

10.B. Flujos.

3. Flujos basados en bytes.

Este tipo de flujos es el idóneo para el manejo de entradas y salidas de bytes, y su uso por tanto está orientado a la lectura y escritura de datos binarios.

Para el tratamiento de los flujos de bytes, Java tiene dos clases abstractas que son `InputStream` y `OutputStream`. Cada una de estas clases abstractas tiene varias subclases concretas, que controlan las diferencias entre los distintos dispositivos de E/S que se pueden utilizar.

```
class FileInputStream extends InputStream {
```

```
    FileInputStream (String fichero) throws FileNotFoundException;
```

```
    FileInputStream (File fichero) throws FileNotFoundException;
```

```
    ... ..
```

```
}
```

```
class FileOutputStream extends OutputStream {
```

```
    FileOutputStream (String fichero) throws FileNotFoundException;
```

```
    FileOutputStream (File fichero) throws FileNotFoundException;
```

```
    ... ..
```

```
}
```

`OutputStream` y el `InputStream` y todas sus subclases, reciben en el constructor el objeto que representa el flujo de datos para el dispositivo de entrada o salida.

Por ejemplo, podemos copiar el contenido de un fichero en otro:

```
void copia (String origen, String destino) throws IOException {
    try{
        // Obtener los nombres de los ficheros de origen y destino
        // y abrir la conexión a los ficheros.
        InputStream fentrada = new FileInputStream(origen);
        OutputStream fsalida = new FileOutputStream(destino);
        // Crear una variable para leer el flujo de bytes del origen
        byte[] buffer= new byte[256];
        while (true) {
            // Leer el flujo de bytes
            int n = fentrada.read(buffer);
            // Si no queda nada por leer, salir del bucle
            if (n < 0)
                break;
            // Escribir el flujo de bytes leídos al fichero destino
            fsalida.write(buffer, 0, n);
        }
        // Cerrar los ficheros
        fentrada.close();
        fsalida.close();
    }catch(IOException ex){
        System.out.println(ex.getMessage());
    }
}
```

Mismo código copiable:

```
void copia (String origen, String destino) throws IOException {
```

```
    try{
```

```
// Obtener los nombres de los ficheros de origen y destino
```

```
// y abrir la conexión a los ficheros.
```

```
InputStream fentrada = new FileInputStream(origen);
```

```
OutputStream fsalida = new FileOutputStream(destino);
```

```
// Crear una variable para leer el flujo de bytes del origen
```

```
byte[] buffer= new byte[256];
```

```
while (true) {
```

```
    // Leer el flujo de bytes
```

```
    int n = fentrada.read(buffer);
```

```
    // Si no queda nada por leer, salir del while
```

```
    if (n < 0)
```

```
        break;
```

```
    // Escribir el flujo de bytes leídos al fichero destino
```

```
    fsalida.write(buffer, 0, n);
```

```
}
```

```
// Cerrar los ficheros
```

```
fentrada.close();
```

```
fsalida.close();
```

```
}catch(IOException ex){
```

```
    System.out.println(ex.getMessage()); }
```