Actividad Práctica - Tema 3: Lectura y Escritura Secuencial

Objetivo

Practicar la lectura y escritura de ficheros de texto utilizando `FileWriter`, `BufferedWriter`, `FileReader` y `BufferedReader`.

Escribir líneas en un archivo de texto.
Leer el contenido de ese archivo línea por línea.
Visualizar la salida por consola.

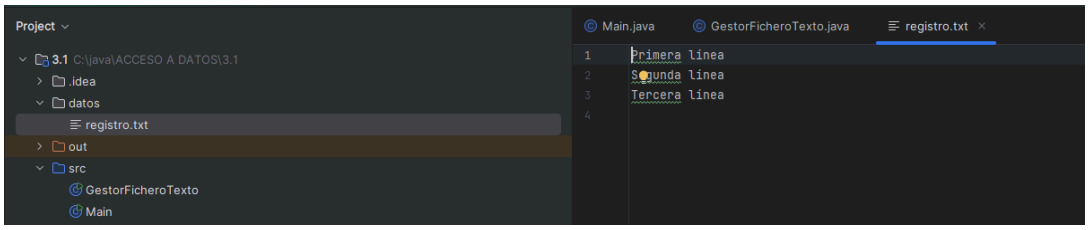
Instrucciones

1. Crea una carpeta llamada `datos` en la raíz del proyecto.
2. Crea una clase Java llamada `GestorFicheroTexto`.
3. Implementa los siguientes pasos en el método `main`:
   Escribe 3 líneas de texto en un archivo llamado `datos/registro.txt`. Lee el contenido de ese archivo y muéstralo por consola.
   Usa `BufferedWriter` y `BufferedReader`.

```
import java.io.*;



public class GestorFicheroTexto {

   public static void main(String[] args) {

       try {

// Escritura

           FileWriter fw = new
FileWriter("datos/registro.txt");

           BufferedWriter bw = new BufferedWriter(fw);

           bw.write("Registro 1");

           bw.newLine();

           bw.write("Registro 2");
```

```
            bw.newLine();
            bw.write("Registro 3");
            bw.newLine();
            bw.flush();
            bw.close();
            System.out.println("Archivo escrito con éxito.");
// Lectura
            FileReader fr = new
FileReader("datos/registro.txt");
            BufferedReader br = new BufferedReader(fr);

            String linea;
            System.out.println("Contenido del archivo:");
            while ((linea = br.readLine()) != null) {
                System.out.println("> " + linea);
            }
            br.close();
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
/*
Preguntas de reflexión con respuestas:
```

```
1. ¿Qué ocurre si se vuelve a ejecutar el programa sin cambiar el nombre del archivo?

  - Si vuelvo a ejecutar el programa, el archivo existente se sobrescribe y se pierden las líneas anteriores. Solo quedan "Registro 1", "Registro 2" y "Registro 3".



2. ¿Cómo podrías añadir texto sin borrar el contenido anterior?

  - Para añadir texto sin borrar lo que ya existe, puedo abrir el FileWriter en modo append:

    FileWriter fw = new FileWriter("datos/registro.txt", true);

    Esto agrega nuevas líneas al final del archivo sin borrar lo que ya estaba.



3. ¿Qué diferencias observas si eliminas el BufferedWriter y usas solo FileWriter?

  - Si uso solo FileWriter, el programa sigue funcionando, pero es menos eficiente porque cada escritura va directo al disco. BufferedWriter guarda los datos en un buffer y los escribe de golpe, además permite usar newLine() fácilmente.



4. ¿Por qué es importante cerrar los buffers después de usarlos?

  - Es importante cerrar los buffers para liberar los recursos del sistema y asegurarse de que odo lo que estaba en memoria se escriba realmente en el archivo. Si no se cierra, podríamos perder información.

*/
```


Project
3.1 C:\java\ACCESO A DATOS\3.1
.idea
datos
registro.txt
out
src
GestorFicheroTexto
Main
Main.java
GestorFicheroTexto.java
registro.txt
Primera linea
Segunda linea
Tercera linea