



Gestión de procesos 3.1

Hecho por: Virgilio Jesus
Dominguez Gonzalez



Índice

1. Introducción	2
2. Requisitos previos	3
3. Ejecución de cada aplicación por separado	4
3.1 lectorTexto	4
3.2 filtraLineas	4
3.3 contadorPalabras	5
4. Ejecución encadenada con tuberías	5
Explicación:	6
5. Conclusión	6



1. Introducción

El presente manual describe el funcionamiento de tres aplicaciones en Java:

- **lectorTexto:** Lee un archivo de texto y envía su contenido a la salida estándar.
- **filtraLineas:** Filtra las líneas de entrada que tenga más de 20 caracteres.
- **contadorPalabras:** Cuenta el número total de palabras en las líneas recibidas por la entrada estándar.

También se explica cómo ejecutar cada programa de forma independiente y cómo combinarlos mediante tuberías (|) en consola, emulando un procesamiento encadenado de datos.

2. Requisitos previos

Antes de ejecutar los programas es necesario:

1. Tener instalado **Java JDK 17 o superior** y configurada la variable de entorno **PATH**.

Comprobar con:

```
java -version
```



2. Haber generado el archivo `.jar` mediante Maven. En este caso, el comando usado en IntelliJ (o en consola) es:

```
mvn clean package
```

Esto crea un archivo en la carpeta `target/`, por ejemplo:

```
target/miapp.jar
```

3. Tener un archivo de prueba llamado `entrada.txt` en el directorio del proyecto, con contenido de varias líneas y que este dentro de un directorio "Resources".

3. Ejecución de cada aplicación por separado

3.1 lectorTexto

Lee el archivo `entrada.txt` y muestra su contenido en consola.

Comando:

```
java -jar lectorTexto.jar
```

Ejemplo esperado:

```
C:\Users\Usuario1\.jdk\openjdk-25\bin\java.exe -Dfile.encoding=UTF-8 -Dsur
Hola mundo
Esta es una línea bastante larga que debería pasar el filtro
Otra línea corta
Un texto adicional con varias palabras y más de veinte caracteres

Process finished with exit code 0
```



3.2 filtraLineas

Filtra las líneas recibidas desde la entrada estándar, mostrando únicamente aquellas que superen los 20 caracteres.

Comando (introduciendo texto manualmente):

```
java -jar filtrarLineas.jar
```

Ejemplo de entrada:

```
No debe de devolver ningún dato
```

3.3 contadorPalabras

Cuenta el número total de palabras en las líneas recibidas por la entrada estándar.

Comando (entrada manual):

```
java -jar contadorPalabras.jar
```

Ejemplo de entrada:

```
No debe de devolver ningún dato o devuelve 0
```




4. Ejecución encadenada con tuberías

Los tres programas pueden conectarse mediante el operador `|` de la siguiente forma:

```
java -jar lectorTexto | java -jar filtrarLineas.jar | java -jar  
contadorPalabras.jar
```

Explicación:

1. **lectorTexto** lee el archivo `entrada.txt` y manda cada línea a la salida estándar.
2. **filtrarLineas** recibe esas líneas y descarta las que tengan menos de 20 caracteres.
3. **contadorPalabras** recibe solo las líneas filtradas y cuenta el total de palabras.

Ejemplo esperado (suponiendo `archivo.txt` con 5 líneas, de las cuales 2 tienen más de 20 caracteres):

```
Total de palabras: 22
```

5. Conclusión

Este ejercicio demuestra cómo trabajar con **entrada y salida estándar en Java**, permitiendo construir programas que, aunque simples por separado, pueden combinarse mediante **tuberías** para realizar procesos más complejos.

La metodología es equivalente al uso de comandos en Linux como `cat`, `grep` y `wc`, pero implementada íntegramente en Java.