MEDAC
Instituto Oficial de Formación Profesional

PSP. Actividad 2, Tema 3. Gestión de procesos 3.1

Alejandro Cantos Molina

Práctica 2

/1. Aplicación LectorTexto

```java
import java.io.*;

/**
 * LectorTexto: lee 'entrada.txt' y envía su contenido a salida estándar.
 * Funciona de manera individual con archivo o en tubería con System.out.
 */
public class LectorTexto {
    public static void main(String[] args) {
        File archivo = new File( pathname: "entrada.txt");
        if (!archivo.exists()) {
            System.err.println("Error: 'entrada.txt' no encontrado.");
            return;
        }

        try (BufferedReader br = new BufferedReader(new FileReader(archivo))) {
            String linea;
            while ((linea = br.readLine()) != null) {
                System.out.println(linea); // salida estándar
            }
        } catch (IOException e) {
            System.err.println("Error leyendo 'entrada.txt': " + e.getMessage());
        }
    }
}
```

LectorTexto es una aplicación que lee un archivo de texto llamado **entrada.txt** línea por línea y envía cada línea a la salida estándar (**System.out**). Si el archivo no existe, muestra un mensaje de error y termina. La clase puede ejecutarse de manera individual leyendo directamente **entrada.txt** o integrarse en una tubería, de modo que su salida pueda ser procesada por otra aplicación, como FiltraLineas.

Para ejecutar el archivo individualmente, usaremos: "**java -cp target/classes org.example.LectorTexto"**, lo cual devolverá:

```
PS C:\Users\Usuario1\IdeaProjects\2DAM_Maven> java -cp target/classes org.example.LectorTexto
Hola Mundo.
Esta frase deberia aparecer porque tiene muchos caracteres.
Esta frase tambien tiene muchos caracteres.
Esta no.
```

/2. Aplicación FiltraLineas

```java
import java.io.*;

/**
 * FiltraLineas: lee de la entrada estándar o archivo redirigido.
 * Muestra solo líneas con más de 20 caracteres.
 */
public class FiltraLineas {
    public static void main(String[] args) {
        try (BufferedReader br = args.length > 0
                ? new BufferedReader(new FileReader(args[0]))  // lectura de archivo opcional
                : new BufferedReader(new InputStreamReader(System.in))) {

            String linea;
            while ((linea = br.readLine()) != null) {
                if (linea.length() > 20) {
                    System.out.println(linea); // salida estándar
                }
            }

        } catch (IOException e) {
            System.err.println("Error leyendo la entrada: " + e.getMessage());
        }
    }
}
```

FiltraLineas lee líneas de texto desde la entrada estándar (**System.in**) o desde un archivo si se le pasa como argumento. Su función es mostrar únicamente las líneas que contengan más de 20 caracteres. Puede ejecutarse de manera independiente pasando un archivo, o integrarse en una tubería para procesar la salida de LectorTexto y enviar el resultado a otra aplicación, como ContadorPalabras.

Para ejecutar el archivo individualmente, usaremos: "**java -cp target/classes org.example.FiltraLineas salida.txt"**, lo cual devolverá:

```
PS C:\Users\Usuario1\IdeaProjects\2DAM_Maven> java -cp target/classes or
g.example.FiltraLineas salida.txt
Esta frase deberia aparecer porque tiene muchos caracteres.
Esta frase tambien tiene muchos caracteres.
```

/3. Aplicación ContadorPalabras

ContadorPalabras lee líneas de texto desde la entrada estándar (**System.in**) o desde un archivo proporcionado como argumento y calcula el número total de palabras, considerando como palabra cualquier secuencia separada por espacios. Muestra el resultado en la consola. La aplicación puede ejecutarse individualmente usando un archivo como entrada o integrarse en una tubería para contar palabras de la salida filtrada por FiltraLineas.

Para ejecutar el archivo individualmente, usaremos: "**java -cp target/classes org.example.ContadorPalabras filtradas.txt"**, lo cual devolverá:

```
PS C:\Users\Usuario1\IdeaProjects\2DAM_Maven> java -cp target/classes or
g.example.ContadorPalabras filtradas.txt
Número total de palabras: 14
```

/4. Uso de la tubería "|"

Por último, también podremos ejecutar todos los archivos en tubería mediante: "**java -cp target/classes org.example.LectorTexto | java -cp target/classes org.example.FiltraLineas | java -cp target/classes org.example.ContadorPalabras**" lo cual, leerá **entrada.txt**, filtrará las frases largas en **filtradas.txt** y finalmente ContadorPalabas devolverá cuántas palabras hay en las frases largas.

```
PS C:\Users\Usuario1\IdeaProjects\2DAM_Maven> java -cp target/classes or
g.example.LectorTexto | java -cp target/classes org.example.FiltraLineas
 | java -cp target/classes org.example.ContadorPalabras
Número total de palabras: 14
PS C:\Users\Usuario1\IdeaProjects\2DAM_Maven>
```