

Bases de datos geogràfiques

Albert Botella Plana
Rafael Camps Paré
Anna Muñoz Bolas

PID_00153925

Índice

Introducción	5
Objetivos	6
1. Los tres mundos: el real, el conceptual y el de las representaciones	7
2. Introducción a las Bases de Datos	11
2.1. Ficheros	11
2.2. Bases de datos relacionales	11
2.3. Ventajas en el uso de bases de datos	13
2.4. Sistemas gestores de bases de datos (SGBD)	13
2.5. Consultas en la base de datos	15
2.6. La operación SELECT	16
2.7. Otras operaciones con SQL	17
2.8. Diseño de bases de datos	17
2.8.1. Fases de diseño de una base de datos	18
2.9. Normalización	20
3. Bases de datos geográficos	23
3.1. Almacenamiento de datos vectoriales	24
3.2. Almacenamiento de geometrías vectoriales	26
3.2.1. Almacenamiento de puntos	26
3.2.2. Almacenamiento de líneas	27
3.2.3. Almacenamiento de áreas	29
3.2.4. Almacenamiento de geometrías complejas	31
3.3. Modelos de datos de almacenamiento	32
3.4. Principales formatos de almacenamiento	33
3.4.1. Ficheros CAD (Microstation y AutoCAD)	33
3.4.2. General Electric Smallworld	34
3.4.3. ESRI Shapefile	35
3.4.4. Geomedia Warehouse	36
3.4.5. GML	37
3.4.6. ESRI ArcSDE	38
3.4.7. PostGIS	40
3.4.8. Oracle Spatial	40
4. Consulta e indexación de datos geográficos	41
4.1. Relaciones espaciales y filtrado	41
4.2. Operadores y funciones espaciales	42
4.2.1. Operadores	43
4.2.2. Funciones	46

4.3. Índices espaciales	50
4.3.1. <i>Quadtree</i>	52
4.3.2. Los <i>R-trees</i>	53
4.4. Consultas espaciales	55
4.4.1. Ejemplos de consultas Oracle Spatial	56
5. Modelos avanzados de almacenamiento	59
5.1. Topología	59
5.1.1. Almacenamiento de topología	61
5.1.2. Topología geométrica frente a topología relacional	63
5.1.3. Explotación de la topología	64
5.2. <i>Lineal referencing system</i> (LRS)	65
5.2.1. Almacenamiento de LRS	66
5.2.2. Usos de LRS	68
6. Almacenes <i>raster</i>	69
6.1. Información <i>raster</i>	69
6.2. Modelo de datos <i>raster</i>	71
6.3. Bandas y capas	72
6.4. Pirámides	73
6.5. Mosaicos	76
6.6. Formatos de almacenamiento <i>raster</i>	77
6.6.1. Archivos World	77
6.6.2. GeoTiff	78
6.6.3. MrSID	78
6.6.4. ECW	78
6.6.5. Georaster (Oracle Spatial)	79
7. Software para bases de datos geográficas	80
7.1. Ejemplos de bases de datos	80
7.2. Comparativa entre SQL Server 2008, MySQL y PostGIS	82
7.3. Comparativa entre Oracle Spatial y PostGIS	83
7.3.1. Importación de datos espaciales	84
7.3.2. Especificación SFS	84
7.3.3. Tablas espaciales	84
Resumen	86
Ejercicios de autoevaluación	89
Solucionario	91
Glosario	91
Bibliografía	92

Introducción

En estos materiales tratamos la problemática de cómo almacenar los datos geográficos y cómo explotarlos mediante consultas.

Empezaremos por repasar los conceptos de datos, información y conocimiento, el concepto de base de datos y la diferencia que existe entre bases de datos y ficheros. Más adelante, completaremos los conceptos de bases de datos con una visión general del diseño de bases de datos y las formas normales. Esta visión general os permitirá entender el modo como se almacenan los datos geográficos. Los datos geográficos son muy variados y no todos se almacenan de la misma manera.

Una vez hayáis visto cómo se guardan los objetos discretos sobre el territorio, veréis cómo se almacenan las relaciones topológicas, los acontecimientos lineales, las coberturas o las imágenes satélite. Finalmente, presentaremos el panorama actual de productos de base de datos geográficos, discutiendo, para cada producto, sus puntos fuertes y débiles.

Objetivos

En los materiales didácticos de este módulo encontraréis los elementos imprescindibles para alcanzar los objetivos siguientes:

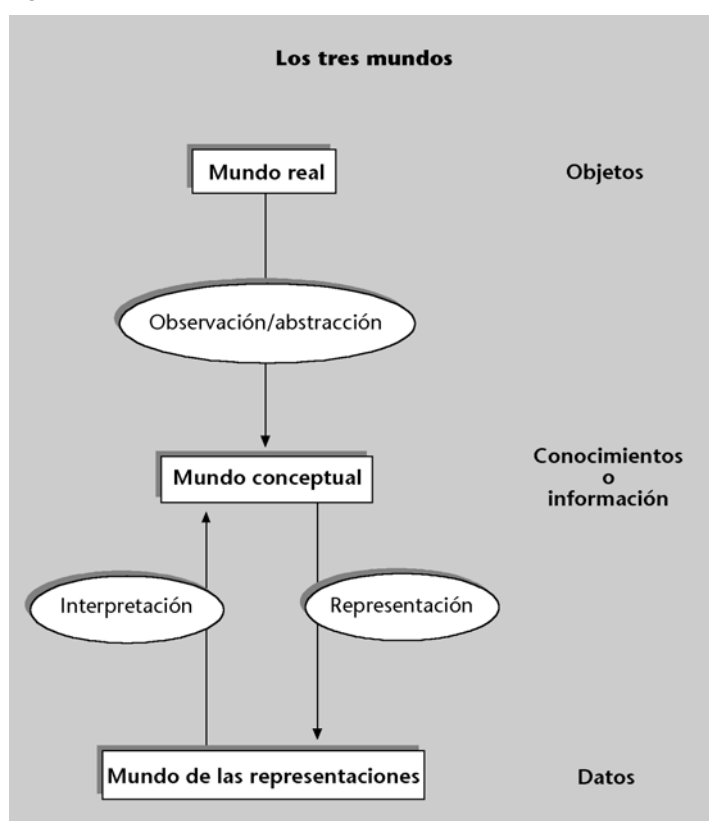
1. Conocer los formatos y las técnicas de almacenamiento de los datos y en particular de los geográficos.
2. Poder diseñar bases de datos para guardar de manera eficiente y ordenada datos geográficos adaptados a las necesidades de cada sistema o proyecto.
3. Entender el modo de guardar las geometrías dentro de un entorno de bases de datos relacionales.
4. Aprender las dificultades y las técnicas de almacenamiento de imágenes *raster*.
5. Saber realizar consultas espaciales optimizadas y relacionando diferentes conceptos.
6. Conocer los formatos de datos y los diferentes fabricantes de bases de datos geográficos del mercado.

1. Los tres mundos: el real, el conceptual y el de las representaciones

Para tener un marco donde situar los términos y los conceptos que explicaremos en la asignatura, distinguiremos tres ámbitos diferentes:

- El mundo real con los objetos de nuestro interés.
- El mundo de las conceptualizaciones lógicas.
- El mundo de las representaciones informáticas.

Figura 1



1.1. La realidad: los objetos

Como informáticos que debemos analizar o construir un sistema de información (SI) determinado, necesitamos conocer el mundo real al que este SI debe hacer referencia o modelizar; así, nuestro mundo real podrá ser un hospital, una empresa distribuidora de productos alimenticios, la matriculación de los alumnos de una universidad, etc.

El **mundo real**, la parte de la realidad que nos interesa, es lo que percibimos con nuestros sentidos y está compuesto por objetos concretos, físicos o no.

Ejemplos de objetos concretos

El enfermo Juan García, la cama 34 de la segunda planta, el almacén de Soria, el camión B-3452-AG, la alumna María Pi, la asignatura *Química I*, la enfermedad meningitis, la devolución de un pedido concreto, un determinado accidente de tráfico, etc. son algunos ejemplos de objetos que pertenecen al mundo real.

Dado que trabajamos en el campo de los SI, los mundos reales que nos interesan son las organizaciones: empresas, instituciones, etc.

1.2. Las concepciones: la información


A partir de la observación del mundo real, los humanos somos capaces de deducir conocimientos e información. La observación de los objetos del mundo real nos conduce a su análisis y su síntesis; después, obtenemos abstracciones de ello, hacemos clasificaciones (podemos saber que dos objetos son de la misma clase a pesar de que sean diferentes), deducimos propiedades e interrelaciones, etc.

El conjunto de los conocimientos obtenidos a partir de la observación de un mundo real se denomina **mundo conceptual** o **mundo de las concepciones**. En la esfera de las concepciones construimos un modelo abstracto, conceptual, del mundo real, y esto nos ayuda a razonar y a expresarnos.


El **proceso de observación/abstracción** es básicamente un proceso para modelizar la estructura, las propiedades y el funcionamiento de la realidad.

De la observación obtenemos la información

La observación del campo de la matriculación en una universidad nos permite reconocer diferentes clases o tipos de objetos, como por ejemplo el *alumno* o la *asignatura*. Deducimos que todo alumno tendrá las propiedades (son abstracciones) *fecha de nacimiento*, *DNI*, *nombre*, etc., y de este modo obtenemos informaciones como las siguientes: el alumno de nombre Juan García tiene el DNI 34.567.854 y su año de nacimiento es 1979.

De hecho, existen diferencias entre *conocimiento* e *información*. La **información** es un conocimiento transmisible, es decir, que se puede representar. Los únicos conocimientos que nos interesarán aquí son, por lo tanto, las informaciones. 

Un mismo mundo real puede ser visto, concebido y modelizado de distintas formas por diferentes observadores (incluso por un mismo observador) según su entorno o marco de referencia. Por ejemplo, un profesor no ve del mismo modo el ámbito de la gestión de un centro universitario que un administrativo de secretaría. Tienen marcos de referencia diferentes. No están interesados en los mismos conceptos. El profesor, a diferencia del administrativo, no necesitará conocer el importe de la matrícula, no querrá distinguir las abstracciones *alumno con beca* y *alumno sin beca*. Los profesores estarán interesados en la calificación numérica, mientras que el servicio administrativo tal vez sólo tendrá en cuenta la forma textual de la calificación.

Así, en el paso del mundo real al de las concepciones encontramos pluralismo. Es posible que la observación y el análisis de una misma parte de una organización o empresa lleven a concepciones diferentes, todas igualmente válidas, y que pueden tener que coexistir. 

1.3. Las representaciones: los datos

El mundo de las concepciones o de los conocimientos es un mundo mental. Sin embargo, para trabajar con estos conocimientos y comunicarlos, necesitamos proyectar los pensamientos al exterior, representándolos físicamente de alguna manera. Éste es el **mundo de las representaciones**.


Podemos representar conocimientos...

... escribiendo a mano sobre un papel, grabando *bytes* en un disco magnético según un formato y una codificación determinados, etc.

Nosotros aquí nos ocuparemos de las representaciones informáticas, y hablaremos de datos, ficheros, bases de datos, registros, campos, *bytes*, discos, etc.

Damos el nombre de **datos** a las representaciones físicas de los conocimientos que tenemos de los objetos del mundo real. El paso de los conocimientos a los datos, o de una concepción a una representación informática, no es automático. Es un proceso humano: un proceso de diseño.

Obviamente, en este caso como en el caso del paso del mundo real al de las concepciones, también hay pluralismo. Un mismo conjunto de conocimientos se puede representar de muchas formas; por ejemplo, en forma de base de datos relacional o como ficheros tradicionales, con vectores o sin ellos, con longitud fija o variable, con codificación ASCII o EBCDIC, etc. Una visión o concepción del mundo real de un hospital, de una universidad o de una distribuidora de productos se podrá representar de muchas formas sobre soportes físicos informáticos.

Sin ningún tipo de duda, las tareas más importantes del analista/diseñador de SI o de aplicaciones informáticas son las siguientes: 

- 1) Analizar los objetos del mundo real, y hacer abstracciones y obtener una concepción lógica de ellos.
- 2) Diseñar una representación informática concreta que se pueda tratar eficientemente.

Diseños diferentes

Se pueden hacer muchos diseños diferentes de representación informática que correspondan a un único modelo conceptual de una realidad. Todos pueden representar la misma realidad, pero tendrán una eficiencia diferente según la utilización que se haga de ellos.

El hecho de saber observar la realidad y hacer de ella las abstracciones lógicas más adecuadas, así como la habilidad para el análisis y la síntesis, llegan a ser cualidades fundamentales que debe tener el desarrollador de SI. Y estas cualidades se deben educar y cultivar.

Evolución del diseño de aplicaciones

El paso de un mundo conceptual a un mundo de representaciones informáticas se hizo más sencillo a medida que la tecnología informática avanzaba y se simplificaba su utilización. En los años sesenta y setenta, el desarrollador de aplicaciones se veía obligado a tener en cuenta una multitud de detalles físicos de la representación informática. Actualmente, la simplificación del proceso de diseño de la representación hace que el proceso de observación/abstracción se convierta en la tarea principal del desarrollador de SI.

1.4. La interpretación

Acabamos de ver el camino que nos conduce de la realidad a los conocimientos, y de éstos a los datos o las representaciones. Sin embargo, nos hará falta interpretar la representación. El proceso inverso al de representación se denomina **interpretación**.

¿Cómo se puede obtener conocimientos e información de una representación?

Si consideramos un dato, una representación que consta de la serie de símbolos 1 9 9 9, y que está extraída de una base de datos relativa a la matriculación de alumnos, no podremos obtener ninguna información de ella si no sabemos si hace referencia al año de matriculación, al año de nacimiento, al importe de la matrícula, al número de la matrícula, etc.; además, no sabremos de qué alumno concreto (de qué objeto del mundo real) se trata. Vemos, entonces, que para poder interpretar los datos se debe saber además a quién y a qué (a qué conceptos) hacen referencia.

Hemos dicho que una información es un conocimiento que se puede representar, pero ahora, teniendo en cuenta el camino inverso, podremos decir que la **información** es el significado que le damos a los datos.

2. Introducción a las Bases de Datos

Antes de introducir las bases de datos geográficos y el almacenamiento de datos espaciales, es conveniente repasar los conceptos principales de almacenamiento de datos informáticos. De esta manera, podréis entender cómo se ha llegado al modelo de datos actual para los datos geográficos. Hay dos grandes familias de formatos de almacenes en los sistemas de información: los ficheros y las bases de datos.

Los ficheros se usan principalmente para informaciones con poco volumen que se suelen consultar en bloque. Las bases de datos se utilizan para grandes bloques de información que se quieren explotar y consultar de manera parcial.

2.1. Ficheros

Un fichero informático es un conjunto de información que se almacena en algún medio que permite al sistema operativo la lectura y escritura. Un fichero se identifica por un nombre y una extensión, que indica el formato. Los ficheros se organizan en carpetas dentro del sistema de ficheros que gestiona el sistema operativo.

Los ficheros almacenan información binaria en un orden y formato que permita a ciertas aplicaciones interpretar su contenido. La lectura y escritura del fichero pasa por funciones de bajo nivel del sistema operativo. Por ello la velocidad de acceso o la edición concurrente del fichero se ven limitados por las capacidades del sistema de ficheros del sistema operativo.

Edición concurrente

La edición concurrente se da cuando dos usuarios modifican un mismo conjunto de datos simultáneamente.

2.2. Bases de datos relacionales

Para definir qué es una base de datos relacional, debemos empezar por saber qué es una base de datos. Una **base de datos** es un conjunto de datos del mismo contexto almacenados y ordenados para que un sistema de información pueda seleccionar rápidamente una porción de éstos. Las bases de datos geográficos que veréis más adelante son las bases de datos que almacenan información para sistemas de información geográficos o SIG.

En contraposición con el resto de datos que podéis encontrar en un ordenador, las bases de datos se organizan y se gestionan mediante motores de bases de datos independientes del sistema de ficheros del sistema operativo denominados *sistemas de gestión de bases de datos* o *SGBD*.

Hay distintos modelos y tipos de bases de datos, pero actualmente hay uno que es el más extendido y el que usan todas las bases de datos geográficas: las bases de datos relacionales.

No repasaremos el resto de tipos de bases de datos, ya que no se aplican a las bases de datos geográficas.

Las bases de datos relacionales son aquellas que están organizadas en tablas y relaciones entre tablas.

- Las **tablas** se organizan en campos y registros de manera similar a las columnas y filas de una hoja de cálculo.
- Cada uno de los **registros** de la tabla representa un objeto con las mismas propiedades.
- Cada una de las propiedades está almacenada en un **campo** de un tipo determinado.
- Toda tabla tiene una **clave primaria** formada por un campo o conjunto de campos. La clave primaria permite identificar inequívocamente cada uno de los registros según su valor.

Ejemplo de tabla

En la figura 2 podéis ver representada una tabla con sus campos y sus registros.

Figura 2. Ejemplo de tabla de bases de datos

	DNI	Nombre	Apellidos	Teléfono
	1234567X	Pep	Camps	897876765
	2345671X	Maria	Camps	897876765
	3456712X	Pau	Valls	–
	4567123X	Pol	Prats	243562817
	5671234X	Pep	Brou	–
	6712345X	Marta	Miro	345654765

Esta tabla almacena datos de personas y tiene 4 campos (*DNI*, *Nombre*, *Apellidos* y *Teléfono*) de diferentes tipos. *DNI*, *Nombre* y *Apellidos* son cadenas de texto, mientras que *Teléfono* es un número. Cada uno de los registros de la tabla representa a una persona. Observad algunas características de los datos:

El campo *Nombre* tiene dos valores repetidos (el valor repetido es “Pep”) y el campo *Apellidos* también (el valor repetido es “Camps”).

El campo *Teléfono* tiene valores vacíos, ya que no todo el mundo tiene teléfono. Cuando un registro no tiene valor para un campo se dice que tiene el valor “NULL”.

DNI es el único campo que tiene valores diferentes y no “NULL” para cada uno de los registros. *DNI* es el mejor (y único) candidato para ser la clave primaria de esta tabla.

Cada tabla se puede relacionar con otras tablas mediante claves foráneas. Una clave foránea es un campo que contiene una referencia a la clave primaria de

otra tabla. Esta relación nos permite indicar que un registro de una tabla está relacionado con un registro de otra tabla. Una tabla también se puede relacionar con clave foránea con ella misma.

Podríamos añadir un campo a la tabla de personas representada en la figura 2 que fuera una clave foránea al padre de la persona. El campo contendría el valor del DNI del padre de la persona.

2.3. Ventajas en el uso de bases de datos

El uso de bases de datos relacionales tiene ventajas e inconvenientes respecto al uso de ficheros:

- Ventajas
 - Facilidad en el manejo de grandes volúmenes de datos.
 - Alto rendimiento de consulta gracias a los índices.
 - Independencia del tratamiento de la información por parte del sistema operativo.
 - Seguridad de acceso a la información.
 - Integridad referencial y no duplicidad de los datos.
 - Acceso concurrente y transaccional a los datos.
 - Copias de seguridad y recuperación de datos.
- Inconvenientes
 - El coste de licencia y el mantenimiento de algunos SGBD es muy elevado.
 - Se deben poseer conocimientos de administración y explotación de bases de datos, ya que un mal diseño puede suponer futuros problemas.
 - Tareas sencillas como copia o edición de la información requieren conocimientos avanzados.

En general, usaremos ficheros cuando no queremos filtrar la información, cuando sólo haya un usuario editando los datos o cuando el conjunto de datos sea discreto. Cuando queramos una base de datos corporativa, la edición por parte de varios usuarios a la vez o la ejecución de consultas y filtrados sobre los datos, entonces la mejor apuesta son las bases de datos.

2.4. Sistemas gestores de bases de datos (SGBD)

Un sistema gestor de base de datos (SGBD, *data base management system* DBMS) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a ellos. El objetivo principal de un SGBD es proporcionar un entorno conveniente y eficiente que permita manipular, extraer y almacenar información de una base de datos.

Los índices se tratarán en el apartado 4 de este módulo.



ANSI

El Instituto Nacional de Estándares de Estados Unidos (American National Standards Institute, ANSI) es una organización sin ánimo de lucro que supervisa el desarrollo de estándares para productos, servicios, procesos y sistemas en Estados Unidos.

ISO

La Organización Internacional para la Estandarización (International Organization for Standardization, ISO) es una organización internacional no gubernamental que produce normas internacionales industriales y comerciales. Estas normas se conocen como normas ISO y su finalidad es facilitar el comercio, facilitar el intercambio de información y contribuir al desarrollo y la transferencia de tecnologías.

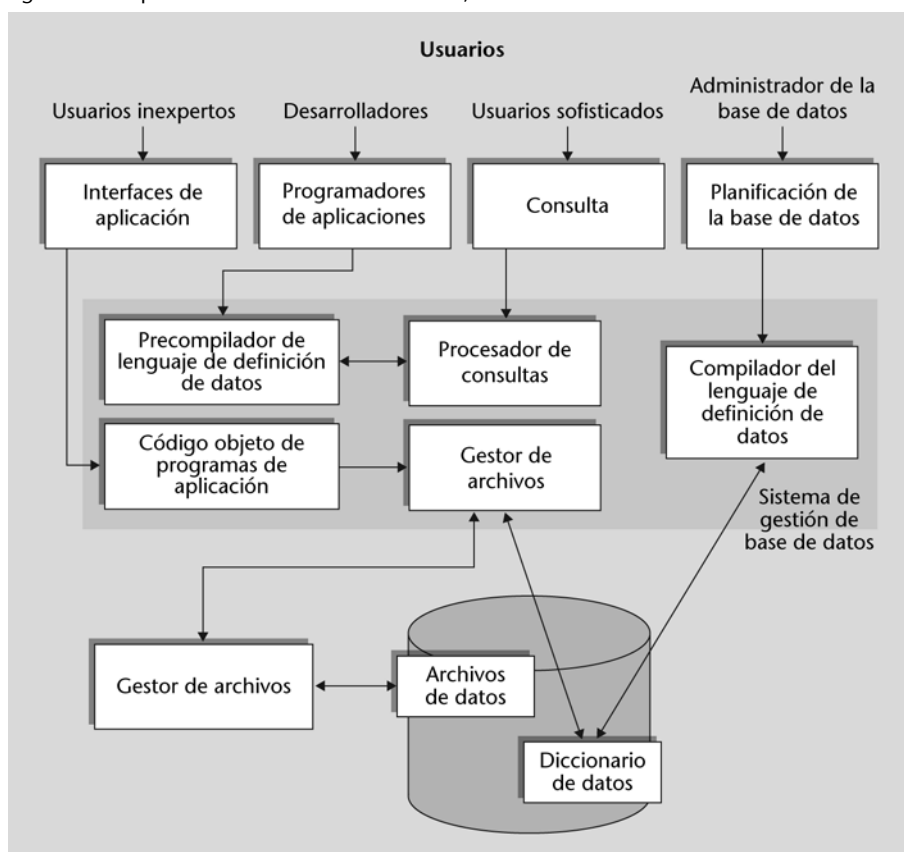
Un SGBD es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, y proporciona acceso a la misma.

Un SGBD debe incluir, por lo menos, las siguientes funciones:

- **Definición de los datos.** El SGBD debe ser capaz de aceptar las definiciones de datos en su versión original y convertirlas en la versión objeto. El SGBD debe incluir componentes procesadores para cada uno de los lenguajes de definición de datos (DDL, *data definition language*).
- **Manipulación de los datos.** El SGBD debe atender las solicitudes de los usuarios para extraer, actualizar, añadir o suprimir datos. También debe incluir un procesador del lenguaje de manipulación de datos (DML).
- **Seguridad e Integridad de los datos.** El SGBD debe supervisar las solicitudes de los usuarios y rechazar los intentos de violar las medidas de seguridad e integridad definidas por el administrador de la base de datos.
- **Recuperación y concurrencia de los datos.** El principal objetivo de la implantación de una base de datos es poner a disposición de un gran número de usuarios un conjunto integrado de datos. Estos usuarios podrán manipular los datos y el SGBD debe garantizar su coherencia tras las operaciones que hayan podido llevarse a cabo con ellos.

En la zona central de la figura siguiente podréis ver los componentes funcionales de un SGBD:

Figura 3. Componentes del SGBD. Fuente: Korth, 1995.



- **Código objeto de programas de aplicación.** Se encarga de actuar de intermediario entre los usuarios y el gestor de archivos. Este componente suele contar con una interfaz de aplicación con algunas tareas predefinidas para facilitar el acceso a los usuarios inexpertos. También acceden a él los desarrolladores.
- **Procesador de consultas.** Traduce sentencias en un lenguaje de consultas en SQL a instrucciones de bajo nivel que entiende el gestor de la base de datos. A este componente acceden los usuarios avanzados, que son capaces, no sólo de llevar a cabo consultas, sino de crear sus propias consultas.
- **Gestor de archivos.** Gestiona la asignación de espacio en la memoria de disco y de las estructuras de datos usadas para representar la información almacenada en disco.
- **Pre-compilador del lenguaje de manipulación de datos (DML).** Para manipular los datos, los SGBD proporcionan lo que se llama lenguaje de manipulación de datos o DML (del inglés *data manipulation language*). El pre-compilador del DML convierte estas sentencias, que están incorporadas en un programa de aplicación, en llamadas normales a procedimientos en el lenguaje principal que utilice el SGBD.
- **Compilador del lenguaje de definición de datos (DDL).** Para definir los datos y esquemas de datos, los SGBD proporcionan también un lenguaje que sería el lenguaje de definición de datos o DDL (del inglés *data definition language*). El compilador de DDL convierte sentencias en este lenguaje en un conjunto de tablas de metadatos o “datos sobre los datos”.
- **Gestor del diccionario de datos.** Almacena metadatos sobre la estructura de la base de datos.

Además de estos componentes, se necesita un **gestor de la base de datos**, que proporcione la interfaz entre los datos de bajo nivel almacenados en la base de datos y los programas de aplicación y las consultas que se hacen en el sistema.

Para acabar, cabe decir que el SGBD accede a donde están almacenados los datos.

2.5. Consultas en la base de datos

Las bases de datos relacionales mediante los SGBD están estructuradas con el fin de poder acceder a los datos y manipularlos de manera distinta y eficiente. Las organizaciones en ficheros normalmente se diseñan para estar optimizadas para un tipo de acceso y pueden implicar algoritmos secuenciales para realizar las consultas.

Uno de los hechos que más han impulsado las bases de datos relacionales es que desde el principio han dispuesto de un lenguaje estándar para realizar las consultas, el SQL.

Evolución del SQL

La primera versión del SQL fue desarrollada en IBM por Donald D. Chamberlin y Raymond F. Boyce a comienzos de los setenta. Más tarde fue adoptada por ORACLE y los diferentes SGBD que fueron surgiendo. La primera versión estándar data de 1986, aunque ha ido sufriendo modificaciones, de las cuales la más importante fue la de 1992, que dio lugar al llamado SQL-92.

SQL (*structured query language*, lenguaje de consultas estructurado) es un lenguaje declarativo diseñado para la obtención y el manejo de datos sobre sistemas gestores de bases de datos relacionales. SQL es un estándar tanto para ANSI como para ISO y lo implementan todos los sistemas de bases de datos relacionales actuales.

SQL se compone de un conjunto discreto de sentencias que permiten realizar tanto consultas como modificaciones sobre una base de datos relacional. A continuación, veréis las sentencias más habituales. Éstas son:

`SELECT`: para obtener datos.

`INSERT`: para añadir datos.

`UPDATE`: para modificar datos ya introducidos.

`DELETE`: para borrar datos.

`CREATE TABLE`: para añadir una nueva tabla.

`DROP TABLE`: para borrar una tabla.

2.6. La operación `SELECT`

La operación más común con el SQL es la consulta de información que se ejecuta mediante la operación `SELECT`. La operación `SELECT` devuelve información de una o más tablas filtrada según unos determinados criterios, pero no modifica los datos. SQL permite al usuario definir cuál es el conjunto de datos que quiere obtener, pero deja la responsabilidad de planificar y optimizar la captura y el cálculo físico de los datos al sistema gestor de bases de datos.

Una consulta SQL incluye una serie de cláusulas para especificar el resultado esperado:

- Justo después de la operación `SELECT` se especifican los campos de resultado de la consulta. Para indicar todos los campos disponibles en las tablas consultadas, se debe escribir el carácter asterisco (*). Recordad que el resultado de una consulta `SELECT` tiene la forma de una tabla con campos y registros.
- La cláusula `FROM` permite indicar las tablas de las que se extraerá la información. Con la cláusula `JOIN` se pueden relacionar tablas entre ellas según el criterio del usuario.
- La cláusula `WHERE` incluye el filtro que restringe los registros de la consulta. Esta cláusula elimina todos los registros para los cuales la expresión indicada por el usuario se evalúa como falso.

La cláusula `ORDER BY` nos permite determinar el orden de los registros de la consulta.

2.7. Otras operaciones con SQL

Las principales operaciones de modificación y manipulación de los datos son:

INSERT: se usa para introducir nuevos registros en una tabla existente

```
INSERT INTO Personas(nombre, dirección, teléfono) VALUES  
( 'Pep', 'Av Diagonal 12', '48485825')
```

UPDATE: se usa para modificar registros de una tabla existente

```
UPDATE Personas SET nombre= 'Pep' WHERE dni= '44332223A'
```

DELETE: se usa para borrar registros de una tabla existente

```
DELETE FROM Personas WHERE dni= '44332223A'
```

CREATE TABLE: se usa para crear una tabla nueva

```
CREATE TABLE Personas  
(nombre varchar(30),  
apellidos varchar(60),  
direccion varchar,  
edad int(3) )
```

DROP TABLE: se usa para borrar una tabla existente

```
DROP TABLE Personas
```

Éstas son sólo algunas de las operaciones que pueden realizarse con una base de datos mediante sentencias SQL. Existen otras operaciones que permiten asignar o revocar permisos a usuarios, gestionar transacciones, etc. SQL tiene muchas opciones que sería interesante dominar pero que no son objeto de esta asignatura.

2.8. Diseño de bases de datos

Ahora que ya hemos visto cómo se estructura y gestiona una base de datos relacional, veremos cómo se deben organizar los datos para obtener el mejor resultado. Repasaremos y haremos una introducción al diseño de bases de datos para poder entender cómo se almacenan los datos geográficos dentro del modelo relacional, aunque este tema se debería cubrir con asignaturas enteras, dada su amplitud.

Mediante tablas y relaciones entre tablas, se pueden representar realidades complejas en forma de modelo. Un modelo de datos es una estructura de tablas y relaciones entre ellas.

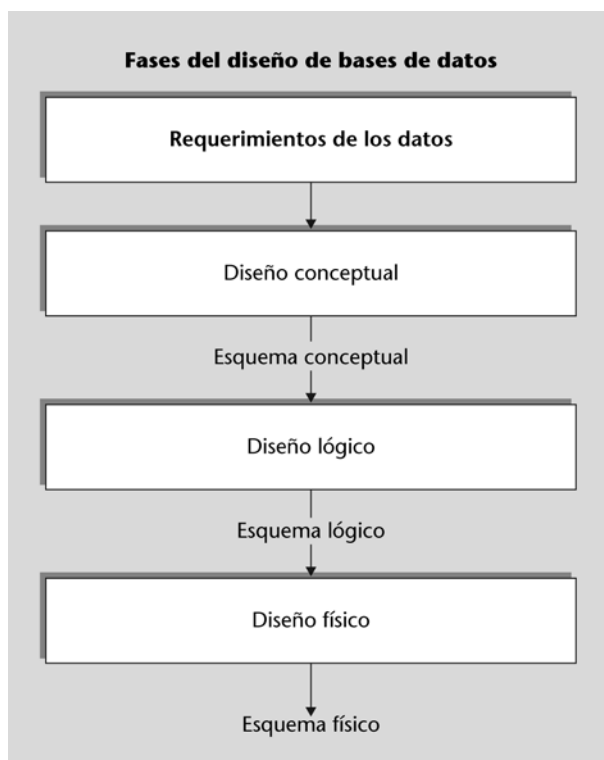
2.8.1. Fases de diseño de una base de datos

El diseño de bases de datos es el proceso de creación de un modelo de datos detallado para representar una realidad. El proceso consta de las siguientes fases:

- 1) Determinación de los datos que se van a almacenar
- 2) Diseño conceptual: determinar las relaciones entre los datos
- 3) Diseño lógico: representar en un modelo lógico los datos y sus relaciones
- 4) Diseño físico: convertir el modelo lógico en un diseño físico de base de datos

La siguiente figura muestra la relación entre las diferentes fases del diseño que se explican a continuación.

Figura 4. Fases del diseño de una base de datos. Enfoque orientado a los datos.



- **Requerimientos de los datos.** Para determinar qué datos se quieren almacenar, se ha de determinar qué conceptos se deben almacenar, qué atributos queremos guardar y qué atributos diferencian un elemento de otro. De esta manera, podremos determinar qué tablas se necesitan, con qué campos y qué clave primaria.
- **Diseño del modelo conceptual.** El modelo conceptual se refiere a la forma como están caracterizados los elementos del mundo real; es decir, parte de los requerimientos y su resultado es el esquema conceptual de la base de datos. Un **esquema conceptual** es una descripción de alto nivel de la estructura de la base de datos, independiente del software de SGBD que se use para manipularla.

Los diagramas de datos más usados para el diseño conceptual de base de datos son los diagramas entidad-relación (ER), los de lenguaje de modelado unificado (UML, *unified modeling language*) y las técnicas de modelado de objetos (OMT, *object modeling techniques*).

- **Diseño del modelo lógico.** El modelo lógico es el desarrollo completo y detallado del modelo conceptual en el que se encuentra una descripción detallada de cada una de las entidades, el diseño de las tablas con sus atributos, identificadores, relaciones, tipo de datos, que constituyen la base de datos. Es decir, tiene como origen el esquema conceptual y da como resultado un esquema lógico. Un esquema lógico es una descripción de la estructura de una base de datos que puede procesar el software de SGBD.

Al igual que el modelo conceptual, es independiente de los programas y equipos que se vayan a utilizar y de su correcta concepción depende el éxito de la base de datos.

- **Diseño del modelo físico.** El modelo físico corresponde a la implementación de la base de datos en un programa o software específico. El modelo físico convierte las relaciones del modelo lógico a tablas, y las relaciones foráneas siguiendo el modelo relacional. Esto se conoce como esquema físico. Un esquema físico es una descripción de la implantación de la base de datos; describe las estructuras de almacenamiento y los métodos usados para tener un acceso efectivo a los datos. Depende, por tanto, del SGBD con el que se trabaje.

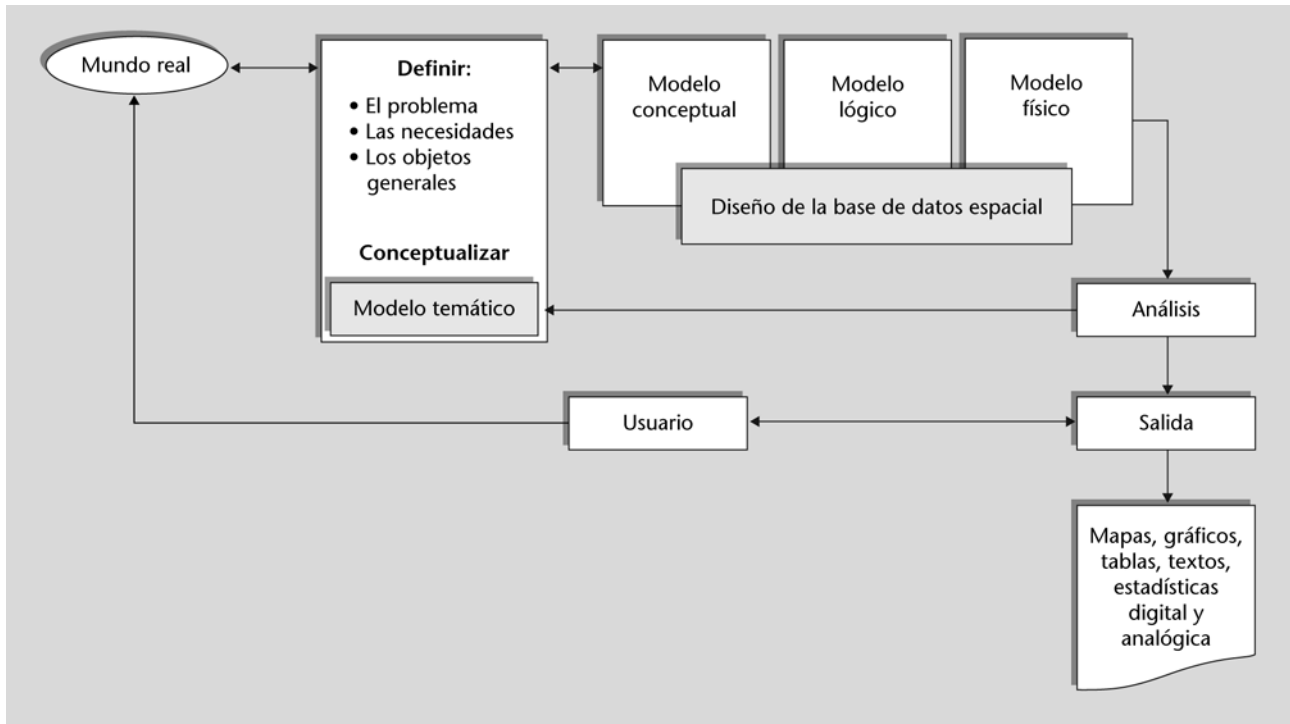
WEB

Podéis consultar un diagrama UML de ESRI correspondiente a un caso de estudio de datos náuticos en la siguiente página web: <http://dusk.geo.orst.edu/djl/arccgis/diag.html>

Para el diseño de una base de datos se debe tener en cuenta el desarrollo de su estructura, la definición de sus contenidos y la determinación de los datos. Dicho de otro modo: el diseño conceptual, el diseño lógico y el diseño físico.

La siguiente figura resume todos los conceptos explicados en este apartado:

Figura 5. Fases del diseño de una base de datos (espacial).



2.9. Normalización

El diseño de bases de datos permite producir modelos variados para una misma realidad. Algunos de estos modelos pueden ser incorrectos, producir anomalías en los datos almacenados y, de esta manera, comprometer la consistencia y el valor de los datos.

La normalización de las bases de datos es una técnica para diseñar bases de datos que establece diferentes niveles incrementales para determinar la consistencia y la integridad de un modelo. Estos niveles incrementales se conocen como **formas normales**. Una base de datos normalizada a cierto nivel reduce la duplicación de datos y la inconsistencia que puede implicar errores y falta de integridad. Además, la normalización simplifica el desarrollo y el mantenimiento de la base de datos y favorece su escalabilidad.

Escalabilidad

El escalabilidad es la capacidad de crecimiento de un modelo o software para asumir nuevas funcionalidades.

Por otra parte, la normalización provoca la creación de tablas y amplía el número de vínculos necesarios para obtener una información. Esto penaliza el rendimiento de las consultas en la base de datos.

En consecuencia, el nivel de normalización se debe diseñar en cada caso particular. Como en muchos otros aspectos, encontraréis que decidir el nivel de normalización al diseñar una base de datos no es blanco o negro, sino que se mueve cómodamente en una gama de grises. Para bases de datos con muchas transaccio-

nes puntuales es recomendable más normalización, mientras que para bases de datos muy estáticas podemos reducir la normalización al mínimo.

Normalizamos para intentar evitar que:

- Los datos estén duplicados, es decir, que la misma información esté representada en registros diferentes. Este hecho puede conllevar inconsistencias si se modifica el valor en uno de los registros y no en el otro, lo cual puede provocar la duda de cuál es el dato correcto.
- Al introducir un dato, dependamos de la introducción previa o simultánea de otro dato que no tenemos obligación de saber.
- Al borrar un dato, hayamos de borrar otros que no queremos perder.

Idealmente, una base de datos relacional no debe sufrir las anomalías descritas.

Las formas normales son requisitos incrementales –es decir, que cada uno supone el cumplimiento de los anteriores– que nos aseguran el diseño correcto de una base de datos para no sufrir anomalías.

Las formas normales las empezó a definir Edgar F. Codd proponiendo la primera forma normal. No se aplican sólo a las bases de datos relacionales, sino también a cualquier tipo de base de datos o a cualquier organización de información en general. Se representan habitualmente con las siglas *NF* (*normal form*) precedida del número de forma normal. Por ejemplo, 1NF es la primera forma normal.

Formas normales

No es el objetivo de este módulo que conozcáis las formas normales detalladamente, pero a continuación tenéis su descripción por si estáis interesados:

- **1NF:** bajo la primera forma normal todos los registros deben contener el mismo número de campos.
- **2NF:** bajo la segunda forma normal el valor de un campo no clave no puede depender directamente de un subconjunto de la clave primaria.
- **3NF:** bajo la tercera forma normal el valor de un campo no clave no puede depender directamente de otro campo no clave.
- **BCNF:** bajo la forma normal de Boyce-Codd el valor de un campo clave no puede depender directamente de otro campo clave.

Ejemplo de aplicación de formas normales

Veamos un caso práctico de la diferencia que aportan las formas normales. Imaginad que necesitamos almacenar datos de exámenes. Nos interesa almacenar:

- El nombre del alumno que realiza el examen.

- Las asignaturas que cursa el alumno.
- Los idiomas que habla el alumno.
- La nota que ha sacado en el examen.
- En qué aula se ha hecho el examen.
- Cuánta gente cabía en el aula.
- El número de alumnos matriculados en la asignatura.
- De qué asignatura es el examen.

Un primer diseño de bases de datos podría ser una tabla con la siguiente definición: Exámenes (***Asignatura***, ***Alumno***, *Nota*, *Aula*, *Capacidad_aula*, *Numero_alumnos*, *Asignatura_alumno*, *Idioma_alumno*).

Este primer diseño no está normalizado y puede provocar varias anomalías al no cumplir las formas normales en los puntos siguientes:

- No cumple con la 2NF, ya que el campo *Numero_alumnos* depende directamente de un subconjunto de la clave primaria (***Asignatura***).
- No cumple con la 3NF, ya que el campo *Capacidad_aula* depende directamente de otro campo no clave (*Aula*).

En consecuencia:

- El valor de la capacidad de un aula estaría repetido tantas veces como exámenes se realizaran.
- El valor del número de alumnos matriculados estaría repetido por tantos alumnos como hicieran el examen.
- Seríamos incapaces de almacenar más de una asignatura e idioma por alumno.

Aplicando las formas normales llegaríamos a un diseño como:

- Exámenes (***Asignatura***, ***Alumno***, *Nota*, *Aula*)
- Aulas (***Aula***, *Capacidad_aula*)
- Asignaturas (***Asignatura***, *Numero_alumnos*)
- Alumnos (***Alumno***)
- Asignaturas_alumno (***Asignatura***, ***Alumno***)
- Idiomas_alumno (***Idioma***, ***Alumno***)

Con este modelo la información no se repite y no se producirán anomalías.

3. Bases de datos geográficos

Ahora que hemos repasado el almacenamiento de datos en general, veremos cómo se almacenan los datos geográficos. En éste y los próximos apartados veréis cómo se almacenan datos vectoriales, topología, segmentación lineal imágenes *raster* y coberturas.

Tal como hemos visto en el apartado anterior, hay dos maneras de almacenar datos: los ficheros y las bases de datos. Todos los datos geográficos se pueden guardar tanto en un modelo como en el otro. En este apartado veréis que, aunque los datos se pueden almacenar en ficheros, las herramientas de los sistemas de información geográfica explotan y consultan los datos como si se almacenaran en bases de datos. Veremos primero cómo se almacenan en bases de datos y después veremos las particularidades para los ficheros.

Los datos geográficos son la representación digital de entidades, objetos o fenómenos que ocurren sobre la superficie de la Tierra o cerca de ésta.

Las bases de datos geográficos deben almacenar y permitir la exportación, modificación y consulta de entidades que recopilen información tanto geográfica como alfanumérica. Un elemento geográfico es, pues, el conjunto de la geometría que lo representa sobre el territorio y los atributos alfanuméricos que describen las propiedades.

Básicamente, hay dos maneras de modelizar la realidad:

- Almacenamiento de datos vectoriales
 - **Objetos puntuales.** Son objetos que se encuentran en un punto concreto del territorio. Por ejemplo, un árbol, la posición actual de un coche o un restaurante. Objetos de dimensión cero.
 - **Objetos lineales.** Son objetos que se distribuyen linealmente sobre el territorio. Por ejemplo, un río, una carretera, una red de distribución eléctrica o una curva de nivel. Objetos de dimensión uno.
 - **Objetos área.** Son objetos que ocupan un área de territorio. Por ejemplo, el mar, una zona boscosa, una parcela agrícola o las zonas verdes de una ciudad. Objetos de dimensión dos.
- Almacenamiento de datos *raster*
 - **Coberturas.** Información geoespacial que representa fenómenos que varían en el espacio. Por ejemplo, la cantidad de nitratos del suelo, la cota o

Diferentes tipos de representaciones

La misma realidad se puede representar con diferentes tipos de geometría según el nivel de detalle que se quiera. Un río se puede representar como una línea o como un área. Un restaurante puede ser un punto o algo mucho más complejo.

En este apartado veremos con detalle el almacenamiento de datos vectoriales. El almacenamiento de datos *raster* se trata en el apartado 6.



la calidad del aire. Es un fenómeno en el que cada punto del territorio tiene un valor diferente.

- **Imágenes *raster*:** Imágenes o fotografías aéreas que se sitúan sobre el territorio.

Todos los datos geográficos tienen en común una localización sobre la Tierra que puede ser capturada y almacenada, y unos atributos y propiedades que caracterizan estos datos.

Los datos geográficos, al igual que los datos simplemente alfanuméricos, se pueden guardar en multitud de formatos y modelos. Elegir un formato u otro es tarea de los diseñadores y analistas y en muchas ocasiones es la clave del éxito de un proyecto SIG.

Existe una gran diversidad de formatos sobre ficheros. Los archivos CAD, los archivos *raster*, el formato GML, el formato KML, el formato *shapefile*, etc. La tendencia actual, sin embargo, es tratar todas las fuentes de datos como si fueran tablas de bases de datos. El modelo relacional de tabla con campos y registros lo encontramos en cualquier fuente de datos geográfica y las herramientas de explotación utilizan SQL para acceder a los datos, tanto si están dentro de un sistema gestor de bases de datos como si no.

Por ejemplo, podemos ejecutar una consulta SQL filtrante por nivel o color sobre un archivo CAD (DGN de Microstation, por ejemplo) usando las herramientas que nos da un SIG comercial como ArcMap o Geomedia.

En este apartado aprenderéis las diferentes opciones de almacenamiento vectorial disponibles con el fin de poder elegir aquel formato que más se adapte a vuestras necesidades.

Ahora veréis qué dificultades se han de superar para almacenar datos geográficos en formato vectorial. Después veréis cómo se almacenan los diferentes formatos vectoriales y, finalmente, haremos un repaso de los formatos de almacenamiento de datos geográficos más relevantes para ver sus particularidades.

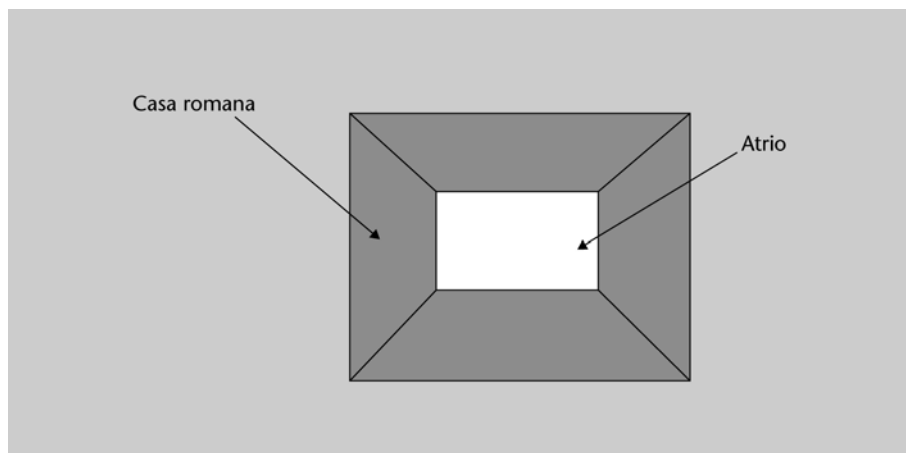
3.1. Almacenamiento de datos vectoriales

Tal como habéis visto en el apartado anterior, con el fin de sacar jugo a las bases de datos relacionales es muy importante su diseño. Las geometrías que representan los elementos sobre el territorio son realidades muy variadas y difíciles de modelizar relacionamente.

Para que os hagáis una idea, tomaremos, por ejemplo, el área constructiva de una casa romana. Si la vemos desde el aire, el área de la casa es un polígono

con un agujero en medio, el atrio o patio central de la casa. Podéis ver una representación de la vista aérea en la figura siguiente:

Figura 6. Casa romana desde el aire



Si diseñamos una base de datos de casas romanas, cada uno de los registros de esta tabla deberá contener una geometría. Por lo tanto, la geometría de la casa son dos polígonos, uno para la zona exterior de la casa y otro para el agujero del atrio. Cada polígono es un conjunto de coordenadas sobre el territorio (eso siempre que no tenga paredes curvas). Cada punto es un conjunto de dos o tres coordenadas (en función de si queremos guardar también la cota). A este tipo de almacenamiento lo llamamos *vectorial*.

Esta realidad es representable en un modelo relacional, de forma compleja pero no imposible. El problema reside en el hecho de que la consulta para ver el gráfico de la casa implicaría entre 5 y 6 tablas con relaciones complejas entre ellas y un número de registros que depende de la complejidad de la geometría.

El almacenamiento de geometrías vectoriales en bases de datos relacionales en forma normal crea un número elevadísimo de tablas que reduce el rendimiento de las consultas y complica el mantenimiento de los datos hasta niveles intolerables. La gran variedad de posibilidades geométricas provoca que el modelo relacional normalizado sea demasiado complejo.

En resumen, almacenar las geometrías en un modelo relacional no es práctico y es muy complejo. Por ello, los SGBD no optan por esta opción.

Todos los SGBD han optado por soluciones mixtas en las que se modeliza con tablas y relaciones los datos alfanuméricos, mientras que las geometrías se guardan en unos campos de tipos complejos o binarios asociados a los datos alfanuméricos.

3.2. Almacenamiento de geometrías vectoriales

Como ya hemos comentado, las geometrías, tanto si se almacenan en ficheros como en bases de datos relacionales, se guardan en formatos binarios complejos. A pesar de haber un estándar definido dentro de los estándares OGC, *simple feature type*, hay multitud de formatos diferentes que tratan de abordar las problemáticas de almacenamiento de manera ligeramente diferente. Muchos de ellos no solucionan todos los problemas, por lo que tienen limitaciones a la hora de guardar algunas geometrías y esto provoca errores al migrar los datos entre formatos.

OGC

El tipo de entidad simple (*simple feature type*, OGC) es el estándar para el almacenamiento de información geográfica.

No es objetivo de este módulo que seáis capaces de diseñar un nuevo formato que solucione todos los problemas, pero sí que lo es que entendáis la complejidad de las geometrías para que seáis capaces de entender y evitar los problemas en el almacenamiento y la migración de datos geográficos.

Cada tipo de geometría tiene sus propias características y, por lo tanto, abordaremos la descripción de cada uno por separado. Aquí sólo trataremos el caso de almacenar geometrías vectoriales: puntos, líneas y áreas; y dejaremos para el apartado 5 el almacenamiento de coberturas e imágenes *raster*.

3.2.1. Almacenamiento de puntos

Un elemento puntual representa una realidad geoméricamente discreta y mínima sobre el territorio. Se puede almacenar con un solo conjunto de coordenadas. Las coordenadas dependen del sistema de coordenadas aplicado.

- Para sistemas proyectados
 - Coordenada X: representa el desplazamiento desde el origen de coordenadas en el eje X.
 - Coordenada Y: representa el desplazamiento desde el origen de coordenadas en el eje Y.
- Para sistemas geográficos
 - Latitud: representa la latitud del punto.
 - Longitud: representa la longitud del punto.
- Aplicables a los dos sistemas de coordenadas
 - Coordenada Z: representa la cota o altura sobre el terreno del punto.
 - Coordenada M: representa la medida para el almacenamiento lineal (LRS).



El LRS se trata en el apartado 5.

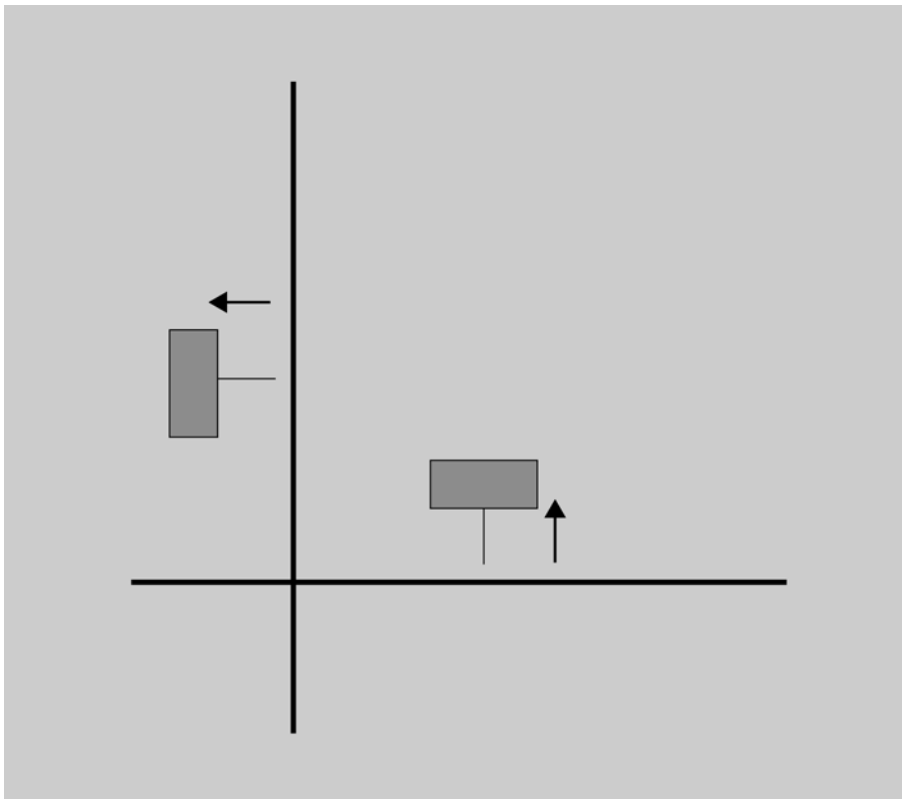
Algunas realidades puntuales requieren además la orientación para poder representar el punto sobre un mapa, lo que denominamos *geometría de puntos*

orientados, y ampliar la geometría con la orientación en grados o radianes de cada punto.

Ejemplo

Queremos representar en una base datos las paradas de autobús de Barcelona y realizar un mapa sobre la cartografía de la ciudad donde se representen las paradas con una señal ortogonal en la calle. No sólo será necesario guardar el punto donde se encuentra la parada, sino también la orientación, para poder representar la señal de manera que quede perpendicular a la calle en el mapa. La orientación del punto nos permite representar un símbolo sobre el punto con una orientación concreta. Podéis ver una representación de esto en la figura 7.

Figura 7. Paradas orientadas perpendicularmente a la calle



Como podéis ver, la representación de un punto no es muy compleja. Por ello hay formatos que almacenan las coordenadas como campos de una tabla dentro del modelo de datos relacional. Sin embargo, por unidad y consistencia con el resto de geometrías, la mayoría de formatos almacenan los puntos también en formato binario dentro de un campo.

3.2.2. Almacenamiento de líneas

En general todo elemento lineal se puede representar como un conjunto de segmentos donde cada segmento se puede definir como:

- **Polilínea.** Una polilínea es un conjunto ordenado de puntos sucesivos unidos por un segmento entre cada punto y el siguiente. Cada punto se representa de la forma descrita en el apartado anterior. Podéis ver un ejemplo de polilínea en la figura 8.

Figura 8. Ejemplo de polilínea



- **Arco de circunferencia.** Un arco de circunferencia es un conjunto de tres puntos que definen los extremos del arco y el centro de la circunferencia donde se encuentra este arco. En un arco de circunferencia, si el punto inicial y el punto final coinciden tenemos una circunferencia. En la figura 9 podéis ver un arco de 90 grados.
- **Curvas de Bézier.** Una curva de Bézier es una curva definida por un conjunto de puntos de paso y puntos de control que determinan el grado. En la figura 10 podéis ver una curva cúbica de Bézier con sus dos puntos de control.

Una curva de Bézier lineal (con ningún punto de control) es una polilínea; aumentando los puntos de control conseguimos más suavidad en el trazado, pero más complejidad de cálculo.

Tanto los arcos como las curvas de Bézier permiten una descripción vectorial sencilla de trazados muy esmerados. Esto reduce el espacio necesario para almacenar geometrías curvas.

Estas técnicas, sin embargo, complican el proceso de representación de la geometría y pueden moderar la representación de mapas. Por ello algunos formatos sencillos no toleran este tipo de representación vectorial y se limitan a las polilíneas. Una polilínea puede aproximarse al trazado de la curva con el nivel de precisión que se quiera trazando puntos por encima de la curva o arco. Este proceso se denomina *trazar* (o en inglés *stroke*).

Problemas al almacenar líneas

Hay geometrías que, a pesar de cumplir la definición de polilínea, presentan problemas de representación o inconsistencias. Dependiendo del fabricante y las herramientas que utilicemos, esto puede llevar a geometrías corruptas. En general los problemas que pueden surgir deberían estar controlados por las herramientas comerciales, pero estos casos son los que aparecen más a menudo en las listas de problemas y mejoras de las nuevas versiones de las herramientas. Para las líneas os podéis encontrar principalmente con los problemas siguientes:

- **Puntos repetidos.** Se da cuando dentro de la serie de puntos de una polilínea un punto se repite dos veces seguidas. En la figura 11 podéis ver cómo el punto más oscuro está repetido.

Figura 9. Ejemplo de arco

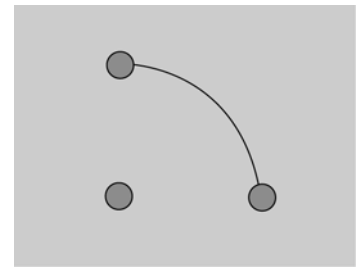
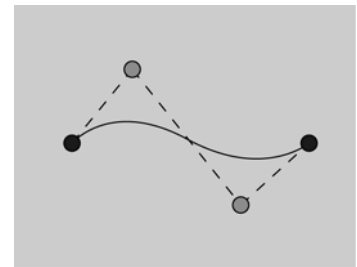


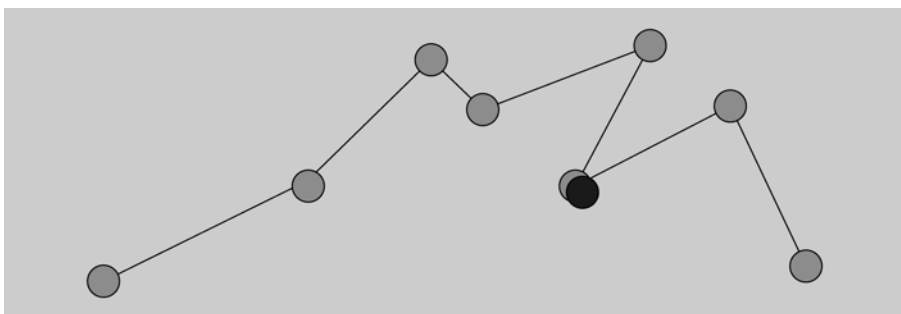
Figura 10. Ejemplo de curva de Bézier



Proceso de trazar

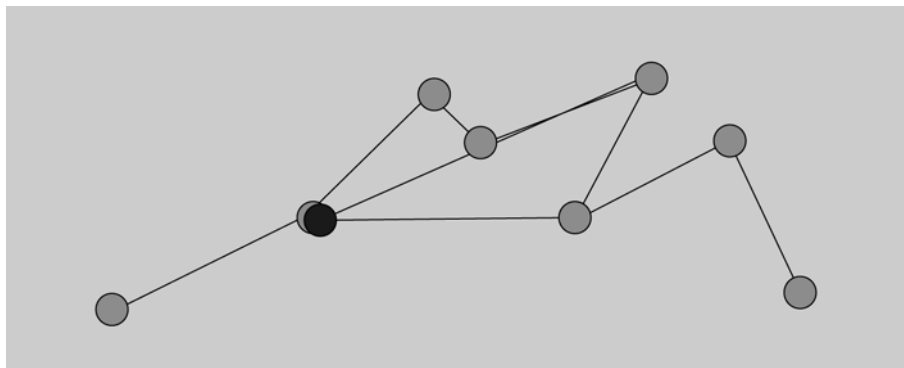
El proceso de trazar es muy habitual al pasar información de archivos CAD a formato SIG, es decir, de un almacenamiento en fichero a un almacenamiento en base de datos cuando el rendimiento es más importante que la precisión.

Figura 11. Ejemplo de puntos repetidos



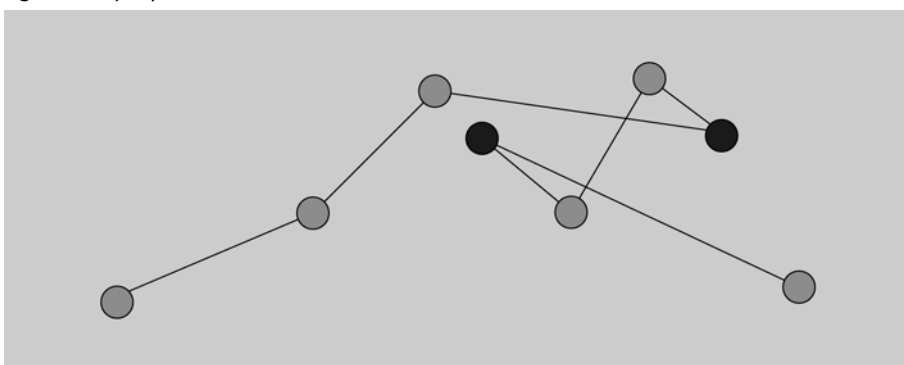
- **Kickbacks** (o vueltas atrás). Se da cuando dentro de la serie de puntos de una polilínea se repiten unos o más puntos anteriores. En la figura 12 podéis ver cómo el punto más oscuro es el mismo que un punto anterior y provoca una desviación anormal del trazado.

Figura 12. Ejemplo de vuelta atrás



- **Loops** (o bucles). Se da cuando dentro de la serie de puntos de una polilínea por alguna razón, se han intercambiado las posiciones ordenadas de dos o más puntos. En la figura 13 podéis ver que los dos puntos más oscuros se han intercambiado las posiciones que tenían en la figura 8 y han creado ciclos al trazado.

Figura 13. Ejemplo de bucle



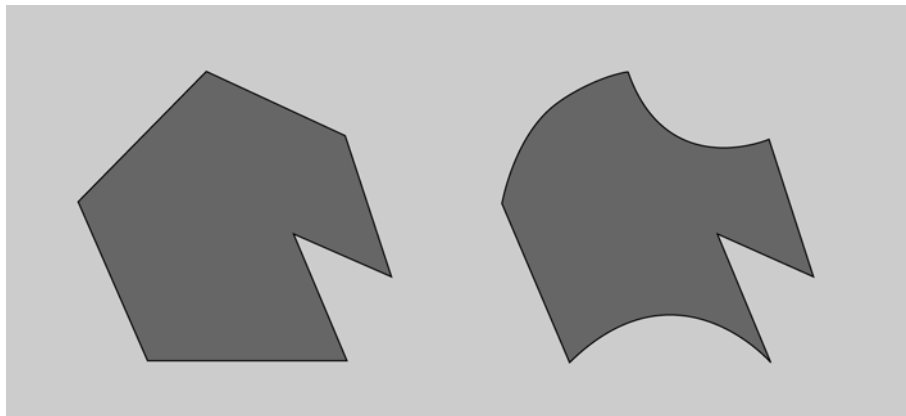
Estos casos aparecen principalmente en los procesos de migración y conversión de datos. Hay muchas herramientas de validación y verificación de geometrías para intentar evitarlos. Tened en cuenta estos casos cuando estéis manejando geometrías vectoriales, ya que una verificación os puede salvar de horas de quebraderos de cabeza por un mal funcionamiento o falta de rendimiento. Las bases de datos no suelen controlar estos casos de forma nativa, por lo que la responsabilidad de que no suceda recae en las herramientas SIG que veréis más adelante.

3.2.3. Almacenamiento de áreas

Las áreas se almacenan en polígonos. La definición matemática de polígono es: un **polígono** es una figura geométrica plana y cerrada formada por un número finito de segmentos. Sin embargo, esta definición no nos permite que los

segmentos sean curvas y por ello se extiende el concepto hablando de polígonos compuestos cuando éstos son figuras geométricas planas y cerradas formadas por un número finito de líneas. Notad que hemos ampliado el concepto segmento por el de línea. En la figura 14, matemáticamente sólo es polígono el dibujo de la izquierda; en cambio, las bases de datos geográficos entienden los dos dibujos como polígonos.

Figura 14. A la izquierda, un polígono matemático y a la derecha, un polígono con curvas en los lados



A pesar de esta ampliación, el concepto de polígono sigue quedando corto para representar entidades de área sobre el territorio. Por ejemplo, si queremos representar el área que ocupa el país de Italia en una sola geometría, lo podemos representar con un polígono compuesto con cierto grado de precisión vectorial. Para ser políticamente correctos, sin embargo, le deberemos restar el Vaticano y San Marino. Es decir, tendremos un área con agujeros.

El uso del concepto de polígono en las bases de datos geográficos ha sido deformado hasta entender un polígono como un concepto mucho más amplio de lo que un matemático toleraría. Tened en cuenta esta deformación del concepto al trabajar con bases de datos geográficos con el fin de entender el tipo de geometrías que se pueden almacenar.

Dentro de los sistemas de información geográficos, un polígono es una superficie plana definida por una frontera exterior y una o más fronteras interiores que definen los agujeros. Cada frontera es un conjunto ordenado de líneas conectadas donde el extremo final de la última línea conecta con el extremo final de la primera. Cada línea puede estar definida con cualquiera de las herramientas para almacenar líneas vistas en el apartado anterior.

Para almacenar áreas no conexas nos hace falta, en ocasiones, más de un polígono y entonces diríamos que un área está formada por un conjunto de polígonos en su visión más amplia. Muchas veces denominamos el conjunto de polígonos *multipolígono*. Esta agrupación nos permite almacenar en un único registro de la base de datos superficies sobre la tierra no conexas y con agujeros.

En la figura 15 podéis ver un área formada por un multipolígono. El multipolígono contiene 3 polígonos, de los cuales uno es un círculo, uno tiene un agujero y el otro no tiene ninguno.

Figura 15. Ejemplo de multipolígono



Ejemplo de multipolígono

Para representar el área de Baleares, necesitaremos un polígono para Mallorca, uno para Menorca, uno para Formentera, etc. El conjunto de todos los polígonos se representará en la base de datos como un solo registro.

3.2.4. Almacenamiento de geometrías complejas

Hay realidades de las que queremos guardar información alfanumérica que tiene una representación sobre el territorio compleja y de distintos tipos a la vez (puntual, lineal y áreas). Este tipo de representación compleja se evita, ya que en un mapa siempre se han buscado representaciones sencillas que eviten este tipo de problemas.

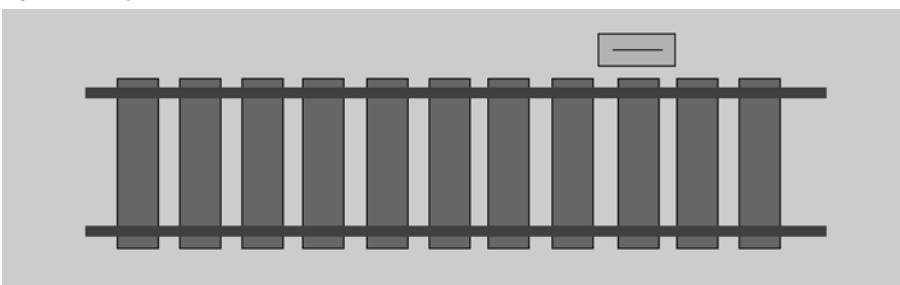
Las geometrías compuestas son una colección de geometrías del mismo o diferente tipo.

No todos los formatos de almacenes geográficos permiten geometrías compuestas.

Siempre que podáis, tenéis que evitar este tipo de geometrías, ya que afectan al rendimiento de representación del mapa y os pueden acarrear problemas a la hora de migrar o trabajar con los datos.

Ponemos por caso que queréis representar las vías del metro y os queréis guardar información de cada tramo de vía, como el punto kilométrico, el responsable del mantenimiento, entre qué paradas está, etc. La representación, sin embargo, la queréis bastante similar a la realidad y que incluya las vías, los listones, las agujas, etc. Un tramo de vía podría aparecer como se muestra en la figura 16.

Figura 16. Representación de las vías del metro



Esta geometría contiene varias áreas y varias líneas a la vez. Sería aconsejable, si queréis seguir las buenas prácticas de las bases de datos geográficos, representar el tramo con una sola línea y la aguja con un punto. El tramo se podría simbolizar con un patrón que simulara la vía del tren, mientras que la aguja se puede representar con un símbolo.

3.3. Modelos de datos de almacenamiento

Aparte de la geometría de los datos geográficos en sí misma, hay otra información que los SIG necesitan de cara a poder explotar los datos. Con el fin de entender una geometría dentro del territorio, necesitamos saber qué representa, de qué tipo es, con qué sistema de coordenadas está medida, etc. Llamamos a este tipo de información *información geográfica de apoyo*.

Dentro de cada formato de almacenamiento se debe definir cómo se guarda la lista de capas que contiene, los sistemas de coordenadas, cuáles son los atributos que representan la geometría, etc.

Algunas de las problemáticas sobre cómo modelizar la información geográfica de apoyo están solucionadas por el estándar *simple feature type* (tipo de entidad sencilla) de OGC, pero en general ningún formato lo sigue y el estándar no abarca todas las posibilidades.

A continuación, veréis qué tipos de datos geográficos de apoyo se deben guardar.

Información geográfica de apoyo

Un SIG, al conectarse a una fuente de datos, debe poder extraer:

- **Lista de capas.** No todas las tablas de la base de datos han de contener información geográfica. Debe existir una lista de qué capas se almacenan en la base de datos, qué nombre tienen y cómo se almacenan. En algunos casos, una capa no corresponde exactamente a una tabla, por lo que además del nombre de la tabla es necesario guardar qué campos forman parte de la capa, por ejemplo.
- **Sistemas de coordenadas.** Hay una gran variedad de sistemas de coordenadas, pero los SIG siempre dan la libertad al usuario para que se defina uno nuevo. Como el usuario puede definir nuevos, los parámetros de los sistemas de coordenadas (o como mínimo la referencia al sistema de coordenadas que se usa) se deben guardar en la base de datos. Para cada capa se deberá guardar también sobre qué sistema de coordenadas está referenciada.
- **Campos donde se guarda la geometría.** Como hemos dicho, en la mayoría de los formatos la geometría se guarda en un campo en formato binario. Muchos formatos dan libertad para definir cómo se llama el campo que contiene la geometría. Algunos permiten incluso tener una tabla con más de un campo de geometría y poder definir varias capas sobre la misma ta-

bla, una para cada campo que contiene una geometría. Así pues, se debe guardar para cada capa cuál es el campo que contiene la geometría y qué tipo de geometría contiene.

Según el formato puede haber otras informaciones que haya que guardar, pero las mencionadas anteriormente son el mínimo esencial.

Cada SGBD y formato tiene su propia manera de almacenar la geometría y el resto de información geográfica, motivo por el que en el subapartado siguiente veréis cuáles son los principales actores y sus formatos de almacenamiento.

3.4. Principales formatos de almacenamiento

Hasta aquí hemos visto que hay distintos problemas para almacenar bases de datos. Para resolverlos hay varias empresas y organizaciones que marcan tendencias y establecen formatos de almacenamiento. Cada compañía tiene su historia y ha ido aportando varios formatos a medida que ha ido mejorando sus productos.

A continuación tenéis una pequeña descripción de los actores principales del mercado, su historia y los formatos que impulsan; esto os ayudará a conocer la variedad de formatos con los que podréis trabajar. El orden de presentación de los formatos se aproxima al orden de aparición histórica para que podáis ver la evolución. Los formatos que estudiaremos son:

- Ficheros CAD (Microstation y AutoCAD)
- General Electric Smallworld
- ESRI Shapefiles
- Geomedia Ware houses
- GML
- ESRI ArcSDE
- PostGIS
- Oracle Spatial

3.4.1. Ficheros CAD (Microstation y AutoCAD)

Los ficheros CAD (*computer aided design* o diseño asistido por ordenador) no sólo almacenan datos geográficos, sino que son el formato de almacenamiento de las herramientas de diseño de arquitectos, ingenieros y diseñadores. Las herramientas CAD son herramientas computacionales de soporte al diseño con geometrías. Las más conocidas y usadas son AutoCAD y Microstation.

Desde finales de los ochenta, las herramientas CAD no sólo permiten diseñar geometrías, sino que también permiten ubicarlas en el territorio georreferen-

AutoCAD:
<<http://www.autodesk.es/adsk/servlet/index?siteID=455755&id=9034707>>

ciándolas. De esta manera, sus ficheros pueden almacenar datos ubicados sobre el territorio.

Cada herramienta CAD tiene un formato propio que normalmente cambia de una versión de la herramienta a otra. Los formatos de archivo CAD más habituales son:

- AutoCAD DXF. Es un formato de archivo CAD diseñado por Autodesk de cara a facilitar la interoperabilidad de AutoCAD con otros programas. Se definió en diciembre de 1982 junto con el lanzamiento de AutoCAD 1.0 y su formato es público.
- DWG. Es un formato de archivo que permite guardar datos de diseños en dos y tres dimensiones y es el formato nativo e interno de trabajo de AutoCAD.
- DGN. Es un formato de archivo CAD con las mismas capacidades que DWG, pero para la herramienta Microstation de Bentley.

*AutoCAD DXF corresponde a las siglas de **drawing interchange format** o formato de dibujo de intercambio.*

*DWG viene de la palabra inglesa **drawing** (o dibujando).*

*DGN viene de la palabra inglesa **design** (o diseñando).*

La estructura interna de almacenamiento de los archivos CAD está orientada al diseño industrial y no responde a la organización de una base de datos relacional. Los ficheros están organizados en capas y cada capa puede contener cualquier dibujo. Los elementos gráficos se almacenan individualmente sin tener que responder a ningún grupo o familia de elementos dentro del archivo. No se pueden almacenar atributos sobre los elementos gráficos dentro de los formatos de archivo CAD.

Este formato de almacenamiento no cumple con muchas de las características generales de los formatos de almacenamiento geográficos, pero es necesario tenerlo en cuenta, ya que:

- Hay mucha cartografía en formato CAD. Por ejemplo, el Instituto Cartográfico de Cataluña (ICC) genera y provee la base topográfica de Cataluña en formato CAD. Es importante, pues, poder importar información de ficheros CAD.
- Una funcionalidad muy solicitada por los usuarios es poder extraer datos SIG en formato CAD para trabajar con estos datos desde las herramientas de ingeniería. Es importante, por tanto, poder exportar información SIG a ficheros CAD.

Instituto Cartográfico de Cataluña:
<<http://www.icc.cat>>

3.4.2. General Electric Smallworld

Smallworld es el nombre de una compañía fundada en 1989 en Cambridge, Inglaterra. Esta compañía desarrolló un SIG muy orientado a las utilidades y la gestión de redes y comunicaciones que llegó a ser el líder del mercado SIG en cuanto a gestión de redes. General Electric compró la compañía y ahora el

nombre de Smallworld se asocia sólo a la aplicación SIG que desarrollaron y a su formato de almacenamiento de datos.

Smallworld está basado en dos tecnologías:

- Se ejecuta sobre el lenguaje Magik, un lenguaje orientado a objetos que se ejecuta sobre una máquina virtual que interpreta el código fuente.
- El formato de almacenamiento está basado en VMDS (*version managed data store*, almacenamiento de datos gestionado por versiones), que es un formato propietario de General Electric diseñado para ser una base de datos relacional orientada a versiones. Una de las capacidades principales de este formato es la capacidad de modificar los datos sobre una versión local que después se consolida contra la versión de datos central.

El formato de Smallworld permite almacenar puntos, líneas, polígonos y *rasters* y tiene amplias capacidades topológicas. Quizá el hecho más característico de este formato es que trata todos los elementos como objetos, no como simples datos. La diferencia radica en el hecho de que los objetos, aparte de sus propiedades, tienen métodos para transformarlos, enlazarlos o, en general, para trabajar con ellos.

Por sus características, este formato es muy usado para representar redes de servicios, ya que está optimizado para almacenar redes topológicas y realizar cálculos sobre estas redes.

Hasta ahora todas las herramientas de importación y exportación del formato de datos Smallworld pasaban por código en Magik, pero últimamente ya se dispone de algunas herramientas de exportación a otros formatos (como Oracle Spatial). El hecho de trabajar sobre un lenguaje interpretado ha creado muchas barreras a los usuarios, ya que exige muchos conocimientos informáticos a la hora de trabajar sobre el sistema y los datos. Las funcionalidades que ofrece el sistema compensan este esfuerzo adicional a muchos usuarios.

3.4.3. ESRI Shapefile

Shapefile es uno de los formatos más populares y sobradamente soportados. Fue diseñado y lo mantiene ESRI como formato interoperable y de intercambio de información entre las herramientas de ESRI y otras herramientas SIG. Un *shapefile* se almacena en varios ficheros, pero su contenido se puede entender como una tabla de una base de datos relacional.

Los datos de un *shapefile* se almacenan en una combinación de varios ficheros:


- **Fichero .shp.** Contiene las geometrías de los elementos.
- **Fichero .shx.** Contiene el índice de los elementos.

- **Fichero .dbf.** Contiene los atributos alfanuméricos de los elementos. Son estos datos los que se pueden considerar como una tabla de base de datos relacional.
- **Fichero .prj.** Contiene el sistema de coordenadas del *shapefile*.
- **Fichero .shx.** Contiene los índices espaciales, si los hay.

Además contiene otros ficheros con información de metadatos y otros índices complementarios. Podéis ver el apéndice A (Metadatos).

De todos éstos sólo son obligatorios los ficheros .shp, .shx y .dbf. Los otros ficheros son opcionales. Al no ser una base de datos relacional propiamente dicha, tiene algunas limitaciones:

- No permite modelos de almacenamiento avanzados que incluyan topología y LRS.
- No permite contener valores nulos.
- No permite el acceso concurrente para la modificación de los datos.



En el apartado 5 de este módulo podéis ver modelos de almacenamiento avanzados.

A pesar de las limitaciones ha llegado a ser, y todavía es, uno de los formatos más extendidos. Los motivos son su simplicidad y su formato abierto. Se ha convertido en un estándar *de facto* y casi todas las herramientas SIG tanto comerciales como de código libre pueden trabajar con este formato.

3.4.4. Geomedia Warehouse

Geomedia es una familia de herramientas SIG provista por Intergraph, una de las compañías “clásicas” de desarrollo e innovación en SIG.

En las herramientas de Geomedia se accede de la misma manera a todos los almacenes de datos SIG, a excepción de los formatos *raster*. Entendemos por almacén Geomedia (Geomedia Warehouse) el conjunto de formatos propios de Geomedia para el almacenamiento de datos. Geomedia también soporta formatos otras compañías, como el formato Shapefile visto en el subapartado anterior, pero no los podemos considerar almacenes propios de la herramienta.

Entendemos por Geomedia Warehouse (o almacén Geomedia) un conjunto de formatos diferentes con la característica común de ser formatos propietarios de Geomedia y poder ser leídos con sus herramientas comerciales. Ahora veamos cada uno de ellos por separado.

Geomedia SmartStore

SmartStore es un formato de almacenamiento en fichero en formato binario orientado a minimizar el tiempo de consulta de los datos. Es un formato sólo accesible mediante herramientas Geomedia pero el hecho de trabajar con es-

tas herramientas nos puede ayudar a mejorar el rendimiento de una aplicación. De la misma manera que el formato *shapefile*, se compone de un conjunto de archivos:

- Archivo DDC y DTC: almacenan los datos y son obligatorios.
- Archivo CSF: almacena el sistema de coordenadas y es opcional.

Geomedia Access Warehouse

El formato Acces Warehouse permite almacenar datos geográficos dentro de un archivo de Microsoft Access. Según su fabricante y de cara al usuario, Access es un SGBD, pero su motor de bases de datos, el hecho de que se guarde en un fichero y sus pocas capacidades provocan que muchos no lo consideren realmente un SGBD. Estas mismas características hacen, sin embargo, que sea muy cómodo y sencillo trabajar con este formato, ya que se dispone de las mismas herramientas de consulta que en un SGBD (SQL, diseño de tablas, etc.) con la comodidad de un fichero.

Geomedia define un modelo de datos de almacenamiento (recordad el subapartado 3.4) dentro del archivo Access que permite definir en qué tablas se guardan los atributos de las capas y en qué campo de la tabla se guarda la geometría de cada capa. La geometría se guarda en formato binario en un campo de tipo BLOB (Binary, etc.) que sólo saben interpretar las herramientas de Geomedia.

Geomedia Oracle Warehouse

El formato Oracle Warehouse responde al mismo patrón que el anterior, pero sobre una base de datos Oracle. Oracle es ya un SGBD reconocido y de gran potencia. El modelo de datos de almacenamiento es similar al definido sobre Microsoft Access y también guarda la geometría en un campo de tipo binario. Como después veréis, Oracle ha definido su propio formato pero Geomedia ya permitía guardar datos SIG en formato Oracle antes de que se definiera este formato. Actualmente, Geomedia es capaz de trabajar tanto con el formato propio de Geomedia Warehouse como con el formato propio de Oracle.

Hay otros formatos en otros SGBD, como SQL-Server, pero todos responden al mismo modelo de datos de almacenamiento.

3.4.5. GML

GML (*geography markup language* o lenguaje de marcas geográfico) es una gramática XML definida por OGC para almacenar y transmitir información geográfica. GML se ha establecido como el estándar para el intercambio de información geográfica en las transacciones por Internet. Una de las claves del éxito de GML, aparte del hecho de ser un estándar OGC, es la habilidad de in-

WFS

El servicio en red de entidades (*web feature service*, WFS) de OGC es un estándar de servicio en Internet para el intercambio de datos geográficos en formato GML.

tegrar todos los tipos de datos geográficos, desde objetos discretos o información vectorial hasta *rasters* o coberturas.

GML define dos conceptos principales. Por una parte, las entidades (o *features*) son objetos que representan una realidad física, por ejemplo, un edificio, una carretera o un árbol. Por otra parte, los objetos geométricos (o *geometry objects*) definen una localización o región. Muchas herramientas SIG hacen convergir estos dos conceptos en uno solo donde las entidades tienen un objeto geométrico que las ubica en el mundo. GML es más flexible y permite que una entidad no tenga varios objetos geométricos o ninguno que la representen. GML se usa principalmente sobre Internet como formato tanto para el intercambio como para la modificación de información geográfica con el servicio WFS de OGC.

En el ejemplo siguiente de GML podéis ver cómo se almacenan un polígono cerrado, un punto y una polilínea:

```
<gml:Polygon>
  <gml:outerBoundaryIs>
    <gml:LinearRing>
      <gml:coordinates>0,0 100,0 100,100 0,100
      0,0</gml:coordinates>
    </gml:LinearRing>
  </gml:outerBoundaryIs>
</gml:Polygon>
<gml:Point>
  <gml:coordinates>100,200</gml:coordinates>
</gml:Point>
<gml:LineString>
  <gml:coordinates>100,200 150,300</gml:coordinates>
</gml:LineString>
```

En este ejemplo podéis ver cómo se almacena un polígono (<gml:Polygon>) como una frontera exterior (<gml:outerBoundaryIs>) definida por una polilínea (<gml:LinearRing>) formada por una serie de coordenadas (<gml:coordinates>). También podéis ver cómo se almacena un punto (<gml:Point>) y una polilínea. Las coordenadas se separan entre ellas con una coma y cada par de coordenadas dentro de una serie se separa con espacios.

3.4.6. ESRI ArcSDE

El ESRI (Environmental Systems Research Institute) es una de las compañías líder en el mercado del SIG, sobre todo para administraciones públicas y ges-

ción ambiental. El paquete de herramientas SIG de ESRI se denomina ArcGIS. Está formado por una decena de productos con el prefijo *Arc*. Hay varias herramientas dentro de ArcGIS. Las principales son:

- ArcMap: herramienta de visualización y análisis de datos geográficos.
- ArcCatalog: herramienta de tratamiento y organización de datos geográficos.
- ArcIMS: servidor de mapas por Internet.
- ArcGIS Server: servidor de mapas y procesamiento geográfico por Internet.
- ArcSDE: conector del resto de herramientas con almacenes sobre bases de datos relacionales.

Estas herramientas se pueden adquirir para diferentes usos más avanzados o más simples. Por ello hay diferentes niveles de licencia, que son:

- ArcView.
- ArcEditor.
- ArcInfo.

Si alguien os dice que usa ArcView, de hecho trabaja con ArcMap con un conjunto limitado de funcionalidades. De la misma manera, si trabajáis con ArcInfo, trabajáis sobre ArcMap, pero ahora con todas las funcionalidades.

ESRI ArcSDE es un conector para establecer la comunicación entre un SGBD y ArcGIS. Esta pieza se encarga de conocer el formato específico de almacenamiento de cada SGBD y ofrecer una interfaz estándar para el acceso a los datos de las herramientas ArcGIS.

ArcSDE permite, de la misma manera que los Geomedia Warehouse vistos en el subapartado anterior, almacenar datos SIG tanto en Oracle como en SQL Server. El modelo de datos de almacenamiento de ArcSDE tiene la particularidad de que permite guardar versiones y consolidarlas (de la misma manera que el formato Smallworld). Esta propiedad es muy útil para el trabajo en delegaciones remotas con una mala conexión a la Red o para el trabajo de campo. Permite llevarse un subconjunto de los datos, modificarlo y después integrarlo dentro de la base de datos corporativa. Durante la integración de los datos se deben resolver los conflictos que puedan aparecer como, por ejemplo, que dos personas diferentes hayan modificado el mismo elemento el mismo día de modo diferente.

La capacidad de versionar de ArcSDE provoca que el modelo de datos de almacenamiento sea un poco más complejo. La geometría no se guarda en un campo binario en la misma tabla donde están los atributos, sino que se guarda en varias tablas internas con las altas, las bajas y las modificaciones. Con el fin de consolidar esta información en la tabla principal, se debe ejecutar un proceso costoso en tiempo.

3.4.7. PostGIS

PostGIS es el primer desarrollo de código libre para el almacenamiento de geometrías sobre una base de datos relacional. Añade el soporte para datos geográficos sobre el SGBD PostgreSQL. PostgreSQL es un SGBD de código libre bajo licencia de uso GNU.

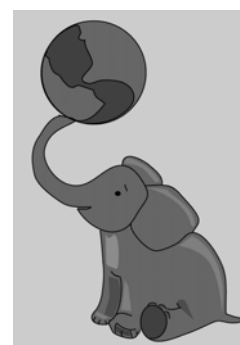
PostgreSQL:
<<http://www.postgresql.org/>>

PostGIS:
<<http://postgis.refrations.net/>>

Podéis entender PostGIS como la opción de código libre similar a ArcSDE o Oracle Spatial que permite que un SGBD se use como base de datos de un SIG.

PostGIS sigue las especificaciones de OGC y ha sido certificado para cumplir el estándar *simple feature type* (o tipo de entidad simple). Ha sido desarrollado por Refrations Research como proyecto de código libre y todavía está en evolución. Aunque no tiene la potencia de otras opciones comerciales, tiene las funcionalidades principales de almacenamiento y es una opción más que válida para muchos proyectos.

Para explotar los datos, es necesario disponer de herramientas SIG de código libre, ya que la mayoría de las herramientas comerciales no pueden leer datos en formato PostGIS. Existen multitud de proyectos de código libre que permiten trabajar con estos datos y convertirlos a otros formatos.



Logotipo de PostGIS

3.4.8. Oracle Spatial

Oracle es el principal proveedor y líder del mercado en SGBD. La primera versión de Oracle Spatial venía como una opción denominada “Spatial Data Option” en la versión Oracle 7. En agosto del 2007 Oracle lanzó al mercado Oracle Spatial 11g. Oracle Spatial se vende como licencia de software complementaria de la licencia básica de Oracle. Incluye herramientas para almacenar datos geográficos y además lleva distintas y potentes herramientas SIG.

La apuesta de Oracle por los SIG ha sido muy fuerte y se ha situado rápidamente entre uno de los formatos de almacenamiento más escalables, fiables y de alto rendimiento de los disponibles.

Una de las ventajas de este formato es que guarda los datos geográficos como un tipo de campo más y extiende SQL para poder realizar consultas espaciales sobre esta información. El hecho de que las consultas espaciales las ejecute un motor de buscas como Oracle de forma nativa y no lo hayan de realizar las herramientas SIG acelera el proceso de filtrado.

Los datos geográficos se guardan en campos en un formato de campo definido para almacenar geometrías (`SDO_GEOMETRY`). Esto permite que se pueda tratar como un campo más de la base de datos y se puedan realizar consultas y filtros.

4. Consulta e indexación de datos geográficos

La organización y el almacenamiento de los datos tiene como objetivo que se puedan consultar y recuperar cuando y como sea necesario. La ordenación de las bases de datos relacionales responde, en gran manera, a ser consultada de modo eficiente para maximizar el rendimiento de las aplicaciones. Para obtener la información que queremos, se deben explotar las relaciones entre los datos para extraer aquel subconjunto que nos interesa.

Por ejemplo, podemos guardar una lista de empresas y una lista de personas con relación a donde han trabajado. Una consulta que permite obtener más información y explota la relación sería obtener una lista de las personas que trabajan en una empresa o la lista de empresas donde ha trabajado una persona.

Las bases de datos geográficos aportan además una riqueza de relaciones superior al añadir el concepto territorial a los datos. De esta manera, dos elementos pueden estar relacionados por el hecho de estar cerca, estar el uno sobre el otro o tocarse, sin necesidad de indicarlo explícitamente con una relación entre tablas.

Por ejemplo, podemos guardar una lista de parcelas agrícolas y una cobertura de nivel de nitratos del suelo. Una consulta geográfica sería obtener todas aquellas parcelas agrícolas que tienen un nivel de nitrato en el suelo superior a un umbral perjudicial para la salud.

Ahora que ya habéis visto cómo se almacenan los datos geográficos, en este apartado veréis qué consultas espaciales se pueden realizar y cómo realizarlas. También veréis cómo se debe ordenar una base de datos relacional con el fin de hacer eficientes las consultas.

Primero veremos las posibles relaciones entre los datos, repasaremos las relaciones entre datos no geográficos y añadiremos las relaciones geográficas. Después veremos los operadores y las funciones que se pueden utilizar para explotar estas relaciones.

Para acceder a los datos y realizar consultas, será necesario ordenar e indexar los datos para un acceso más rápido. Veremos los índices geográficos más usados y, para acabar, veremos cómo realizar consultas geográficas.

4.1. Relaciones espaciales y filtrado

Los datos espaciales pueden tener más tipos de relaciones entre ellos que los datos no espaciales. Antes de ver las relaciones espaciales, cabe recordar las relaciones que son posibles en una base de datos cualquiera.

Físicamente, las bases de datos relacionales sólo pueden tener un tipo de relación, la clave foránea, pero desde el punto de vista conceptual se pue-

den representar diferentes relaciones con estas claves foráneas. Conceptualmente, tenemos las relaciones de herencia, composición, relación de uno a muchos y relación de muchos a muchos. A continuación las vemos con detalle:

- **Herencia.** Tenemos una relación de herencia cuando los elementos de una clase tienen todos los atributos de otra clase más algunos más. Se denomina clase *padre* la clase de la que se heredan los atributos.
- **Composición.** Tenemos una relación de composición cuando un elemento de una clase está compuesto por varios elementos de otra clase.
- **Relación de uno a muchos (1-N).** Tenemos una relación de uno a muchos cuando un elemento de una clase *A* tiene relación con varios elementos de otra clase *B* y cada elemento de la clase *B* sólo tiene relación con un elemento de la clase *A*.
- **Relación de muchos a muchos (N-M).** Tenemos una relación de muchos a muchos cuando un elemento de una clase *Z* tiene relación con varios elementos de otra clase *B* y cada elemento de la clase *B* tiene relación con varios elementos de la clase *A*.

Como ya sabéis, este conjunto de relaciones conceptuales se representan sobre el modelo físico de base de datos mediante tablas con relaciones foráneas entre ellas. Las relaciones espaciales, por contra, no se representan con relaciones sobre el SGBD. Las relaciones espaciales se pueden convertir en relaciones conceptuales realizando un cálculo y añadiendo la relación con claves foráneas. Por ejemplo, una relación espacial como “se tocan” se puede convertir en relación no espacial añadiendo una relación de muchos a muchos entre los elementos que se tocan. Vistas las relaciones generales de datos, veamos seguidamente las relaciones basadas en el territorio.

4.2. Operadores y funciones espaciales

Con el fin de explotar las relaciones espaciales –es decir, basadas en el territorio– disponemos de los operadores y las funciones espaciales. Los operadores y funciones espaciales nos permiten realizar consultas entre los elementos teniendo en cuenta su posición en el mundo. Para poder ejecutar consultas con relaciones espaciales, necesitaréis una herramienta SIG o un SGBD que soporte consultas espaciales.

No todos los SGBD ni herramientas SIG dan soporte a las mismas relaciones espaciales. Para darles soporte, estas herramientas deben ofrecer un conjunto de operadores y funciones que permitan acceder y consultar sobre estas relaciones.

Entendemos por operadores espaciales aquellos métodos que nos permiten preguntar propiedades de las geometrías y nos responden cierto o falso.

Entendemos por funciones espaciales aquellos métodos que nos permiten analizar las geometrías y nos responden con una nueva geometría fruto del cálculo.

La diferencia entre los operadores y las funciones es que los operadores nos responden una pregunta de cierto o falso, mientras que las funciones calculan una nueva geometría a partir de los datos espaciales.

Cuando queráis realizar consultas sobre vuestros datos deberéis revisar la documentación del formato con el que los tenéis almacenados y las herramientas SIG con las que queréis realizar la consulta para ver qué operadores y funciones os ofrecen.

En los siguientes subapartados podréis ver una recopilación de los posibles operadores y funciones que hay disponibles. La lista se basa en el estándar *simple feature access* (tipo simple de entidad) de OGC, pero podéis encontrar algún formato de bases de datos geográficos que añada otras funciones no previstas por el estándar. En particular, veremos los elementos siguientes:

- Operadores: Igual, Disjuntos, Interseca, Toca, Cruza, Incluida, Contiene y Encabalgadas.
- Funciones: Centro, Caja Contenedora, *Buffer*, Envolvente Convexa, Intersección, Unión, Diferencia y diferencia simétrica, Distancia y Vecino más próximo.

4.2.1. Operadores

Los operadores espaciales son métodos booleanos (es decir, que devuelven cierto o falso) que nos indican si se cumple una relación específica entre dos objetos geométricos.

Las relaciones espaciales se describen entre dos elementos *A* y *B* con geometrías sobre el plano. Las operaciones espaciales se calculan después de aplicar el sistema de coordenadas para proyectar las geometrías.

Para entender las relaciones espaciales, se debe tener en cuenta que una geometría tiene un interior, una frontera y un exterior. Hay relaciones que se describen a partir de estos conceptos:

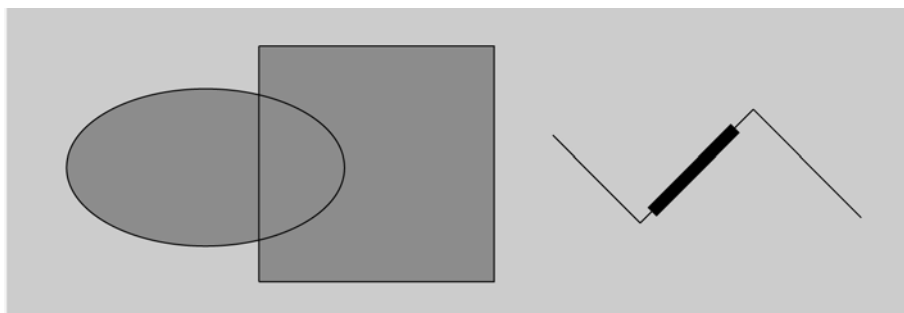
- La frontera es la delimitación del objeto. Es otro objeto geométrico pero de una dimensión inferior. Por ejemplo, la frontera de un polígono de dimensión 2 es una línea de dimensión 1.

- El interior es el área estrictamente incluida, es decir, sin la frontera del objeto. Por ejemplo, el área interior de un polígono o la línea interior de un segmento.
- El exterior es el espacio estrictamente fuera de la frontera.

A continuación, repasamos cada una de las operaciones espaciales:

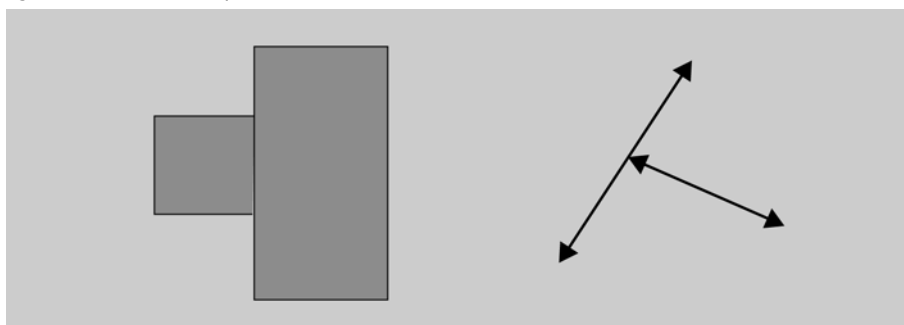
- **Igual:** A es espacialmente igual a B si su geometría es igual.
- **Disjuntos:** A es disjunta de B si las dos geometrías no tienen ningún punto en común en el plano.
- **Se interseca:** A se interseca con B si las dos geometrías tienen algún punto en común en el plano. La relación de intersección es el operador contrario al operador “Disjuntos”. El operador de intersección es bastante general e incluye los resultados de las operaciones “Toca”, “Cruza”, “Incluida” y “Contiene” que veréis a continuación. En la figura 17 podéis ver dos polígonos y dos líneas que tienen la relación de intersección.

Figura 17. Geometrías en intersección



- **Toca:** A se toca con B si el interior de las geometrías no tiene ningún punto en común sobre el plano y las geometrías sí que lo tienen. En la figura 18 podéis ver dos polígonos que se tocan. Para un polígono la frontera es el perímetro, mientras que el área contenida es el interior. En la figura podéis ver cómo los rectángulos comparten parte de la frontera pero no el interior. En el segundo ejemplo podéis ver dos segmentos que se tocan.

Figura 18. Geometrías que se tocan

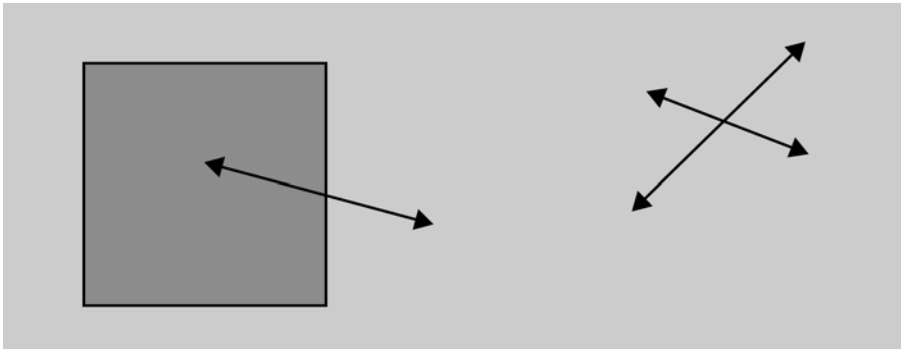


Cruza: A se cruza con B si tienen algún punto en común y la dimensión de la intersección es menor que el máximo de las dimensiones de A y B . Recordad que la dimensión de un punto es 0, la de un segmento o línea es 1 y la de un

polígono es 2. Lo que dice la definición de este operador es que las dos figuras geométricas se intersectan y la dimensión de esta intersección debe ser menor que el máximo de las dimensiones de A y B .

Por ejemplo, en la figura 19 podéis ver un polígono y un segmento que se cruzan; la dimensión del polígono es 2 y la del segmento es 1; por lo tanto, el máximo de dimensiones en este caso es 2. Por otra parte, tenemos que la intersección es una parte del segmento y, por ende, la intersección es de dimensión 1, que es menor que 2.

Figura 19. Geometrías que se cruzan

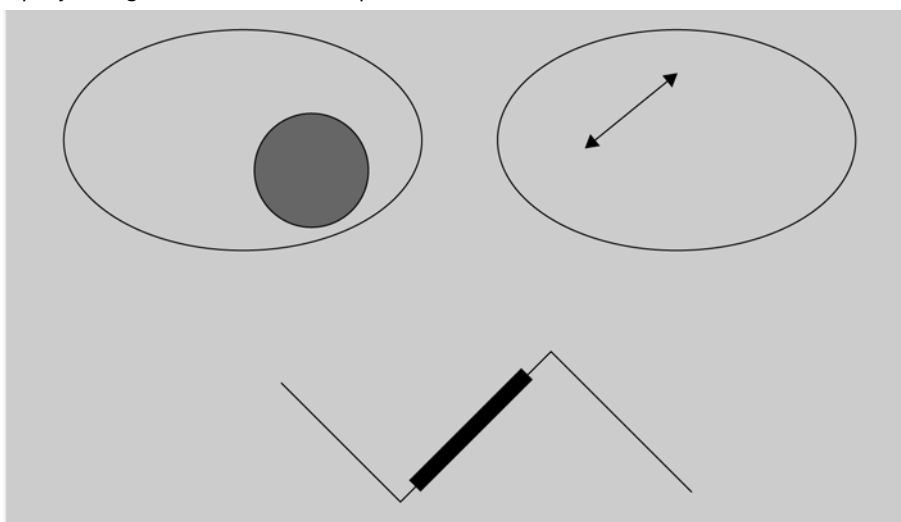


En la figura 19 podéis ver también cómo dos segmentos de dimensión 1 se intersectan en un punto de dimensión 0.

Esta definición excluye el caso de dos polígonos con intersección de sus interiores. La intersección sería un área de dimensión 2 y, por lo tanto, tendría la misma dimensión que las dos figuras geométricas que se intersectan.

- **Incluida:** A está incluida en B cuando la intersección entre A y B es exactamente A . Es decir, que todo A está dentro de B . En la figura 20 podéis ver un polígono en forma de círculo incluido en un polígono en forma de elipse. También podéis ver una línea incluida dentro de un polígono y una línea incluida dentro de otra línea.

Figura 20. Geometrías incluidas. Un círculo incluido en una elipse, un segmento incluido en una elipse y un segmento incluido en una polilínea



- **Contiene:** A contiene B si B está incluida en A . Este operador es recíproco en el operador de inclusión.
- **Solapadas:** A se solapa con B si se intersecan y la dimensión de A , B y su intersección son iguales. Esta función sólo se puede aplicar entre objetos geométricos de la misma dimensión (polígonos con polígonos, líneas con líneas y puntos con puntos) y, por lo tanto, excluye los casos del operador “Tocan” y el operador “Cruzan”.

Para puntos con puntos, se solapan, se cruzan y se encabalgan tiene el mismo resultado.

4.2.2. Funciones

Ahora que ya habéis visto las operaciones, veamos las funciones. Las funciones son aquellos métodos que nos permiten analizar las geometrías y nos responden con una nueva geometría fruto del cálculo. A continuación, tenéis una lista de las operaciones geométricas que se pueden realizar. Esta lista está basada en el estándar *simple feature access* de OGC y con algunas funciones muy habituales en las bases de datos geográficas. En particular tenemos:

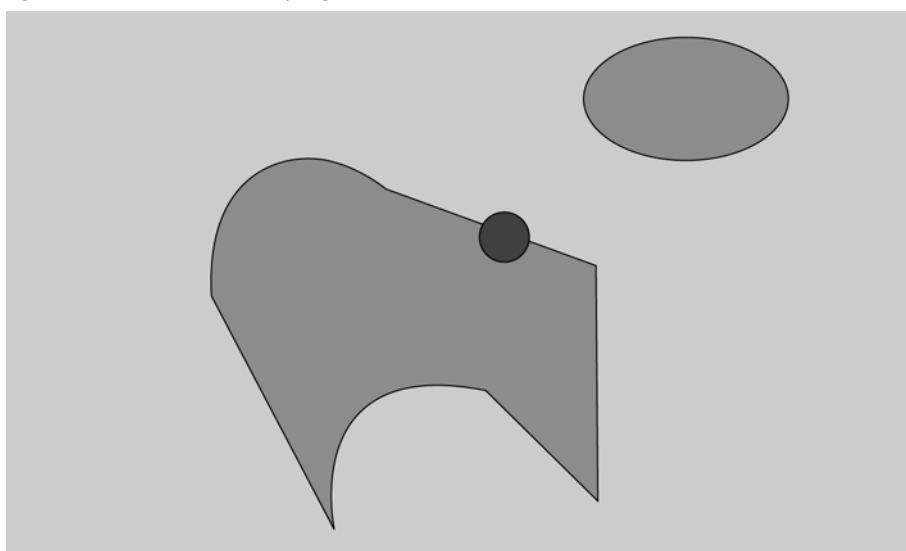
La descripción de cada una de las funciones se realizará sobre objetos geométricos A y B . Recordad los conceptos de interior, frontera y exterior vistos al explicar los operadores.

- **Centro.** El centro de A es el centroide (también conocido como centro de gravedad) de A . Para un triángulo el centroide es el baricentro. Para un conjunto de polígonos el centroide tiene en cuenta el peso (área) de cada polígono en el cálculo del centroide. El centroide no tiene por qué caer dentro del área de un polígono. Por ejemplo, el centroide de la letra C cae fuera del trazado de la letra. En la figura 21 podéis ver un multipolígono con su centroide representado con un punto.

Baricentro y mediana

La mediana de un triángulo es el segmento que va de un vértice al centro del lado opuesto. Mientras que el baricentro de un triángulo es el punto de intersección de las medias.

Figura 21. Centro de un multipolígono



La función Centro se usa sobre todo para convertir geometrías de más de una dimensión en puntos.

- **Caja contenedora.** La caja contenedora de A es el rectángulo mínimo con lados paralelos a los ejes de coordenadas del sistema de coordenadas de

proyección que incluye A . También se conoce esta función como MBR (*minimum bounding rectangle* o rectángulo mínimo contenedor). La definición se basa en dos condiciones simples:

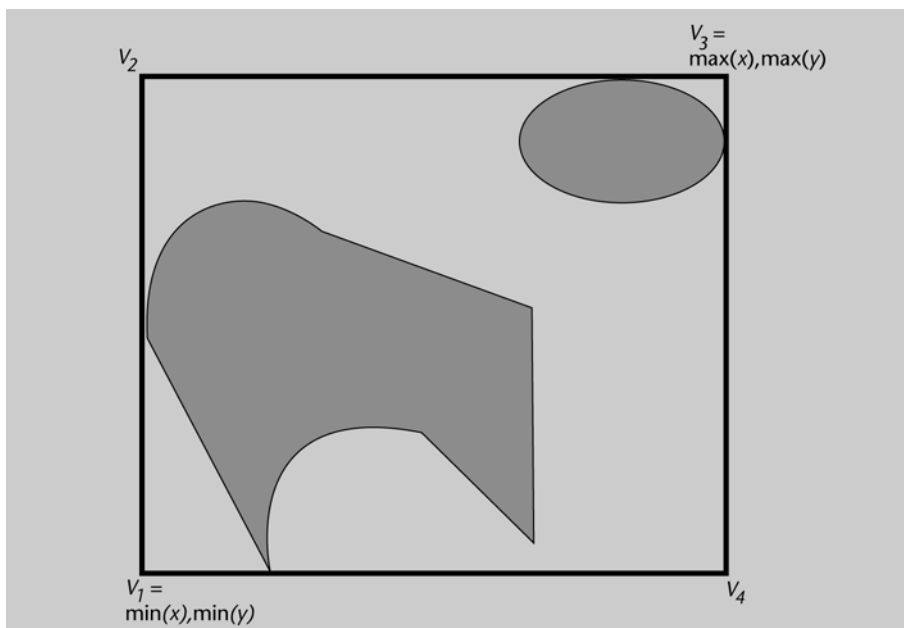
- El rectángulo mínimo que incluye A .
- Que el rectángulo esté orientado según los ejes de coordenadas.

Una manera sencilla de calcular la caja contenedora de A sobre un sistema de coordenadas proyectado es definir un rectángulo con unos vértices (V_1 , V_2 , V_3 y V_4) donde:

- $V_1 = (\min(x), \min(y))$
- $V_2 = (\min(x), \max(y))$
- $V_3 = (\max(x), \max(y))$
- $V_4 = (\max(x), \min(y))$

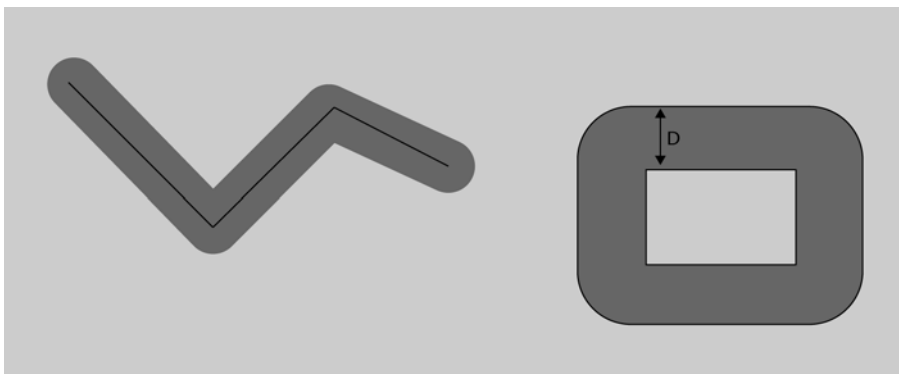
La función $\min(x)$ nos devuelve el valor mínimo de las coordenadas x de una geometría, mientras que la función $\max(x)$ nos devuelve el valor máximo de las coordenadas x . En la figura 22 podéis ver un multipolígono encuadrado con su caja contenedora.

Figura 22. Caja contenedora de un multipolígono



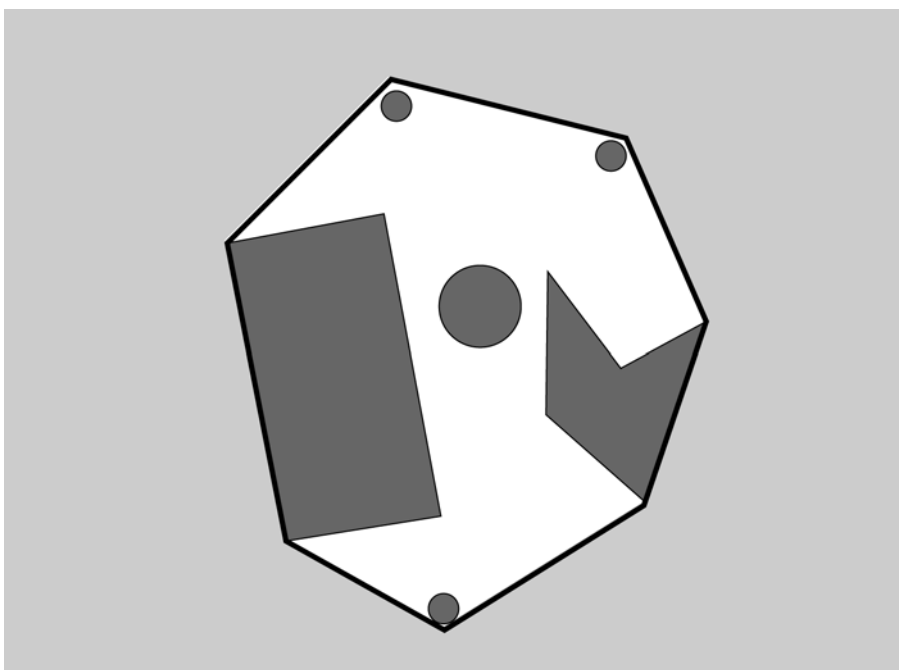
La función de caja contenedora se utiliza principalmente para poder encuadrar un elemento y centrar el mapa sobre el elemento. Es una de las funciones más utilizadas en las herramientas SIG.

- **Buffer.** El *buffer* (o zona de influencia) de un objeto A , que puede ser un punto, una línea o un polígono, nos devuelve una nueva geometría que cubre todos los puntos desde la frontera de A hasta una cierta distancia de A . La función recibe como parámetros la geometría A y la distancia D y genera una figura como la de la figura 23, donde la zona oscura corresponde al *buffer*.

Figura 23. Geometrías con *buffer*

- **La envolvente convexa (*convex hull*).** Entendemos por envolvente convexa de A , en el plano, el polígono convexo mínimo que incluye todos los puntos de A . Muchas definiciones de la envolvente convexa recurren a la idea de una goma elástica que, situada rodeando la geometría y estirada, tomaría la forma de la envolvente convexa. En la figura 24 podéis ver un ejemplo, donde la línea representa la goma elástica que define la envolvente convexa:

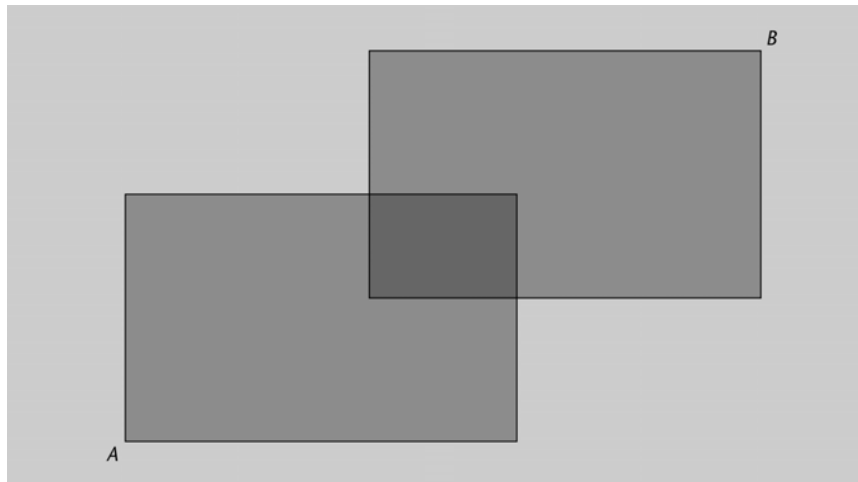
Figura 24. Envolvente convexa de un conjunto de geometrías



La envolvente convexa se utiliza, por ejemplo, para definir el ámbito de actuación de una obra o para generalizar un conjunto de datos y verlo desde grandes escalas.

- **Intersección.** La intersección de A y B devuelve un objeto geométrico que incluye todos los puntos del plano incluidos tanto a A como B . Es la operación lógica AND. En la figura 25 podéis ver la intersección, en color más oscuro, de los rectángulos A y B .

Figura 25. Intersección de dos geometrías



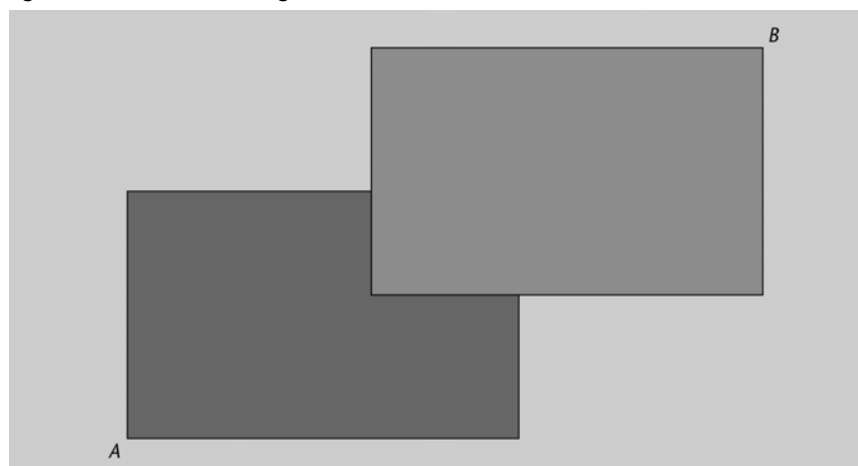
- **Unión.** La unión de A y B devuelve un objeto geométrico que incluye todos los puntos del plano incluidos en A o en B . Es la operación lógica OR. En la figura 26 podéis ver la unión de los polígonos A y B .

Figura 26. Unión de dos geometrías



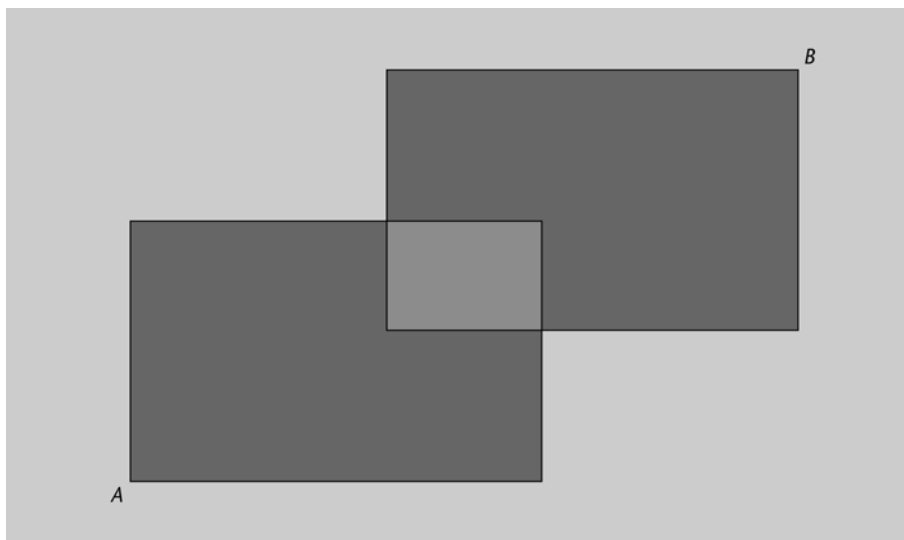
- **Diferencia y diferencia simétrica.** La diferencia de A y B devuelve un objeto geométrico que incluye todos los puntos del plano incluidos en A y no en B . Es la operación lógica de resto. En la figura 27 podéis ver la diferencia, en color más oscuro, de A respecto a B .

Figura 27. Diferencia de dos geometrías



La diferencia simétrica de A y B devuelve un objeto geométrico que incluye todos los puntos del plano incluidos en A y no en B o en B y no en A . Es la operación lógica XOR. En la figura 28 podéis ver la diferencia simétrica, en color más oscuro, entre A y B .

Figura 28. Diferencia simétrica de dos geometrías



- **Distancia.** La distancia entre A y B nos devuelve un número real que indica la distancia entre los dos puntos más próximos de A y B .
- **Vecino más próximo.** El vecino más próximo a A devuelve la geometría B más próxima a A de un conjunto de geometrías. Con esta función se puede calcular, por ejemplo, cuál es el restaurante que está más cerca de mi casa.

4.3. Índices espaciales

Habéis visto ya las posibles relaciones espaciales entre los datos. Dadas estas relaciones, es necesario que ordenemos los datos de manera eficiente para acelerar la obtención de datos. Los índices son una herramienta de los SGBD para acelerar el proceso de busca sobre las tablas. Sin índices, con el fin de obtener el resultado de una consulta, el SGBD debe recorrer toda la tabla para ver qué registros cumplen el criterio de busca. Los índices, entre otras herramientas, son una de las principales ventajas en el uso de bases de datos relacionales por encima de los ficheros, ya que, en general, para leer la información de un fichero se debe leer y cargar todo en memoria, mientras que las bases de datos permiten acceder a partes de un conjunto de datos sin leerlos todos.

Los índices clásicos se construyen sobre lo que se conoce como árboles B+. Estos árboles permiten buscar en un directorio sin tener que acceder a todos los registros. Los árboles son estructuras de datos organizados en forma de árbol con un nodo raíz que permite acceder a sus hijos, que al mismo tiempo pueden tener hijos o son ya un nodo final u hoja. En un árbol B+ cada nodo funciona como delimitador de los nodos hijo y contiene la información para

indicar hacia qué hijo se debe ir para buscar un valor concreto. En el ejemplo que presentamos a continuación, podréis ver un caso de índice clásico y una figura con esta estructura de árbol (figura 29).

Antes de entrar a describir los índices para el acceso y las consultas espaciales, veréis un ejemplo del uso de índices generales para que podáis entender su potencia.

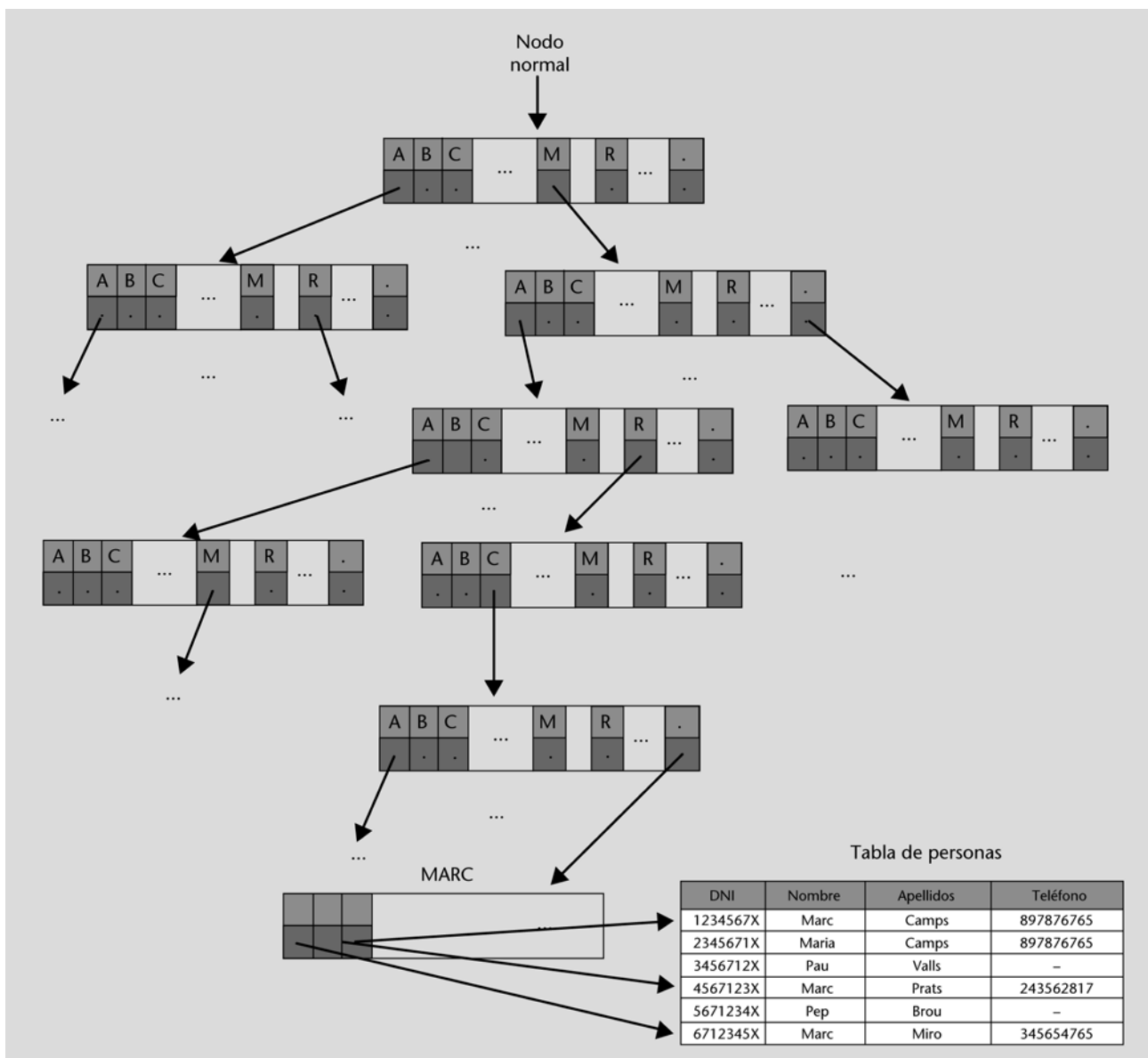
Ejemplo de uso de índices generales

Imaginad una tabla con información de personas donde queremos ejecutar la consulta de aquellas personas que tienen por nombre Marc. El formato SQL de esta consulta sería:

```
SELECT Apellidos, Direccion FROM Personas WHERE Nombre = 'Marc'.
```

Esta consulta nos devolverá una tabla con dos columnas con la información de los apellidos y la dirección de todas las personas que tengan por nombre Marc. Supongamos que la tabla tiene todas las personas del Estado español y, por lo tanto, tiene 46 millones de registros. Si construimos un índice sobre el campo Nombre de esta tabla, el índice podría tener la apariencia de la figura 29.

Figura 29. Árbol de un índice



En la figura podéis ver cómo el algoritmo de busca accede al nodo raíz y busca el primer carácter del texto que se quiere buscar. El árbol de busca lo dirige a un nodo hijo que engloba todos los registros con un nombre que empiece por *M*. Aquí el algoritmo busca el segundo carácter del nombre que queremos buscar y eso lo redirige hacia otro nivel del árbol a un nodo que engloba todos los registros de la tabla con un nombre que empiece por *MA*. Y así avanza el algoritmo hasta que llega al nodo que engloba todos los registros que empiezan por *MARC*. Desde este nodo el árbol tiene apuntadores a todos los registros de la tabla que contiene *MARC*.

Para comparar el rendimiento con índice o sin índice, pensad que sin índice deberíamos recorrer 46 millones de registros, mientras que con índice sólo hemos de acceder a 5 nodos del árbol. Cabe señalar que los índices basados con árboles B+ sirven para consultas sobre campos de tipo numérico o texto, pero no sirven para optimizar consultas espaciales.

Los índices espaciales son índices diseñados para optimizar las buscas basadas en criterios espaciales sobre la información geográfica.

Hay distintos tipos de índices sobre información geográfica. Sólo veremos dos con detalle, ya que no es objetivo de esta asignatura que sepáis el funcionamiento interno de todos los posibles índices espaciales. Sí que es importante, sin embargo, que conozcáis cómo funcionan los índices habituales de cara a utilizarlos de la mejor manera posible.

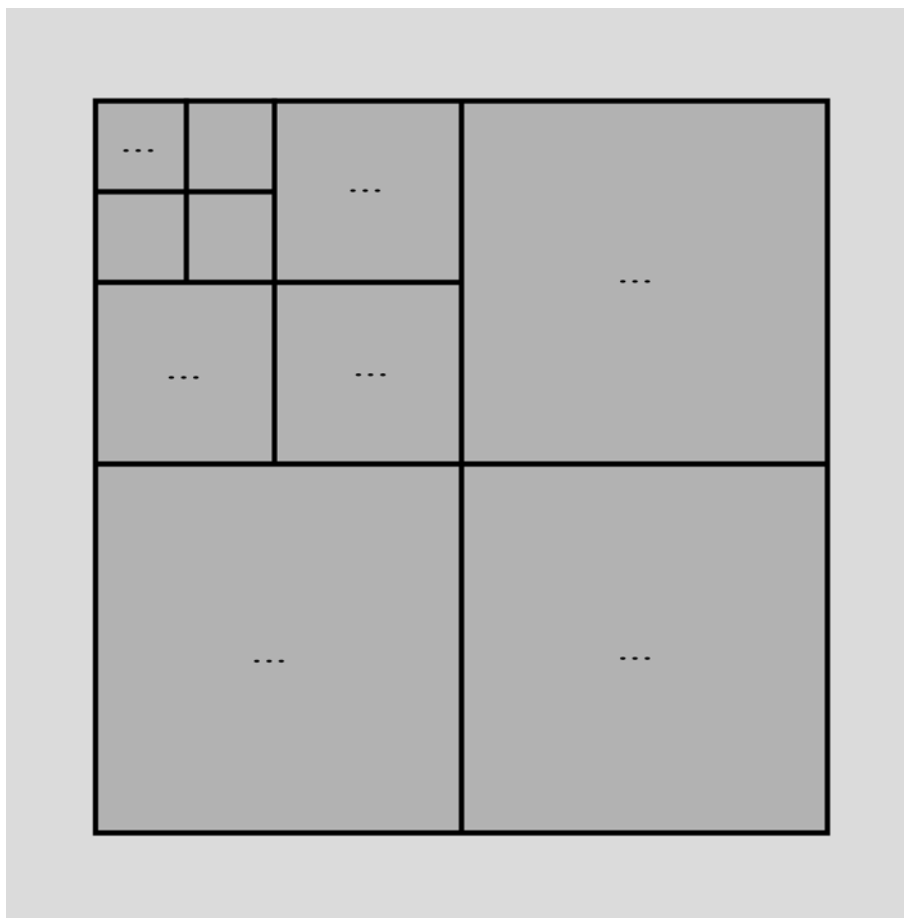
Recordad que cuando manejáis grandes cantidades de información geográfica es muy importante crear índices, tanto espaciales como no espaciales, de cara a optimizar el acceso a la información geográfica.

4.3.1. *Quadtree*

Un *quadtree* (o árbol cuaternario) es una estructura de datos en forma de árbol en la que cada nodo del árbol tiene hasta cuatro nodos. Los *quadtrees* se utilizan habitualmente para dividir espacios de dos dimensiones en piezas recursivamente pequeñas y, de esta manera, dividir el espacio en cuadrantes.

Los *quadtrees* son una manera natural de dividir el espacio por regiones y sirven sobre todo para dividir geometrías puntuales. No son tan eficientes para indexar geometrías de tipo área o línea, que pueden estar distribuidas en muchos cuadrantes a la vez. En la figura 30 podéis ver cómo divide el espacio un *quadtree*.

Los *quadtrees* se han utilizado mucho para indexar geometrías pero actualmente están siendo sustituidos por el *R-tree*, que mejora sustancialmente el rendimiento de las buscas.

Figura 30. Separación del espacio de un *quadtree*

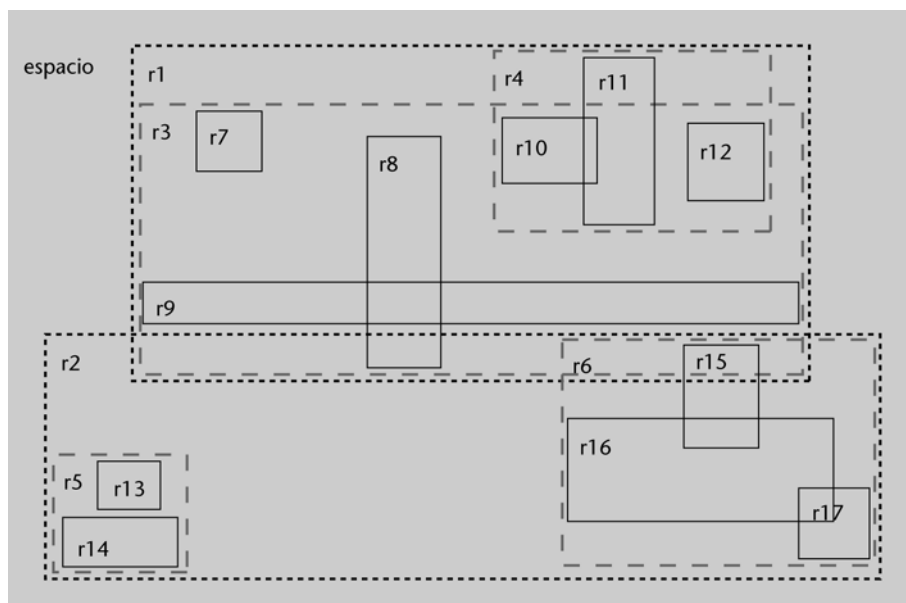
4.3.2. Los *R-trees*

Los *R-trees* son estructuras de datos en forma de árbol que se utilizan para acceder a datos espaciales indexando información multidimensional. Se usan, por ejemplo, para indexar información sobre un sistema de coordenadas de dos dimensiones. Mediante los *R-trees* se pueden optimizar búsquedas del estilo “dónde está el hospital más cercano a mi oficina”.

El árbol divide el espacio en cajas contenedoras de las geometrías en *minimum bounding rectangles* (o MBR) imbricadas jerárquicamente y con solapamiento.

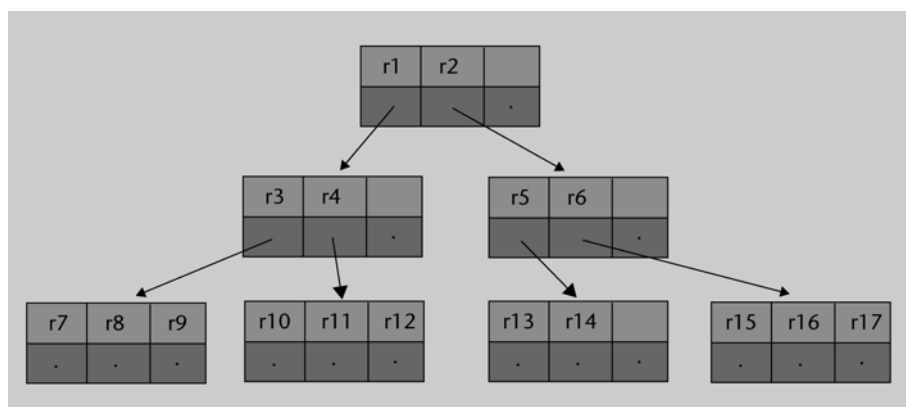
En la figura 31 podéis ver cómo divide el espacio un índice *R-tree*. Los rectángulos negros representan cajas contenedoras de las geometrías que hay que indexar. Los rectángulos en línea discontinua representan las cajas contenedoras de conjuntos de como máximo 3 geometrías (el número máximo de geometrías se puede elegir de cara a optimizar el índice). Los rectángulos en línea punteada (r1 y r2) son las cajas contenedoras de más alto nivel e incluyen como máximo 3 cajas del nivel inferior (los rectángulos en línea discontinua).

Figura 31. Agrupación de geometrías para MBR



Como veis en la figura, las cajas contenedoras se pueden solapar. Cada nodo del árbol tiene un número de entradas máximo que se ha predefinido (en la figura este máximo es 3). Cada nodo no terminal del árbol contiene la lista de nodos hijo con la caja contenedora que lleva a cada uno de los hijos. De esta manera se organiza un árbol como el de la figura 32.

Figura 32. Organización del *R-tree*



Los algoritmos de inserción y borrado de elementos del índice aseguran que cada nuevo elemento se ubica en el árbol junto con los elementos que tiene más cerca para así mantener el árbol arreglado.

Para entender cómo funciona la busca sobre un *R-tree*, planteamos el caso de querer representar sobre el mapa las geometrías de una región de territorio. En la figura 33 podéis ver representada en línea gruesa la región que queremos dibujar.

El algoritmo de busca accede al árbol y compara la caja contenedora de cada uno de los nodos hijo (r1 y r2) con el territorio que vamos a dibujar. Podéis ver en la figura que el territorio se interseca con dos cajas contenedoras, r1 y r2. El algoritmo decide que se deben examinar tanto el nodo que representa r1 como el que representa r2. Al comparar el territorio con las cajas contene-

doras hijos de r_1 (r_3 y r_4) sólo se interseca con r_3 . Al comparar el territorio con los hijos de r_2 (r_5 y r_6), sólo se interseca con r_6 . El algoritmo continúa bajando por los nodos que representan r_3 y r_6 . Al examinar los hijos de r_6 no se encuentra ninguna caja contenedora que se interseque con el territorio. Al examinar r_3 se encuentra intersección con r_8 y r_9 . Como ya se ha llegado a nodos finales del árbol, ya tenemos las geometrías con una caja contenedora que se interseca con el territorio que hay que representar sobre el mapa, r_8 y r_9 . Podéis ver representado en la figura 34 el camino que examina el algoritmo.

Figura 33. Consulta de un ámbito territorial sobre un *R-tree*

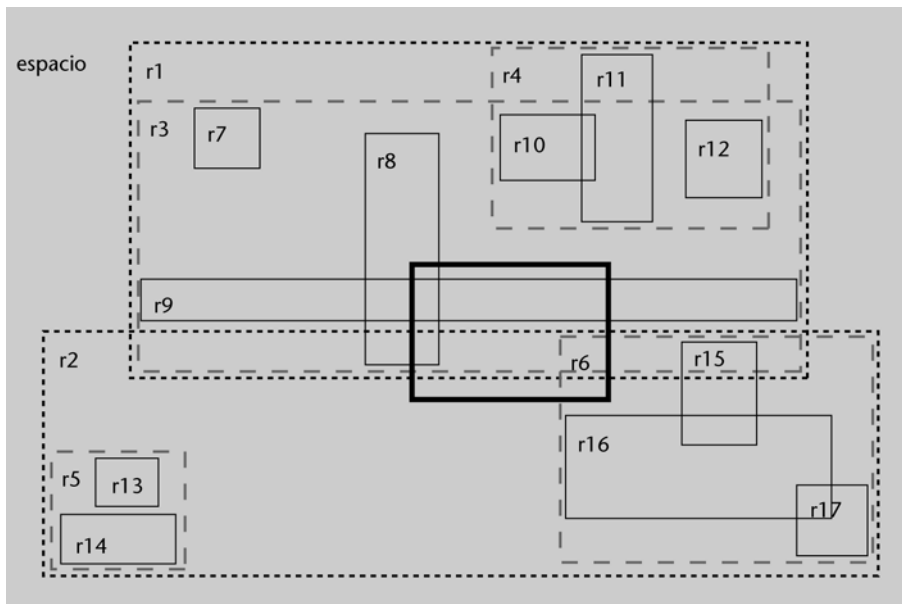
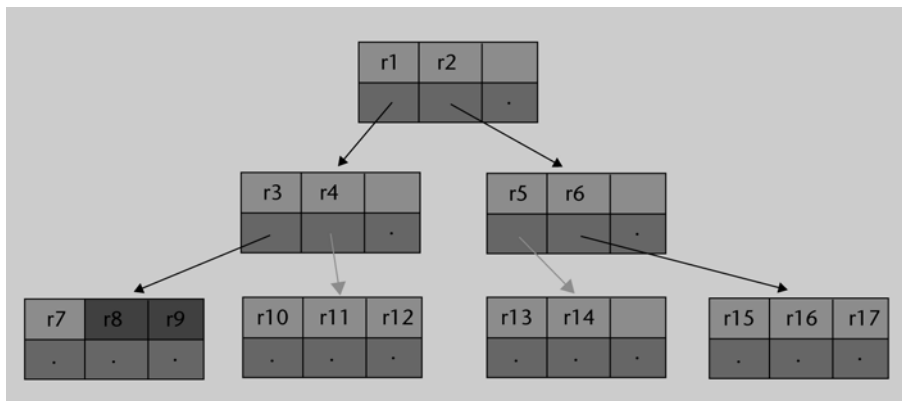


Figura 34. Accesos a un *R-tree* para una consulta espacial



En este caso hay pocas geometrías y no podéis ver claramente el uso del índice, pero en un caso en el que, por ejemplo, hubiéramos de representar todas las fincas de Barcelona sobre el mapa, un índice como éste nos podría ahorrar tener que recorrer todas las fincas para decidir si las hemos de pintar o no.

4.4. Consultas espaciales

Los datos han de estar ordenados e indexados para poder consultarlos. Veamos, pues, cómo consultar los datos espaciales. Las consultas espaciales

son consultas sobre las bases de datos que usan operadores y funciones espaciales como criterio de busca. Ya habéis visto los operadores y las funciones espaciales pero no habéis visto cómo aplicarlas. La aplicación de estos operadores y funciones depende de las herramientas SIG que utilicéis y muchas de estas herramientas no permiten aplicarlos a escala de SGBD. Los SGBD tienen un motor interno de busca y obtención de datos altamente optimizado. Todo aquello que se pueda ejecutar sobre este motor ahorrará recursos de sistema e irá más rápido. Muchas herramientas SIG ofrecen estas funcionalidades de consulta sobre sus programas y fuera del motor de bases de datos. Esta opción, a pesar de ser la más habitual, consume más recursos y es menos eficiente.

Un operador o función se puede aplicar en un SGBD o en una herramienta SIG. Las herramientas SIG suelen ser más lentas en calcular las funciones espaciales, ya que deben cargar todos los datos en memoria de trabajo para poder realizar la busca. Desgraciadamente, sin embargo, lo más habitual es tener que decantarse por esta opción, puesto que son pocos los SGBD que permiten consultas espaciales hoy en día. Las consultas espaciales calculadas por el mismo SGBD sacan todo el partido de los índices espaciales y permiten obtener los resultados con mejor rendimiento y menos consumo de recursos.

Para que seáis capaces de diseñar y aplicar vuestras propias consultas espaciales, deberéis adaptaros al lenguaje de la herramienta SIG que utilicéis. Veréis a continuación, como ejemplo, cómo realizar estas consultas con Oracle Spatial. Hemos elegido Oracle Spatial por lo siguiente:

- Porque es el lenguaje de consultas espaciales más integrado con el motor de bases de datos.
- Porque es el lenguaje de consultas más próximo al estándar SQL.
- Porque está deviniendo un estándar como formato de almacenamiento geográfico.

4.4.1. Ejemplos de consultas Oracle Spatial

Imaginad una base de datos donde tenemos dos entidades geográficas almacenadas en dos tablas. La primera entidad son las carreteras con su geometría de tipo lineal y su nombre. La segunda es una entidad puntual que almacena los puntos de interés (restaurantes, hoteles, miradores, etc.) con su nombre. Tendremos las tablas siguientes (el campo en negrita es la clave primaria de la tabla):

- carreteras (**id:entero**, nombrecarretera:texto, recorrido:geometria)
- puntosinteres (**id:entero**, nombre_puntointeres:texto, ubicacion:geometria)

La consulta siguiente nos devolvería una lista de los 5 primeros restaurantes más próximos a la carretera C17. Observad cómo se usan órdenes propias del motor espacial del SGBD dentro de las consultas SQL.

```
SELECT nombre_puntointeres
FROM
  (SELECT P.nombre_puntointeres,
    SDO_GEOM.SDO_DISTANCE(P.ubicacion, C.recorrido, 0.5)
  distancia
  FROM carreteras C, puntosinteres P
  WHERE C.nombrecarretera = 'C17'
  ORDER BY distancia )
WHERE ROWNUM <= 5;
```

Observad que hay dos consultas imbricadas (hay dos instrucciones `SELECT`). La consulta interior permite obtener el nombre del punto de interés y otro campo calculado la distancia del punto de interés a la carretera C17. Denominamos este segundo campo *distancia*. El resultado de la consulta se ordena por distancia. La segunda consulta sólo filtra los 5 primeros registros de la consulta interior.

Fijaos en el uso del operador espacial `SDO_GEOM.SDO_DISTANCE` para obtener la distancia de cada punto de interés a la carretera. `P.ubicacion` y `C.recorrido` son campos que almacenan geometrías y, por lo tanto, se pueden aplicar operaciones espaciales.

La consulta siguiente nos devolvería los puntos de interés que se encuentran a cierta distancia de la carretera. Concretamente a 50 km.

```
SELECT P.nombre_puntointeres
FROM carreteras C, puntosinteres P
WHERE
  SDO_ANYINTERACT
  (
    P.ubicacion,
    SDO_GEOM.SDO_BUFFER(C.recorrido, 50, 0.5, 'UNIT=KM')
  ) = 'TRUE'
AND C.nombrecarretera='C17';
```

Fijaos, a diferencia del ejemplo anterior, que en esta segunda consulta no usamos los operadores y funciones espaciales para calcular el valor de un campo, sino para filtrar registros (cláusula `WHERE`). El operador `SDO_ANYINTERACT` tomará el valor `TRUE` (o cierto) sólo para los puntos de interés que estén a menos

de 50 km de una carretera (cualquiera). La condición siguiente nos obliga, además, al hecho de que la carretera sea la C17.

Para conseguirlo, se ha tenido que calcular la geometría del área de influencia con diámetro 50 km de las carreteras. Para ello se usa la función `SDO_GEOM.SDO_BUFFER`.

`SDO_ANYINTERACT` compara las geometrías del campo `P.ubicacio` y las del resultado de la función `SDO_GEOM.SDO_BUFFER`.

5. Modelos avanzados de almacenamiento

Ya habéis visto cómo se almacenan la mayor parte de los datos geográficos vectoriales, pero dado que las características de la recogida de los datos o de la explotación que queremos hacer no es siempre igual, no todos los datos geográficos se almacenan de la misma manera. Hay modelos avanzados de almacenamiento de información geográfica que nos permiten introducir los datos de manera más cómoda o más funcionalidades de consulta.

Ahora que ya habéis visto cómo se almacenan los datos geográficos en la mayor parte de los casos, veréis dos casos especiales de almacenamiento de datos geográficos: la topología y la referencia lineal.

5.1. Topología

La topología es una parte de las matemáticas que se preocupa de los objetos en el espacio y de las relaciones entre ellos. Las relaciones topológicas incluyen relaciones como Contener, Dentro de, Cubre, Cubierto por, Se tocan y Se solapan. Las relaciones topológicas se mantienen al transformar o deformar el espacio. Hay, sin embargo, varias relaciones entre objetos en el espacio que no son topológicas, como Longitud de, Distancia entre o El área de.

La topología en matemáticas se define como:

“Rama de las matemáticas que trata especialmente de la continuidad y de otros conceptos más generales originados en ella, como las propiedades de las figuras con independencia de su tamaño o forma”.

Diccionario de la Lengua Española (Real Academia Española)

La topología asume que las entidades geográficas se producen sobre el plano de proyección en dos dimensiones. Proyectadas sobre el plano, las entidades se pueden representar con nodos, aristas (o arcos) y caras. Los puntos se convierten en nodos. Las líneas se convierten en aristas y si se crean con otra línea, se convierten en dos aristas y un nodo sobre el punto de intersección. Los polígonos se convierten en caras. En la tabla siguiente tenéis una recopilación de la transformación:

Geometría vectorial simple	Geometría vectorial con topología
Punto	Nodo
Línea	Aristas y nodos a las intersecciones
Polígono	Cara con aristas y nodos por frontera

Esta transformación nos crea un grafo sobre el plano donde:

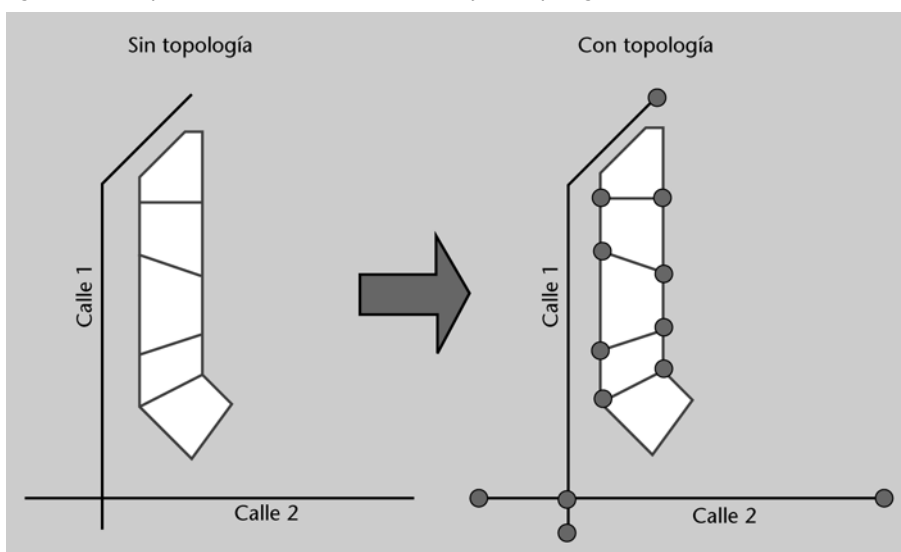
- Los puntos son nodos.
- Las líneas son conjuntos de nodos conexas con las aristas que los comunican.
- Los polígonos son caras o regiones cerradas del grafo, rodeadas por un conjunto de vértices conexas. Las aristas comunican estos vértices, rodean el polígono y el último vértice es el primero.

Grafo

Un grafo es un conjunto de objetos denominados *vértices* o *nodos* unidos por enlaces denominados *aristas*.

En la figura 35 podéis ver cómo un conjunto de datos geográficos se pueden almacenar con topología.

Figura 35. Comparación de almacenamiento con y sin topología



Observad en la figura que, sin topología, guardamos 5 polígonos con un total de 21 caras; es decir, 42 coordenadas. Con topología, en cambio, guardamos 12 aristas y 8 nodos; esto es, guardamos la misma información con un total de 13 coordenadas (una coordenada para cada nodo y 5 más para las caras de las aristas exteriores que no son líneas rectas). Las aristas en topología no son polilíneas, sino enlaces entre coordenadas, por lo que ocupan muy poco.

Dentro del mundo de los SIG almacenar los datos geográficos con topología se considera ventajoso porque:

- Proporciona herramientas para detectar errores de digitalización y edición.
- Reduce el espacio de almacenamiento al guardar sólo una vez las fronteras entre los polígonos.
- Permite análisis de los datos sobre la conectividad, inclusión o adyacencia de los elementos. Por ejemplo, el cálculo de rutas sobre las carreteras.

- Evita solapamientos por el propio modelo de almacenamiento.

Veréis seguidamente cómo se almacena la topología, qué opciones de almacenamiento de topología hay y cómo podemos explotar las ventajas de almacenar con topología.

5.1.1. Almacenamiento de topología

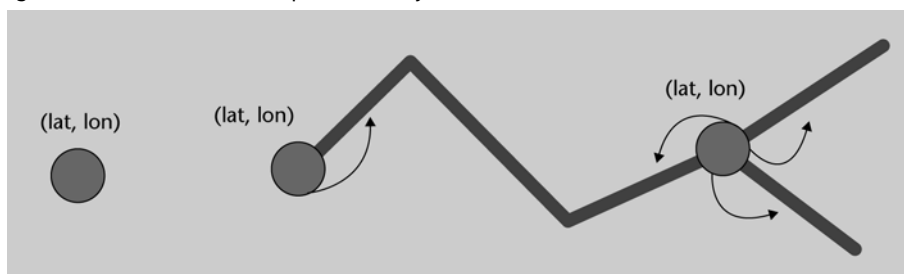
Los elementos básicos de la topología son los nodos, las aristas y las caras. Sobre el SGBD se almacenan estos elementos y las relaciones entre ellos para guardar la topología. Las entidades hacen referencia a estos elementos (nodos, aristas y caras) para indicar cuál es su posición sobre el territorio. Veamos ahora estos elementos con detalle.

Nodo

Un nodo, representado por un punto (es decir, un par de coordenadas que lo georreferencian), puede representar un elemento aislado o puede representar un margen de una arista. Dos o más aristas se encuentran en un nodo no aislado.

En la figura 36 podéis ver los nodos como punto aislado o como inicio y final de una arista. Las flechas indican una referencia o enlace.

Figura 36. Datos almacenados por un nodo y una arista



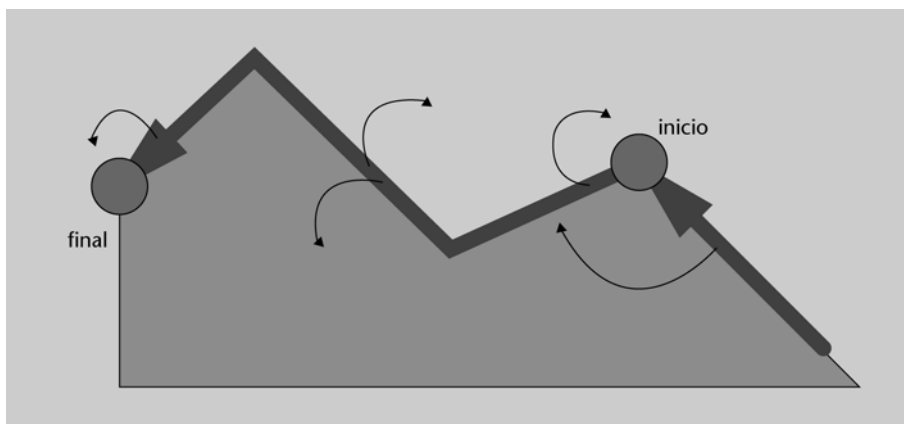
Ejemplos de entidades que se pueden representar con nodos pueden ser puntos de interés, principios y finales de calles.

Arista

Una arista tiene por márgenes dos nodos: el nodo de origen y el nodo de fin. Una arista tiene asociado un elemento de tipo línea, normalmente una polilínea, que describe el recorrido de la arista sobre el territorio. El orden de las coordenadas en la polilínea nos da la dirección de la arista. La dirección de la arista es importante a la hora de determinar las relaciones topológicas. El sentido positivo sigue las coordenadas de la polilínea, mientras que el sentido negativo sigue el orden contrario. Toda arista con sentido es el espejo de otra arista con sentido que tiene por nodo inicial el final y por nodo final el inicial de la otra arista. Una arista se ubica entre dos caras y hace referencia a las dos. Cada arista con sentido hace referencia a la arista siguiente en el perímetro contiguo de la cara en su lado izquierdo.

En la figura 37 podéis ver una arista con todas las referencias a los elementos que la rodean. Cada flecha en la figura representa una referencia.

Figura 37. Referencias de las aristas para rodear las caras



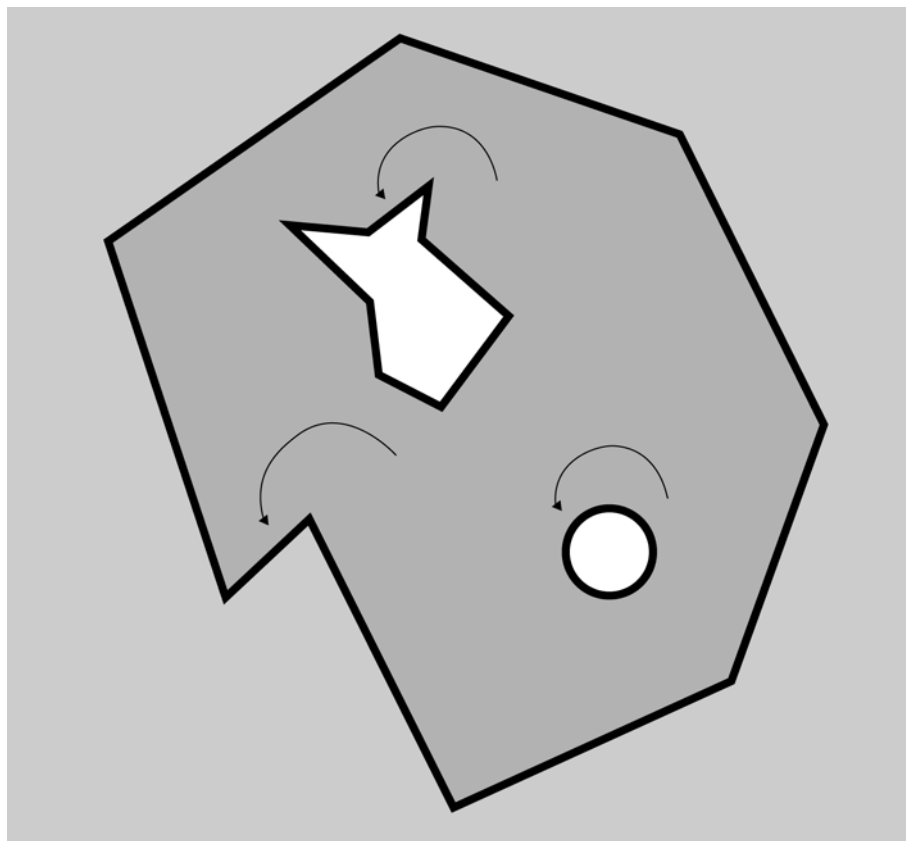
Ejemplos de entidades representables con aristas pueden ser tramos de calle o ríos.

Cara

Una cara, correspondiente a un polígono, hace referencia a una de las aristas con sentido de su perímetro. Si el polígono tiene agujeros o islas en el interior, la cara también hace referencia a una de las aristas del perímetro del agujero o isla.

En la figura 38 podéis ver una cara con las referencias a una de las aristas de cada una de sus fronteras. Cada una de las flechas representa una referencia.

Figura 38. Referencias de las caras en las aristas



Ejemplos de entidades que se pueden representar como caras son lagos, países, parcelas agrícolas, fincas, etc.

La topología sobre los formatos de datos de SIG se almacena normalmente en formatos propietario. De todos modos, la mayoría de los formatos almacenan la definición en tres tablas relacionales para los nodos, las aristas y las caras. También se usa una cuarta tabla que almacena las coordenadas. La tabla para los nodos almacena información de los nodos. La tabla de las aristas contiene la información de las aristas, que incluye el nodo de inicio y el de final y la cara de la derecha y de la izquierda. La tabla de caras guarda la información de las caras como una arista del perímetro exterior y una arista de cada uno de los perímetros interiores.

5.1.2. Topología geométrica frente a topología relacional

No todos los formatos de almacenamiento soportan la topología. De hecho, sólo los formatos más sofisticados la soportan. Sin embargo, hay modelos simples de topología que se pueden guardar por encima del SGBD.

Una red de carreteras, por ejemplo, se puede guardar como un grafo dirigido donde las aristas son tramos de carretera y los nodos son todos los cruces o desviaciones que pueda haber en la red. Las herramientas SIG permiten almacenar este modelo con topología geométrica o con topología relacional.

Entendemos por topología geométrica aquella que requiere que los objetos geométricos se toquen con el fin de tener una relación entre ellos.

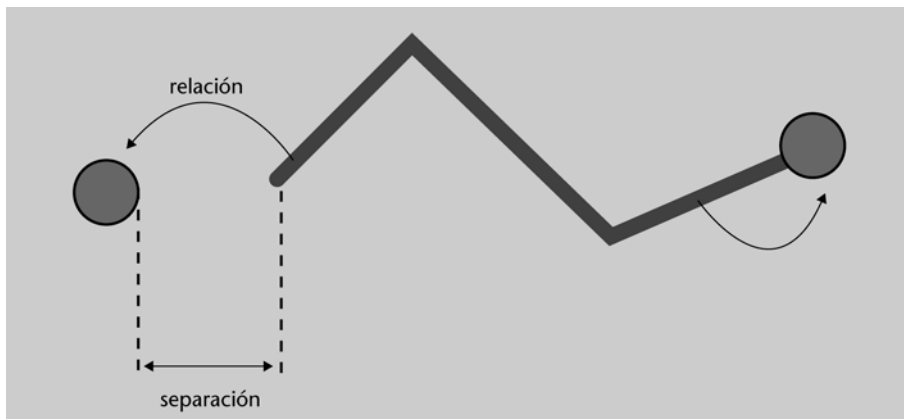
Entendemos por topología relacional aquella que almacena una relación funcional entre dos elementos geográficos sin forzar a que se toquen gráficamente.

La topología relacional permite guardar con una relación entre tablas de base de datos que una arista está relacionada con un nodo o una arista. De esta manera, podríamos tener un tramo de canalización de agua con la presa de agua que tiene al principio y con la canalización de agua que tiene después. Esto lo haríamos sin forzar a que gráficamente la canalización de agua se toque con la presa de agua. En la figura 39 podéis ver cómo se puede relacionar una línea con un punto sin forzar a que se toquen. Esto sería imposible con topología geométrica, ya que ésta obliga a que estén en contacto.

El modelo relacional es más tolerante con la información, lo que facilita la edición, pero no asegura la calidad topológica de los datos, y esto puede dar lugar a mapas poco minuciosos. Podéis entender la topología relacional como si

construyéramos sobre el SGBD una topología propia que sólo tiene en cuenta las relaciones que nos interesan.

Figura 39. Separación de elementos con topología



La topología geométrica, por otra parte, tiene como inconveniente principal la complejidad de edición y construcción de la topología y las restricciones que aporta.

5.1.3. Explotación de la topología

Utilizad el almacenamiento con topología cuando las restricciones que os aporta generen valor a vuestro sistema o aplicación y el coste de computación de editar la topología sea asumible. Estos casos son normalmente redes donde la conectividad es esencial (redes de agua, redes de gas, redes de carreteras para cálculos de ruta, etc.) No es necesario que al definir un almacén de datos geográficos lo almacenéis todo con topología o sin. Podéis almacenar algunas entidades con topología, mientras que el resto se puede almacenar normalmente.

Una vez tenemos los datos almacenados con topología, le hemos de sacar partido. Aparte del ahorro en espacio de disco al guardar la información con menos redundancias, la topología nos aporta relaciones que de otra manera serían imposibles de calcular. A continuación veremos tres ejemplos:

- **Cálculo de rutas.** El caso de explotación de la topología más claro es el cálculo de rutas. A partir de un punto de una red, por ejemplo de carreteras, queremos saber cómo llegar a otro punto. Si consideramos los tramos de calle como aristas y las intersecciones como nodos, podemos representar toda la red de carreteras con un grafo dirigido. Para saber el mejor recorrido, sólo hemos de asignar un peso a cada arista, por ejemplo el tiempo que se tarda en recorrerla teniendo en cuenta la longitud y la velocidad máxima, y aplicar, por ejemplo, el algoritmo de Dijkstra sobre el grafo. La conectividad y direccionalidad del grafo nos la da la topología.
- **Flooding.** El *flooding* o inundación es un cálculo muy útil para ver afectaciones en una red (por ejemplo de aguas, de gas o de electricidad). Se utiliza

Algoritmo de Dijkstra

El algoritmo de Dijkstra es un algoritmo que soluciona el problema del camino mínimo sobre un grafo dirigido con pesos de arista no negativos.

para saber qué afectación tiene sobre la red un corte, una fuga o cualquier otra incidencia. A partir de un punto de la red, se van recorriendo las aristas conexas hasta llegar a los nodos que nos indican que no se puede continuar (un nodo que represente una válvula cerrada, por ejemplo).

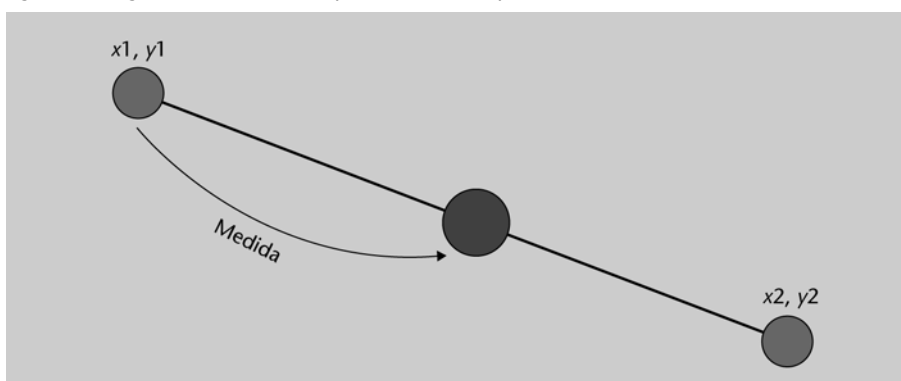
- **Vecinos.** La topología también es muy útil para calcular los vecinos de una parcela o conjunto de parcelas.

5.2. *Lineal referencing system (LRS)*

Otra opción avanzada de almacenamiento, aparte de la topología, es la referencia lineal. El sistema de referencia lineal (*lineal referencing system* o LRS) es un modo conveniente de asociar atributos o éxitos a localizaciones o tramos de una entidad lineal. Se usa mucho en aplicaciones de transportes como gestión del metro, redes de carreteras, ferrocarriles, etc. o en redes de utilidades como distribución de gas o agua.

La ventaja más importante de la referencia lineal es la capacidad de localizar atributos a lo largo de la entidad lineal con sólo un parámetro, la coordenada M de medida, en vez de dos coordenadas. Los tramos de la entidad lineal se pueden georreferenciar dinámicamente indicando la medida del inicio del tramo y la medida del final del tramo. En la figura 40 podéis ver que, para saber dónde está el punto central, sólo necesitamos las coordenadas de los extremos del segmento y la medida, entendida como la distancia, siguiendo el segmento desde el origen al punto que queremos ubicar. Así nos ahorramos una coordenada, a la vez que si corregimos el segmento se corrige automáticamente la ubicación del punto.

Figura 40. Segmento lineal con un punto localizado por medida



LRS, como sucedía con la topología, no está soportado por la mayoría de los formatos y sólo SGBD o formatos de ficheros muy especializados permiten el almacenamiento de geometrías en LRS.

Las ventajas de LRS con respecto a otros métodos de almacenamiento son:

- Facilidad de introducción de datos.

- Capacidad de georrefenciar elementos sin herramientas de medida como satélites y sin puntos de referencia. Muy útil para túneles y medidas subterráneas.
- Ahorro de espacio al no tener que almacenar las coordenadas x e y de todos los elementos, sino sólo de los puntos de referencia. Del resto de elementos sólo almacenamos la coordenada M (medida lineal).

A continuación veremos las características de este almacenamiento y sus usos.

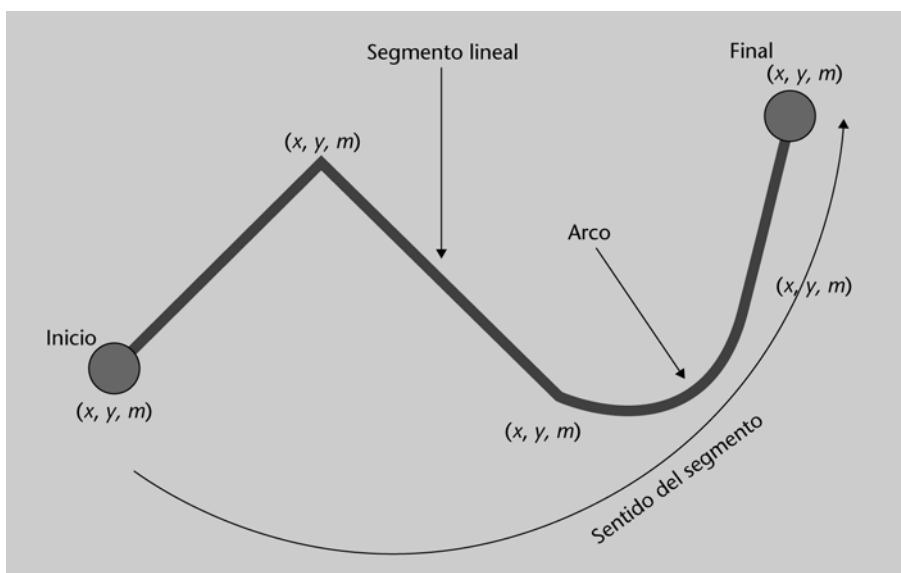
5.2.1. Almacenamiento de LRS

La base de LRS son los segmentos geométricos. Un segmento geométrico puede ser una línea simple, una multilínea (es decir, una serie de líneas simples no conectadas como, por ejemplo, una carretera cortada por un lago) o un polígono (por ejemplo, para representar un recorrido turístico circular que empieza y acaba en el mismo punto).

El segmento geométrico debe tener al menos medidas en coordenadas de su punto origen y su punto final definiendo una recta. Se pueden añadir medidas para puntos de interés o cambios de dirección durante la ruta del segmento para definir el trazado. Estas medidas sirven como punto de referencia para ubicar elementos sobre el segmento geométrico.

En la figura 41 podéis ver un segmento geométrico con sus medidas. Observad que hay varias medidas para el segmento indicadas por la etiqueta (x, y, m) . En este caso se ha realizado una medida para cada cambio de dirección. El número y la situación de las medidas viene dado, muchas veces, por las medidas que se han realizado. A continuación repasaremos los conceptos que definen el almacenamiento con segmentación lineal.

Figura 41. Segmento lineal con LRS



Puntos de medida

Los puntos de medida son puntos a lo largo del trazado del segmento geométrico definidos cuando el segmento se construye y se utilizan para medir la información. Normalmente, estos puntos de medida de uso interno del almacenamiento LRS coinciden con puntos de medida físicos reales. Si el segmento geométrico representa una carretera o una vía de ferrocarril, seguramente los puntos de medida serán los puntos kilométricos. Los puntos de medida son la referencia sobre la que se ubican el resto de elementos y sucesos.

Punto kilométrico

Un punto kilométrico es una marca en la carretera que indica el kilometraje desde el comienzo de la carretera.

Dirección del segmento geométrico

La dirección está determinada por el orden de los vértices, de inicio a fin, de la definición geométrica, es decir, siguiendo el orden de almacenamiento de las coordenadas. Las medidas de los puntos en un segmento geométrico siempre incrementan o decrementan linealmente a lo largo de la dirección del segmento geométrico.

Medida

La medida de un punto a lo largo del segmento geométrico es la distancia lineal medida desde el punto de inicio para valores que incrementan y desde el punto de final para valores que decrementan. Algunos sistemas LRS llaman *desplazamiento* a la medida, pero como hay otro concepto que denominamos *desplazamiento* (la distancia ortogonal en el segmento geométrico), esto nos podría confundir.

La medida no tiene por qué coincidir con la distancia geográfica sobre el plano. Al no coincidir, nos permite introducir datos reales de medida sin tener el trazado del segmento geométrico bien referenciado.

Desplazamiento

El desplazamiento de un punto a lo largo de un segmento geométrico es la distancia ortogonal entre el punto y el segmento geométrico. Sirve para indicar que el punto no se encuentra exactamente sobre el segmento, sino que se encuentra a una cierta distancia. El desplazamiento es positivo si el punto se encuentra a la izquierda del segmento y, negativo si se encuentra a la derecha. Para los puntos sobre el segmento geométrico el desplazamiento es igual a cero.

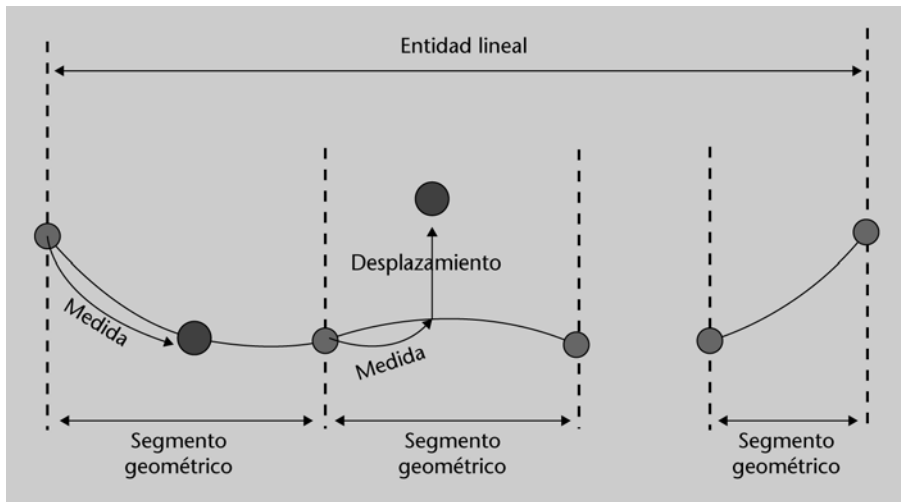
Entidades lineales

Las entidades lineales son cualquier objeto espacial que se pueda tratar como conjunto de segmentos geométricos lineales. Ejemplos de entidades lineales son las autopistas, las redes de agua o gas, las vías ferroviarias, etc. Una entidad lineal es una colección de los segmentos geométricos que la forman.

En la figura 42 podéis ver representados los conceptos de entidad lineal, segmento geométrico, medida y desplazamiento. La entidad lineal es todo el tra-

zado discontinuo formado por los 3 segmentos geométricos. Los puntos mayores representan elementos que hay que georreferenciar respecto a la entidad lineal y se representan su medida y desplazamiento.

Figura 42. Entidad lineal con sus segmentos geométricos



5.2.2. Usos de LRS

El LRS se usa principalmente por comodidad y precisión a la hora de introducir y manipular los datos. Las herramientas de medida de coordenadas no son o no han sido tan precisas como las herramientas de medida lineal. Además, muchas herramientas de medida de coordenadas dependen de estar en el exterior o de tener cobertura satélite. ¿Cómo medimos, pues, dónde están las salidas de emergencia de un túnel de metro? Se debe referenciar la vía de metro, definir puntos kilométricos midiendo la distancia a las estaciones de metro y ubicar las salidas de emergencia según a qué distancia (medida) y desplazamiento de la vía se encuentran.

Cuando se ha de cambiar la información también es una gran ventaja. Pongamos que la primera vez que queríamos inventariar las salidas de emergencia de metro no teníamos manera de georreferenciar el trazado de la vía. Sólo podíamos referenciar con GPS o topología las paradas de metro y medir la distancia de la vía de una parada a la otra. La distancia sí que la podíamos medir con detalle. Una vez tenemos todos los elementos de inventario referenciados en la vía, disponemos de las herramientas para georreferenciar todo su trazado. Al añadir esta información, todos los elementos se ubican allí donde les corresponde sin tener que volver a tomar las medidas.

Este método de almacenamiento también es muy cómodo para consultas como qué elementos están “cerca de” o “a una distancia x de”.

Fijaos cómo la topología se preocupaba de las relaciones de contacto, mientras que el LRS se preocupa de las relaciones de distancia. Sus usos son, por lo tanto, diferentes pero no excluyentes.


6. Almacenes *raster*

Hasta ahora habéis visto cómo se almacena la información espacial en formato vectorial. En este apartado veremos cómo se almacena la información de tipo *raster*. El hecho de dedicarle menos espacio dentro de este módulo no significa que los datos en este formato sean menos relevantes o menos abundantes que los datos vectoriales. Tanto los datos *raster* como los datos vectoriales tienen el mismo peso dentro del mundo de los SIG. Eso sí, los datos *raster* tienen menos diversidad y se almacenan normalmente de manera similar en la mayor parte de los formatos.

En este apartado veremos primero las características generales de los datos *raster*. Después veremos cómo se almacenan. Para hacerlo veremos los conceptos de bandas, capas, pirámides y mosaicos. Finalmente, haremos un repaso de los formatos más usuales de almacenamiento de *rasters*.

6.1. Información *raster*

Las entidades geográficas se pueden representar en formato vector, en formato *raster* o en los dos. Como habéis visto en el apartado 2, en formato vectorial los puntos se almacenan en coordenadas explícitas, las líneas se almacenan en un conjunto ordenado de puntos y las áreas se representan como polígonos cuyas fronteras son líneas.

En formato *raster* se pueden representar objetos espaciales cubriendo todo el espacio con una matriz de celdas y asignando valores concretos a las celdas que cubren el objeto. Este tipo de formato tiene menos precisión que el formato vectorial, ya que se ve limitado por el tamaño de las celdas, pero es ideal para ciertos tipos de análisis y cierto tipo de información. 

Entendemos como datos *raster* tanto las imágenes georreferenciadas como la información almacenada en parrillas como las coberturas. Dependiendo de la herramienta SIG que se utilice, puede dar más peso a un tipo o al otro, pero principalmente entendemos que se engloban dentro de los almacenes *raster*:

1) **Imágenes georreferenciadas.** Ortofotomágenes, imágenes satélite o cualquier otro tipo de fotografía que represente parte del territorio y se pueda ubicar sobre él. Estos datos, por la propia naturaleza de las fotografías, ya contienen información en forma de matriz de píxeles. En la figura 43 podéis ver una imagen aérea de la plaza Cataluña en Barcelona.


 En el apartado 6.6 veréis los formatos de almacenamiento.

Figura 43. Ortofotoimagen aérea

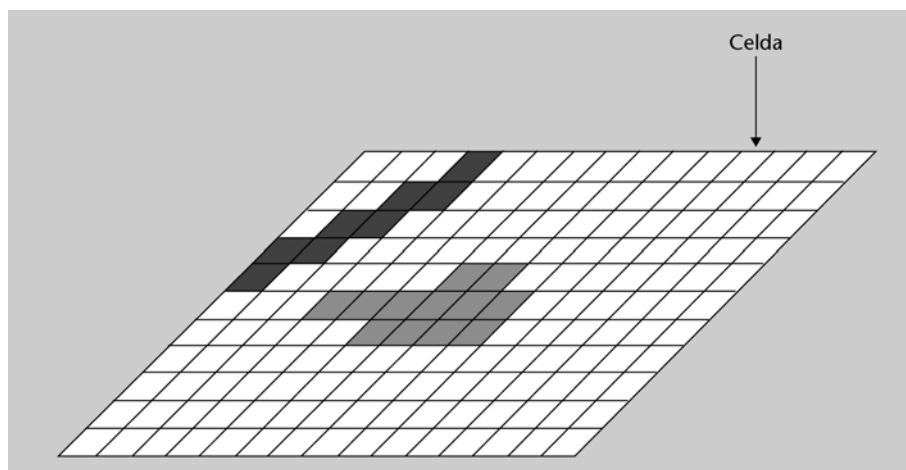


2) **Coberturas.** Matriz de valores sobre el territorio donde el valor almacenado en una celda representa el valor que toma una variable sobre la ubicación del territorio que cubre la celda. Las coberturas se pueden utilizar para representar variables distribuidas uniformemente sobre el territorio, como la altura del terreno, la contaminación del suelo, etc. También se pueden usar para representar objetos espaciales como lagos o ríos asignando el mismo valor a todas las celdas que cubren la entidad. Las diferencias con las imágenes son:

- Las coberturas representan una variable distribuida sobre el territorio. El valor del píxel no es un color, es el valor de la variable.
- Las coberturas pueden tener múltiples valores para un mismo píxel. Así, pueden representar la evolución de la variable durante el tiempo o varias variables.

En la figura 44 podéis ver una representación de una cobertura con un lago y una carretera.

Figura 44. Cobertura con sus celdas



6.2. Modelo de datos *raster*

A continuación veremos el modelo de almacenamiento de los datos *raster*. Los datos *raster* tienen algunos o todos los elementos siguientes:

- Las celdas o píxeles.
- Dominio espacial, es decir, qué área de territorio se cubre.
- Información de las bandas.
- Atributos de las celdas.
- Información de proceso.

El núcleo del almacenamiento *raster* es una matriz multidimensional de celdas. Entendemos un almacén *raster* como un conjunto de matrices de distintas dimensiones.

Cada una de las matrices tiene:

- Un número de dimensiones: normalmente son dos dimensiones para representar las coordenadas del plano. Poder disponer de más de dos dimensiones sirve, por ejemplo, para almacenar la evolución temporal de un atributo en una cobertura.
- La medida de cada dimensión: nos indica cuántas celdas de ancho y de alto tiene la matriz.
- Una profundidad de celda: el tamaño de los datos de todos los valores de celda.

Las matrices están formadas por celdas. Cada celda es un elemento de la matriz y su valor se denomina *valor de celda*. Si el almacén *raster* representa una imagen, la celda también se puede llamar *píxel*.

6.3. Bandas y capas

La información de las matrices de un almacén *raster* se agrupa en bandas y capas. En los almacenes *raster* los conceptos de lado o capa son similares. Una banda es una dimensión física del conjunto de datos *raster* multidimensional, es decir, es una ordenada dentro del espacio de una celda. Por ejemplo, una celda podría tener la fila, la columna y la banda por ordenadas.

La capa es un concepto lógico que se hace coincidir con las bandas. Típicamente, una capa corresponde a una banda. En resumen:

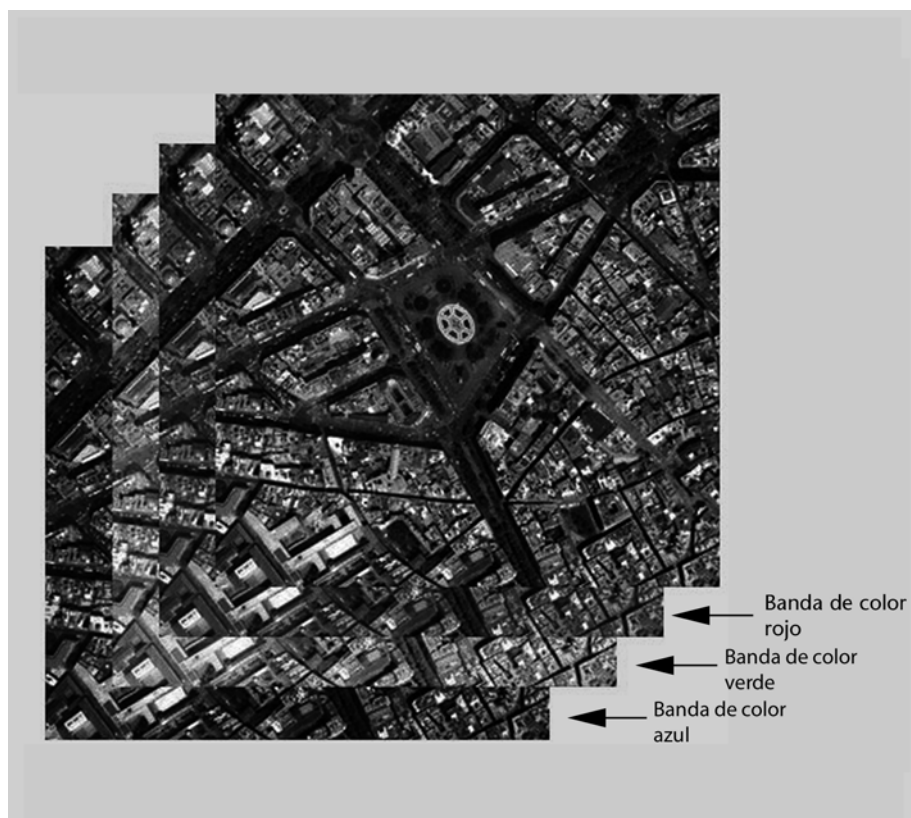
Una **capa** es una realidad que almacenar que se debe poder escribir, leer y representar. Dentro de un mapa almacenado como imagen *raster* se organizan los conceptos lógicos (ríos, casas, carreteras, etc.) como capas. Una capa es un concepto lógico.

Una **banda** es una ordenada dentro del formato de almacenamiento de imágenes *raster* que se puede usar para almacenar capas. Una banda es un concepto físico que se puede utilizar para almacenar capas u otra información (como los diferentes espectros de una fotografía en RGB).

Las bandas también sirven para almacenar valores de diferentes canales por los sensores que capturan imágenes o valores geográficos en diferentes canales. Las imágenes multiespectrales, por ejemplo, pueden contener muchos canales. En una imagen fotográfica normal podemos separar la información en un canal para el color rojo, un canal para el color verde, un canal para el color azul y un canal para el color negro. La composición de todos los canales nos da la imagen resultante. Cada canal corresponderá a una banda del almacén *raster*.

En la figura 45 podéis ver las diferentes bandas de una ortofotoimagen almacenada en RGB.

Figura 45. Representación de las bandas de colores de una imagen RGB



6.4. Pirámides

Las imágenes *raster* pueden tener mucha resolución. Al generar mapas a una escala muy grande, podríamos tener que acceder a gran cantidad de imágenes para cubrir todo el territorio que queremos representar. Como recorrer todos los píxeles de todas las imágenes puede ser muy costoso, se construyen las pirámides.

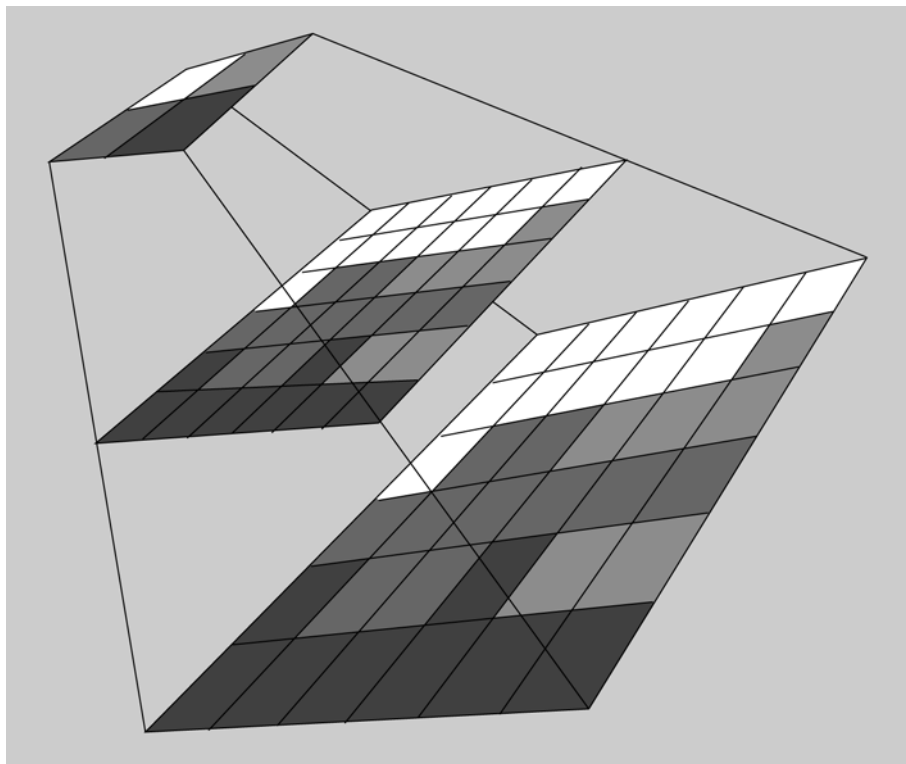
Las pirámides son una serie de representaciones de la imagen *raster* con resolución reducida que principalmente se usan para mejorar el rendimiento cuando se trabaja en escalas que no requieren toda la resolución de la imagen. Contiene un conjunto de niveles cada uno de los cuales es una representación de la misma imagen a menos resolución. En la figura 46 podéis ver una representación de una pirámide de 3 niveles, cada nivel con menos resolución que el anterior.

Las pirámides permiten a las herramientas SIG tomar la información a la misma resolución que necesitan para representar el mapa. Si se intenta abrir una imagen de alta resolución aunque sea para verla desde una visión general, podemos colapsar el sistema operativo, ya que se ha de leer muchísima información para representar poca.

Cuando la imagen *raster* se modifica, se debe rehacer toda la pirámide. Este proceso se realiza “en frío” antes de que las herramientas SIG hayan de con-

sultar la nueva información. La producción de la pirámide requiere tomar una muestra de varios píxeles para producir menos píxeles de salida. De esta manera se generan los niveles de menos resolución. En cada nivel de la pirámide la medida de un píxel se duplica y así se reduce el número de píxeles a una cuarta parte.

Figura 46. Pirámide *raster*



La pirámide empieza en la base o nivel 0 con la imagen *raster* con la resolución original. A partir de este nivel, se van generando los niveles superiores hasta que sólo quedan 4 píxeles o hasta un número límite de niveles. Cada uno de los nuevos niveles generados es una matriz multidimensional como si se tratara de un almacén *raster* independiente. Hay distintos métodos para reducir la resolución de la imagen que funcionan por interpolación. A continuación os mostramos algunos ejemplos:

1) **Vecino más próximo (*nearest neighbour*)**. El método del vecino más próximo es un método de interpolación que emplea el valor del píxel más próximo para interpolar el valor del píxel del nivel siguiente (figura 47). Es el método más rápido y el más idóneo para información nominal (como usos del suelo).

2) **Bilineal**. El método de interpolación bilineal toma el valor de la media de los cuatro píxeles que corresponden al píxel del nivel superior (figura 48). Este método es idóneo para información de altura del territorio, nivel de nitrato, calidad del suelo o cualquier otra variable que se reparta uniformemente por el territorio.

Figura 47. Interpolación por vecino más próximo

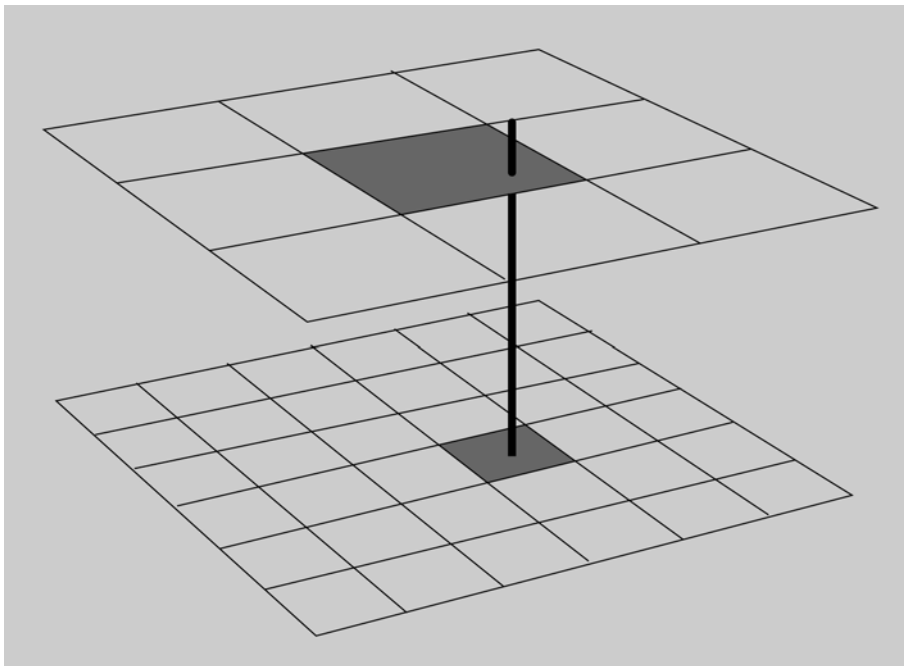
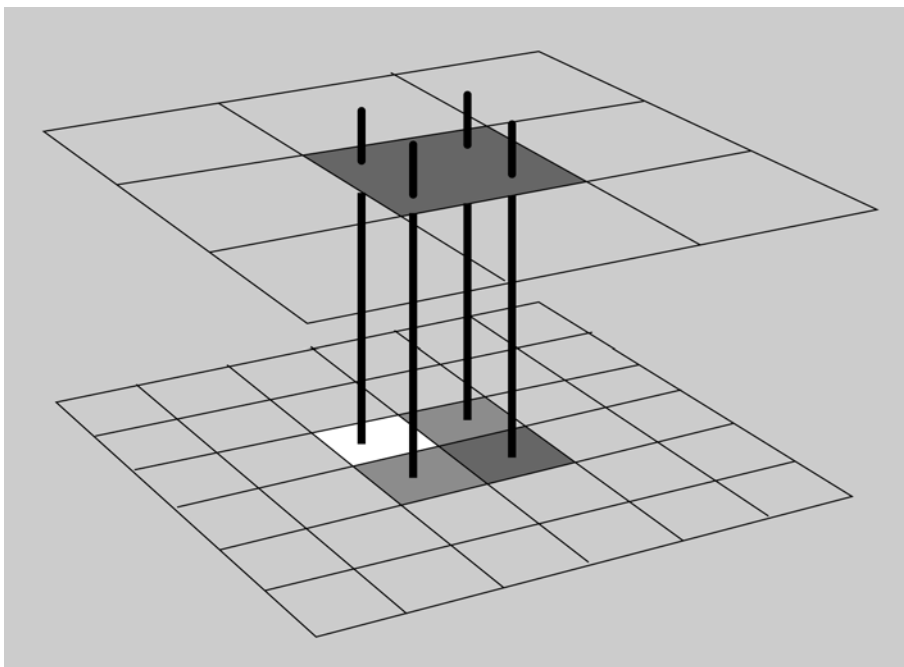


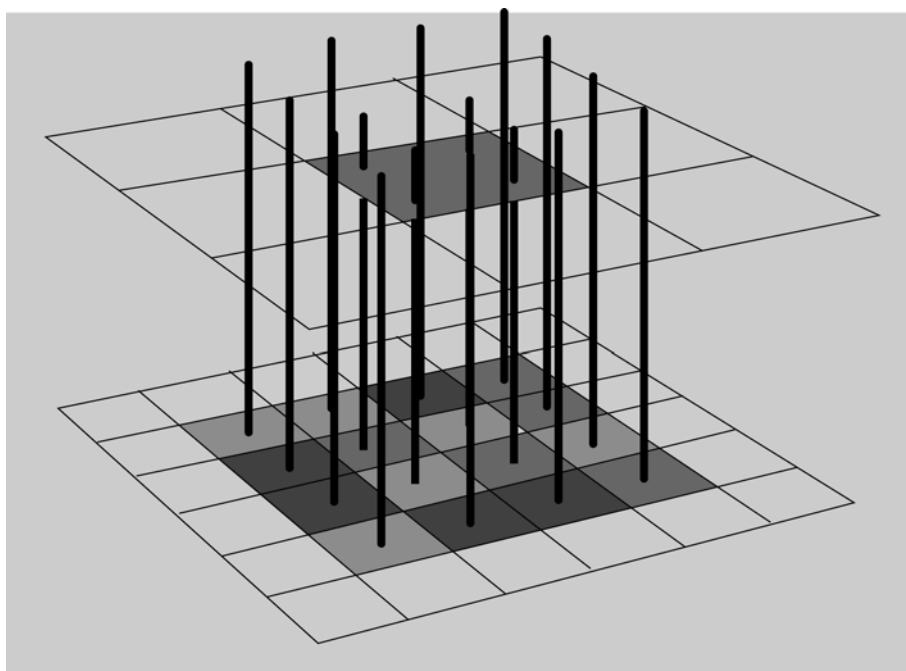
Figura 48. Interpolación bilineal



3) Bicúbico. El método de interpolación bicúbico estima el valor del píxel destino según la media de los 16 píxeles de origen que rodean el píxel destino (figura 49). Esta interpolación genera una sensación de continuidad y transiciones suaves y es idóneo para las imágenes satélite u ortofotoimágenes.

En resumen, el objetivo de las pirámides es que las herramientas SIG no hayan de consultar en ningún momento una imagen con una resolución demasiado elevada que comprometa su rendimiento, pero sin perder calidad o resolución al navegar.

Figura 49. Interpolación bicúbica



Es muy habitual generar pirámides a partir de mosaicos de imágenes (los veréis en el subapartado siguiente). Encontraréis la explicación de las pirámides con mosaico después de ver el concepto de mosaico.

6.5. Mosaicos

En combinación con las pirámides también se utiliza otro concepto para la organización de los datos *raster*, los mosaicos. Los mosaicos son imágenes que se han creado juntando dos o más imágenes en una sola. Sobre todo en ortofotoimágenes o mapas topográficos en formato *raster*, sirven para dar la sensación de continuidad en el mapa. Las herramientas SIG pueden consultar imágenes *raster* de diferentes fuentes y representarlas todas juntas, pero el pintado realizado por herramientas SIG tiene los problemas siguientes:

- Al consultar muchas fuentes, las herramientas SIG no tienen tanto rendimiento porque se deben mantener muchas conexiones abiertas.
- Al tratar las imágenes de manera independiente, las transiciones entre imágenes son bruscas. Las imágenes aéreas, por ejemplo, se toman en diferentes días con diferentes condiciones de iluminación. La transición entre una fotografía aérea y otra puede ser muy brusca.
- Al tratarse como imágenes diferentes, no puede construirse una pirámide global que permitiría ver las imágenes a mucha escala. Si se construye una pirámide para cada imagen independientemente, las transiciones entre imágenes pueden ser todavía más bruscas.

Para evitar estos problemas, es muy habitual construir mosaicos de imágenes. En un mosaico de imágenes cada una de las imágenes de origen se denomina *corte*. El proceso de generación de un mosaico a partir de sus cortes puede ser muy costoso, ya que se trata de suavizar las transiciones entre cada corte y sus vecinos.

Ahora que ya hemos visto las pirámides y los mosaicos, podemos ver las dos técnicas aplicadas a la vez. Aplicar las dos técnicas permite generar índices de acceso a información *raster* muy extensa y es muy habitual.

Las pirámides con mosaico son pirámides en las que cada nivel está formado no por una imagen, sino por una matriz de imágenes. Cada nivel de la pirámide es un mosaico dividido en una imagen por cada celda. El mosaico aporta su valor de continuidad y suavidad entre los cortes. La pirámide aporta su valor de rendimiento a la hora de visualizar las imágenes a una escala elevada. La separación de cada nivel en celdas permite no tener que cargar todo el nivel de la pirámide para representar sólo una parte. Con la conjunción de estas tres técnicas se puede generar una imagen *raster* de alta definición de todo el mundo que permite acercarse a cualquier lugar del mundo y con un alto rendimiento. Un ejemplo de este método es la imagen de fondo satélite de Google Maps.

Google Maps: <<http://maps.google.com>>

6.6. Formatos de almacenamiento *raster*

Ahora que ya hemos visto las técnicas de almacenamiento *raster*, haremos un repaso de los formatos comerciales más habituales.

6.6.1. Archivos World

Un archivo World es un fichero de texto plano que se usa para georreferenciar la imagen contenida en un fichero *raster*. Este archivo no contiene información de la imagen *raster*, sino que acompaña al fichero que la contiene. Su extensión depende del formato que acompaña. Las más habituales son .tfw, .sdw y .jpw y corresponden a los formatos de imagen .tif, .sid y .jpg, respectivamente.

Los archivos World contienen seis líneas con números decimales que describen la referencia espacial de la imagen:

- Línea 1: medida del píxel en el eje de coordenadas x
- Línea 2: rotación sobre el eje de las y
- Línea 3: rotación sobre el eje x
- Línea 4: medida del píxel en la herramienta de coordenadas y
- Línea 5: coordenada x de la esquina superior izquierda de la imagen
- Línea 6: coordenada y de la esquina superior izquierda de la imagen

6.6.2. GeoTiff

El formato GeoTiff es un estándar de metadatos público que permite la georreferenciación de la información contenida en un archivo Tiff. Permite incluir dentro del fichero Tiff la información necesaria para establecer la referencia espacial del fichero. Esto incluye la proyección, el sistema de coordenadas, el elipsoide y el datum. El formato GeoTiff cumple con la especificación de los ficheros Tiff generales y por ello se pueden abrir desde cualquier aplicación no SIG que permita ver ficheros Tiff.

6.6.3. MrSID

El formato de datos MrSID (conocido popularmente como Mister SID) es un acrónimo de *multiresolution seamless image database*. Se trata de un formato de almacenamiento de imágenes sobre fichero desarrollado y propiedad de LizardTech. Su extensión es .sid y normalmente lo acompaña un archivo World de referencia espacial con la extensión .sdw.

LizardTech:
<<http://www.lizardtech.com>>

Las características principales del formato son la capacidad de contener mosaicos y pirámides internos al formato. Con el mosaico se consigue una alta compresión de las imágenes digitales con una pérdida mínima de resolución.

Este formato se utiliza mucho con las herramientas SIG, ya que permite mover imágenes de alta calidad y resolución con tamaños de ficheros muy comprimidos (con una proporción de compresión de hasta 22:1). También lleva implementado el concepto de pirámide dentro del fichero, por lo que, al consultarlo, las herramientas SIG sólo han de leer la información esencial para obtener la imagen a la resolución que necesiten.

6.6.4. ECW

El ECW (*enhanced compression wavelet* u ola de alta frecuencia de mayor compresión) es un formato de almacenamiento de imágenes estándar abierto con compresión con *wavelets* (u olas de alta frecuencia). El formato está optimizado para imágenes aéreas o satélite y comprimirlas con la mínima pérdida posible. Contiene la información de la referencia espacial dentro del fichero. Al generar el archivo se comprime a alta velocidad y se puede comprimir con proporciones de 10:1 hasta 100:1. Además, se puede descomprimir una parte del archivo sin tener que descomprimirlo todo.

El formato ha sido desarrollado por Earth Resource Mapping y lo pueden leer la mayoría de las herramientas SIG comerciales. A pesar de ser un estándar abierto, no hay herramientas gratuitas que permitan trabajar con este formato.

Earth Resource Mapping:
<<http://www.ermapper.com>>

6.6.5. Georaster (Oracle Spatial)

Georaster es una capacidad de Oracle Spatial que permite almacenar, indexar y consultar imágenes *raster* y coberturas. Este formato permite almacenar las imágenes dentro de un SGBD accediendo a las imágenes con las capacidades de consulta e indexación del motor relacional.

Georaster almacena la información de la referencia espacial en tablas y objetos internos del SGBD. Permite la generación de pirámides y mosaicos.

Con el fin de introducir los datos al formato Georaster, se deben migrar de uno de los formatos de ficheros anteriores o *rasterizar* a partir de información vectorial. Los procesos de carga y actualización del SGBD pueden ser costosos, mientras que con los formatos de archivo sólo hay que copiar las nuevas actualizaciones.

Introducir las imágenes *raster* dentro de una base de datos relacional tiene ventajas como la capacidad de indexar, realizar copias de seguridad o disponer de todos los datos ordenados. Sin embargo, los inconvenientes de la dificultad de carga y mantenimiento, junto con el hecho de que no se mejora el rendimiento de consulta respecto a los ficheros, provocan que este formato no se haya impuesto y todavía en la actualidad los formatos en fichero, al contrario que el almacenamiento en vectorial, dominan el mercado.

7. Software para bases de datos geográficas

Hace algunos años, la elección de una base de datos era bastante simple: si se disponía de *mainframes* y hardware de IBM, DB2 era la elección correcta. Si se disponía de sistemas operativos Unix, Oracle era la elección adecuada, y si la infraestructura se basaba en la plataforma Windows, SQL Server era la elección favorita. Sin embargo, con los años cada fabricante ha ido incorporando nuevas funcionalidades y mayores capacidades para aumentar el rendimiento, así como la compatibilidad con otras plataformas, con lo que la plataforma ya no condiciona tanto la elección de la base de datos.

El objetivo de este apartado es hacer un repaso al panorama actual del software de bases de datos geográficas. Para ello hemos escogido los productos de bases de datos más conocidos, tanto de código abierto como de tipo privativo, y comparamos las funcionalidades espaciales básicas de algunos de ellos.

En el capítulo “Principales formatos de almacenamiento”, ya habréis oído hablar de varias de las bases de datos que aparecen aquí. Sin embargo, allí nos centrábamos más en el aspecto del almacenamiento y aquí ofrecemos una visión más general de sus funcionalidades. De esta manera dispondréis de una hoja de ruta que esperamos que os sea de ayuda en la decisión del uso de uno u otro SGBD geoespacial según las especificaciones del proyecto a desarrollar.

El capítulo se estructura en dos bloques:

- En el primer bloque os presentamos las bases de datos geográficas.
- En el segundo bloque tenéis una comparativa entre ellas.

7.1. Ejemplos de bases de datos

En este apartado os presentamos las siguientes bases de datos:

- MySQL
- PostgreSQL/PostGIS
- Oracle Spatial
- Microsoft SQL Server 2008 Spatial
- Spatial Extender para IBM DB2

Hemos creado una ficha para cada una en la que podéis encontrar los siguientes puntos:

- URL: el enlace a la web del producto.

- Versión estable: la última versión estable a fecha del cierre del presente texto.
- Licencia: el tipo de licencia con el que se comercializa la base de datos.
- Descripción corta: una breve descripción que ayuda a situar la base de datos.
- Finalmente, encontraréis algunas características de la base de datos que os pueden orientar a la hora de escogerla o desestimarla.

MySQL	
URL	http://mysql.com/
Versión estable	[5.1] [http://dev.mysql.com/downloads/]
Licencia	GPL, GNU General Public License 2.0. Dispone de licencia dual, con funcionalidades extra en la versión no libre.
Descripción corta	MySQL es la base de datos libre de Sun Microsystems.

MySQL es la base de datos más popular dentro del ámbito del software libre, especialmente en combinación con el servidor web Apache y el lenguaje de programación PHP. En el ámbito geoespacial es mucho menos utilizada, ya que no cumple con los estándares internacionales, ni incorpora toda la funcionalidad y potencia que ofrece PostGIS.

Aun así, en ocasiones puede ser interesante utilizar sus funcionalidades geoespaciales, por ejemplo, en la integración con sistemas ya existentes.

PostGIS	
URL	http://postgis.refrations.net/
Versión estable	[1.4.0] July 24, 2009
Licencia	GPL, GNU General Public License 2.0.
Descripción corta	PostGIS es la extensión geoespacial de la base de datos libre PostgreSQL

No hay duda de que el buque insignia en el ámbito de los gestores de bases de datos del software libre es PostgreSQL. Con PostGIS, el módulo para PostgreSQL desarrollado principalmente por *Refrations Research Inc.* PostgreSQL adquiere la capacidad de almacenar información geoespacial y de realizar operaciones de análisis geográfico. Además, utilizando la extensión pgRouting se puede emplear como un potente motor de cálculo de rutas y otras operaciones geoespaciales.

El cumplimiento del estándar SFSS es uno de los factores donde radica su potencia, convirtiendo a PostGIS en la base de datos espacial casi por excelencia, que se enlaza con múltiples herramientas SIG. Así, los datos PostGIS se pueden analizar con GRASS, se pueden publicar con Mapserver y MapGuide, pueden visualizarse con uDig, gvSIG or QGIS entre otros SIG de escritorio y se pueden exportar a formatos propietarios con OGR y FDO.

Oracle Spatial	
URL	http://www.oracle.com/lang/es/database/spatial.html
Versión estable	[11g] http://www.oracle.com/technology/software/products/database/index.html
Licencia	OTN License.
Descripción corta	Oracle Spatial es la extensión espacial de la BD Oracle.

Oracle Spatial se ha incluido en cada edición de la base de datos Oracle. Proporciona servicios para administrar los datos y modelos geoespaciales, ya sean los datos vectoriales o ráster, así como modelos de red y reglas de topología.

Los objetos espaciales, ya sean simples o compuestos, se almacenan mediante el tipo de datos SDO_GEOMETRY, que garantiza la interoperabilidad de Oracle con otros sistemas, como por ejemplo los sistemas de información geográfica.

Referencia bibliográfica

Los datos de este apartado se han tomado de:

http://www.construmatica.com/construpedia/Bases_de_Datos_Geoespacial_es._Herramientas_de_SIG_Libre

IBM DB2 con Spatial Extender	
URL	http://www-01.ibm.com/software/data/spatial/db2spatial/features.html
Versión estable	DB2 9.7
Licencia	IBM DB2 License
Descripción corta	Extensión espacial de la base de datos relacional DB2.
<p>Spatial Extender genera y analiza información espacial, además de almacenar y gestionar los datos a partir de los que se ha generado dicha información. Integra la herramienta "Spatial Index Optimizer for DB2" para optimizar el rendimiento de los índices espaciales. Se puede combinar con herramientas de <i>business intelligence</i> propias de IBM, que conjuntamente con los sistemas de información geográfica, ayudan a la toma de decisiones de negocio.</p>	

Microsoft SQL Server 2008 Spatial	
URL	http://www.microsoft.com/sqlserver/2008/en/us/spatial-data.aspx
Versión estable	R2 CTP
Licencia	SQL Server Licensing.
Descripción corta	SQL Server 2008 Spatial proporciona soporte de datos espaciales para SQL Server 2008.
<p>SQL Server 2008 Spatial extiende los tipos de datos soportados por SQL Server 2008, tanto para la versión Express como para la Enterprise Edition. Soporta la definición y el tratamiento de objetos espaciales de gran complejidad, así como consultas espaciales. Optimiza estas consultas mediante la creación de índices espaciales, siguiendo los mecanismos estándares de OGC para las bases de datos geoespaciales.</p> <p>Aunque ofrece las funciones básicas de geoproceto, como intersección y creación de geometrías, no soporta geocodificación, ni cálculo de rutas ni modelos topológicos.</p> <p>Podéis ver una entrevista con relación a la presentación de la extensión espacial de SQL Server 2008 en el siguiente enlace:</p> <p>http://www.directionsmag.com/editorials.php?article_id=2477&trv=1</p>	

7.2. Comparativa entre SQL Server 2008, MySQL y PostGIS

En la comparativa presentada, hemos prestado especial atención a las capacidades de análisis espacial que poseen las distintas bases de datos geográficas. La comparación se realiza entre SQL Server 2008 Spatial, MySQL y PostGIS. Se han escogido estas bases de datos por tener un menor coste de instalación y mantenimiento en comparación con DB2 y Oracle Spatial.

En cualquier caso, dada la importancia de Oracle Spatial, en el siguiente apartado compararemos PostGIS con Oracle Spatial.

	SQL Server 2008	MySQL	PostGIS
Sistema operativo	Windows XP, Windows 2000+	Windows XP, Windows 2000+ (excepto la versión 2008), Linux, Mac.	Windows XP, Windows 2000+ (incluyendo Vista y 2003. Falta probar en 2008), Linux, Unix, Mac
Licencia	Microsoft License	GPL	GPL
Instalación y recursos	Moderado	Fácil	Moderado

	SQL Server 2008	MySQL	PostGIS
Lectura por SIG <i>open source</i>	Shp data loader for SQL Server 2008	OGR2OGR, shp2mysql.pl script	shp2pgsql, OGR2OGR, SPIT, SHP loader para PostGIS, ...
Lectura por SIG privativo	Safe FME Objects, ESRI ArcGIS 9.3	Safe FME Objects	Safe FME Objects, ESRI ArcGIS 9.3
Disponibilidad de controladores específicos para componentes espaciales.	No	GDAL C++, AutoCAD FDO, SharpMap via OGR.	SharpMap.Net, JDBC postgres.jar incluido con postGIS, JTS etc. GDAL para C++, AutoCAD FDO BetaSupport.
Disponibilidad de drivers: ODBC, JDBC, ADO.NET	Sí	Sí	Sí
Visores de escritorio y editores gratuitos.	SQL Manager	GvSIG	OpenJump, QuantumGIS, gvSIG, uDIG
Visores de escritorio y editores comerciales.	Server SDE en el último service pack, Manifold, FME	FME	ESRI ArcGIS 9.2 Server, ZigGIS para escritorio, Manifolds, FME
Herramientas de webmapping	Manifold, MapDotNet, ArcGIS 9.3, UMN MapServer, MapGuide Open Source	UMN MapServer, GeoServer, MapGuide Open Source	Manifold, MapDotNet, ArcGIS 9.3, UMN MapServer, GeoServer, Feature Server, MapGuide Open Source
Funciones espaciales	Unas 70 funciones básicas de proceso espaciales basadas en el OGC SFS.	Funciones básicas de relaciones espaciales sólo para 2D.	Más de 300 funciones y operadores OGC SFS, sin soporte geodésico total.
Índices funcionales	No	Sí	Sí
Indexado espacial	Sí, basado en R-Tree	Sí, R-Tree quadratic splitting	GIST, variante de R-Tree.
Soporte geodésico real	Sí, con limitaciones.	No	No
Herramienta gráfica sin coste adicional	SQL Management Studio	No	PgAdmin III.

Un análisis rápido de las funcionalidades espaciales contrastadas nos revela PostGIS como la mejor de las tres opciones. Vamos ahora a comparar la funcionalidad de este producto con la de Oracle Spatial.

7.3. Comparativa entre Oracle Spatial y PostGIS

Oracle Spatial y PostGIS son dos de los grandes productos de software de bases de datos espaciales. Analizaremos brevemente sus puntos comunes y sus divergencias.

Uno de los puntos a los que se prestará especial atención es a la compatibilidad con el estándar de OGC para el almacenamiento y acceso a datos en bases de datos relacionales. Es lo que se conoce como el estándar SFS, del inglés *simple features interface*.

Aparte de este aspecto, también compararemos la capacidad de cada una para la importación de datos espaciales, así como el tratamiento que posteriormente se hace de éstos.

7.3.1. Importación de datos espaciales

Tanto Oracle Spatial como PostGIS disponen de un comando para la importación de ficheros en formato *shape* (*shp2pgsql* en PostGIS y *shp2sdo* en Oracle Spatial). Estos comandos se invocan desde la consola del sistema e importan tanto los datos espaciales como los temáticos asociados (.dbf).

7.3.2. Especificación SFS

Aunque en la documentación de ambos productos se asegura que siguen la especificación SFS del OGC (que han pasado con éxito el test de comprobación del OGC*), hay que matizar alguna cosa al respecto:

- Oracle Spatial sigue la especificación pero de una manera relajada, es decir, los métodos definidos en SFS tienen su correspondencia en métodos de Oracle Spatial pero éstos tienen diferente el nombre, número o tipo de argumentos. Esto es una gran limitación, ya que se necesita traducir el código SQL de Oracle si se quiere exportar a otros sistemas, incluso aunque éstos soporten de una manera estricta la especificación del OGC.
- PostGIS sigue de una manera mucho más fiel la especificación SFS, respetando los nombres de los métodos (salvo alguna excepción) y el número de argumentos. PostGIS almacena también la información de metadatos de las columnas de geometría de las tablas (GEOMETRY_COLUMNS) y de los sistemas de referencia espacial siguiendo el estándar SFS (SPATIAL_REFERENCE_SYSTEMS).
- Ambos sistemas incorporan métodos no definidos por el OGC en su especificación, pero necesarios si se quieren realizar análisis complejos o procedimientos almacenados que incrementen la funcionalidad del sistema.

* Open GIS Consortium, Inc.
(1998): Conformance Test
Guidelines for OpenGIS Simple
Features Specification (SFS) for
SQL.Rev. 1.0

7.3.3. Tablas espaciales

PostGIS y Oracle Spatial realizan de forma diferente la creación de una tabla de geometría:

- PostGIS utiliza el método 'addGeometryColumn' (método propuesto en la SFS), que se encarga de añadir el campo de geometría del tipo seleccionado a una determinada tabla.

- Oracle no implementa el método 'addGeometryColumn', sino que define la tabla directamente.

Por tanto, PostGIS sigue más fielmente el estándar SFS. Esto, que puede parecer una ventaja, puede comportar problemas porque el método 'addGeometryColumn' de SFS es más rígido que el de Oracle Spatial. Por ejemplo, PostGIS limita el tipo de geometrías almacenadas en una tabla de manera que, si se quiere insertar un tipo de geometría no coincidente con la restricción de la tabla, el sistema dará un error de violación de restricción. Este tipo de problemas no se da en Oracle Spatial, que no establece este tipo de restricciones en las tablas.

Resumen

En este módulo hemos estudiado cómo se almacenan los diferentes datos que manejan las herramientas SIG. Hemos empezado por ver que la diferencia entre el almacenamiento en fichero o en base de datos radica en el volumen de datos que se pueden almacenar y el rendimiento de acceso a los datos. Después de repasar las características principales de las bases de datos relacionales, hemos visto cómo se almacenan las geometrías vectoriales tanto dentro de un fichero como en una base de datos. Las geometrías vectoriales se almacenan como colecciones de puntos donde cada punto se define como un conjunto de coordenadas. Se ha visto la complejidad de diseñar bases de datos consistentes e íntegras para almacenar realidades complejas como motivo principal que lleva a que la mayoría de los formatos para almacenar las geometrías no sigan las formas normales.

Una vez hemos visto cómo se guardan los datos vectoriales, hemos visto los operadores y las funciones que permiten relacionar los datos teniendo en cuenta el territorio. También cómo se arreglan los datos mediante índices espaciales como el *R-tree* con el fin de minimizar el tiempo de consulta. Finalmente, hemos podido ver cómo se pueden realizar consultas con Oracle Spatial con el fin de explotar las relaciones espaciales y aprovechando la optimización de los índices como ejemplo de motor de consulta de datos.

También hemos repasado formatos avanzados de almacenamiento vectorial, como son la topología y la segmentación lineal, que, a pesar de ser complementarios y optativos, aportan grandes ventajas en algunos casos y son imprescindibles para algunas funcionalidades. Por ejemplo, no se puede realizar un cálculo de rutas sin topología.

Para acabar, hemos visto cómo se almacenan las imágenes *raster*, que completan, con los datos vectoriales, la fuente de datos de las herramientas y aplicaciones SIG. Hemos repasado los conceptos que permiten el almacenamiento de imágenes *raster* como son las capas, las bandas, las pirámides y los mosaicos.

Se ha enumerado la lista de los formatos de datos principales que os podéis encontrar al trabajar con herramientas SIG tanto para datos vectoriales como para datos *raster*. Hemos visto desde *shapefiles* a Oracle Spatial, pasando por archivos GeoTiff o MrSID, sin olvidar las marcas comerciales más extendidas como son ESRI, SmallWorld y Geomedia o el formato de intercambio más aceptado como es GML.

Hemos visto que en general para los datos vectoriales, con un volumen de datos pequeño, el almacenamiento en ficheros es más cómodo y no aporta nin-

gún inconveniente. A medida que los volúmenes de datos van aumentando, crece la necesidad de ordenar e indexar los datos y se hace necesario el almacenamiento en una base de datos relacional. Para sistemas geográficos corporativos con muchos usuarios accediendo a ellos y modificando los datos de manera concurrente, el almacenamiento en bases de datos es imperativo.

Los datos *raster*, por contra, no suelen ser muy variables y el almacenamiento en bases de datos relacionales puede ser más un inconveniente que una ventaja. De todos modos, para sistemas con grandes volúmenes de datos y procesos de negocio geográficos con alta criticidad, el almacenamiento en bases de datos relacionales también es recomendable.

Ejercicios de autoevaluación

1. Diseñad una base de datos normalizada (es decir, que cumpla las formas normales) para almacenar datos de empleados, departamentos y sedes, teniendo en cuenta lo siguiente:

- Un empleado tiene nombre, apellido, DNI y cargo.
- Los departamentos están distribuidos en las distintas sedes.
- Una sede tiene una dirección y puede acoger más de un departamento.
- Hay que almacenar el número de empleados que trabajan en una sede.

2. ¿A qué tabla de vuestro diseño para el ejercicio anterior creéis que se podría añadir un campo geográfico?

3. Debéis almacenar los siguientes datos. ¿Qué formato creéis que sería mejor para cada uno?:

- Una red de gas.
- Lagos.
- Recorridos con bicicleta.
- Ríos.
- Los extintores de un túnel de metro.
- El mapa topográfico.
- Lugares de interés.
- Fotografías por satélite.
- La elevación del territorio.
- Zonas verdes.
- Guardarraíles de carretera.
- Hospitales.

Puntos	Líneas	Áreas	Topología	LRS	Imagen <i>raster</i>	Cobertura

4. ¿Qué datos se deben guardar para almacenar las entidades geográficas siguientes (fijaos en el ejemplo)?:

- Un punto vectorialmente: “La coordenada x , la coordenada y y opcionalmente la cota”.
- Un punto con LRS.
- Las vías del tren.
- Una superficie rectangular.
- Una superficie rectangular con un agujero rectangular.
- Dos superficies cuadradas limítrofes con topología.

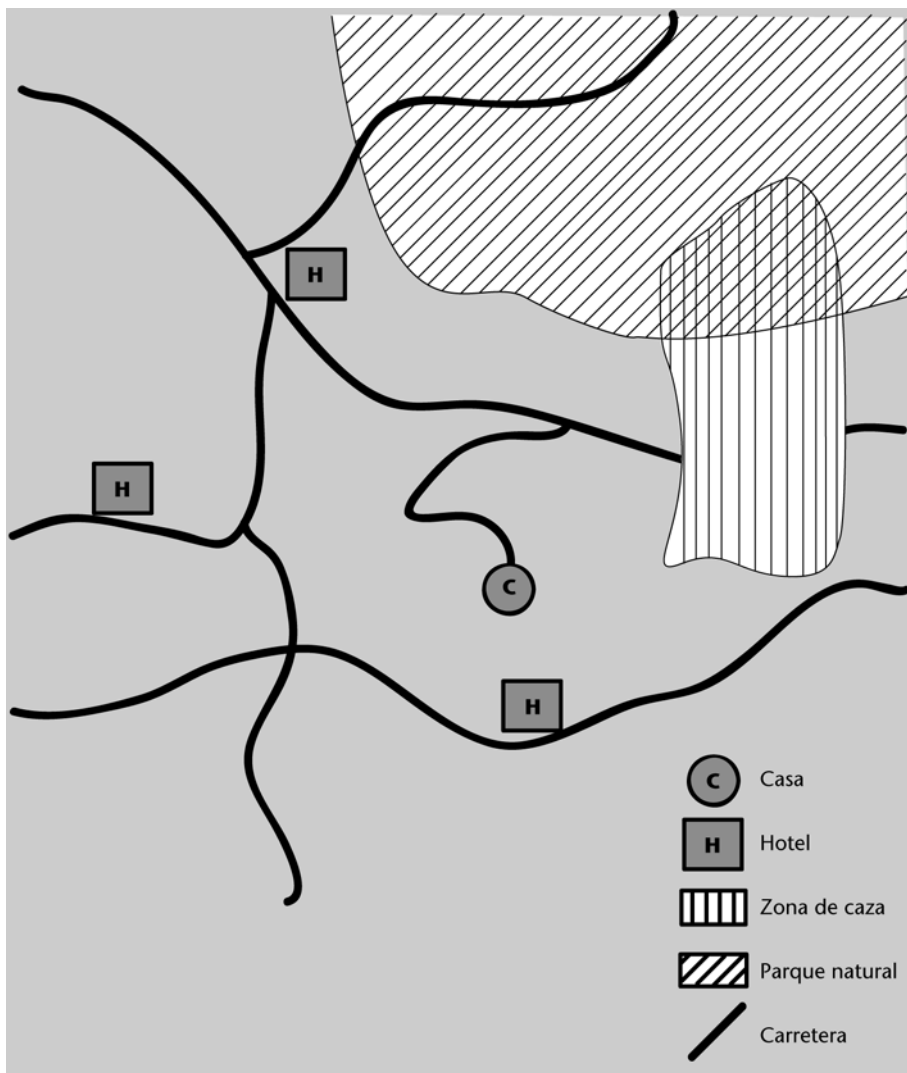
5. Clasificad los formatos comerciales de almacenamiento geográfico siguientes según si tienen o no capacidad para almacenar formato vectorial y formato *raster* (pensad que pueden tener ambas capacidades).

- ArcSDE.
- MrSID.
- Smallworld.
- ESRI shapefile.
- Oracle Spatial.
- GeoTiff.
- GML.

Formato vectorial	Formato <i>raster</i>

6. Dado un mapa como el siguiente –donde tenemos las capas Hoteles, Casas, Carreteras, Zonas de caza y Parques Naturales–, ¿qué operadores y funciones usaríais y sobre qué entidades para calcular lo siguiente?:

- Los hoteles más próximos a la casa C.
- Si hay zonas de caza en los parques naturales.
- Qué carreteras cruzan el parque natural.
- El área mínima para representar la casa y las zonas de caza.



7. La consulta de Hoteles más próximos del ejercicio anterior no responde a la pregunta de qué hotel está más cerca yendo en coche por carretera desde la casa C. Para poder responder a esta pregunta, ¿qué modelo de almacenamiento creéis que habría que usar y qué algoritmo aplicaríais?

Solucionario

1.
 - a) Empleados (**dni**, nombre, apellido, cargo, departamento)
 - b) Departamentos (**nombre_departamento**)
 - c) Sedes (**nombre_sede**, dirección, número empleados)
 - d) Departamentos_sedes (nombre_departamento, nombre_sede)

Nota: sería mejor identificar las tablas con códigos numéricos y así las claves foráneas serían también numéricas. Si lo habéis hecho así, mejor.

2. La única entidad que tiene componente geográfico son las sedes. Añadiríamos, pues, las coordenadas de la ubicación en la Tierra de la sede en la tabla de Sedes.

3.

Puntos	Líneas	Áreas	Topología	LRS	Imagen raster	Cobertura
Lugares de interés Hospitales	Recorridos con bicicleta Ríos	Lagos Zonas verdes	Una red de gas	Los extintores de un túnel de metro Guardarraíles de carretera	El mapa topográfico Fotografías por satélite	La elevación del territorio

4.

- b) El segmento lineal, la coordenada M y el desplazamiento.
- c) Una línea formada por una serie de puntos. De cada punto se debe guardar la coordenada x y la coordenada y .
- d) Una línea formada por 5 puntos donde el primer punto es igual al último.
- e) Se debe almacenar una frontera exterior y una frontera interior. Las dos definidas como una línea formada por 5 puntos donde el primer punto es igual al último.
- f) Se deben guardar todos los vértices de los cuadrados sin repetición, es decir, 6 vértices. Se han de guardar también todos los segmentos entre los vértices, es decir, 7 segmentos. También se deben guardar las dos caras de los rectángulos.

- De cada vértice guardaremos las coordenadas x e y .
- De cada segmento guardaremos las referencias al vértice de origen, al vértice final y al segmento siguiente.
- De cada cara guardaremos la referencia a una arista de su perímetro.

5.

Formato vectorial	Formato raster
ArcSDE	ArcSDE
Smallworld	MrSID
ESRI shapefile	Smallworld
Oracle Spatial	Oracle Spatial
GML	GeoTiff

6.

- a) La función “Vecino más próximo entre las entidades Casas y Hoteles”. Se debe aplicar un filtro adicional que indique que de la entidad Casas sólo se debe tener en cuenta la Casa C.
- b) El operador “Interseca” entre las entidades Zonas de caza y Parques Naturales.
- c) La función “Intersección” entre las entidades Parques Naturales y Carreteras.
- d) La función “Caja contenedora” sobre el resultado de aplicar la función de “Unión” entre las entidades Casa y Zonas de caza. Se debe aplicar un filtro adicional que indique que de la entidad Casas sólo se debe tener en cuenta la Casa C.

7. Se deberían almacenar las carreteras, los hoteles y las casas con topología. Almacenando los datos con topología, obtendríamos un grafo conexo sobre el que podríamos aplicar un algoritmo de cálculo de rutas.

Glosario

ArcGIS m Paquete de soluciones SIG de ESRI.

base de datos relacional f Conjunto de datos del mismo contexto almacenados y ordenados en tablas y relaciones entre tablas para que un sistema de información pueda seleccionar rápidamente una porción de éstas.

ESRI *f* Empresa dedicada al desarrollo y la comercialización de herramientas SIG.

fichero CAD *m* Fichero de trabajo y almacenamiento de las herramientas de soporte al diseño usadas principalmente por arquitectos, ingenieros y diseñadores industriales. Se utilizan en cartografía por su precisión.

Geomedia *m* Paquete de soluciones SIG de Intergraph.

GML *m* Formato estándar para la modelización, el transporte y el almacenamiento de información geográfica.

índice espacial *m* Índice para acelerar las consultas sobre datos en un SGBD que se basan en la ubicación sobre el territorio para ordenar los datos.

Intergraph *f* Empresa dedicada al desarrollo y la comercialización de herramientas SIG.

mosaico *m* Conjunto de imágenes que se han creado juntando dos o más imágenes en una sola.

Oracle Spatial *m* Extensión del SGBD Oracle para el almacenamiento de datos geográficos.

pirámide *f* Serie de representaciones de una imagen *raster* con resolución reducida, principalmente usadas para mejorar el rendimiento cuando se trabaja en escalas que no requieren toda la resolución de la imagen.

PostGIS *m* Extensión de código libre sobre el SGBD PostgreSQL, también de código libre, para el almacenamiento de datos geográficos.

raster *m* Malla o matriz regular de celdas que se utiliza para almacenar fotografías o imágenes.

referencia lineal *f* Modo conveniente de asociar atributos o éxitos a localizaciones o tramos de una entidad lineal.

R-tree *m* Índice espacial en forma de árbol que se utiliza para acceder a datos espaciales indexando información multidimensional. Se caracteriza por organizarse según cajas contenedoras con encabalgamiento.

Shapefile *m* Formato propietario abierto para el almacenamiento de datos geográficos desarrollado por ESRI.

Smallworld *m* Paquete de herramientas SIG orientadas a la gestión de redes. También es el nombre de la empresa que comercializa este paquete de herramientas y que desde el año 2007 pertenece al grupo de empresas General Electric.

topología *f* Parte de la matemática que estudia aquellas propiedades de los conjuntos de puntos de la recta, del plano, del espacio o de espacios de dimensiones superiores que no son alteradas por transformaciones continuas.

Bibliografía

Beaujardiere, Jeff de la (2004, gener). "OGC 03-109r1 Web Map Service Interface 1.3.0". Open Geospatial Consortium Inc.

Herring, John R. (2005, octubre). "OpenGIS Implementation for Geographic information - Simple feature access - Part 1: Common architecture". OpenGIS Implementation Specification.

Longley Paul A.; Goodchild Michael, F.; Macguire David, J.; Rhind David, W. (2005). *Geographic Information systems and Science*. Ed. Wiley, ESRI PRESS.

Murray, Ch. (2005, juny). *Oracle Spatial User's Guide and Reference, 10g Release 2 (10.2)*. Oracle.

Murray, Ch. (2005, juny). *Oracle Spatial Topology and Network Data Models, 10g Release 2 (10.2)*. Oracle.

Vretanos, Panagiotis A. (2005, maig). "OGC 04-094 Web Feature Service Implementation Specification 1.1.0". Open Geospatial Consortium Inc.