

Ejercicio 01.11

El programa desarrollado tiene como objetivo leer el contenido de un fichero binario y copiar únicamente los caracteres imprimibles a un nuevo fichero de texto. De esta forma, se eliminan del resultado todos los bytes que no correspondan a caracteres visibles, es decir, aquellos cuyo valor ASCII esté fuera del rango comprendido entre 32 y 127.

Para poder probar el funcionamiento del programa, se ha creado una clase llamada FicheroBinario. En ella se genera un fichero binario de ejemplo llamado input.bin. Este fichero contiene una mezcla de bytes imprimibles, como letras y símbolos, y bytes no imprimibles, como saltos de línea, tabulaciones o valores fuera del rango ASCII visible. El método crearFicheroBinario() utiliza un FileOutputStream para escribir directamente en el fichero una secuencia de bytes definida en un array. Entre estos bytes se incluyen los caracteres A, B, C, !, @, #, junto con otros no imprimibles como el byte 200 o el valor nulo (0), con el fin de comprobar que el programa es capaz de filtrarlos correctamente.

```

1  import java.io.FileOutputStream;
2  import java.io.IOException;
3
4  public class FicheroBinario { 1 usage
5      public static void crearFicheroBinario() { 1 usage
6          String nombreFichero = "input.bin";
7
8          byte[] datos = {
9              65, 66, 67, // A B C
10             10, // salto de línea (no imprimible)
11             32, // espacio
12             33, 64, 35, // ! @ #
13             (byte)200, // byte no imprimible (>127)
14             68, 69, 70, // D E F
15             9, // tabulador (no imprimible)
16             97, 98, 99, // a b c
17             (byte)0, // null (no imprimible)
18             100, 101, 102 // d e f
19         };
20
21         try (FileOutputStream fos = new FileOutputStream(nombreFichero)) {
22             fos.write(datos);
23             System.out.println("Fichero binario '" + nombreFichero + "' creado.");
24         } catch (IOException e) {
25             e.printStackTrace();
26         }
27     }
28 }

```

La clase principal, Main, se encarga de leer el fichero binario generado y crear el nuevo fichero de salida en formato texto, denominado output.txt. Para ello, se utiliza un flujo de entrada (FileInputStream) y un escritor de texto (FileWriter). Mediante un bucle while, se van leyendo los bytes del fichero uno a uno. Cada byte leído se evalúa para comprobar si su

valor está entre 32 y 127. Si cumple esa condición, se escribe en el fichero de salida; en caso contrario, se ignora.

```
import java.io.*;

public class Main {
    public static void main(String[] args) {
        FicheroBinario.crearFicheroBinario();
        String inputFileName = "input.bin";
        String outputFileName = "output.txt";

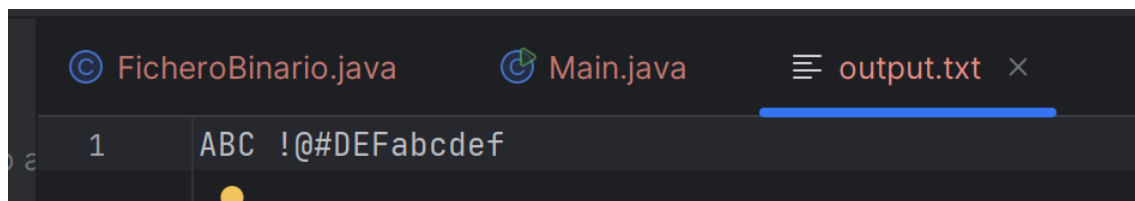
        try (FileInputStream fis = new FileInputStream(inputFileName);
            FileWriter fw = new FileWriter(outputFileName)) {

            int byteRead;
            while ((byteRead = fis.read()) != -1) {
                if (byteRead >= 32 && byteRead <= 127) {
                    fw.write(byteRead);
                }
            }
        }
    }
}
```

Una vez finalizado el proceso, el programa muestra un mensaje por consola indicando que el contenido imprimible se ha escrito correctamente. De este modo, en el fichero output.txt solo aparecen los caracteres legibles, como "ABC !@#DEFabcdef", mientras que los bytes no imprimibles han sido descartados.

```
System.out.println("El contenido imprimible se ha escrito correctamente a " + outputFileName);
} catch (IOException e) {
    System.err.println("Error al procesar los archivos: " + e.getMessage());
}
}
```

En conjunto, este programa demuestra el uso de flujos de entrada y salida en Java, así como el manejo de ficheros binarios y de texto. Además, sirve como ejercicio práctico para trabajar con el control de rangos de valores ASCII y la escritura selectiva de datos según determinadas condiciones.



The screenshot shows an IDE with three tabs: 'FicheroBinario.java', 'Main.java', and 'output.txt'. The 'output.txt' tab is active, showing the text 'ABC !@#DEFabcdef' on line 1. A yellow cursor is visible at the end of the line.