

UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA



Modelos Avanzados de Base de Datos

FUNCIONALIDAD 3

BASES DE DATOS WEB Y BASES DE DATOS MULTIMEDIA

Juan Andrada Romero
Juan José Antequera Flores
Jose Domingo López López

16 de marzo de 2010

Índice

1. Bases de datos Web	1
1.1. Introducción	1
1.2. Páginas Web estáticas y dinámicas	2
1.3. Integración Web-SGBD	2
1.3.1. Modelo de Intercambio de Objetos (OEM - Object Exchange Model)	4
1.3.2. XML - eXtensible Markup Language	4
1.4. Ventajas y desventajas de la integración Web-SGBD	5
1.5. Técnicas para la integración Web-SGBD	6
1.5.1. CGI - Common Gateway Interface	6
1.5.2. Lenguajes de script	6
1.5.3. ODBC	7
1.5.4. Java	7
1.5.5. ASP.NET	9
2. Bases de datos multimedia	11
2.1. Introducción	11
2.2. Sistemas de gestión de bases de datos multimedia	12
2.3. Operaciones sobre las bases de datos multimedia	13
2.3.1. Almacenamiento de datos multimedia	13
2.3.2. Recuperación de datos multimedia	14
2.4. Ventajas e inconvenientes de una base de datos multimedia	16
3. Comparativa	18
3.1. Similitudes	18
3.2. Diferencias	18
Referencias	19

1. Bases de datos Web

1.1. Introducción

La World Wide Web (Web, de manera abreviada) aparece a finales de los años 90, proporcionando una técnica simple para explorar grandes volúmenes de páginas de información que residen en Internet. La información que puede contener dichas páginas puede ser texto, gráficos, imágenes, sonidos y video, además de los hipervínculos que una página web posee hacia otra página.

Por todo ello, la Web puede definirse como *un sistema basado en tecnología hipermmedia que proporciona un método de explorar información en Internet de forma no secuencial, utilizando hipervínculos*. [6]

La Web está compuesta por una red muy numerosa de computadores, donde unos actúan como *servidores*, para proporcionar la información, y otros como *clientes* o *navegadores Web*, que solicitan a los servidores la información, utilizando un protocolo de comunicación. El protocolo que se utiliza para transferir las páginas Web a través de Internet es **HTTP**, que es un protocolo genérico, orientado a objetos y sin memoria del estado, es decir, que el servidor no almacena ninguna información entre una solicitud y la siguiente. Sin embargo, esta carencia de memoria de HTTP hace que sea difícil soportar el concepto de *sesión*, necesario para poder realizar las operaciones necesarias sobre el SGBD (Sistema Gestor de Bases de Datos).

Por otra parte, para componer y poder publicar un documento en la Web, se utiliza el lenguaje HTML, el cuál define la información que se transmite entre nodos de la red. Dicho lenguaje ha ido evolucionando a lo largo de su historia, utilizándose actualmente la versión HTML 4.01, que tiene soporte para hojas de estilo, scripts y objetos incrustados.

Con todo ello, se puede resumir el funcionamiento de un navegador Web en los siguientes puntos:

1. El usuario, haciendo uso del navegador web (sistema cliente), solicita visualizar una página web.
2. El navegador solicita esta página (que está codificada en lenguaje HTML, el lenguaje que puede interpretar) al servidor web (sistema servidor), utilizando el protocolo HTTP.
3. El servidor web envía la información que le han pedido, en HTML, al navegador del usuario.
4. El navegador muestra la página web en la pantalla del usuario.

Como se puede comprobar, el intercambio de información entre el cliente y el servidor es posible gracias a que, entre otras cosas, las páginas Web utilizan el lenguaje HTML. Sin embargo, tradicionalmente, se han utilizado documentos HTML estáticos, para los cuales se creaban las posibles respuestas ante las peticiones del cliente. Pero en la actualidad surge la posibilidad de utilizar aplicaciones que permitan acceder a información de forma dinámica, con contenidos y formatos muy diversos. Esto ha hecho que sea necesaria la integración de los SGBDs en los servidores Web, para poder recuperar la información de la base de datos según la petición que envíe el cliente.

Así, en la sección 1.2 se van a comentar las diferencias existentes entre una página Web estática y una página Web dinámica, además de comentar brevemente los servicios Web. En la sección 1.3 se van a comentar las características o requisitos que debe tener una base de datos para su uso en la Web, así como sus ventajas y desventajas. Por último, se describirán brevemente algunas de las técnicas y lenguajes de programación que se emplean hoy en día para acceder a las bases de datos desde aplicaciones web.

1.2. Páginas Web estáticas y dinámicas

Una página web estática se puede definir como un documento HTML que contiene la información sobre la página y donde dicha información no cambia, a no ser que se cambie el propio documento. Por esta razón, no es necesario acceder a una base de datos para recuperar la información, ya que se ésta se encuentra contenida dentro del propio documento que se va a enviar al cliente.

Sin embargo, la información de una página Web dinámica se genera cada vez que se accede a dicha página, realizando, por ejemplo, consultas a una base de datos. De este modo, las páginas Web dinámicas tienen características que no tienen las páginas Web estáticas:

- Los datos que se muestran en la página Web son personalizados para cada usuario.
- Se puede responder a los datos de entrada que el usuario introduce a través del navegador.

En la actualidad, prácticamente todas las páginas que existen son dinámicas, pero existe otro paradigma que está cobrando una gran importancia a la hora de construir y utilizar aplicaciones Web: **Servicios Web**. Un servicio Web puede definirse como *un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web, intercambiando datos entre sí con el objetivo de ofrecer unos servicios*. [5] En la Figura 1.1 se puede observar un ejemplo de servicios Web.

Las tecnologías estándares que utiliza un servicio Web para esta interoperabilidad son:

- Lenguaje XML (eXtensible Markup Language).
- Protocolo SOAP (Simple Object Access Protocol), para la comunicación en Internet.
- Lenguaje WSDL (Web Services Description Language), para describir el servicio Web.
- Protocolo UDDI (Universal Discovery, Description and Integration), que se utiliza para registrar el servicio y que pueda ser utilizado.

En el caso de los servicios Web, como al fin y al cabo son aplicaciones Web, es también necesario el uso de base de datos, para poder acceder a la información que necesita la aplicación para su funcionamiento.

1.3. Integración Web-SGBD

En la Web, no se puede hablar de un tipo de SGBD específico, como puede ocurrir en otros tipos de bases de datos, como pueden ser las orientadas a objetos. Si bien es cierto que existen

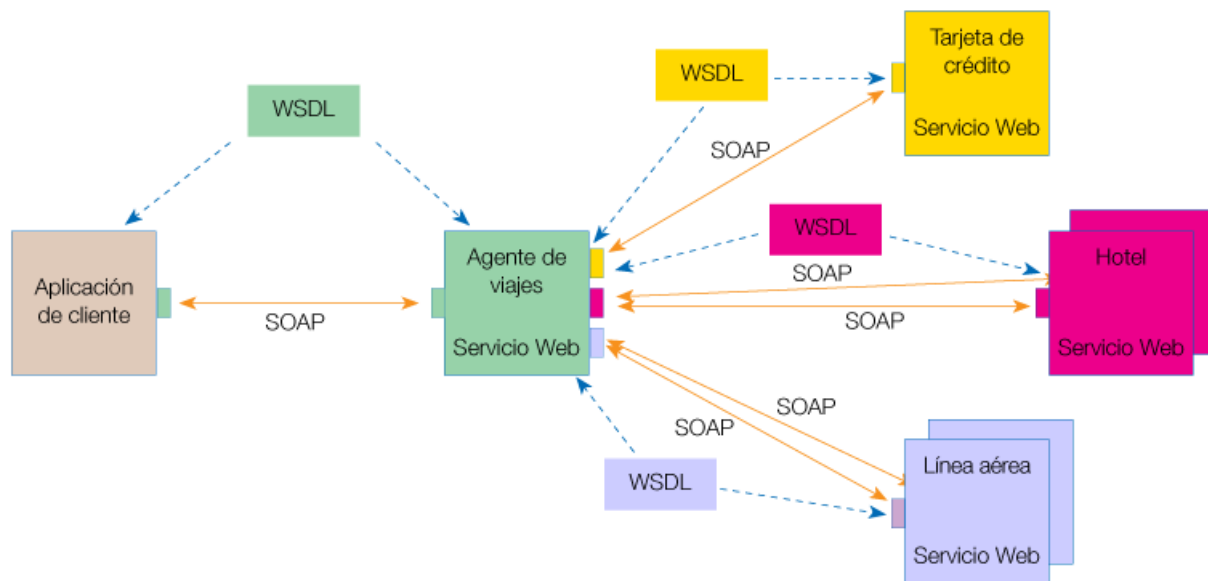


Figura 1.1: Interacción entre Servicios Web

fabricantes que tratan de proporcionar soluciones propietarias de conectividad, la mayoría de empresas prefieren utilizar un SGBD más general, para no atarse a una tecnología concreta. Así, las bases de datos Web no son más que SGBD utilizados y orientados a la Web, de tal modo que se utiliza el SGBD que más se adapte a la naturaleza de la información que debe manejar la aplicación Web correspondiente.

Los requisitos que debe cumplir un SGBD para poder integrarse en una aplicación Web, son los siguientes:

- Incorporar algún tipo de *middleware* (capa de software que se sitúa sobre el SGBD) para añadir conectividad a la base de datos a través de Internet. Dicha conectividad debe ser independiente de los datos y del fabricante.
- Soporte para transacciones que abarquen múltiples solicitudes HTTP (es decir, acceso concurrente a los datos).
- Soporte para mecanismos de autenticación basados en la sesión y en la aplicación.
- Arquitectura abierta que permita la interoperabilidad con diversos sistemas y tecnologías (servicios Web, RMI, CORBA, XML, etc.)

No obstante, existe un tipo especial de SGBD que se utiliza para datos semi-estructurados, que son datos que pueden ser irregulares o incompletos, cuya estructura puede cambiar de forma rápida o impredecible y son débilmente tipados. [4].

Por este motivo, un SGBD como puede ser un SGBD relacional u orientado a objetos pueden tener problemas a la hora de manejar datos de este tipo, ya que dichos SGBD almacenan los datos siguiendo un esquema fijo, como puede ser una tabla, en el modelo relacional.

Por tanto, para tratar con este tipo de datos, que abunda en la Web, se utilizan los siguiente modelos y SGBD específicos:

1.3.1. Modelo de Intercambio de Objetos (OEM - Object Exchange Model)

OEM ([3]) es un modelo de objetos anidados que fue diseñado para su uso en diferentes proyectos del *Grupo de Bases de Datos de la Universidad de Stanford*, para soportar la integración de datos de diferentes orígenes de datos. Los datos OEM son carentes de esquema y pueden considerarse como un grafo dirigido etiquetado (ver Figura 1.2), donde los nodos son objetos. A su vez, cada objeto está compuesto por un identificador, una etiqueta descriptiva, un tipo de datos y un valor. En cuanto a los tipos de datos, éstos pueden ser simples (*string*, *int*, etc.) y no tienen conexiones con otros nodos del grafo, o pueden ser otros objetos, que se conectan con otros nodos del grafo.

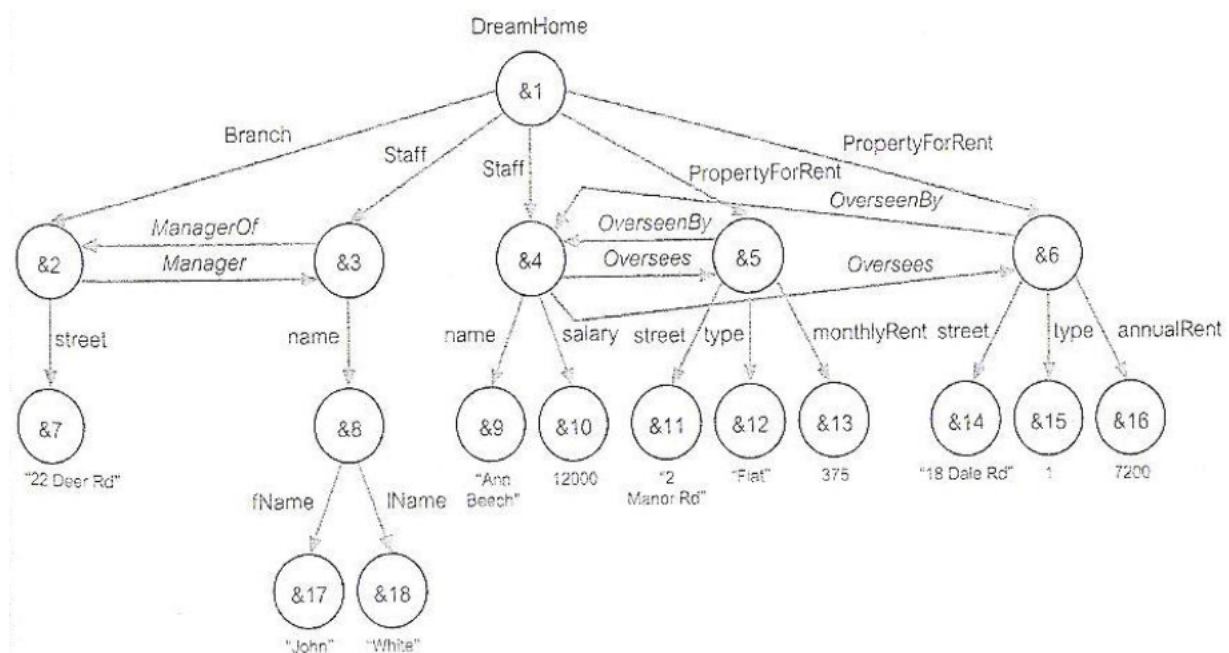


Figura 1.2: Ejemplo de datos en el modelo OEM

1.3.2. XML - eXtensible Markup Language

XML ([7]) es un metalenguaje definido por el World Wide Web Consortium (W3C) que permite la definición de etiquetas personalizadas para proporcionar funcionalidad no disponible en HTML. En XML existen dos tipos de modelos: el modelo centrado en los datos y el modelo centrado en los documentos.

Si se utiliza como modelo de documentos, XML se utiliza como formato de almacenamiento e intercambio para datos que estructurados. En este caso, los datos podrían almacenarse en un SGBD relacional, transformando el documento XML en el esquema relacional. Esto puede hacer con una extensión del SQL, llamada SQL/XML.

En el caso de que XML se use para codificar datos semi-estructurados, se usa una base de datos XML nativa (*NXD*, *Native XML Database*), las cuáles definen un modelo de datos para un

documento XML, almacenando y extrayendo documentos de acuerdo con dicho modelo. Ejemplos de estos SGBD son Lore, Infonyte DB o Tamino XML Server. Además, se pueden distinguir dos tipos de NXD, según como almacenen la información:

- **Basados en texto:** almacenan el código XML como texto, como un archivo de un sistema de archivos o como un dato de tipo CLOB en un SGBD relacional.
- **Basados en modelo:** almacenan el código XML en alguna representación interna en forma de árbol.

Por otra parte, estas bases de datos XML utilizan un lenguaje de consulta de datos llamado *XQuery*. Un ejemplo de consulta para obtener una lista de identificadores de empleados de un documento XML, es el siguiente:

```
FOR $S IN doc(\staff_list.xml")
WHERE $S/SALARY >15000 AND $S/@branchNo="B005"
RETURN $S/STAFFNO
```

1.4. Ventajas y desventajas de la integración Web-SGBD

En lo que se refiere a las ventajas y desventajas de utilizar un SGBD en una aplicación Web, podemos señalar las siguientes:

- **Ventajas**

- Permite proporcionar a una página HTML de toda la funcionalidad y de todos los mecanismos de protección que proporciona el SGBD.
- Independencia entre el SGBD y la aplicación Web que lo utilice.
- Poder acceder a la base de datos desde cualquier parte del mundo, utilizando la Web.
- Implantación escalable, ya que se pueden distribuir diferentes bases de datos.
- Acceso distribuido a los datos, lo que puede aumentar la velocidad de acceso a éstos cuando se solicita una página Web al servidor.

- **Desventajas**

- Fiabilidad, ya Internet es un medio de comunicación poco fiable.
- Seguridad en el acceso a los datos.
- Coste que supone mantener los servicios Web, los servidores y las bases de datos.
- Carencia de memoria del protocolo HTTP, lo que obliga a mantener información adicional en el servidor para manejar el acceso a la base de datos.
- Dificultad para manejar grandes volúmenes de datos con la base de datos, debido a que el ancho de banda de la red es limitado, lo que puede aumentar la latencia de la aplicación Web.

1.5. Técnicas para la integración Web-SGBD

En las siguientes subsecciones se comentarán algunas de las técnicas y tecnologías más utilizadas para la integración de bases de datos en entornos web.

1.5.1. CGI - Common Gateway Interface

La interfaz de pasarela común es un protocolo estándar que define cómo los *scripts CGI* se comunican con el servidor Web. Un script CGI está diseñado para aceptar y devolver datos que sean conformes con la especificación de CGI, pudiendo así reutilizar los scripts independientemente del servidor Web que se esté utilizando.

A continuación se describe la forma de actuación de un CGI de forma esquemática:

1. En primera instancia, el servidor recibe una petición (el cliente ha activado un URL que contiene el CGI), y comprueba si se trata de una invocación de un CGI.
2. El servidor comprueba los archivos de configuración y de acceso para asegurarse de que el solicitante tiene acceso al script CGI.
3. El servidor prepara el entorno para ejecutar el script.
4. El servidor ejecuta el script, leyendo las variables de entorno y su entrada estándar.
5. El script realiza su función. Como consecuencia de su actividad se va generando un objeto MIME que la aplicación escribe en su salida estándar.
6. Cuando el script finaliza, el servidor envía la información producida, junto con información propia, al cliente, que se encontraba en estado de espera.

Actualmente, ésta es la solución que más se está utilizando para la creación de interfaces Web-SGBD. Entre las ventajas de la programación CGI, se pueden destacar su sencillez, ya que es muy fácil de entender; es un lenguaje de programación independiente, soportando lenguajes como C, C++, Java, Perl, Python, etc. y es un estándar para usarse en todos los servidores Web.

Como desventaja se puede destacar la baja eficiencia que tiene el uso de CGI, debido a que el servidor Web tiene que cargar el script CGI y establecer la conexión con la base de datos cada vez que se recibe una petición.

1.5.2. Lenguajes de script

Un *script* es un código escrito en un lenguaje de programación que se incrusta en el código HTML de una página Web y que se puede ejecutar en el lado del servidor o del cliente, añadiendo nuevas funcionalidades a la página Web. De esta forma, se pueden automatizar funciones para acceder a los datos de la base de datos, creando nuevo código HTML y mostrando así las páginas Web personalizadas con los datos que pide el cliente.

Algunos de los lenguajes de script más utilizados actualmente se comentan a continuación.

JavaScript

JavaScript es un lenguaje de scripts basado en Java, orientado a objetos, interpretado, débilmente tipado e incrustado en código HTML. Permite incluir en las páginas HTML funciones que respondan a los sucesos del usuario, como clics de ratón, eventos de teclado, etc., o acceso a la base de datos si se ejecuta en el lado del servidor.

VBScript

VBScript es otro lenguaje de scripts procedimental (basado en subrutinas), muy similar a JScript en cuanto a funcionalidad se refiere, pero con la sintaxis de Visual Basic.

Perl y PHP

Perl (Practical Extraction and Report Language) es un lenguaje de programación de alto nivel interpretado que combina características de C y de diferentes órdenes de sistemas UNIX. Hoy en día es uno de los lenguajes de script más utilizados para su ejecución en el lado del servidor.

Por otra parte, PHP (Hypertext Preprocessor) es otro lenguaje de script de código abierto que se incrusta en HTML y que es soportado por multitud de servidores Web. Uno de los mayores usos que tiene es el acceso a bases de datos en el lado del servidor, utilizando como SGBD *MySQL* y como servidor, *HTTP Apache*.

1.5.3. ODBC

ODBC (Open Database Connectivity) proporciona una interfaz común, desarrollada por Microsoft, para acceder a bases de datos SQL. Gracias a ella, el acceso de las aplicaciones a las bases de datos se hace de un modo totalmente transparente. ODBC actúa como una capa intermedia, por lo que es necesario configurar el origen de datos cargando el driver que necesite para interactuar con el SGDB en cuestión.

1.5.4. Java

Java es un lenguaje de programación, orientado a objetos e interpretado desarrollado por Sun Microsystems, que se ha convertido en un estándar *de facto* para la programación basada en la Web, permitiendo crear tanto aplicaciones Web (**applets**), como aplicaciones de servidor (**servlets**).

Además, Java es un lenguaje independiente de la plataforma, ya que es interpretado en su máquina virtual (JVM - Java Virtual Machine). Otras características de este lenguaje es que

posee un recolector de basura, mecanismos de seguridad con comprobación de tipos y la ausencia de punteros.

En 1999, Sun crea **J2EE**, una arquitectura pensada para aplicaciones empresariales robustas, escalables, multiusuario y seguras. Esta plataforma incluye todo lo necesario para el desarrollo de aplicaciones Web, aplicaciones de servidor, presentación en el cliente y las comunicaciones necesarias para comunicar el cliente, el servidor y la bases de datos (ver Figura 1.3).

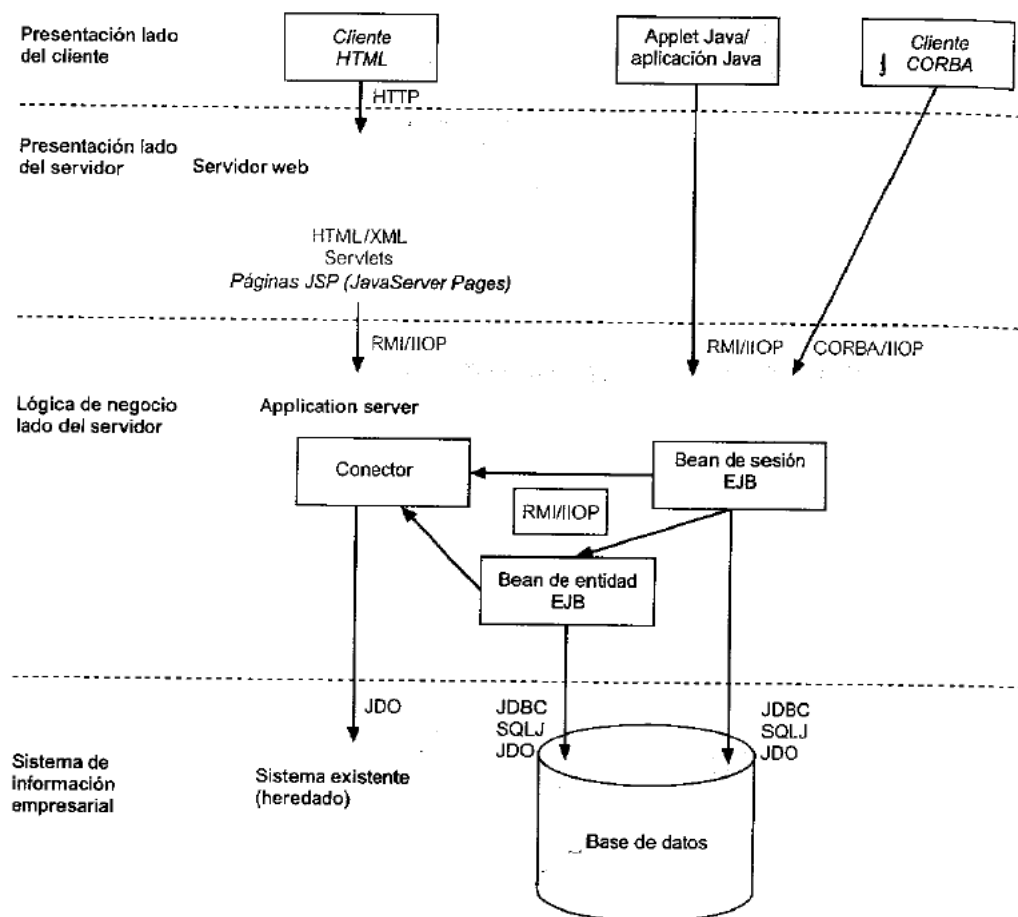


Figura 1.3: Arquitectura J2EE simplificada

JSP

JSP (Java Server Pages) es un lenguaje de *script* basado en Java que se ejecuta en el lado del servidor. Permite mezclar HTML estático con HTML generado dinámicamente de un modo similar a PHP.

Cuando un cliente hace una petición de una página JSP, el servidor procesa todo el código Java que hay en su interior para generar un servlet y devuelve al cliente un flujo de código HTML. Por esto, es necesario que el servidor Web sea capaz de compilar y ejecutar código Java. Algunos

servidores de los que se dispone son Apache Tomcat, JBoss y WebSphere (un plug-in de IIS - Microsoft Internet Information Server).

JDBC

JDBC es la técnica mas utilizada y madura para el acceso a un SGBD relacional desde Java. JDBC define una API para acceder a diferentes SGBD relacionales, soportando toda la funcionalidad SQL básica. Además, por encima del JDBC se construyen interfaces de aplicación a mayor nivel como, por ejemplo, SQL embebido en Java o una correspondencia directa de tablas de una base de datos relacional a clases Java (Hibernate).

Para el acceso a la base de datos por parte de las aplicaciones, hay diferentes opciones:

- Usar el puente JDBC-ODBC, donde el controlador ODBC actúa como nivel mediador entre el SGBD y el controlador JDBC. Es necesario cuando no se puede acceder directamente a las bibliotecas del fabricante del SGBD.
- Usar el controlador JDBC parcial, que convierte llamadas JDBC directamente a llamadas del API del SGBD.
- Usar JDBC puro para middleware de bases de datos, que traduce las llamadas del controlador JDBC al protocolo del middleware, el cual es traducido de nuevo al protocolo del SGBD. De este modo, existe más variedad a la hora de usar diferentes SGBD.
- Usar JDBC puro con conexión directa al SGBD. Esto conlleva utilizar un controlador diferente, según el SGBD que se utilice.

Sea cual sea la opción de comunicación elegida, el proceso de interacción con la base de datos es el siguiente:

1. Registrarse en el controlador JDBC, utilizando el método `Class.forName()`.
2. Abrir la conexión con el SGBD, utilizando el método `DriverManager.getConnection()`, indicando la URL del SGBD, que tiene el siguiente formato: `protocolo:subprotocolo:subnombre`. Por ejemplo: `jdbc:odbc:www.dreamhome.es.co.uk/dhdatabase`.
3. Ejecutar una sentencia SQL sobre la base de datos, con los métodos `createStatement` y `execute()`.
4. Procesar los resultados obtenidos (si la sentencia era una consulta).
5. Cerrar la conexión, con el método `close()`.

1.5.5. ASP.NET

ASP.NET es un modelo de programación que permite crear páginas web interactivas y dinámicas de un modo similar a JSP y PHP. Es la versión avanzada de ASP (Active Server Pages) que ha sido rediseñada para mejorar sus prestaciones y la escalabilidad. Esto es debido a la incorporación

de la biblioteca de clases de .NET, que es una colección de clases reutilizables, interfaces y tipos que se integran dentro de un entorno común para proporcionar funcionalidades como la gestión de cadenas de caracteres, las comunicaciones por red, la gestión de hilos, etc.

Además, con ASP.NET se dispone de ADO.NET que es un conjunto de clases que exponen los servicios de acceso a datos para que puedan ser utilizados por el programador de .NET.

2. Bases de datos multimedia

2.1. Introducción

El término **multimedia** se aplica a *"algo que utiliza conjunta y simultáneamente diversos medios, como imágenes, sonidos y texto, en la transmisión de una información"* [2]. Por tanto, un **dato multimedia** es cualquier objeto que cumpla con la definición anterior. De esta definición también podemos extraer los tipos de datos multimedia e inferir acerca de sus características:

- **Texto:** estructurado o no estructurado, hipermedia.
- **Gráficos:** dibujos e ilustraciones codificados según un estándar de descripción de archivos (CGM, PICT, PostScript, etc.).
- **Imágenes:** elementos gráficos codificados según algún formato estándar (mapa de bits, jpeg, png, tiff, etc).
- **Animaciones:** secuencias temporales de imágenes o datos (gif, swf, etc).
- **Video:** secuencia temporal de imágenes a una determinada velocidad (fps).
- **Audio Estructurado:** secuencia de sonidos identificados según nota, tono, duración, etc.
- **Audio Digital:** secuencia de sonidos digitales.
- **Datos multimedia compuestos o mezclados:** datos multimedia agregados junto con información sobre los datos (metadatos).

Tradicionalmente, las bases de datos se han utilizado para almacenar datos de tipo textual y numérico, pero con la aparición de los datos multimedia se hacen necesarias nuevas bases de datos que puedan almacenar los datos multimedia que se han descrito. De este modo, los sistemas gestores de bases de datos multimedia tienen mucha relación con sistemas de bases de datos objeto-relacionales y los orientados a objetos, ya que pueden almacenar datos multimedia en los siguientes tipos de datos:

- **BLOB:** (Binary Large Objects): elemento utilizado en las bases de datos para almacenar datos de gran tamaño que cambian de forma dinámica, este es un objeto binario que puede tratar una cantidad de datos variables.
- **CLOB:** (Character Large Object) Una columna también puede ser del tipo CLOB, que es capaz de almacenar grandes cantidades de datos en formato carácter. Usualmente tiene límites con respecto a la data que guarda, en el orden de los 2 GB. Se diferencia del BLOB prácticamente en la forma de encriptado que se usa. En ciertos manejadores, permite los comandos de SQL: LIKE y SUBSTRING al igual que métodos alternativos para acceder a la data.
- **NCLOB:** (National Character Large Object) Un tipo de datos NCLOB de Oracle contiene datos de caracteres basados en el conjunto de caracteres nacional del servidor. Su tamaño máximo es de 4 gigabytes.

Por todo lo anterior, en el momento en el que se quiere diseñar y utilizar una base de datos multimedia, el objetivo es poder almacenar grandes volúmenes de información y de diferente naturaleza como son textos, sonidos, imágenes, videos, etc. permitiendo que estos volúmenes de datos

estén distribuidos en diferentes plataformas y localizaciones geográficas. Por esta última razón, las bases de datos multimedia tienen dos características principales: un entorno de fácil utilización por parte del usuario y el establecimiento de redes de telecomunicaciones de alta velocidad.

Para terminar, podemos distinguir entre dos tipos de bases de datos multimedia:

- **Bases de datos referenciales:** son bases de datos que pueden almacenar información como películas, imágenes, música, etc. En la mayoría de los casos, la información que se almacena hace referencia a cuestiones descriptivas (autor, título, duración, productor, etc.) o a cuestiones técnicas (formato, duración, etc.).
- **Bases de datos descriptivas:** son bases de datos que además de mantener las cuestiones descriptivas o técnicas, como las bases de datos anteriores, aportan información específica sobre el contenido indicando, por ejemplo, dónde se sitúan los cambios de plano dentro de una película, etc.

2.2. Sistemas de gestión de bases de datos multimedia

Los sistemas de gestión de bases de datos multimedia, también llamados *gestores de información hipertexto*, se basan en un SGBD objeto-relacional, como ya se ha comentado en la sección anterior. Sin embargo, un SGBD multimedia cuenta con los siguientes componentes:

1. **Base de presentación:** parámetros a aplicar para mostrar información al usuario.
2. **Base de estructura:** visión lógica del hipertexto, según un modelo.
3. **Base de contenido:** conjunto de documentos que se integran en el hipertexto.
4. **Base de utilización:** información sobre hábitos y comportamiento de cada usuario.

Por otra parte, un SGBD multimedia debe cumplir con los siguientes requisitos:

- Encontrar rápidamente la información multimedia.
- Conectar documentos según su contenido.
- Facilitar el acceso instantáneo a la información multimedia, incluyendo los mecanismos necesarios de compresión y descompresión para su manipulación.
- Crear relaciones entre los distintos tipos de datos multimedia.
- Búsqueda por texto completo.
- Administración de sinónimos.
- Búsqueda de similares.

2.3. Operaciones sobre las bases de datos multimedia

Debido a la naturaleza tan variada de los datos que debe manejar este tipo de bases de datos, se hace necesario encontrar un modelo que permita describir la estructura de la diferente información multimedia. Es decir, según la naturaleza del dato multimedia, sería más adecuado utilizar uno u otro modelo, esto es, uno u otro SGBD.

Por ejemplo, para almacenar texto, se puede utilizar un SGBD relacional, mientras que para almacenar objetos de video o imágenes, una mejor opción sería un SGBD orientado a objetos.

A continuación, se comentará el modo de almacenar y recuperar los tipos de datos multimedia más extendidos, como son las imágenes, vídeo y texto.

2.3.1. Almacenamiento de datos multimedia

Indexación de texto

Para almacenar un objeto multimedia de tipo texto, se emplea un SGBD relacional para almacenar, como atributos, los diferentes términos que se pueden extraer del texto del documento. Además, si se almacena el documento completo, éste se puede guardar en una base de datos orientada a objeto.

Para obtener los términos que se almacenan en la base de datos, el texto se somete a un proceso de eliminación de ruido, para sólo almacenar aquellos términos con significado y que sean relevantes. Para esto, se aplican técnicas de *stemming* (agrupación de términos según su raíz), eliminación de palabras sin sentido, como artículos o preposiciones (llamadas *stopwords*), etc.

Indexación de imágenes

A la hora de indexar la información de una imagen, se puede optar por técnicas de procesamiento automático de imágenes o por la asignación manual de características. La primera de estas técnicas facilita la indexación de grandes colecciones de imágenes, ya que se asignan descriptores (color, brillo, forma, etc.) a las imágenes de manera automática.

Existen varias implementaciones a la hora de extraer los descriptores de las imágenes. Generalmente consiste en crear una representación del descriptor en forma de cadena de caracteres.

Sin embargo, esta técnica es menos fiable, ya que pueden existir características que no puedan ser identificadas de manera automática. En este caso, la asignación manual puede resolver esta carencia, ya que los descriptores se asignan según el conocimiento de la persona que procesa la imagen. Sin embargo, este método resulta inviable a la hora de procesar grandes colecciones de imágenes.

Para almacenar este tipo de dato multimedia, se suele utilizar una base de datos objeto-relacional, debido al tamaño de los descriptores, que suelen almacenarse como un tipo *BLOB*, como se puede ver en la Figura 2.1

id	nombre	cedd	fcth
38	1165_escocia-noche.jpg	BLOB	BLOB
39	15_19_1--Tree-Sunrise-Northumberland_web.jpg	BLOB	BLOB
40	20061023230254-0083-noche-mar-ftld.jpg	BLOB	BLOB
41	400_1192928691_limon.jpg	BLOB	BLOB
42	image_gallery.jpg	BLOB	BLOB
43	lechuga.jpg	BLOB	BLOB
44	limon.jpg	BLOB	BLOB
45	MANZANA VERDE.jpg	BLOB	BLOB

52 rows fetched in 0.0193s (0.0122s)

Edit Apply Changes Discard Changes First Last Search

Figura 2.1: Ejemplo de base de datos con descriptores de imágenes

Indexación de video

Para indexar un video, en primer lugar éste se divide en **shots**, que pueden definirse como *un conjunto de frames grabados secuencialmente que representan una acción continua en el tiempo y espacio y que ha sido tomada por una sola cámara* ([1]). Una vez tomados estos *shots*, se obtienen los *key frames* (ver figura 2.2) de cada uno de ellos y, a partir de ellos, como los frames ya son imágenes, se aplican las técnicas de extracción de descriptores comentados en el apartado anterior.

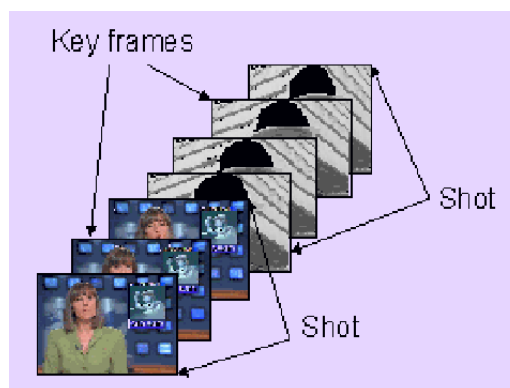


Figura 2.2: Ejemplo de shots y key frames de un video

Cabe destacar que para este caso, se utiliza un SGBD objeto-relacional para los descriptores, y un SGBD orientado a objetos donde se almacena el archivo de video en sí y los diferentes *key frames*.

2.3.2. Recuperación de datos multimedia

La recuperación de información multimedia de la bases de datos es una cuestión importante que hay que resolver de forma eficiente. En esta sección se explican algunas técnicas de recuperación de datos multimedia basándose en el contenido de éstos.

Recuperación de texto

Para la recuperación de texto existen numerosas aproximaciones. Las más famosas son el modelo booleano, el modelo estadístico y el modelo vectorial, siendo este último -con algunas mejoras- el más utilizado.

Cuando el usuario introduce una cadena de texto -denominada vector pregunta- para realizar una búsqueda, esta cadena se someterá al mismo preprocesamiento de texto al que se sometieron los textos que forman parte de la base documental en el momento de la indexación [2.3.1](#), con el objetivo de eliminar ruido y buscar únicamente términos relevantes.

En el modelo vectorial se utilizan vectores multidimensionales. La pregunta y cada uno de los documentos de la base documental se representan mediante un vector, en el que cada término define una dimensión. Para calcular la similitud entre dos vectores (el vector pregunta y un vector documento) se calcula el ángulo que forman ambos vectores. Si ambos vectores son perpendiculares, el grado de similitud es nulo; si por el contrario son paralelos, significa que son muy similares.

No obstante, cuando se trabaja con texto existen problemas relacionados con la ambigüedad del lenguaje, el contexto, las faltas de ortografía, etc.

Recuperación de imágenes

La recuperación de imágenes consiste en extraer imágenes relevantes para los usuarios de una gran colección de datos. La relevancia está determinada por la naturaleza de la aplicación. Por ejemplo, en una base de datos de imágenes de tejidos, las imágenes relevantes serían aquellas que sean similares en términos de textura y color. Muchos SGBD relacionales soportan campos para BLOBs (Binary Large Objects) y facilitan el acceso a atributos definidos por el usuario (fecha, hora, tipo de dato, resolución de imagen, etc.). En definitiva, estos sistemas analizan el contenido visual de las imágenes e indexan los descriptores que inferen.

A continuación se muestran algunos tipos de consultas para la recuperación de imágenes:

- *Consulta de descriptores*: el usuario indica los valores de determinados descriptores como el brillo, contraste, forma, color, textura, etc.
- *Consulta de combinación de descriptores*: el usuario combina diferentes descriptores e indica sus valores y pesos.
- *Consulta de descriptores específicos*: el usuario especifica los valores y localizaciones de determinados descriptores que deben darse en la imagen.
- *Consulta por similitud*: la similitud puede ser determinada basándose en una serie de descriptores seleccionadas por el usuario. En este tipo de consultas el sistema genera un conjunto aleatorio de imágenes, el usuario selecciona una y el sistema devuelve imágenes similares (ver figura [2.3](#)).
- *Comparación de objetos e imágenes*: el usuario puede describir los descriptores de un objeto que debe aparecer en una imagen.
- *Consulta de relación de objetos*: el usuario especifica objetos, sus atributos y relaciones entre ellos.

- *Consultas de conceptos*: algunos sistemas permiten al usuario que definan conceptos sencillos basándose en los valores y localizaciones de descriptores con los que trabaja el sistema.



Figura 2.3: Ejemplos de búsqueda por similitud

Recuperación de video

Como se ha visto en la sección 2.3.1, cuando se trabaja con video se manejan los conceptos *shot* y *key frame*.

Cuando un sistema indexa un video realiza un proceso de extracción de descriptores. Los descriptores que se pueden extraer de un video son texto, iconos e imágenes y serán indexadas del mismo modo que si fuesen procesos separados. Por tanto, cuando se trata de recuperar un video se utiliza una combinación de las técnicas de recuperación de textos 2.3.2 y recuperación de imágenes 2.3.2.

2.4. Ventajas e inconvenientes de una base de datos multimedia

• Ventajas

- La posibilidad de integrar en un único sistema una gran diversidad de formatos (imágenes, texto, video, sonido, etc).
- Ofrecen mayor variedad a la hora de representar la información.
- Un gran, y creciente, mercado potencial que augura que se siga investigando activamente en el futuro.

• Desventajas

- Necesita grandes espacios para almacenar toda la información que queremos.

- Este tipo de bases de datos necesitan grandes anchos de banda para obtener un rendimiento óptimo.
- Complejidad en cuanto a programar operaciones, o incluso la interfaz, debido a la alta cantidad de formatos que hay que manejar, lo que puede repercutir en su rendimiento.

3. Comparativa

3.1. Similitudes

- Ambas bases de datos deben soportar grandes volúmenes de datos.
- Aunque no es estrictamente necesario que las bases de datos multimedia sean concurrentes, es aconsejable, y para las bases de datos web es obligatorio, ya que debe dar servicio a muchos usuarios al mismo tiempo.
- Una base de datos multimedia puede utilizarse como base de datos web siempre y cuando disponga de la capa de comunicación entre el SGBD y el usuario que está utilizando la aplicación.
- Los dos tipos de bases de datos pueden utilizar sistemas gestores de bases de datos tanto relacionales, como objeto-relacionales, etc. . .

3.2. Diferencias

- Web
 - Las bases de datos web deben disponer obligatoriamente de sistemas de seguridad.
 - Deben estar orientadas a sesión.
 - Las bases de datos web pueden estar basadas en XML, y para ello utilizan sistemas gestores de bases de datos nativos de XML (NXD, que utiliza el lenguaje XQuery).
- Multimedia:
 - Mientras que en las bases de datos web debemos conocer exactamente como es el objeto que estamos intentando recuperar, en las multimedia no ocurre lo mismo, ya que las búsquedas pueden estar basadas en ejemplo y en las web recuperamos un objeto en concreto (conociendo de antemano sus características).
 - De lo anterior puede derivarse que el uso de bases de datos multimedia aporta más funcionalidad al poder buscar por descriptores, similitud, metadatos, etc.

Referencias

- [1] Oscar Díaz Mario Piattini. *Advanced Database Technology and Design*. ‘Artech House.
- [2] Definición de Multimedia. http://buscon.rae.es/draeI/SrvltGUIBusUsual?TIPO_HTML=2&TIPO_BUS=3&LEMA=multimedia.
- [3] Roy Goldman, Sudarshan Chawathe, Arturo Crespo, Jason McHugh. A Standard Textual Interchange Format for the Object Exchange Model (OEM).
- [4] Modelo semi-estructurado y XML. <http://www.ldc.usb.ve/~ruckhaus/materias/ci7453/clase3.pdf>.
- [5] Guía Breve de Servicios Web. <http://www.w3c.es/divulgacion/guiasbreves/ServiciosWeb>.
- [6] Carolyn E. Begg Thomas M. Connolly. *Sistemas de bases de datos. Un enfoque práctico para diseño, implementación y gestión*. ‘Pearson. Addison Wesley.
- [7] Extensible Markup Language (XML). <http://www.w3.org/XML/>.