

## 1. Ficheros **secuenciales binarios**

### Byte a byte

FileOutputStream y FileInputStream

#### Escritura

##### Sintaxis:

```
FileOutputStream salida;  
salida= new FileOutputStream(nombre);  
  
FileOutputStream(File Objeto_File)  
  
FileOutputStream(String nombre_fichero, boolean añadir)
```

##### Excepciones que lanza el constructor:

FileNotFoundException

##### Métodos:

```
write(int i): IOException  
  
close(): IOException
```

#### Lectura

##### Sintaxis:

```
FileInputStream entrada;  
entrada=new FileInputStream(nombre);  
entrada=FileInputStream(objeto_File);
```

##### Excepciones que lanza el constructor:

FileNotFoundException

##### Métodos:

```
read(): IOException  
  
close(): IOException
```

---

## Datos pasados a byte

DataOutputStream y DataInputStream

### Escritura

#### Sintaxis:

```
DataOutputStream salida;  
salida=new DataOutputStream(new FileOutputStream(nombre));
```

```
DataOutputStream salida=new DataOutputStream  
    (new BufferedOutputStream  
    (new FileOutputStream(nombre)));
```

**Excepciones que lanza el constructor:** Las del FileOutputStream

#### Métodos:

```
writeInt(variable_tipo_entero)  
writeUTF(objeto_tipo_cadena)  
writeDouble(variable_tipo_doble)  
writeFloat(variable_tipo_float)  
writeChar(variable_tipo_carácter)  
writeBoolean, writeByte, writeLong, writeShort, etc.
```

**Excepciones que lanzan:** IOException

```
close():IOException
```

### Lectura

#### Sintaxis:

```
DataInputStream entrada;  
entrada=new DataInputStream(new FileInputStream(nombre));
```

```
DataInputStream entrada=new DataInputStream(new  
    BufferedInputStream(new FileInputStream(nombre)));
```

**Excepciones que lanza el constructor:** Las del FileInputStream

#### Métodos:

```
readChar(), readDouble(), readInt(), readFloat(), readUTF()  
readBoolean(), readByte(), readShort(), readLong(), etc.
```

**Excepciones que lanzan los métodos:** EOFException y IOException

```
close():IOException
```

---

## 2. Ficheros secuenciales de texto

### BufferedReader y PrintWriter

#### Escritura

##### Sintaxis:

```
PrintWriter salida;  
salida =new PrintWriter(new FileWriter(nombre))  
  
FileWriter (nombre, añadir)
```

**Excepciones que lanza el constructor FileWriter:** IOException

##### Métodos:

```
println()  
print()  
  
close():IOException
```

#### Lectura

##### Sintaxis:

```
BufferedReader entrada;  
entrada= new BufferedReader(new FileReader(nombre));
```

**Excepciones que lanza el constructor FileReader:** FileNotFoundException

##### Métodos:

```
readLine(): null  
read(): -1  
  
BufferedReader entrada=new BufferedReader(  
                                new FileReader("fichero.txt");  
char car;  
car=(char)(entrada.read());
```

**Excepciones que lanzan los métodos:** IOException

```
close():IOException
```

---

## Scanner y PrintWriter (a partir de la versión 5.0 de Java)

### Escritura

#### Sintaxis:

```
PrintWriter salida;  
salida = new PrintWriter(new FileWriter(nombre))
```

```
FileWriter (nombre, añadir)
```

**Excepciones que lanza el constructor FileWriter:** IOException

#### Métodos:

```
println()  
print()  
printf()  
  
close():IOException
```

### Lectura

#### Sintaxis:

```
Scanner entrada;  
entrada = new Scanner (new FileReader(nombre));    o  
entrada = new Scanner (new File(nombre));
```

**Excepciones que lanza el constructor:** FileNotFoundException

#### Métodos:

```
useLocale (Locale.US)  
next (), nextLine()  
nextInt(), nextDouble(), nextFloat(), ...  
hasNextInt (), hasNextDouble (), hasNextFloat (), ...  
close()
```

---

### 3. Ficheros de acceso directo

RandomAccessFile

#### Creación

```
RandomAccessFile(File objeto_fichero, String modo)
```

```
RandomAccessFile(String nombre, String modo)
```

modo: "r" (*read*) y "rw" (*read-write*)

**Excepciones que lanza el constructor:** FileNotFoundException

#### Métodos:

```
void seek(long posición)
```

```
long getFilePointer()
```

```
int skipBytes(int desplazamiento)
```

```
long length()
```

**Excepciones que lanzan los métodos:** IOException

```
close(): IOException
```

#### Escritura

```
RandomAccessFile salida;
```

```
salida=new RandomAccessFile(nombre,"rw");
```

#### Métodos:

```
writeInt(entero), writeDouble(doble), writeBytes(cadena)
```

```
writeUTF(String), etc
```

**Excepciones que lanzan los métodos:** IOException

```
close(): IOException
```

#### Lectura

```
RandomAccessFile entrada;
```

```
entrada=new RandomAccessFile(nombre, "r");
```

#### Métodos:

```
readInt(), readDouble(), readUTF(), readFloat(),readShort(), etc.
```

**Excepciones que lanzan los métodos:** EOFException y IOException

#### Movimiento en un fichero:

```
posicion=(n-1)*l_registro;
```

```
posicion=n*l_registro;
```

---

## 4. La clase File

```
File(String nombre)
File(String directorio, String nombre)
File(File directorio, String nombre)
```

```
File fichero=new File("datos.dat")
```

### Métodos:

```
exists()
getName()
length()
lastModified()
list()
delete()
```

## 5. Ficheros y objetos

```
class Ejemplo implements Serializable {
--- Código para la clase Ejemplo ---
}
```

### Creación de un *stream* de objetos para salida:

```
ObjectOutputStream salida;
salida=new ObjectOutputStream(new FileOutputStream(nombre));
```

### Creación de un *stream* de objetos para entrada:

```
ObjectInputStream entrada;
entrada=new ObjectInputStream(new FileInputStream(nombre));
```

### Métodos:

```
writeObject(Objeto)
readObject()
```

### Ejemplo:

```
ObjectOutputStream salida;
salida=new ObjectOutputStream
    (new FileOutputStream("datos.dat"));
salida.writeObject(obj1);
```

```
ObjectInputStream entrada;
entrada=new ObjectInputStream(new FileInputStream("datos.dat"));
obj2=(Ejemplo) entrada.readObject();
```