With Application To GIS*. Academic Press, 2001. ISBN: 1-55680-588-6.

[Sal63] G. Salton. Associative Document Retrieval Techniques Using Bibliographic Information. *J. ACM*, 10(4):440–457, 1963.

[Sam06] H. Samet. *Multidimensional and Metric Data Structures*. Morgan Kaufmann, 2006. ISBN: 0123694469.

[SC01] D. Smith y G. Crane. Disambiguating geographic names in a historical digital library. En *ECDL'01: Proc. of the 5th European Conference on Research and Advanced Technology for Digital Libraries*, pp. 127–136, 2001.

[SIG07] SIGIR. Special Interest Group on Information Retrieval. Fecha de consulta: Enero de 2007. Disponible en: http://www.sigir.org/.

[SL65] G. Salton y M.E. Lesk. The SMART automatic document retrieval systems - an illustration. *Commun. ACM*, 8(6):391–398, 1965.

[SL68] G. Salton y M. Lesk. Computer Evaluation of Indexing and Text Processing. *J. ACM*, 15(1):8–36, 1968.

[Sow99] J. Sowa. *Knowledge representation: Logical, philosophical, and computational foundations*. Brooks/Cole Publishing Co., Pacific Grove, 1999.

[SRF87] T.K. Sellis, N. Roussopoulos, y C. Faloutsos. The R+-Tree: A Dynamic Index for Multi-Dimensional Objects. En *VLDB'87: Proc. of the 13th International Conference on Very Large Data Bases*, pp. 507–518, Brighton, England, 1987.

[Tag09] A. Tagliasacchi. Kd-tree for Matlab. Fecha de consulta: Enero de 2009. Disponible en: http://www.mathworks.com/matlabcentral/fileexchange/21512.

[The08] Y. Theodoridis. The R-tree-portal. Fecha de consulta: Noviembre de 2008. Disponible en: http://www.rtreeportal.org/.

[TM97] A. Turpin y A. Moffat. Fast file search using text compression. En *Proc. of the 20th Australasian Computer Science Conference*, pp. 1–8, 1997.

[TSM03] E. Tjong, K. Sang, y F. Meulder. Introduction to the CoNLL-03 shared task: Language-independent named entity recognition. En *CoNLL'03: Proc. of the 7th Conference on Natural Language Learning*, pp. 142–147, 2003.

[VJJS05] S. Vaid, C.B. Jones, H. Joho, y M. Sanderson. Spatio-Textual Indexing for Geographical Search on the Web. En *SSTD'05: Proc. of the 9th Int. Symp. on Spatial and Temporal Databases*, volume 3633 of *LNCS*, pp. 218 – 235, 2005.

[Wor04] M.F. Worboys. *GIS: A Computing Perspective*. CRC, 2004. ISBN: 0415283752.

[Wor08] World Wide Consortium. OWL Web Ontology Language Reference. Fecha de consulta: Marzo de 2008. Disponible en: http://www.w3.org/TR/owl-ref/.

[Yah07] Yahoo. Mapas. Fecha de consulta: Marzo de 2007. Disponible en: http://espanol.maps.yahoo.com/.

[YC07] B. Yu y G. Cai. A Query-Aware document Ranking Method for Geographic Information Retrieval. En *GIR'07: Proc. of the workshop on Geographic Information Retrieval*, 2007.

[Zip49] G.K. Zipf. *Human Behavior and the Principle of Least Effort*. Addison-Wesley, 1949.

[ZM06] J. Zobel y A. Moffat. Inverted files for text search engines. *ACM Comput. Surv.*, 38(2):6, 2006.

[ZXW+05] Y. Zhou, X. Xie, C. Wang, Y. Gong, y W.Y. Ma. Hybrid index structures for location-based web search. En *CIKM'05: Proc. of the 13th ACM CIKM Conference*, pp. 155–162, New York, USA, 2005.

Apéndice A

Publicaciones y proyectos relacionados con este trabajo de tesis

A.1. Publicaciones

A.1.1. Conferencias internacionales

- Brisaboa, N. R., Luaces, M. R., Navarro, G., Seco, D. A New Point Access Method Based on a Wavelet Tree. To appear in ER 2009 Workshops Proceedings – Advances in Conceptual Modeling Challenges and Opportunities. LNCS. Gramado (Brazil), 2009.

- Brisaboa, N. R., Luaces, M. R., Pedreira, O., Places, A. S., Seco, D. Indexing Dense Nested Metric Spaces for Efficient Similarity Search. In Proc. of the 7th International Andrei Ershov Memorial Conference (Perspectives of System Informatics) (PSI'09). Novosibirsk (Russia), 2009.

- Montserrat, L., Cotelo-Lema, J. A., Luaces, M. R., Seco D. Collecting, Analyzing, and Publishing Massive Data about the Hypertrophic Cardiomyopathy. To appear in Revised Selected Papers of the 2nd International Conference on Health Informatics (HEALTHINF 2009) – Communications in Computer and Information Science (CCIS). Porto (Portugal), 2009.

- Brisaboa, N. R., Cerdeira-Pena, A., Luaces, M. R., Seco, D. Defining a Workflow process for textual and geographic indexing of documents. In Proc. of the 11th International Conference on Enterprise Information Systems (ICEIS'09), HCI, pp. 78–83. Milan (Italy), 2009.

- Luaces, M. R., Pedreira, O., Places, A. S., Seco, D. A Web-Based Version of a Trivial Game to Promote Galician Culture. Revised Selected Papers of the 4th International Conference on Web Information Systems and Technologies (WEBIST'08) – LNBIP 18, 18, pp. 242–252. Madeira (Portugal), 2009.
- De Bernardo R., Guillermo, Cerdeira-Pena, A., Pedreira, O., Places, A. S., Seco, D. SCRABBLE.GZ: A Web-Based Collaborative Game to Promote the Galician Language. In Proc. of the 2nd International Conferences on Advances in Computer–Human Interactions (ACHI'09), pp. 142–147. Cancún (México), 2009.
- Montserrat, L., Cotelo, J. A. , Luaces, M. R., Seco, D. A Document Management System and Workflow to help at the Diagnosis of Hypertrophic Cardiomyopathy. In Proc. of the 2nd International Conference on Health Informatics (HEALTHINF 2009), pp. 3–10. Porto (Portugal), 2009.
- Luaces, M. R., Paramá, J. R., Pedreira, O., Seco, D. An Ontology-based Index to Retrieve Documents with Geographic Information. In Proc. of the 20th International Conference on Statistical and Scientific Database Management (SSDBM'08) – LNCS, 5069, pp. 384–400. Hong Kong (China), 2008.
- Ladra, S, Luaces, M. R., Pedreira, O., Seco, D. A Toponym Resolution Service following the OGC WPS Standard. In Proc. of the 8th International Symposium on Web and Wireless Geographical Information System. LNCS 5373, pp. 75–85. Shanghai (China), 2008.
- Luaces, M. R., Places, A. S., Rodríguez, F. J., Seco, D. Retrieving Documents with Geographic References Using a Spatial Index Structure based on Ontologies. In ER 2008 Workshops Proceedings – Advances in Conceptual Modeling Challenges and Opportunities. LNCS 5232, pp. 395–404. Barcelona (Spain), 2008.
- Brisaboa, N. R., Pedreira, O., Uribe, R., Solar, R., Seco, D. Clustering based similarity search in metric spaces with sparse spatial centers. In Proc. of SOFSEM 2008: Current Trends in Theory and Practice of Computer Science, pp. 186–197. High Tatras (Slovakia), 2008.
- Luaces, M. R., Pedreira, O., Places, A. S., Seco, D. Trivial.gz: A Web-based Collaborative Game To Promote Galician Culture. In Proc. of the 4th International Conference on Web Information Systems and Technologies (WEBIST), 2, pp. 67–73. Madeira (Portugal), 2008.
- Cerdeira-Pena, A., Carpente R., M. Luisa, Fariña, A., Seco, D. New Approaches for the School Timetabling Problem. In Proc. of the 7th International Conference on Artificial Intelligence (MICAI'08)., pp. 261–267. México DF (México), 2008.

- Luaces, M. R., Paramá, J. R., Pedreira, O., Seco, D., Viqueira, J. R. An Index Structure to Retrieve Documents with Geographic Information. In Proc. of the 1st International Workshop on Multimedia Data Mining and Management (MDMM'07) (DEXA Workshops 2007), pp. 64–68. Regensburg (Germany), 2007.

A.1.2. Conferencias nacionales e iberoamericanas

- Brisaboa, N. R., Luaces, M. R., Navarro, G., Seco, D. Indexación espacial de puntos empleando wavelet trees. Aceptado y pendiente de publicación en Actas de las XIV Jornadas de Ingeniería del Software y Bases de Datos. San Sebastián (España), 2009.

- Luaces, M. R., Paramá, J. R., Pedreira, O., Seco, D. LBD LOCAL: Un sistema para la recuperación de documentos con referencias geográficas. Actas de las II Jornadas de SIG Libre. Girona (España), 2008.

- Cerdeira-Pena, A., Luaces, M. R., Pedreira, O., Seco, D. Un Sistema de Gestión Documental y Workflow con Indexación Temática y Geográfica de los Documentos. Actas de las V Jornadas de la Infraestructura de Datos Espaciales de España IDE, aplicaciones al planeamiento y la gestión del territorio. Tenerife (España), 2008.

- Cerdeira-Pena, A., Luaces, M. R., Pedreira, O., Seco, D. Un Servicio de Resolución de Topónimos siguiendo el estándar OGC WPS. Actas de las V Jornadas de la Infraestructura de Datos Espaciales de España IDE, aplicaciones al planeamiento y la gestión del territorio. Tenerife (España), 2008.

- Lamas, J.I., Luaces, M. R., Places, A. S., R. López, Eduardo, Seco, D. Los GIS y servicios IDE en la difusión de la cultura gallega. La Web Cultura Galega. Actas de las V Jornadas de la Infraestructura de Datos Espaciales de España IDE, aplicaciones al planeamiento y la gestión del territorio. Tenerife (España), 2008.

- Luaces, M. R., Paramá, J. R., Seco, D. Servicio web de análisis de redes en sistemas de información geográfica. La Infraestructura de Datos Espaciales de España en 2007. Proyectos, servicios y nodos, pp. 119–127. Santiago de Compostela (España), 2007.

- Luaces, M. R., Paramá, J. R., Pedreira, O., Seco, D., Viqueira, J. R. Una estructura de indexación para la recuperación de documentos con referencias geográficas. Actas del taller Evolución de la Investigación y la Docencia en Bases de Datos (EIDBD 2007), 1(7), pp. 19-26. Zaragoza (España), 2007.

- Uribe, R., Solar, R., Brisaboa, N. R., Pedreira, O., Seco, D. SSSTree: búsqueda por similitud basada en clustering con centros espacialmente dispersos. Encuentro nacional en computación de Chile (ENC'07), pp. 1–13. Iquique (Chile), 2007.

- Brisaboa, N. R., Places, A. S., Luaces, M. R., Seco, D. Desarrollo de Webs interactivas con filosofía AJAX: El TRIVIAL.GZ. Actas del XII Congreso Argentino de Ciencias de la Computación (CACIC'06). San Luis (Argentina), 2006.

A.1.3. Revistas

- Brisaboa, N. R., Luaces, M. R., Places A. S., Seco, D. Exploiting Geographic References of Documents in a Geographical Information Retrieval System Using an Ontology-based Index Aceptado, pendiente de publicación en GeoInformatica.

- Luaces, M. R., Pedreira, O., Places, A. S., Seco, D. Los sistemas de Información geográfica en turismo. ROTUR Revista de Ocio y Turismo(1), pp. 117–134. A Coruña (España), 2008.

- Places, A. S., Brisaboa, N. R., Paramá, J. R., Pedreira, O., Seco, D. Managing the workflow of massive feeding of digital libraries. Research in Computer Science, 32, pp. 352–362. Mexico, 2007.

A.2. Proyectos de fin de carrera

Título:	Desarrollo de un juego de Monopoly online con capacidades de promoción turística
Alumno:	Miguel Montero Fernández
Directores:	Miguel R. Luaces y Diego Seco
Fecha:	Junio de 2009
Calificación:	Sobresaliente (9)

Título:	O xogo da Curuxa: unha versión didáctica do xogo da oca para o web
Alumno:	Pablo González Ramos
Directores:	Ángeles S. Places y Diego Seco
Fecha:	Junio de 2009
Calificación:	Sobresaliente (9)

Título: Desarrollo de un juego web de ahorcado en gallego
Alumno: F. Javier Fraguio Costoya
Directores: Ángeles S. Places y Diego Seco
Fecha: Diciembre de 2008
Calificación: Sobresaliente (9)

Título: Desenvolvemento dun xogo de encrucillados online
Alumno: Iván Queipo Fernández
Directores: Ángeles S. Places y Diego Seco
Fecha: Septiembre de 2008
Calificación: Sobresaliente (9)

Título: Sistema para a recuperación de documentos con referencias xeográficas
Alumno: César Suárez Suárez
Directores: Miguel R. Luaces y Diego Seco
Fecha: Junio de 2008
Calificación: Sobresaliente (9,5)

Título: Deseño e implementación dun sitio web de podcasting
Alumno: Laura Prol Baña
Directores: Ángeles S. Places y Diego Seco
Fecha: Diciembre de 2007
Calificación: Sobresaliente (9)

Título: Herramienta para el análisis de accesos a sitios web: Módulo de configuración y captura de la información
Alumno: Xurxo Muiños Pantin
Directores: Miguel R. Luaces y Diego Seco
Fecha: Diciembre de 2007
Calificación: Sobresaliente (9)

Título:	Ferramenta para a análise dos accesos a sitios web: Módulo de presentación de estatísticas
Alumno:	Iván Ferreiro Pérez
Directores:	Miguel R. Luaces y Diego Seco
Fecha:	Diciembre de 2007
Calificación:	Matrícula de Honor (10)

A.3. Estancias de investigación

- *Estructuras comprimidas para Sistemas de Información Geográfica*. En el Departamento de Ciencias de la Computación de la Universidad de Chile bajo la supervisión de PhD. Gonzalo Navarro desde el 17 de junio del 2009 (planificada hasta el 17 de diciembre de 2009).

A.4. Software publicado

- LBD Local. Prototipo de un sistema de recuperación de información geográfica. Liberado en la forja de mancomún. Disponible en: http://forxa.mancomun.org/projects/lbdlocal.