

# Introducción y Conceptos Generales

---

## Temas:

- Conceptos Generales.
- Por qué usar BDs ?
- Actores Involucrados
- Usos de DBMSs.
- Cuando no usar DBMS.

## Referencia:

---

- **Capítulo 1 del Elmasri-Navathe [EN]**

# Conceptos Generales

---

- Base de Datos (BD).
  - Conjunto de datos relacionados.
  - Volúmenes muy importantes ( >>> RAM ).

# Conceptos Generales

---

- Database Management System (DBMS).
  - Software especializado en gestión de BDs.

## Procesos involucrados:

---

- ♦ **Definición de una BD.**
- ♦ **Construcción de una BD.**
- ♦ **Manipulación de una BD.**

## Por qué usar “Bases de Datos” ?

---

- ♦ **Qué aporta esta tecnología ?**
  - ♦ Software “sofisticado” para manejo de grandes volúmenes de datos.
- ♦ **Qué se haría si no se puede usar ?**
  - ♦ Desarrollar programas usando lenguajes de propósito general.
    - ♦ C, Pascal, COBOL, C++, ...
- ♦ **... y esto que problemas trae ?**
  - ♦ Más trabajo para manipular datos.
  - ♦ Productividad, Costos, Calidad.

## Ejemplo - Aplicación Bancaria

- Un banco desarrolla un Sistema de Información.
  - Se tiene el siguiente conjunto de programas:
    - Un programa que debita y/o acredita cuentas.
    - Un programa que agrega una nueva cuenta.
    - Un programa que encuentra el saldo de una cuenta.
    - Un programa que genera los movimientos mensuales de una cuenta.
    - Un programa que liste el cliente, la dirección y el saldo de su cuenta.
  - Se usa tecnología de archivos tracionales y lenguajes de tipo propósito general (p. Ej. COBOL, C).

## Problemas que se presentan al no usar BD

- Específicos a la implementación basada en archivos, a resolver por el programador.
  - Inconsistencia de datos.
    - Programar control de Restricciones de Integridad.
  - Dificultades para acceso a información.
    - Programar consultas/listados específicos.
  - Aislamiento de datos.
    - Programar algoritmos de merge (*apareo archivos*).
  - Seguridad en datos.
    - Programar los controles de acceso para los ítems.

## Problemas que se presentan

- **Control de concurrencia.**
  - Qué pasa si se ejecuta el programa de RETIRO desde dos estaciones ?
    - RETIRO(nro\_cuenta, monto).
- **Esquemáticamente sería:**
  - **Read(cuenta);**
    - /\* lectura de la cuenta a fin de visualizar el saldo actual de la misma y poder modificarlo \*/
  - **cuenta.saldo := cuenta.saldo – cant\_retiro;**
    - /\* se actualiza el saldo \*/
  - **Write(cuenta);**
    - /\* se hace efectivo el retiro \*/

## Problemas que se presentan

- **Una ejecución podría ser de la forma:**

### Puesto 1 (RETIRO(#10,50))

- saldo = Read\_saldo(#10); // (\$ 500)  
- saldo = saldo - monto; // (queda \$ 450)  
- write\_saldo(#10,saldo);  
- Fin\_Operación.

**QUEDA SALDO = \$ 450**

### Puesto 2 (RETIRO(#10,100))

- saldo = Read\_saldo(#10); // (\$ 500)

- saldo = saldo - monto; // (queda \$ 400)  
- write\_saldo(#10,saldo);  
- Fin\_Operación.

**QUEDA SALDO = \$ 400**

↓  
tiempo

## Por qué usar “Bases de Datos” ?

- **Organización de la información.**
  - Definición central de datos.
  - Abstracción de Datos.
  - Múltiples vistas de los datos.
  - Almacenamiento de datos y programas.
- **“Programación” de la BD.**
  - Separación entre programas y datos.
  - Control de *Restricciones de Integridad*.
  - Estandarización de modelos y lenguajes.

## Por qué usar “Bases de Datos” ?

- **Explotación de la BD.**
  - Datos compartidos.
  - Control de concurrencia.
  - Seguridad y Recuperación ante fallas.
  - Datos persistentes en modelos avanzados.

## Actores Involucrados

---

- **Visibles.**
  - Administrador de BDs.
  - Diseñador de BDs.
  - Desarrolladores de Sistemas de Info.
  - Usuarios finales.
- **Poco visibles.**
  - Implementadores de DBMS.
  - Operadores de sistemas.

## Por qué no usar DBMSs

---

- **Alto costo en inversión:**
  - Hardware.
  - Software.
  - Capacitación técnica.
- **Costo de administración:**
  - Del DBMS.
  - De las Bases de Datos.

## Por qué no usar DBMSs

---

- Cuando no se justifica usar DBMSs:
  - Muy pocos datos (planillas ?).
  - Datos muy estables.
  - Alta performance (Tiempo Real).
  - No hay acceso concurrente.
- Los DBMSs “de escritorio”:
  - Similares en Modelo de Datos a los grandes.
  - Bajo costo (plataforma PC).
  - Por ejemplo: MS-ACCESS.