1. TEMA 1. INTRODUCCIÓN A BASES DE DATOS
   1. Introducción

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

* 1. Características de una BB.DD.
     1. Integridad: Se resuelve con transacciones.
        1. Asegura que los datos que se encuentren almacenados en una base de datos sean válidos:
           1. No puede aparecer información repetida
           2. No puede existir información incorrecta
           3. Comprueba que los datos sean válidos según restricciones de dominio o las validaciones etc…
     2. Seguridad:
     3. Redundancia: Repetición de información.
     4. Velocidad de acceso: Acceso ágil a la información. Factores:
        1. -----------------------------Falta aquí -----------------------------
     5. Representación de la información:
        1. Usuario final
        2. Interfaz de programación
     6. Migración y compatibilidad
        1. Migración (importación):
           1. Capacidad de la Base de datos de obtener datos generados/tratados con otros medios externos.
        2. Compatibilidad:
           1. La capacidad de la base de datos de exportar datos propios de la base de datos a un formato utilizable por otros medios/herramientas externos.
        3. Independencia de datos:
           1. Física:

Si se realiza un cambio en la estructura física de la aplicación de la base de datos, esta sigue funcionando igual.

* + - * 1. Lógica: Si se agrega alguna columna o dato nuevo, esta tiene que seguir funcionando sin problemas.

1. Tema 2
2. TEMA 3: MANEJO DE FICHEROS.
   1. Información sobre ficheros.

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

* + 1. Creación de un fichero: Crea un fichero con un nombre.
    2. Apertura de un fichero: El SO accede al fichero indicado y toma control para acceder a la información existente en el fichero.
    3. Lectura de datos: La transferencia de información desde el fichero 🡪 memoria principal de un programa (variables).
    4. Escritura de datos: Transferencia de información desde memoria de un programa 🡪 fichero.
    5. Cierre del fichero: El programa finaliza las operaciones sobre le fichero y libera el recurso para que le sistema operativo lo deje disponible.

Una aplicación, puede realizar las siguientes operaciones sobre un fichero:

1. Altas: Añadir información a un fichero.
2. Bajas: Eliminar información de un fichero.
3. Modificaciones: Modificar algún dato existente en el fichero.
4. Consultas: Buscar información en el fichero.
   1. Formas de acceso a un fichero.

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

* + 1. Acceso secuencial: la información en un fichero secuencial está escrita en un orden concreto. Para poder acceder a un dato, es necesario leer los datos anteriores hasta alcanzar el dato requerido. (no pueden leer hacia atrás, solo hacia adelante).
       1. Altas: Posicionar al final y escribir.
       2. Bajas: Se copia el fichero hasta el dato que quieres borrar, se salta el dato y se termina de copiar el resto, luego se borra el fichero anterior.
       3. Modificaciones: Igual que bajas, pero en lugar de ignorar el dato se copia el nuevo a modificar.
       4. Consulta: Recorrer el fichero de inicio a fin buscando el dato que se quiere. Si se quiere buscar uno anterior se cierra el fichero y se abre de nuevo.
    2. Acceso aleatorio: La información está escrita sin ningún orden. 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.
       1. Altas: Posicionar al final y escribir.
       2. Bajas: Se escriben los datos que se encuentran por delante del dato a borrar, compactando la información.
       3. Modificaciones: Se modifica el registro respetando el tamaño establecido.
       4. Consulta: Recorrer el fichero en el orden que se quiera.
  1. Entrada / 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 en ficheros. Se compone de un conjunto de clases que se diferencian según sean ficheros de texto, binarios, con buffer, sin buffer, etc.

* + 1. Gestión de ficheros. Se encarga de manejar ficheros en el SO: Creación, modificación, borrado, consulta, etc.
       1. Se realiza con la clase File. Las operaciones de esta clase son:
          1. getName() 🡪 Devuelve nombre del fichero.
          2. getPath() 🡪 Devuelve la ruta relativa
          3. getAbsolutePath() 🡪 Devuelve la ruta absoluta.
          4. Can Read() 🡪 True si es de lectura.
          5. canWrite() 🡪 True si es de escritura.
          6. Length() 🡪 Tamaño del fichero en bytes.
          7. createNewFile() 🡪 Crea un nuevo fichero asociado a file si no existe un fichero con el nombre indicado.
          8. Delete() 🡪 Borra el fichero asociado.
          9. Exists() 🡪 True si existe.
          10. getParemt() 🡪 Devuelve el nombre del directorio padre o null si no existe.
          11. isDirectory() 🡪 True si el objeto de file es un directorio
          12. isFile() 🡪 True si el objeto de file es un fichero.
          13. Mkdir() 🡪 Crea un directorio con el nombre indicado en la creación de File.
          14. renameTo() 🡪 Cambiar nombre del fichero.
    2. Ficheros de texto.

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

Para manejar los ficheros de texto se emplean dos clases:

* + - 1. FileReader: Ficheros de lectura 🡪 El modo de tratar un FileReader es:
         1. Apertura del fichero:

Creación de un objeto File.

Creación de un objeto FileReader desde file.

* + - * 1. Manejo de datos:

Int read() 🡪 Lee un solo carácter.

Read(char[] buf) 🡪 Lee hasta buf.length caracteres de datos de una tabla y los almacena.

Read(char[] buf, int desp, int n) 🡪 Lee n caracteres de datos de la tabla comenzando por el buf[desp] y devuelve el número de caracteres leído.

* + - * 1. Cerrar el fichero:

Cerrar el FileReader(close).

* + - 1. FileWriter: Ficheros de escritura 🡪 El modo de tratar un FileWriter es:
         1. Apertura del fichero:

Creación de un objeto File.

Creación de un objeto FileWriter desde file.

* + - * 1. Manejo de datos:

write(int c) 🡪 Escribe un carácter-

write(char[] buf) 🡪 Escribe hasta buf.length

write(char[] buf, int desp, int n) 🡪 Escribe n caracteres comenzando por el buf[desp].

write(String s) 🡪 Escribe una cadena de caracteres.

append(char c) 🡪 Añade un carácter a un fichero.

* + - * 1. Cerrar el fichero:

Cerrar el FileWriter (close).

1. T4
2. Tema 5