Tema 1. Introducción a bases de datos.

# Introducción:

Las bases de datos surgen como respuesta al problema de dependencia existente en aplicaciones en las que existen una relación directa entre el modelo de almacenamiento y el modelo lógico. Las bases de datos intentan separar la implementación física de la implementación lógica.

# Modelos de datos:

* Es la manera en la que se representa la información de un sistema. Representa contenidos y relaciones entre los contenidos.
* La implementación del modelo de datos creado como respuesta a unos requerimientos, genera una base de datos.

# Bases de datos:

* Es un conjunto de datos contenido y relaciones entre los contenidos. Los datos de una BBDD pueden ser:
  + **Datos de usuario:** información útil. (los insert)
  + **Datos de sistema**: Datos que necesita la BBDD para gestionar la información. (los metadatos)

# 1.2 Características de una BBDD

## Integridad:

Asegura que los datos que se encuentran almacenados en la BBDD sean válidos.

* + No puede haber información repetida.
  + No puede existir información incorrecta.

La integridad comprueba que los datos sean válidos según restricciones dominio validaciones, etc.

## Seguridad:

Acceso por usuarios autorizados en 3 niveles. La seguridad hace referencia también a la seguridad física: Protección contra los fuegos, los rayos, agua, ventilación.

* **Nivel 1:** Autenticación de usuario. (Acceso por Usuario y contraseña)
* **Nivel 2:** Cifrado de información. (Que la información este cifrada)
* **Nivel 3:** Control de datos íntegros, consistencia.

## Redundancia:

Es la repetición de la información en distintas tablas.

## Velocidad de acceso:

El tiempo de respuesta sea ágil a la información. Factores:

* Tipos de conexión, usuarios conectados.
* Estructura física de la BBDD.
* Estructura lógica de la BBDD.
* Orden de las consultas, gestión de índices, claves.

## Representación de la información.

* Interfaz de programación.
* Interfaz de usuario final.

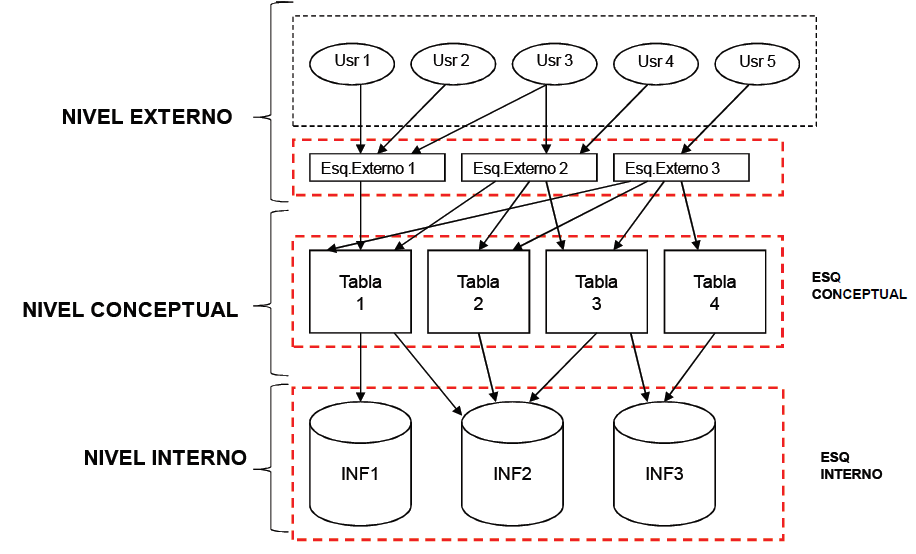
## Migración y compatibilidad

* **Migración (Importación):** capacidad BD de obtener datos generados/tratados con otros medios externos.
* **Compatibilidad:** la capacidad BD de exportar datos propios de la BD a un formato utilizable a un formato por otros medios o herramientas externas.

## Independencia de datos

* Física cambio en la estructura física de la aplicación BD
* Lógica.

# 1.3 Arquitectura de las BBDD



**Tema 3. Manejo de ficheros.**

# 3.1 Información sobre ficheros.

Un fichero se compone de información que tiene algún criterio de asociación. Es un recurso del sistema operativo que suele tener las siguientes primitivas:

* **Creación de fichero:** Crear un fichero con un nombre.
* **Apertura de un fichero:** El sistema operativo accede al fichero indicado y toma control para acceder a la información existente en el fichero.
* **Lectura de datos:** es la transferencia de la información desde el fichero a una memoria principal de un programa (variables).
* **Escritura:** es la transferencia de la información desde la memoria principal de un programa a un fichero.
* **Cierre del fichero:** El programa finaliza las operaciones sobre el fichero y libera el recurso para que el sistema operativo lo deje disponible.

Una aplicación puede realizar las siguientes operaciones sobre un fichero:

* **Altas:** Añadir información a un fichero.
* **Bajas:** eliminar información de un fichero.
* **Modificaciones:** modificar algún dato existente en el fichero.
* **Consultas:** buscar información en el fichero.

# 3.2 Formas de acceso a un fichero.

Hay dos formas de acceder a la información contenida en un fichero:

* **Acceso secuencial:** la información está escrita en un orden concreto. Para poder acceder a un dato, es necesario leer los datos anteriores hasta alcanzar el dato requerido.
  + **Altas:** Se añade al final del fichero
  + **Bajas:** Se copia el fichero.ext
  + **Modificar:** Se copia el fichero.ext
  + **Consultas:** Recorrer el fichero.
* **Acceso aleatorio:** la información está escrita sin ningún orden. En este caso es posible acceder a cualquier dato que se desee del fichero sin necesidad de leer los datos anteriores. Los datos están almacenados en unos registros establecidos que determinan la posición concreta de la información.
  + **Altas:** Posicionar al final del fichero y escribir.
  + **Bajas:** Se escriben los datos que se encuentran por delante del dato a borrar, compactando la información.
  + **Modificar:** Se modifica el registro respetando el tamaño establecido.
  + **Consultas:** Recorrer el fichero en el orden que se quiera.

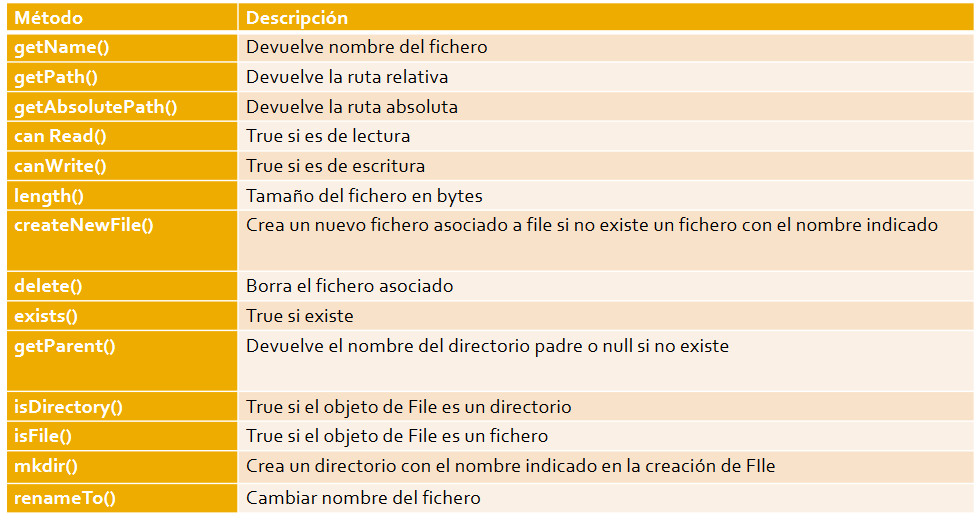
# 3.3 Entrada y salida en Java.

Hay un conjunto de clases que se encuentran en el paquete java.io (Input/output). Este paquete contiene la interfaz entre las aplicaciones y la información organizada por ficheros. Se compone de un conjunto de clases que se diferencian según sean ficheros de texto, binarios, con buffer, sin buffer, etc.

### 3.3.1 Gestión de ficheros.

Se encarga de manejar ficheros en el sistema operativo: creación, modificación, borrado, consulta, etc.

Se realiza con la clase ***File.*** Las operaciones de esta clase son:



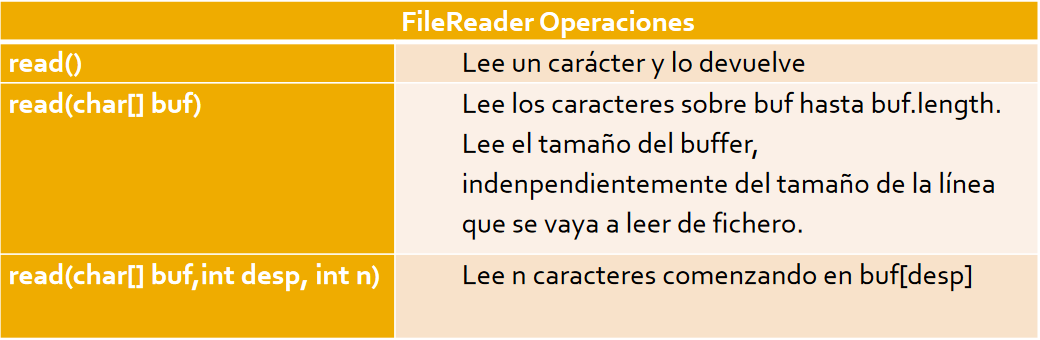
### 3.3.2 Ficheros de texto.

Los ficheros en java se tratan como flujos de información (**Stream**) que contienen los datos con la información a procesar.

Para manejar los ficheros de texto se emplean dos clases:

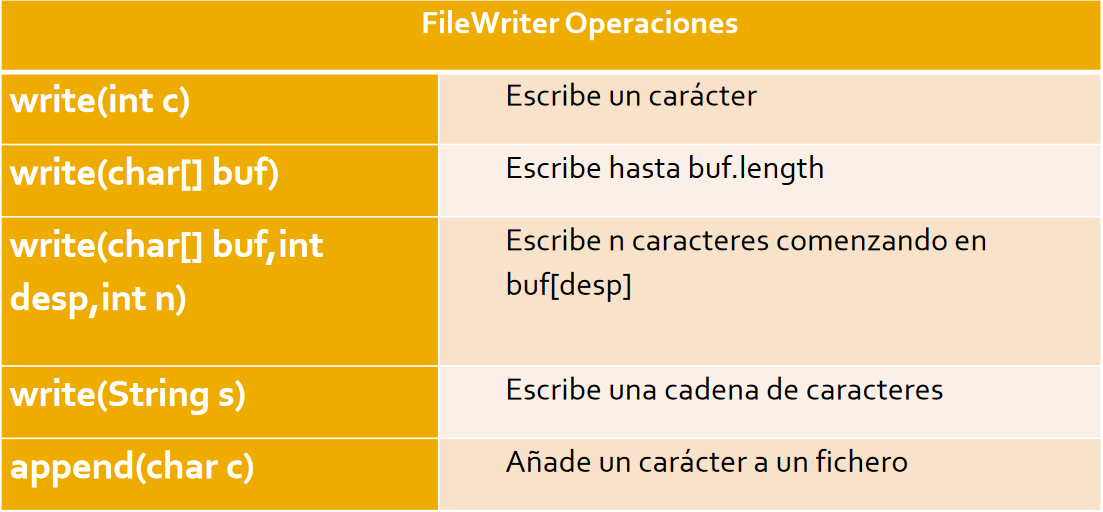
* **FileReader:** Ficheros de lectura.El modo de tratar un FileReader es el siguiente:
  1. **Apertura del fichero:**

1. Creación de un objeto File.
2. Creación de un objeto FileReader desde File.
   1. **Manejo de datos:**



* 1. **Cerrar el FileReader:**
     + - 1. Cerrar el FileReader (close)
* **FileWriter:** Ficheros de escritura.
  1. **Apertura del fichero:**

1. Creación de un objeto File.
2. Creación de un objeto FileWriter desde File.
   1. **Manejo de datos:**



* 1. **Cerrar el FileWriter (close):**
     + - 1. Cerrar el FileWriter (close)

# Tema 7 POO

La programación orientada a objetos (POO) permite cambiar el paradigma de programación, prestando atención en el objeto y sus relaciones con el resto.

 Cambia la filosofía respecto a la programación tradicional en la que el foco de desarrollo era la función.

 Ejemple POO – Tradicional (imprimir tabla)

**Conceptos de poo**

 Clase: Métodos y atributos de un mismo comportamiento.

 Objeto: Instancia de una clase

 Subclase: Clase que deriva de otra clase existente. La clase derivada hereda todos sus métodos y atributos.

 Herencia: Capacidad de recibir métodos y atributos de las clases superiores.

 Encapsulación: Ocultar datos y métodos, para hacer más sencilla la utilización. Permitiendo el acceso a la parte pública.

 Polimorfismo: Capacidad de un objeto de asumir diferentes formas según sea el objeto existente.

**Clasificación**

 Organización de elementos con significado común.

 Produciendo una relación directa con los objetos con los que nos rodean. En P.O.O. permite definir clases que contienen:

 Comportamiento: métodos (funciones)

 Atributos: datos (variables)

**Encapsulacion**

 Organizar diferentes elementos y dotarles de

la misma estructura.

 Se emplea para:

 Combinar métodos y datos en una clase 

Definición (class)

 Controlar el acceso los métodos y datos en una

clase  Control de acceso (public, private)

**Control de acceso**

 tipoAcceso

 public: Accesible por exterior e interior de la clase.

 private: Accesible por interior de la clase.

 protected: Accesible por interior de la clase y los hijos.

**Constructores**

 Clase: Clasificación de elementos que comparten métodos y atributos. Ej: Todas las personas tienen edad, nombre y pueden escribir su nombre.

 Objeto: Instanciación de un elemento de la clase. Ej: Puede haber 20 objetos de la clase persona con edades y nombres diferentes. Todos podrán escribir su nombre.

**Sobrecarga**

 Definición de varios métodos que tengan el mismo identificador, pero se diferencian por distintos parámetros: por número o por tipo diferente.

 Cuando se realza la llamada se determina el método según los parámetros.

 Cuando un mismo método tiene varias versiones con distintos parámetros se dice que está sobrecargado.