

TEMA1. GESTIÓN DE DATOS Y ARCHIVOS

Bases de Datos
CFGS DAW

Raquel Torres
raquel.torres@ceedcv.es

Versión:170922.1330

Licencia



Reconocimiento - NoComercial - CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:



Importante



Atención



Interesante

ÍNDICE DE CONTENIDO


1. ¿Por qué gestionar la información?	4
2. Ficheros	6
2.1 Tipos de ficheros y formatos	6
2.2 Ficheros de texto	7
2.3 Ficheros binarios	8
3. Bases de datos	8
3.1 Conceptos	9
3.2 Estructura de una base de datos	11
3.3 Usos de las bases de datos	11
3.4 Evolución y tipos de base de datos	12
4. Sistemas Gestores de Base de Datos	14
4.1 Concepto de Sistema Gestor de Base de Datos	14
4.2 Funciones de un SGBD	14
4.3 El lenguaje SQL	15
4.4 Tipos de SGBD	16
5. Arquitectura en Niveles de una base de datos	17
6. Bibliografía	18

UD01. GESTIÓN DE DATOS Y ARCHIVOS

1. ¿POR QUÉ GESTIONAR LA INFORMACIÓN?

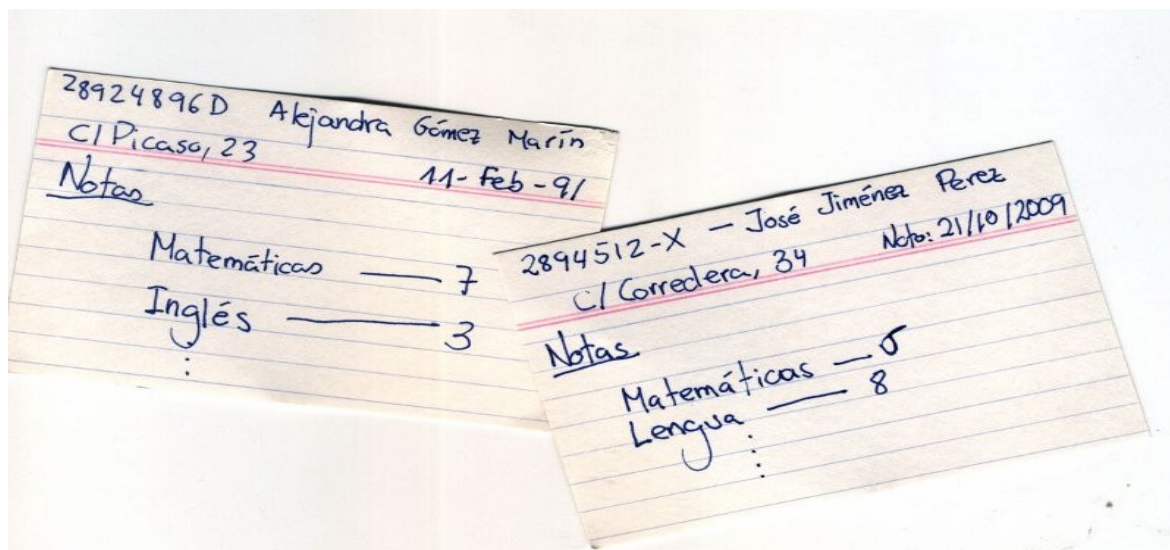
En el mundo actual existe cada vez una **mayor demanda de datos**. Esta demanda siempre ha sido patente en empresas y sociedades, pero en estos últimos años todavía se ha disparado más debido al acceso multitudinario a Internet.

El propio nombre **Informática** (**información + automática**) hace referencia al hecho de ser una ciencia que tiene como objetivo **automatizar el trabajo con información**.

 En informática **se conoce como dato a cualquier elemento informativo que tenga relevancia para el sistema.**

Para poder almacenar datos y cada vez más datos, el ser humano ideó nuevas herramientas como archivos, cajones, carpetas y fichas en las que almacenarlos.

Antes de la aparición del ordenador, el tiempo requerido para manipular esos datos era enorme. Sin embargo el proceso de aprendizaje era relativamente sencillo ya que se usaban elementos que el usuario reconocía perfectamente.



Por esa razón, la informática adaptó sus herramientas para que los elementos que el usuario maneja en el ordenador se parezcan a los que utilizaba

manualmente. Así en informática se sigue hablado de ficheros, formularios, carpetas, directorios,....etc.

En los años 70-80 se utilizaban ficheros de texto donde se guardaba la información:

alumnos.txt			
DNI	NOMBRE	DIRECCIÓN	FECHA NTO
-----	-----	-----	-----
2894512X	José Jiménez Perez	C/ Corredera,34	21-10-90
28924896D	Alejandra Gómez Marín	C/ Picasso, 23	11-02-91
asignaturas.txt			
DNI	NOMBRE	ASIGNATURA	NOTA
-----	-----	-----	-----
----	-----	-----	-----
2894512X	José Jiménez Perez	Matemáticas	5
2894512X	José Jiménez Perez	Lengua	8
.....	-----	-----	-----
28924896D	Alejandra Gómez Marín	Matemáticas	7
28924896D	Alejandra Gómez Marín	Inglés	3

A partir de los años 90 empiezan a utilizarse los Sistemas Gestores de Bases de Datos (**SGBD**) donde se comienza a almacenar la información en forma de tablas:

Alumnos (DNI, Nombre, Dirección, Fecha nacimiento)			
2894512X	José Jiménez Perez	C/ Corredera,34	21-10-90
28924896D	Alejandra Gómez Marín	C/ Picasso, 23	11-02-91
...			
Asignaturas (Código, Nombre)			
001	Matemáticas		
002	Lengua		
003	Inglés		
Notas(DNI, Código_asignatura, nota)			
2894512X	001	5	
2894512X	002	8	
28924896D	001	7	
28924896D	003	3	

Veamos por qué se ha producido esta evolución. ¿Qué ocurriría si quisiéramos conocer el número de alumnos de más de veinticinco años y con nota media superior a siete que además estén matriculados actualmente en la asignatura Bases de datos I?

En un Sistema de Información (SI) sin informatizar, obtener esta información podía requerir mucho tiempo y mucho trabajo, además de ser necesario la realización de cálculos (media,...) así como ir revisando alumno por alumno.

En un SI con ficheros, podríamos crear un programa que fuera obteniendo la información del fichero y realizando los cálculos y devolviendo los resultados.

Sin embargo, en un SI con bases de datos, esta consulta es trivial usando un lenguaje de consulta de datos.

2. FICHEROS

Un ordenador almacena muchos tipos de información, desde datos administrativos, contables o bancarios hasta música, películas, partidas de videojuegos, páginas webs, etc. Toda esta información está almacenada en los dispositivos de almacenamiento del ordenador, esto es, discos duros, DVDs, pen drives, etc.

Para poder organizar la información en estos dispositivos, se utilizan los ficheros o archivos.



Los ficheros son estructuras de información que crean los sistemas operativos de los ordenadores para poder almacenar datos. Suelen tener un nombre y una extensión, que determina el formato de la información que contiene.

2.1 Tipos de ficheros y formatos

El formato y tipo de fichero determina la forma de interpretar la información que contiene, ya que, en definitiva, lo único que se almacena en un fichero es una ristra de bits (ceros y unos), de forma que es necesaria su interpretación para dar sentido a la información que almacena.

Así, por ejemplo, para almacenar una imagen en un ordenador, se puede usar un fichero binario .bmp, que almacena un vector de datos con los colores que tiene cada pixel que forma la imagen.

Además, la imagen posee una paleta de colores y unas dimensiones, información que también hay que almacenar en el fichero. Todos estos datos se ordenan según un formato, y el sistema operativo, o la utilidad que trate los gráficos, debe conocer este formato para poder extraer los píxeles y mostrarlos por pantalla en la forma y dimensiones correctas. Si se abre el gráfico con una utilidad como el bloc de notas, que solo sabe interpretar texto, el resultado será ilegible e incomprensible.

Por ejemplo, busca en tu ordenador un fichero con extensión .doc (del procesador de textos Microsoft Word), y ábrelo con el bloc de notas, pulsando con el ratón derecho sobre él y seleccionando la opción “Abrir con”. Observa que el bloc de notas no conoce el formato del fichero tipo doc, y por tanto, no sabe

interpretar el contenido del fichero, cosa que sí hace la aplicación de Microsoft.

Tradicionalmente, los ficheros se han clasificado de muchas formas, según su **contenido** (texto o binario), según su **organización** (secuencial, directa, indexada) o según su **utilidad** (maestros, históricos, movimientos). El contenido de un fichero puede ser tratado como texto, o como datos binarios, es decir, los bits almacenados en un fichero pueden ser traducidos por el sistema operativo a caracteres alfabéticos y números que entiende el ser humano, o pueden ser tratados como componentes de estructuras de datos más complejas, como ficheros que almacenan sonido, vídeo, imágenes, etc.

La **organización** de un fichero dicta la forma en que se han de acceder a los **datos**, así, los datos de un fichero con **organización secuencial**, están dispuestos siguiendo una secuencia ordenada, es decir, unos detrás de otros. Se caracterizan por tener que recorrer todos los datos anteriores para llegar a uno en concreto.

Los ficheros de **organización directa**, permiten acceder a un dato en concreto sin necesidad de acceder a todos los anteriores.

Finalmente, los de **organización indexada** acceden a los datos consultando un **índice**, es decir, una estructura de datos que permite acceder a la información rápidamente, simulando la forma en que el índice de un libro facilita el acceso a sus contenidos.


Existen también variantes de las anteriores que mezclan las mejores características de cada una de ellas.

Actualmente un sistema operativo trata un fichero desde dos puntos de vista:

1. Según su **contenido** (texto o datos binarios)
2. Según su **tipo** (imágenes, ejecutables, clips de videos, etc.)

2.2 Ficheros de texto

Los ficheros de texto suelen llamarse también ficheros planos o ficheros ASCII.

 El vocablo **ASCII** es un acrónimo de American Standard Code for Information Interchange. Es un estándar que asigna un valor numérico a cada carácter, con lo que se pueden representar los documentos llamados de Texto Plano, es decir, los que son directamente legibles por seres humanos. Puedes ver la tabla en el siguiente enlace: <http://ascii.cl/es/>

Los ficheros de texto, aunque no necesitan un formato para ser interpretado, suelen tener extensiones para conocer qué tipo de texto se halla dentro del fichero, por ejemplo:

- **Ficheros de configuración**: Son ficheros cuyo contenido es texto sobre configuraciones del sistema operativo o de alguna aplicación. Estos pueden tener extensión **.ini, .inf, .conf**
- **Ficheros de código fuente**: Su contenido es texto con programas informáticos. Ejemplos: **.sql, .c, .java**
- **Ficheros de páginas web**: Las páginas webs son ficheros de texto con hipertexto que interpreta el navegador. **.html, .css, .xml**
- **Formatos enriquecidos**: Son textos que contienen códigos de control para ofrecer una visión del texto más elegante: **.rtf, .tex**

2.3 Ficheros binarios

Los ficheros binarios son todos los que no son de texto, y requieren un formato para ser interpretado. A continuación se muestran algunos tipos de formatos de ficheros binarios:

- **De imagen**: .jpg, .gif, .tiff, .bmp, .wmf, .png, .pcx; entre muchos otros.
- **De vídeo**: .mpg, .mov, .avi, .qt.
- **Comprimidos o empaquetados**: .zip, .Z, .gz, .tar, .lhz
- **Ejecutables o compilados**: .exe, .com, .cgi, .0, .a
- **Procesadores de textos**: .doc, .odt

Generalmente los ficheros que componen una base de datos son de tipo binario, puesto que la información que hay almacenada en ellos debe tener una estructura lógica y organizada para que las aplicaciones puedan acceder a ella de manera universal, esto es, siguiendo un estándar. Esta estructura lógica y organizada, generalmente es muy difícil de expresar mediante ficheros de texto, por tanto, la información de una base de datos se suele guardar en uno o varios ficheros:

- El software de gestión de base de datos **Oracle** guarda la información en múltiples tipos de ficheros, llamados "datafiles", "tempfiles", "logfiles", etc.
- Un tipo de tablas del gestor **MySQL** guarda su información en 3 ficheros de datos binarios, con extensión .frm, .myd y .myi.
- **Access** guarda toda la información de una base de datos con extensión "mdb".

3. BASES DE DATOS



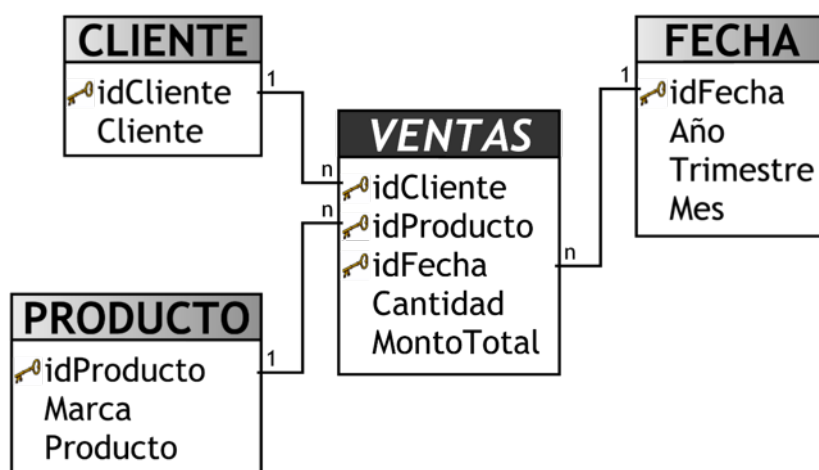
Una Base de Datos (BD) es una colección de información

perteneciente a un mismo contexto, que está almacenada de forma organizada en tablas.

Estas **tablas** se relacionan formando vínculos o relaciones entre ellas, que ayudan a mantener la información de diversos objetos de forma ordenada y coherente (sin contradicciones).

Cada una de estas tablas es una estructura que se parece a las hojas de cálculo, pues está dispuesta mediante filas y columnas. De este modo, cada fila almacena un **registro** con tantos **campos** como columnas tenga la tabla.

Por ejemplo, se podría tener una tabla de *Cliente*, donde cada fila o registro es un cliente de la empresa y cada columna o campo representa un trozo discreto de información sobre cada cliente, por ejemplo su número identificativo o el nombre.



Ejemplo de tablas relacionadas entre sí

3.1 Conceptos

Uno de los grandes problemas al que se enfrentan los informáticos cuando comienzan su aprendizaje, es el gran número de términos desconocidos que debe asimilar, incluyendo el enorme número de sinónimos y siglas que se utilizan para nombrar la misma cosa.

Tratando, a modo de resumen, de aclarar algunos de los componentes que se pueden encontrar en una base de datos, y que se verán en próximos capítulos, se definen los siguientes conceptos:

- **Dato:** El dato es un trozo de información concreta sobre algún concepto o suceso.

Por ejemplo, 1996 es un número que representa un año de nacimiento de


una persona. Los datos se caracterizan por pertenecer a un tipo.

- **Tipo de Dato:** El tipo de dato indica la naturaleza del campo. Así, se puede tener datos numéricos, que son aquellos con los que se pueden realizar cálculos aritméticos (sumas, restas, multiplicaciones...) y los datos alfanuméricos, que son los que contienen caracteres alfabéticos y dígitos numéricos. Estos datos alfanuméricos y numéricos se pueden combinar para obtener tipos de datos más elaborados.
Por ejemplo, el tipo de dato *Fecha* contiene tres datos numéricos, representando el día, el mes y el año de esa fecha.
- **Campo:** Un campo es un identificador para toda una familia de datos. Cada campo pertenece a un tipo de datos.
Por ejemplo, el campo *FechaNacimiento* representa las fechas de nacimiento de las personas que hay en la tabla. Este campo pertenece al tipo de dato *Fecha*. Al campo también se le llama columna.
- **Registro:** Es una recolección de datos referentes a un mismo concepto o suceso.
Por ejemplo, los datos de una persona pueden ser su NIF, año de nacimiento, su nombre, su dirección, etc. A los registros también se les llama tuplas o filas.
- **Campo Clave:** Es un campo especial que identifica de forma única a cada registro.
Así, el *NIF* es único para cada persona, por tanto es campo clave. Hay varios tipos de campos clave como se explicará más adelante.
- **Tabla:** Es un conjunto de registros bajo un mismo nombre que representa el conjunto de todos ellos.
Por ejemplo, todos los clientes de una base de datos se almacenan en una tabla cuyo nombre es *Clientes*.
- **Consulta:** Es una instrucción para hacer peticiones a una base de datos. Puede ser una búsqueda simple de un registro específico o una solicitud para seleccionar todos los registros que satisfagan un conjunto de criterios. Aunque en castellano, consulta tiene un significado de extracción de información, en inglés query, una consulta es una petición, por tanto, además de las consultas de búsqueda de información, que devuelven los campos y registros solicitados, hay consultas (peticiones) de eliminación o inserción de registros, de actualización de registros, cuya ejecución altera los valores de los mismos.

- **Índice:** Es una estructura que almacena los campos clave de una tabla, organizándolos para hacer más fácil encontrar y ordenar los registros de esa tabla. El índice tiene un funcionamiento similar al índice de un libro, guardando parejas de elementos: el elemento que se desea indexar y su posición en la base de datos. Para buscar un elemento que esté indexado, solo hay que buscar en el índice de dicho elemento para, una vez encontrado, devolver el registro que se encuentre en la posición marcada por el índice.
- **Vista:** Es una transformación que se hace a una o más tablas para obtener una nueva tabla. Esta nueva tabla es una tabla virtual, es decir, no está almacenada en los dispositivos de almacenamiento del ordenador, aunque sí se almacena su definición.
- **Informe:** Es un listado ordenado de los campos y registros seleccionados en un formato fácil de leer. Generalmente se usan como peticiones expresas de un tipo de información por parte de un usuario.
Por ejemplo, un informe de las facturas impagadas del mes de enero ordenado por nombre de cliente.
- **Guiones o scripts:** Son un conjunto de instrucciones, que ejecutadas de forma ordenada, realizan operaciones avanzadas de mantenimiento de los datos almacenados en la base de datos.
- **Procedimientos:** Son un tipo especial de script que está almacenado en la base de datos y que forma parte de su esquema.

3.2 Estructura de una base de datos

Una base de datos almacena los datos a través de un esquema.

 El **esquema** es la definición de la estructura donde se almacenan los datos, contiene todo lo necesario para organizar la información mediante tablas, registros (filas) y campos (columnas).

También contiene otros objetos necesarios para el tratamiento de los datos (procedimientos, vistas, índices, etc.) y que se estudiarán más adelante. Al **esquema** también se le suele llamar **metainformación**, es decir, información sobre la información o **metadatos**.

Los gestores de bases de datos modernos *Oracle*, *MySQL* y *DB2*, entre otros, almacenan el esquema de la base de datos en tablas, de tal manera que el

propio esquema de la base de datos se puede tratar como si fueran datos comunes de la base de datos.

3.3 Usos de las bases de datos

Las bases de datos podemos encontrarlas en cualquier tipo de sistema informático, a continuación se exponen solo algunos ejemplos de sus **usos más frecuentes**:

- **Bases de datos Administrativas**: Cualquier empresa necesita registrar y relacionar sus clientes, pedidos, facturas, productos, etc.
- **Bases de datos Contables**: También es necesario gestionar los pagos, balances de pérdidas y ganancias, patrimonio, declaraciones de hacienda.
- **Bases de datos para motores de búsquedas**: Por ejemplo Google o Altavista. Tienen una base de datos gigantesca donde almacenan información sobre todos los documentos de Internet. Posteriormente millones de usuarios buscan en la base de datos de estos motores.
- **Científicas**: Recolección de datos climáticos y medioambientales, químicos, genómicos, geológicos.
- **Configuraciones**: Almacenan datos de configuración de un sistema informático como por ejemplo, el registro de Windows.
- **Bibliotecas**: Almacenan información bibliográfica, por ejemplo, la famosa tienda virtual Amazon o la biblioteca de un instituto.
- **Censos**: Guardan información demográfica de pueblos, ciudades y países.
- **Virus**: Los antivirus guardan información sobre todos los potenciales software maliciosos.
- **Otros muchos usos**: Militares, videojuegos, deportes, etc.

3.4 Evolución y tipos de base de datos

La clasificación de las bases de datos en tipos, está ligada a su evolución histórica, como hemos comentado al principio del tema.

Según ha ido avanzando la tecnología, las bases de datos han mejorado cambiando la forma de representar y extraer la información.

De esta manera, se presenta la evolución sufrida por las bases de datos desde las épocas “prehistóricas” de la informática hasta la actualidad:

En la década de 1950 se inventan las **cintas magnéticas**, que solo podían ser leídas de forma **secuencial y ordenadamente**. Estas cintas, almacenaban ficheros con registros que se procesaban secuencialmente junto con ficheros de movimientos para generar nuevos ficheros actualizados. Estos sistemas se conocen como **aplicaciones basadas en sistemas de ficheros** y constituyen la **generación cero de las bases de datos**, pues ni siquiera entonces existía el concepto de bases de datos.

En la década de 1960 se generaliza el uso de **discos magnéticos**, cuya característica principal es que se podía **acceder de forma directa** a cualquier

parte de los ficheros, sin tener que acceder a todos los datos anteriores. Con esta tecnología aparecen las **bases de datos jerárquicas y en red**, que aprovechan la capacidad de acceso directo a la información de los discos magnéticos para estructurar la información en forma de **listas enlazadas y árboles de información**.

La filosofía de las bases de datos en red es que un concepto principal o padre puede tener numerosas relaciones con conceptos secundarios o hijos. Las bases de datos jerárquicas, evolucionan para admitir varios padres para un concepto hijo.

Edgar Frank Codd, científico informático inglés de *IBM*, publica en 1970 en un artículo “*Un modelo relacional de datos para grandes bancos de datos compartidos*”, donde **definió el modelo relacional**, basado en la lógica de predicados y la teoría de conjuntos. Nacieron, de esta forma, las **bases de datos relacionales**, o segunda generación de bases de datos.

Larry Ellison, fundador de *Oracle*, se inspiró en este artículo para desarrollar el famoso motor de base de datos, que comenzó como un proyecto para la CIA (Central Intelligence Agency) americana.

La potente base matemática de este modelo, es el gran secreto de su éxito. Hoy en día, el modelo relacional de Codd, pese a tener muchas alternativas, sigue siendo el más utilizado a todos los niveles.

En la década de **1980 IBM** lanza su motor de bases de datos **DB2**, para la plataforma *MVS*¹. Unos años después, **IBM** crea el **SQL (Structured Query Language)**, un potente lenguaje de consultas para manipular información de bases de datos relacionales.

A mediados de 1990, **IBM** lanza una versión de **DB2** que es capaz de dividir una base de datos enorme en varios servidores comunicados por líneas de gran velocidad, creándose de este modo las bases de datos paralelas. A esta versión se le llamó **DB2 Parallel Edition**, que ahora, ha evolucionado hasta el **DBZ Data Partition Feature**, único SGBD de este tipo en sistemas distribuidos.

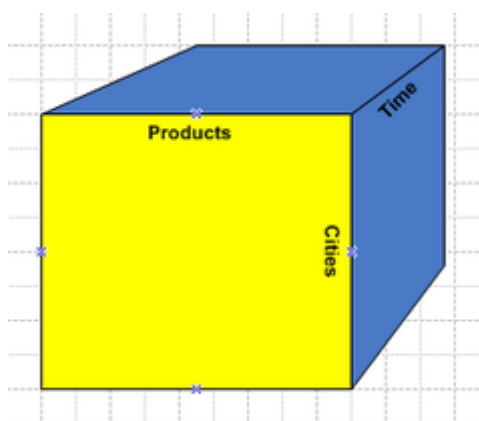
A finales de 1990 **IBM y Oracle** incorporan a sus bases de datos la capacidad de manipular objetos, creando así, las **bases de datos orientadas a objetos**. Estas bases de datos orientadas a objetos se basan en la existencia de objetos persistentes que se almacenan para su procesamiento mediante programas orientados a objetos. En lugar de la filosofía de almacenar relaciones y tablas, se almacenan colecciones de objetos que, además de información, tienen comportamientos (instrucciones sobre cómo procesar los datos).

La aparición de Internet y el comienzo de la era de la información, crean

¹ **MVS** (Multiple Virtual Storage, Múltiple Almacén Virtual en español) fue el sistema operativo, desarrollado por IBM, más usado en los modelos de mainframes System/370 y System/390.

nuevos requerimientos para bases de datos. La cantidad de información comienza a crecer en proporciones desconocidas hasta el momento. De esta forma, se crean las bases de datos distribuidas, que consisten en multiplicar el número de ordenadores que controlan una base de datos (llamados nodos), intercambiándose información y actualizaciones a través de la red.

Este increíble aumento de datos a almacenar, organizados muchas veces en datos estadísticos recopilados con el trascurso de los años, hizo necesaria la aparición de un software llamado **Software de ayuda a la decisión**. Este software avanzado trata de dar respuestas concretas examinando múltiples datos estadísticos que se han recopilado a lo largo del tiempo en bases de datos multidimensionales, formando lo que se denominan **cubos de información**.



Ejemplo de cubo de información

También, a lo largo de la corta historia de la informática, han surgido otros tipos de bases de datos que se enumeran a continuación:

- **Bases de datos espaciales o geográficas**: Son bases de datos que almacenan mapas y símbolos que representan superficies geográficas. Google Earth es una aplicación que lanza consultas a bases de datos de este tipo.
- **Bases de datos documentales**: Permiten la indexación de texto para poder realizar búsquedas complejas en textos de gran longitud.
- **Bases de datos deductivas**: Es un sistema de bases de datos que almacena hechos y que permite, a través de procedimientos de inferencia, extraer nuevos hechos. Se basan en la lógica, por ello también se suelen llamar bases de datos lógicas.

4. SISTEMAS GESTORES DE BASE DE DATOS

4.1 Concepto de Sistema Gestor de Base de Datos

 Se define un **Sistema Gestor de Base de Datos**, en adelante

SGBD², como el conjunto de herramientas que facilitan la consulta, uso y actualización de una base de datos.

Un ejemplo de software Gestor de Base de Datos es **Oracle 11g**, que incorpora un conjunto de herramientas software que son capaces de estructurar en múltiples discos duros los ficheros de una base de datos, permitiendo el acceso a sus datos tanto a partir de herramientas gráficas como a partir de potentes lenguajes de programación (**PL-SQL, php, c++...**).

4.2 Funciones de un SGBD

Los **SGBD** del mercado cumplen con casi todas las **funciones** que a continuación se enumeran:

1. **Permiten a los usuarios almacenar datos, acceder a ellos y actualizarlos de forma sencilla** y con un gran rendimiento, ocultando la complejidad y las características físicas de los dispositivos de almacenamiento.
2. **Garantizan la integridad de los datos**, respetando las reglas y restricciones que dicte el programador de la base de datos. Es decir, no permiten operaciones que dejen cierto conjunto de datos incompletos o incorrectos.

Un **problema** que nos encontramos **a la hora de almacenar datos es la redundancia**, que se define como la cantidad de datos repetidos en la información guardada. Uno de los principales objetivos de una base de datos es reducir todo lo posible la redundancia. Con ello conseguimos dos cosas:

1. Que la información ocupe menos espacio
2. Que sea lo más coherente posible.

La inconsistencia de los datos se produce cuando un dato redundante es diferente en dos o más sitios.

3. **Integran**, junto con el sistema operativo, **un sistema de seguridad que garantiza el acceso a la información exclusivamente a aquellos usuarios que dispongan de autorización.**
4. **Proporcionan un diccionario de metadatos**, que contiene el esquema de la base de datos, es decir, cómo están estructurados los datos en tablas, registros y campos, las relaciones entre los datos, usuarios, permisos, etc. Este diccionario de datos debe ser también accesible de la misma forma sencilla que es posible acceder al resto de datos.
5. **Permiten el uso de transacciones**, garantizan que todas las operaciones de la transacción se realicen correctamente, y en caso de alguna incidencia, deshacen los cambios sin ningún tipo de complicación adicional.
6. **Ofrecen**, mediante completas herramientas, **estadísticas** sobre el uso del gestor, registrando operaciones efectuadas, consultas solicitadas, operaciones fallidas y cualquier tipo de incidencia. Es posible de este modo, monitorizar el uso de la base de datos, y permiten analizar

² En inglés, **DBMS** (DataBase Management System)

hipotéticos funcionamientos erróneos.

7. **Permiten la concurrencia**, es decir, varios usuarios trabajando sobre un mismo conjunto de datos. Además, proporcionan mecanismos que permiten arbitrar operaciones conflictivas en el acceso o modificación de un dato al mismo tiempo por parte de varios usuarios.
8. **Independizan los datos de la aplicación** o usuario que esté utilizándolos, haciendo más fácil su migración a otras plataformas.
9. **Ofrecen conectividad con el exterior**. De esta manera, se puede replicar y distribuir bases de datos. Además, todos los SGBD incorporan herramientas estándar de conectividad. El protocolo ODBC4 está muy extendido como forma de comunicación entre bases de datos y aplicaciones externas.
10. **Incorporan herramientas para la salvaguarda y restauración de la información en caso de desastre**. Algunos gestores, tienen sofisticados mecanismos para poder establecer el estado de una base de datos en cualquier punto anterior en el tiempo. Además, deben ofrecer sencillas herramientas para la importación y exportación automática de la información.

4.3 El lenguaje SQL

La principal herramienta de un gestor de base de datos es la interfaz de programación con el usuario. Este interfaz consiste en un lenguaje muy sencillo mediante el cuál el usuario realiza preguntas al servidor, contestando este a las demandas del usuario.

Este lenguaje comúnmente se denomina **SQL, Structured Query Language**, está estandarizado por la ISO³, es decir, todas las bases de datos que soporten SQL deben tener la misma sintaxis a la hora de aplicar el lenguaje.

Se divide en 4 sublenguajes, el total de todos ellos permite al SGBD cumplir con las funcionalidades requeridas por Cood:

- **Lenguaje DDL**: o lenguaje de definición de datos (**Data Definition Language**). Este lenguaje permite crear toda la estructura de una base de datos (desde tablas hasta usuarios). Sus órdenes son del tipo **DROP** (Eliminar objetos) y **CREATE** (Crear objetos).
- **Lenguaje DML**: o lenguaje de manipulación de datos (**Data Manipulation Language**). Este lenguaje permite, con 4 sentencias sencillas, seleccionar determinados datos (**SELECT**), insertar datos (**INSERT**), modificarlos (**UPDATE**) o incluso borrarlos (**DELETE**). Más adelante se desarrollará la sintaxis de cada una de estas sentencias.
- **Lenguaje DCL**: o lenguaje de control de datos (**Data Control Language**). Incluye comandos (**GRANT** y **REVOKE**) que permiten al administrador gestionar el acceso a los datos contenidos en la base de datos.

3 ISO es el acrónimo de International Standardization Organization

- **Lenguaje TCL**: o lenguaje de control de transacciones (**Transaction Control Language**). El propósito de este lenguaje es permitir ejecutar varios comandos de forma simultánea como si fuera un comando atómico o indivisible. Si es posible ejecutar todos los comandos, se aplica la transacción (**COMMIT**), y si en algún paso de la ejecución sucede algo inesperado, se pueden deshacer todos los pasos dados (**ROLLBACK**).

4.4 Tipos de SGBD

Se pueden clasificar los SGBD de muchas formas, por ejemplo, según las bases de datos que gestionan, clasificando los SGBD según traten bases de datos relacionales, bases de datos orientadas a objetos, etc. Puesto que en la actualidad, la mayoría de los SGBD integran múltiples filosofías y tipos de funcionamiento, en este curso vamos a clasificar los de gestores de bases de datos según la capacidad y potencia del propio gestor:

Los Gestores de Bases de Datos ofimáticas son aquellos que manipulan bases de datos pequeñas (ofimáticas) orientadas a almacenar datos domésticos o de pequeñas empresas. Incluso estos gestores permiten construir pequeñas aplicaciones para ayudar a un usuario inexperto a manipular los datos de una base de datos de forma sencilla e intuitiva.

Un ejemplo de un SGBD ofimático es **Microsoft Access**, que posee tanto una interfaz de usuario muy sencilla como un potente lenguaje de programación (VBA=Visual Basic for Applications) para ofrecer a usuarios avanzados otras posibilidades de gestión mucho más específicas.

Los Gestores de bases de datos Corporativas son aquellas que tienen la capacidad de gestionar bases de datos enormes, de grandes o medianas empresas con una carga de datos y transacciones que requieren un servidor de grandes dimensiones (generalmente un Servidor *Unix*, o un *Windows 200X Server* con altas prestaciones).

Estos gestores son capaces de manipular grandes cantidades de datos de forma muy rápida y eficiente para poder resolver la demanda de muchos (cientos) de usuarios.

Un ejemplo típico de servidor de base de datos Corporativas es el antes comentado **Oracle**, actualmente, junto con **DB2**, el servidor de base de datos más potente del mercado (también el más caro).

Precisamente, ese coste tan alto es el que ha desencadenado que se haya recurrido a una solución intermedia entre gestores de base de datos ofimáticas y corporativas. Entre estas soluciones intermedias se encuentra **MySQL**, un gestor de base de datos que, además de ser gratuito y sencillo, es capaz de manipular gran cantidad de datos cumpliendo prácticamente todos los estándares de la arquitectura ANSI SPARC⁴. Aunque implementa SQL, no tiene un lenguaje de programación propio como *SQL Server* u *Oracle* (aunque está en

⁴ https://es.wikipedia.org/wiki/Arquitectura_ANSI-SPARC

desarrollo), pero a cambio se integra fácilmente en las típicas soluciones **XAMPP**⁵, que son paquetes que incluyen, además de MySQL, una versión del servidor Web Apache y varios lenguajes de script (php, perl. . .) que dotan a MySQL de potentes herramientas para acceso y publicación de los datos.

5. ARQUITECTURA EN NIVELES DE UNA BASE DE DATOS

En cualquier sistema de información se considera que **se pueden observar los datos desde dos puntos de vista**:

- **Vista externa.** Esta es la visión de los datos que poseen los usuarios del Sistema de Información.
- **Vista física.** Esta es la forma en la que realmente están almacenados los datos.

En un sistema orientado a procesos, los usuarios ven los datos desde las aplicaciones creadas por los programadores. Esa vista pueden ser formularios, informes visuales o en papel. Pero la realidad física de esos datos, tal cual se almacenan en los discos queda oculta. Esa visión está reservada a los administradores.

En el caso de los **Sistemas de Base de datos**, se añade una **tercera vista**, que es la **vista conceptual**. Esa vista **se sitúa entre la física y la externa**.

Por tanto, en Bases de datos, estudiaremos **tres niveles** para representar los datos:

- **Esquema, nivel o diseño físico.** Representa la **forma en la que están almacenados los datos**. Esta visión sólo la requiere el administrador/a. El administrador la necesita para poder gestionar más eficientemente la base de datos.

En este esquema se habla de archivos, directorios o carpetas, unidades de disco, servidores, etc.

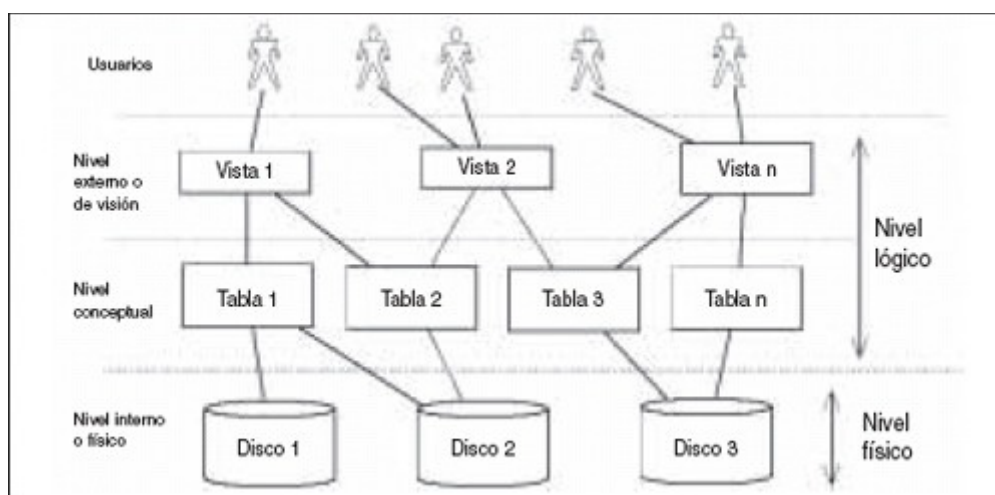
- **Esquema, nivel o diseño conceptual.** Se trata de un **esquema teórico de los datos** en el que figuran organizados en estructuras reconocibles del mundo real y en el que también aparece la forma de relacionarse los datos. **Este esquema es el paso que permite modelar un problema real a su forma correspondiente en el ordenador.**

El diseño conceptual es el primer paso que realizan diseñadores/as o analistas para crear una base de datos.

5 **XAMPP**: El nombre proviene del acrónimo de **X** (para cualquiera de los diferentes sistemas operativos: LAMPP para Linux, WAMPP para Windows, etc.), **A**pache, **M**ySQL, **P**HP, **P**erl.

- **Esquema, nivel o diseño externo.** Se trata de la **visión de los datos que poseen los usuarios y usuarias finales**. Esa visión es la que se obtiene **a través de las aplicaciones**. Las aplicaciones creadas por los desarrolladores abstraen la realidad conceptual, de modo que el usuario no conoce las relaciones entre los datos, como tampoco conoce todos los datos que realmente se almacenan.

Realmente cada aplicación produce un esquema externo diferente (aunque algunos pueden coincidir) o vista de usuario. El conjunto de todas las vistas de usuario es lo que se denomina esquema externo global.



Arquitectura en niveles de una BD

En el siguiente tema se explican más detenidamente estos niveles y los modelos que se utilizan para su definición.

6. BIBLIOGRAFÍA

- [1] *Bases de Datos*. Iván López, M.^a Jesús Castellano. John Ospino. Ed. Garceta
- [2] *Gestión de Bases de Datos. Tema 1: Introducción a las Bases de Datos*. Concepción Guisado, José Domingo, Raúl Ruiz. IES Gonzalo Nazareno.