

Ejercicio 01.05.b

Este programa tiene como objetivo leer el contenido de un fichero de texto llamado *log.txt* y mostrarlo por consola línea a línea. La idea es que, si el archivo existe, se muestre todo su contenido tal cual esté escrito. En el caso de que *log.txt* no exista en el directorio de ejecución, el programa pide al usuario que introduzca otro nombre de fichero, y lo leerá siempre que efectivamente exista. Si tampoco se encuentra ese segundo fichero, el programa finaliza mostrando un aviso.

Para conseguirlo, el código utiliza las clases `File`, `Scanner`, `BufferedReader` y `FileReader`. La clase `File` se usa para representar el fichero en el sistema, mientras que `Scanner` permite interactuar con el usuario por consola. Por otro lado, `BufferedReader` junto con `FileReader` sirven para abrir y leer el archivo línea a línea, de forma eficiente.

El flujo del programa empieza creando un objeto `File` asociado a *log.txt*. A continuación, se comprueba si el fichero existe en el directorio. Si no existe, el programa avisa al usuario y le solicita que escriba el nombre de otro fichero. Ese nuevo nombre se guarda en la variable `fileName` y se crea un nuevo objeto `File` con él. En caso de que el segundo fichero tampoco exista, el programa muestra un mensaje indicando que no se ha encontrado y termina la ejecución sin continuar.

```
1  import java.io.*;
2  import java.util.Scanner;
3
4  public class main {
5      public static void main(String[] args) {
6          File file = new File("log.txt");
7          Scanner scanner = new Scanner(System.in);
8
9          try {
10             if (!file.exists()) {
11                 System.out.println("El fichero 'log.txt' no existe. Introduzca el nombre de otro fichero:");
12                 String fileName = scanner.nextLine();
13                 file = new File(fileName);
14
15                 if (!file.exists()) {
16                     System.out.println("El fichero especificado tampoco existe. Finalizando programa.");
17                     return;
18                 }
19             }
20         }
```

Si el fichero que se ha indicado (ya sea *log.txt* o el alternativo) existe, se procede a leerlo. Para ello se crea un `BufferedReader` que permite leer línea por línea. Cada línea se guarda en una variable temporal llamada `line` y se imprime en la consola hasta que no queden más líneas en el archivo. Al terminar, el lector se cierra para liberar los recursos.

Además, el programa incluye un bloque `try-catch` para capturar posibles excepciones de tipo `IOException` que puedan ocurrir durante la lectura del fichero, como problemas de permisos o errores de acceso. En caso de error, se muestra un mensaje explicando lo sucedido. Finalmente, el bloque `finally` asegura que el `Scanner` utilizado para leer la entrada del usuario se cierre correctamente, evitando fugas de recursos.

```
20
21     BufferedReader reader = new BufferedReader(new FileReader(file));
22     String line;
23     while ((line = reader.readLine()) != null) {
24         System.out.println(line);
25     }
26     reader.close();
27 } catch (IOException e) {
28     System.out.println("Se produjo un error al leer el fichero: " + e.getMessage());
29 } finally {
30     scanner.close();
31 }
32 }
33 }
```

En conclusión, este programa ofrece una forma sencilla y segura de mostrar por consola el contenido de un fichero de texto. Da prioridad a la lectura de *log.txt*, pero también contempla la posibilidad de que no exista y permite al usuario especificar otro archivo alternativo, mejorando así su flexibilidad y robustez.



```
Project Alt+1
  Ejercicio 01.05b C:\Users\anna
    .idea
    out
    src
  Run main x
  C:\Users\annaa\jdk-23\bin\java.exe -javaagent:C:\P
  Esto es un texto de prueba
  Esto es otro texto de prueba
  Process finished with exit code 0
```