

# Tema 1: SISTEMAS DE ALMACENAMIENTO DE LA INFORMACIÓN

Bernat Costa

[Bernat.costa@cesurformacion.com](mailto:Bernat.costa@cesurformacion.com)

# Ejercicio

## Nuestras series favoritas

Vamos a normalizar nuestra encuesta. Vamos a ponernos de acuerdo en que vamos a preguntar.

Preguntemos sobre 3 series a 3 compañer@s

Digitalicemos con el notepad nuestra encuesta en un fichero CSV.

(Una línea por registro, ; para separar los distintos campos)

Importemos ese CSV en el Excel online.

Ahora, unamos todos los CSV en uno.

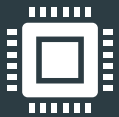
# Índice

1. ¿Cómo guardamos la información?
2. ¿Qué es una bbdd?
3. ¿Qué es un Sistema Gestor de bbdd?
4. SQL
5. BBDD Distribuidas

## 1.1 ¿Cómo guardamos la información en el ordenador?



En informática, tenemos dos tipos de memoria, la volátil y la persistente.



La volátil, (RAM), es de más rápido acceso, pero se pierde al apagar el equipo.



La persistente, es la que guardamos en el disco duro, pendrive, cd...

# Ficheros



Para guardar la información en disco, usamos los ficheros.



Los ficheros son estructuras de información creados por los Sistemas Operativos para poder almacenar datos.

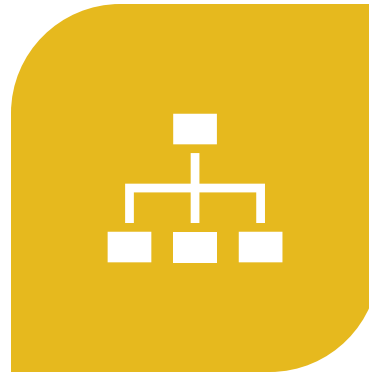


Según el tipo de fichero, guardaremos la información de una manera u otra.

# Tipos de fichero



**CONTENIDO**



**ORGANIZACIÓN**



**UTILIDAD**

# Ficheros por contenido

## Fichero de Texto

Organizado con caracteres alfanumericos. (Se puede leer con el bloc de notas)

## Fichero Binario

Son datos complejos que requieren de un software para interpretarlos. (imagen, sonido, video...)

# Ficheros por Organización

## Secuenciales

- Para acceder a un dato, debemos de recorrer todos los datos anteriores.


## Relativos

- Los datos están posicionados a X posiciones del inicio del fichero. De esta forma, podemos "saltar" de un dato a otro.

## Indexados

- Tenemos un índice, que nos permite saltar a cualquier dato.





# Ficheros según su uso.

- ▶ Fichero Maestro
  - ▶ Contiene la info principal de la app
- ▶ Fichero Historico
  - ▶ Contiene las transacciones historicas de la app, ya no necesarias
- ▶ Fichero Temporal

# Ficheros de texto

## Configuración

- .config, .ini, .inf...

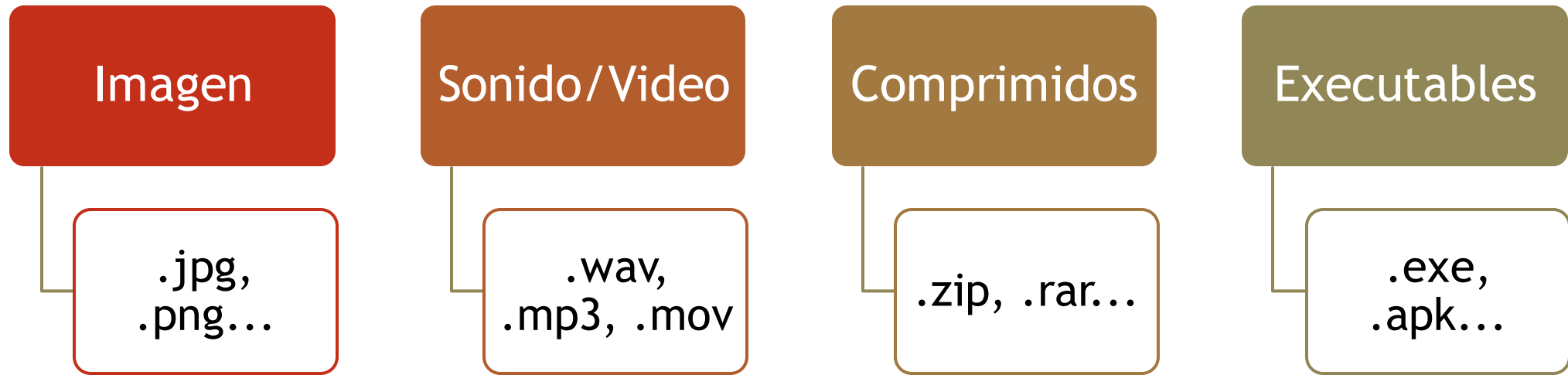
## Código fuente

- .java, .cpp, .py, .js....
- Texto plano .txt

## XML

- HTML -> .html, .css...

# Ficheros Binarios



# ¿Que es una BBDD?

Una base de datos es un **conjunto de datos** ordenados que están **relacionados** entre sí y **almacenados** sistemáticamente para su uso posterior de forma automatizada con un mismo **contexto**

## 1.2 ¿Qué es una BBDD?

Conjunto de  
Datos

Relacionados

Almacenados

Sistematicamente

Automatizado

Mismo  
Contexto

# Conjunto de Datos

Un dato es una parte de la información de un objeto o suceso.

Podremos tener distintos tipos de datos ( fechas, numeros, texto....)



# Relacionados

En una BBDD, tenemos la información relacionada. Los datos tienen relación entre si, y esa relación se "escribe" en la definición de la misma.

# Almacenados

Una BBDD sirve para guardar información, por lo tanto, el almacenamiento es su principal función.



Se deben guardar los datos de forma persistente.



## Sistematicamente

El proceso de creación de una BBDD debe estar parametrizado.

Tenemos un método para crear las BBDD, un patrón.


# Automatizado

Entenderemos por BBDD, un conjunto de datos automatizado. Digitalizado.


Aunque podríamos tener BBDD analógicas, cuando hablemos de BBDD en esta asignatura, nos referiremos a aquellas digitalizadas.

# Uso posterior

Debemos poder consultar y manipular los datos.

A light green arrow pointing downwards from the first box to the second box.

Si no podemos, no estamos hablando de una BBDD.

A light orange arrow pointing downwards from the second box to the third box.

De ahí viene el gran potencial de las BBDD.

# CONTEXTO

---

Es primordial, acordar un contexto para los datos.

---

Un campo fecha, puede indicar, una fecha de un pedido, un cumpleaños, una fecha de registro...

---

Sin ese contexto ( muchas veces nos lo dará el nombre del campo o de la tabla), no podemos interpretar los datos.

## 1.3 Sistemas gestores de BBDD



En los años 70, aparece el primer software para gestionar BBDD, Oracle.



Se define un sistema Gestor de Base de Datos (SGBD) como un conjunto de herramientas que facilitan la consulta, uso y actualización de una Base de Datos.



Algunos ejemplos serian:

MySQL, MariaDB, PosgresSQL, Oracle, Microsoft SQL Server, MongoDB

# Características de un SGDB

Abstracción

Integridad

Seguridad

Metadata

Transacciones

Monitorización

Concurrencia

Independencia

Conectividad

# ABSTRACCION

- ▶ Los SGDB nos harán olvidarnos de que hay debajo.
- ▶ Nos abstraerá del software o de la forma de almacenar los datos.
- ▶ Cuando trabajamos con ficheros, tenemos que estar gestionando como guardamos los datos. En un SGDB nos olvidaremos de esa parte.

# INTEGRIDAD

Nos permitirán garantizar la integridad de los datos.

Tendremos herramientas de recuperación, además de garantizarnos que si un campo es de tipo número, no podremos guardar una letra.



# Seguridad

---

Integra un sistema de seguridad que garantiza el acceso a la Base de datos solo por usuarios autorizados.

---

Tendremos usuarios y permisos para acceder a una BBDD.

---

Permisos para leer, para escribir, para modificar tablas...

# METADATA

---

Nos facilitará toda una estructura de tablas, filas, registros, tipos de datos.

---

Nos proporciona la metadata para crear nuestro esquema de la BBDD.



# TRANSACCIONES

- ▶ Permiten el uso de transacciones que garantizan que todas las operaciones se realizan correctamente. En caso contrario, deshace los cambios.
- ▶ No guardará los cambios si algo falla.
- ▶ Guardar la partida antes de empezar"



# Monitorización

- ▶ Nos dará información sobre el estado de la bbdd.
  - ▶ Consultas realizadas por usuario
  - ▶ Consultas fallidas
  - ▶ Uso de recursos de la máquina...

# CONCURRENCIA

Acceder N personas a la vez.

Tendrán mecanismos, para solucionar los problemas de concurrencia.

Tanto para lectura, como para escritura.

"Cena de los filósofos"

# INDEPENDENCIA

---

Independizan la parte del software de acceso a datos.

---

El desarrollador se olvida del acceso a datos. Además es independiente del lenguaje o plataforma usada.

---

Una consulta SQL, servirá tanto en c#, como en python, como en php.

# Conectividad

Podremos conectarnos a la BBDD desde el exterior de la máquina.

A light green arrow pointing downwards, indicating a flow from the first step to the second.

Tendremos mecanismos para hacer "pública" nuestra BBDD.

A light orange arrow pointing downwards, indicating a flow from the second step to the third.

Podremos replicarla y distribuirla.

# Un poco de historia de las BBDD

- ▶ Desde que el ser humano escribe, documenta, desde que documenta, almacena datos. Biblioteca Alejandria.
- ▶ En **1884**, **Herman Hollerith**, inventa tabulador electromagnético de tarjetas perforadas
- ▶ En un simposium en California en 1963 se usa por primera vez el concepto de Base de datos
- ▶ **Charles Bachman**, desarrolla las bases de datos en red en los años 60
- ▶ **Edgar Frank Codd**, desarrolla el modelo relacional en los 70.
- ▶ Más tarde, **Larry Ellison** basado en el trabajo de **Edgar Frank Codd** creó el **Relational Software System** que actualmente se conoce como **Oracle Corporation**. El primer SGDB
- ▶ En los 80, se creó un **lenguaje de consulta estructurado** (SQL - Structured Query Language)



¿Que sistemas GESTORES de BBDD tenemos?

**Software  
propietario**

**Software  
libre.**

# ¿Qué es el software libre?

- ▶ Proyecto GNU y Richard Stallman
- ▶ Free Software != Software Gratis.
- ▶ 4 libertades
  - ▶ **Libertad 0:** Poder usar el programa con cualquier propósito.
  - ▶ **Libertad 1:** Poder estudiar cómo funciona el programa y poder modificarlo.
  - ▶ **Libertad 2:** Poder distribuir copias del programa.
  - ▶ **Libertad 3:** Poder mejorar el programa y poder compartir dichas mejoras para beneficio de todos.
- ▶ Varias licencias, Lo más estandar es obligar a mantener licencia y a reconocer la autoria de todas las personas.

# SOFTWARE LIBRE

- ▶ Software libre:
  - ▶ MySQL -> más conocido, comprado por Oracle
  - ▶ MariaDB -> Fork de MySQL creado por la comunidad después de la compra de Oracle
  - ▶ PostgreSQL, creciendo en popularidad



# ¿MySQL o PostgreSQL?

## MySQL o MariaDB

- Muy ágil y adecuado para proyectos web
- Muy popular

## PostgreSQL

- Más robusto
- Pensado para BBDD grandes.
- Mejor concurrencia.

# Software propietario

2 opciones predominan el mercado

Oracle, el original.

SQL Server de Microsoft.  
(Azure....)



Son muy parecidos, los dos comparten casi la totalidad de las instrucciones SQL ( con algunas diferencias)

# ORACLE VS SQLSERVER

ases para la mparación	Oráculo	SERVIDOR SQL
ISIC	El lenguaje utilizado por Oracle es PL / SQL (Lenguaje de procedimiento / SQL).	El lenguaje utilizado por SQL Server es T-SQL (Transact-SQL).
ansacción	En Oracle, no se confirma ninguna transacción hasta que el DBA emita explícitamente el comando COMMIT.	Si no se especifican los comandos BEGIN TRANSACTION y COMMIT, se ejecutan y confirman cada comando individualmente.
ganización	La base de datos se comparte entre todos los esquemas y usuarios.	Base de datos no compartida entre los usuarios.
quetes	Los procedimientos, la función y la variable se agrupan en paquetes.	Los paquetes no están allí en SQL.
porte OS	Windows, Linux, Solaris, HP-UX, OS X, z / OS, AIX.	Windows y Linux.
mplejidad	Complejo pero potente.	Más simple y fácil de usar.

# Herramientas de un SGDB



2 herramientas imprescindibles:

El motor

Cliente Una aplicación para consultas.



MYSQL tiene 2 alternativas

phpmyadmin para las consultas, una app web.  
MySQL Workbench, cliente multiplataforma



SQL Server tiene 2 clientes:

Azure Data Studio

SQL Server Management Studio.



Hay alternativas multiplataforma como DBEAVER para conectarse a casi todos los motores.



## ¿Que usaremos en clase?

- ▶ Herramientas de consulta como Azure Data Studio instaladas en nuestro Windows.
- ▶ Principalmente, usaremos SQL Server.
- ▶ Docker para levantar los motores de la BBDD.
- ▶ Sin olvidarnos de DIA Software para hacer nuestros diagramas.



# ¿Y eso de MongoDB? ¿Qué es?

Hay otro tipo de BBDD llamadas NO-SQL donde se rompe toda la estructura y lógica que veremos en esta clase.

SQL no es incompatible con NO-SQL, se suelen complementar.

Hay veces que necesitaremos una BBDD SQL y otras una NO-SQL

En esta asignatura, nos centraremos en las BBDD SQL.

## 1.4 ¿Qué es SQL?

- ▶ Structured Query Language. Lenguaje de consulta estructurado.
- ▶ Es el lenguaje con el que nos comunicaremos con la BBDD.
- ▶ Tiene 4 sublenguajes según su utilidad.

DDL

DML

DCL

TCL

# DDL



## Data Definition Language



Es la parte de SQL que nos va a servir para definir BBDD.



Crear tablas, borrarlas, modificarlas...



3 operaciones básicas:

CREATE, DROP, ALTER  
van a ser sus comandos más habituales.

# DML

Data Manipulation Language

```
graph TD; A[Data Manipulation Language] --> B[Esta parte va a ser la parte con la que trabajaremos con los datos.]; B --> C[4 operaciones básicas]; C --> D[Consulta (SELECT)]; C --> E[Inserción (INSERT)]; C --> F[Modificación (UPDATE)]; C --> G[Borrado (DELETE)];
```

Esta parte va a ser la parte con la que trabajaremos con los datos.

4 operaciones básicas

Consulta (SELECT)

Inserción (INSERT)

Modificación (UPDATE)

Borrado (DELETE)

# DCL

- ▶ Data Control Language
- ▶ Instrucciones para el control de la BBDD.
- ▶ Nos servirá para definir permisos de usuario entre otras cosas.
- ▶ Por ejemplo, tendremos 2 instrucciones para dar permisos a usuarios (GRANT, REVOKE...)



# TCL

Transaction Control  
Language

Las transacciones  
nos van a servir para  
mantener la  
coherencia de los  
datos.

Una transacción es  
"guardar la partida"  
antes de modificar  
datos.

Nos va a servir para  
lanzar  
modificaciones en  
bloque. O se hacen  
todas o ninguna.

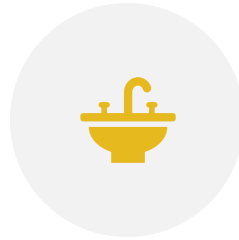
Si hay un fallo,  
poder volver a atrás.

BEGIN TRAN,  
ROLLBACK TRAN

# 1.5 BBDD Distribuidas



Originalmente, las BBDD estaban centralizadas en un solo lugar.



En la actualidad, es habitual tener bbdd distribuidas por la red.



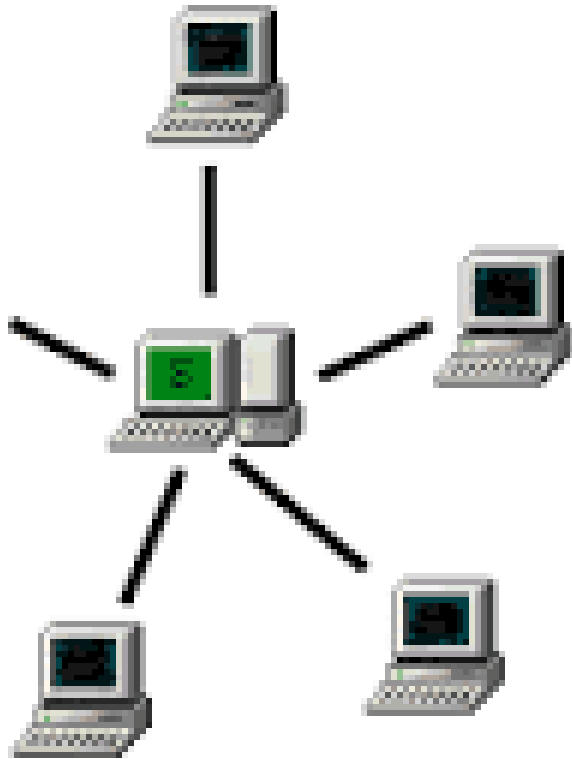
Un Sistema centralizado, simplifica el acceso y la escritura, pero pone en riesgo el sistema ante una posible caída.



Un sistema distribuido, será más resistente a caídas puntuales, pero nos dificulta mucho la escritura.

# cliente-servidor Vs Distribuido

## Estructura cliente-Servidor



## Estructura P2P





# Formas de distribuir una BBDD

---

Replicación

---

Centralizada

---

Particionada

---

Híbrida

# Ventajas Vs Desventajas del modelo distribuido

## Ventajas

- Autonomía local
- Mayor disponibilidad
- Escalabilidad
- Expansión territorial
- + resistente a fallos

## Desventajas:

- Dificultad para garantizar la seguridad
- Errores de red
- + Complejidad
- Privacidad
- Modelos + Nuevos, + vulnerables



*That's all Folks!*