

UT 1 - Confección de interfaces de usuario

Persistencia de datos

INTERFACES DE USUARIO

1. JList y JTable

2. Persistencia de datos

1. Ficheros

2. BBDD

■ **BDR**

■ **BDOR**

■ **BDOO**

■ **SGBD Ntivos XML**

Persistencia:

JList

```
JList lista = new JList();  
lista.setModel(modelo);
```

```
modelo.clear(); // para borrar toda la lista
```

```
modelo.addElement("un item"); // añadir un item a la lista
```

```
modelo.removeElement(0); // Borra el item de la posición 0
```


Persistencia:

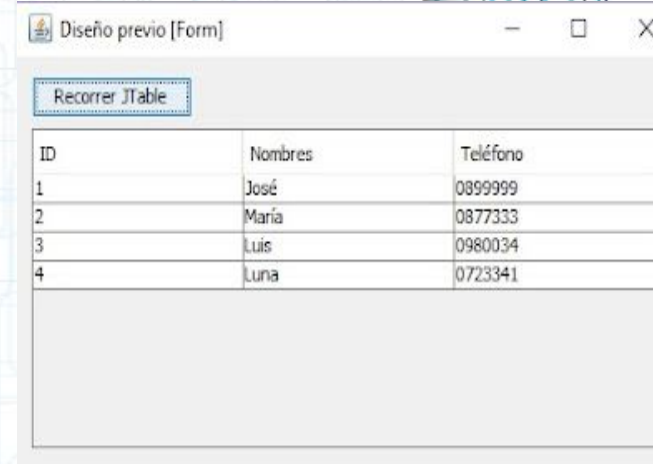
JList

```
MouseListener mouseListener = new MouseAdapter()  
{  
    public void mouseClicked(MouseEvent e)  
    {  
        if (e.getClickCount() == 2) // Se mira si es doble click  
        {  
            int posicion = list.locationToIndex(e.getPoint());  
            System.out.println("La posición es " + posicion);  
        }  
    }  
};
```

Persistencia: JTable

```
DefaultTableModel modelo = new DefaultTableModel();  
JTable tabla = new JTable(modelo);
```

```
modelo.addColumn("etiqueta columna 1");  
modelo.addColumn("etiqueta columna 2");
```



Persistencia:

JTable

```
Object [] fila = new Object[2];  
fila[0] = "dato columna 1";  
fila[1] = "dato columna 3";
```

```
modelo.addRow ( fila ); // Añade una fila al final
```

```
modelo.setValueAt ("nuevo valor", 0, 1); // Cambia fila 1, col 2.
```

```
modelo.removeRow (0); // Borra la primera fila
```

Persistencia:

JTable

```
public void mouseClicked(MouseEvent e)  
{  
int fila = tabla.rowAtPoint(e.getPoint());  
int columna = tabla.columnAtPoint(e.getPoint());  
if ((fila > -1) && (columna > -1))  
    System.out.println(modelo.getValueAt(fila,columna));  
}
```


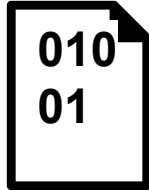
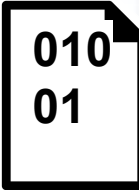

Persistencia:

JTable

```
public void mouseClicked(MouseEvent e)
{
    int fila = tabla.getSelectedRow();
    String columna;
    columna = tabla.getModel().getValueAt(fila, 0).toString();
}
```

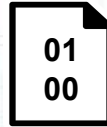
Persistencia:

Ficheros

- Puedo guardar “casa” (caracteres) → 
- Puedo guardar “0111” (bytes) → 
- Puedo guardar “Persona” (objetos) → 
(Serialización)

Persistencia: Ficheros

- Fichero de bytes “0111”
(bytes)

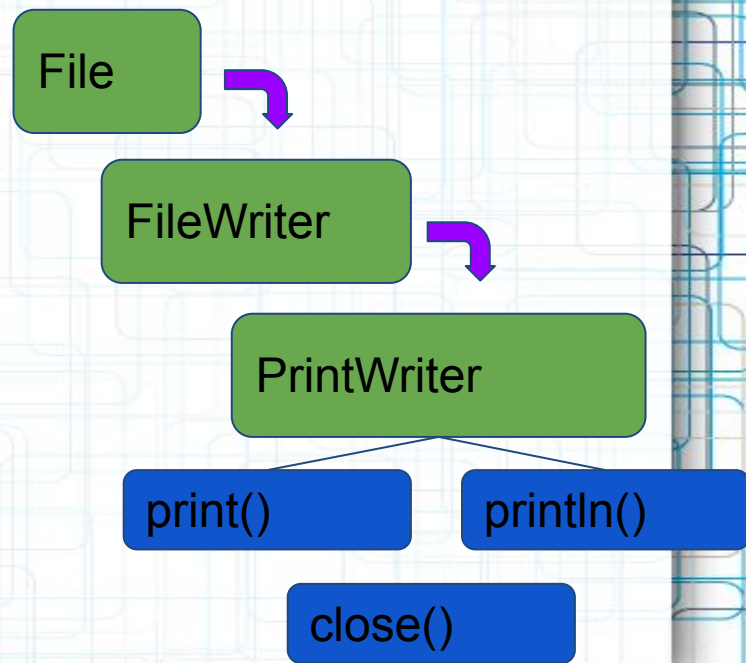
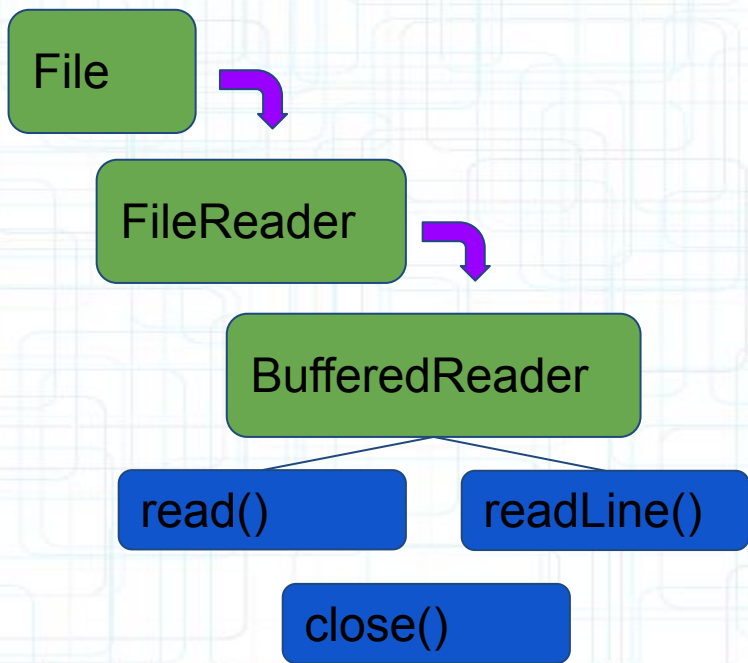


- Ficheros de Texto: Ficheros con caracteres



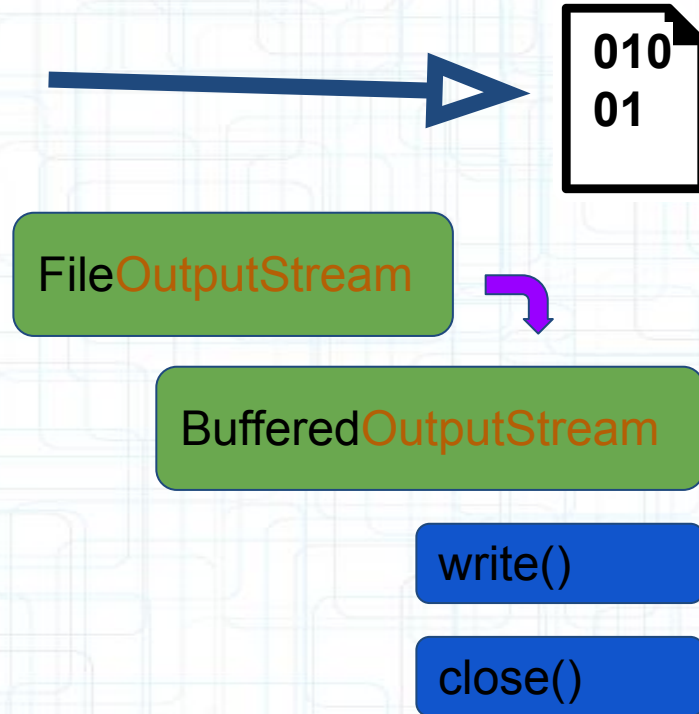
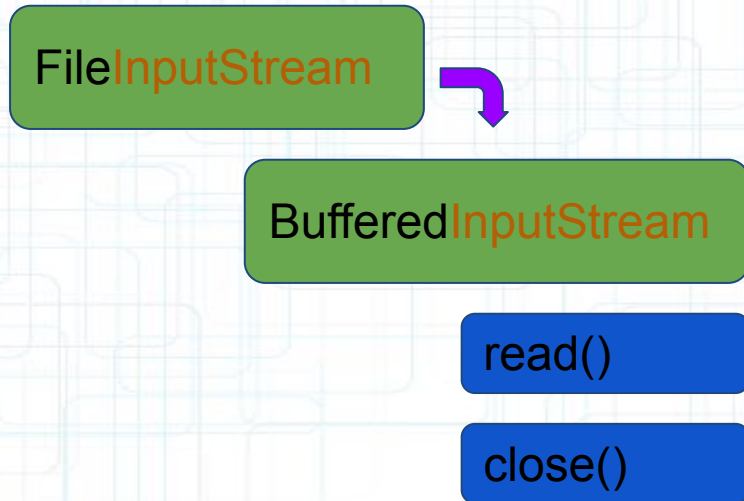
Persistencia: Ficheros

- Ficheros de texto




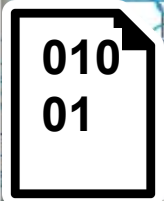
Persistencia: Ficheros

- Ficheros binarios



Persistencia: Ficheros

- Si guardo “Persona” (objetos) ...  (Serialización)



FileInputStream



ObjectInputStream

readObject()

close()

FileOutputStream



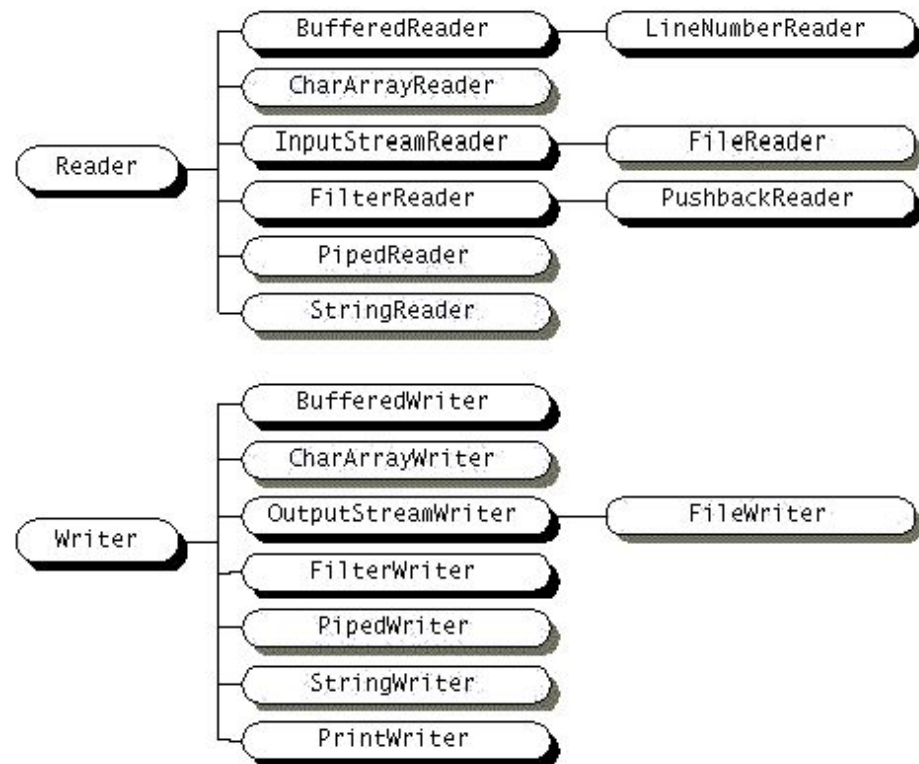
ObjectOutputStream

writeObject()

close()

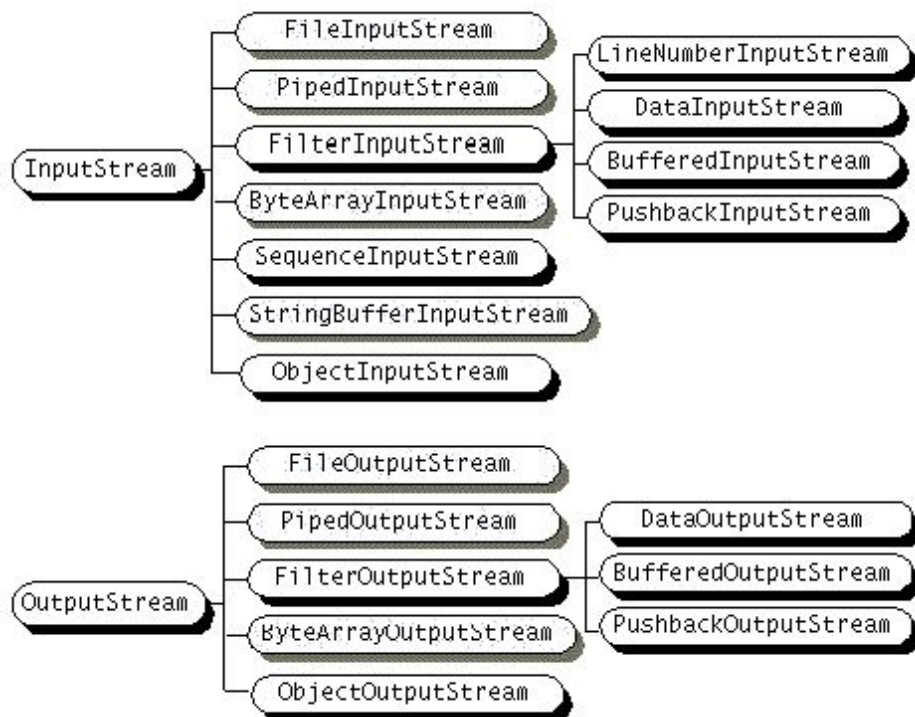
Persistencia: Ficheros

- Ficheros de texto



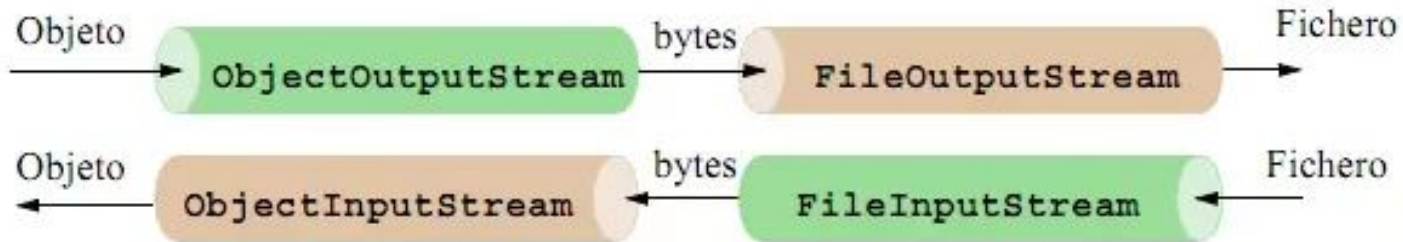
Persistencia: Ficheros

- Ficheros de bytes

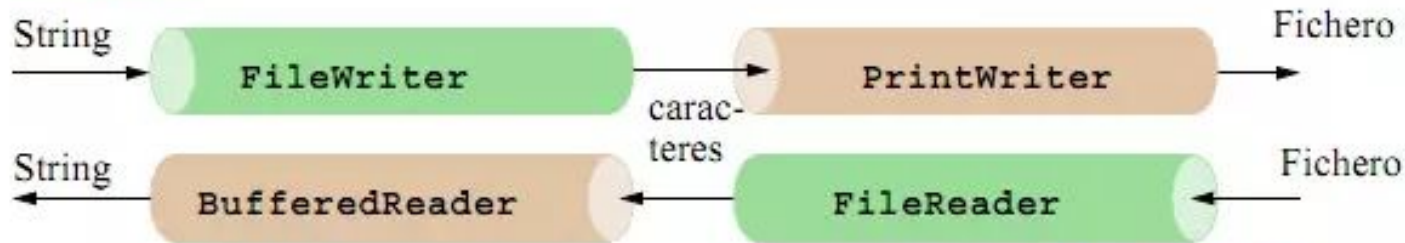


Persistencia: Ficheros

Binarios



De Texto:



1.- REPASO de la Gestión de ficheros en Java

- **Actividad** (busca la información para rellenar esta tabla):

Tipo de fichero	Com guardo los datos	Como sabemos que hemos acabado de leer
Ficheros texto	per characters	
	per linea	
Ficheros de bytes	per bytes	
	per objectes	

Persistencia: Ficheros

Para la gestión de fichero más o menos habéis trabajado bien pero he encontrado

FileReader fr = new FileReader(..)

fr.readLine();

PERO Recordar que cuando gestionamos ficheros debemos CERRAR el elemento que hemos usado *fr.close();*

```
File archivo = null;
FileReader fr = null;
BufferedReader br = null;
try {
    // Apertura del fichero y creacion de Buf
    // hacer una lectura por línea, disponer
    archivo = new File("fl.txt");
    fr = new FileReader(archivo);
    br = new BufferedReader(fr);

    // Lectura del fichero
    String linea;

    linea = br.readLine();
    while (linea != null) {

        System.out.println(linea);
        linea = br.readLine();
    }
    fr.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

Persistencia: Ficheros

**** recordar importante tambien
que hay que englobarlo todo con
`try{ ... }` y `catch(){ ...}`**

```
File archivo = null;
FileReader fr = null;
BufferedReader br = null;
try {
    // Apertura del fichero y creacion de Buf
    // hacer una lectura por linea, disponer
    archivo = new File("fl.txt");
    fr = new FileReader(archivo);
    br = new BufferedReader(fr);

    // Lectura del fichero
    String linea;

    linea = br.readLine();
    while (linea != null) {

        System.out.println(linea);
        linea = br.readLine();
    }
    fr.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

