MEDAC
Instituto Oficial de Formación Profesional

PSP. Actividad Tema 3: Gestión de Procesos 1

Alejandro Cantos Molina

Manual: Cómo Probar la Ejecución de las Aplicaciones Java

Esta guía detalla los pasos para compilar, empaquetar y ejecutar dos programas Java: uno que genera números aleatorios (CrearNumeros) y otro que lee, ordena e imprime esos números (OrdenarNumeros). Lo esencial es la prueba que utiliza el **operador tubería (|)** para conectar la salida del primer programa a la entrada del segundo.

/1. Contexto de las Aplicaciones

El manual asume que los dos fragmentos de código Java del documento han sido guardados en archivos separados, compilados y empaquetados como archivos JAR ejecutables.

Aplicación	Código Fuente	Propósito
CrearNumeros ('aleatorios')	Página 3	Genera una cantidad aleatoria de enteros (entre 40 y 139) y los imprime en la salida estándar.
Ordenariumeros ('ordenarNumeros')	Página 2	Lee enteros desde la entrada estándar hasta el final del flujo, los almacena, los ordena y los imprime.

Nomenclatura: Utilizaremos los nombres de archivo **CrearNumeros.jar** y **Ordenariumeros.jar**, como se indica en la línea de comando de la página 3.

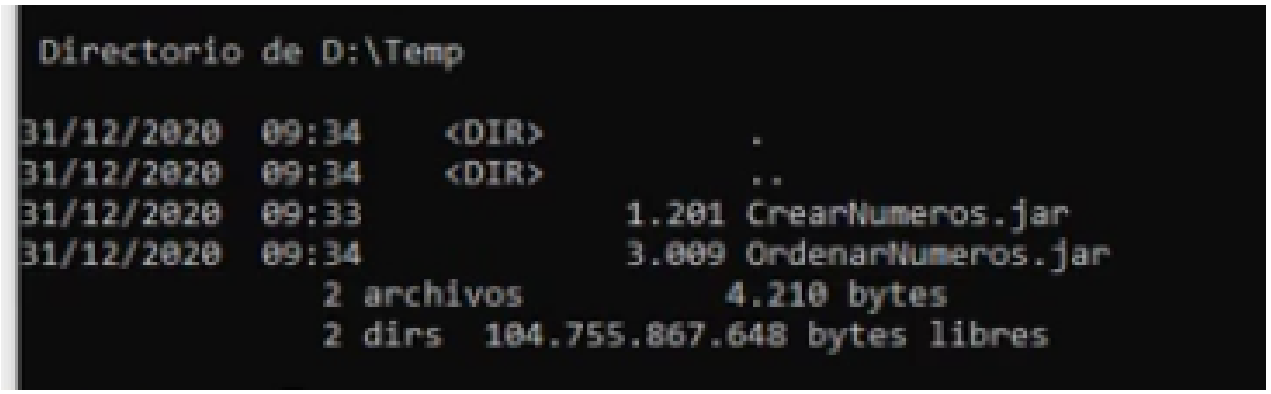
/2. Preparación de los Archivos JAR

Antes de la prueba, debes haber compilado el código (con javac) y empaquetado cada aplicación en su propio archivo JAR ejecutable (con el comando jar).

/2.1: Compilación y Empaquetado

Asegúrate de tener los dos archivos JAR listos para ejecutar.

- **CrearNumeros.jar**: Contiene la clase principal que genera los números aleatorios (el código de la página 3).
- **Ordenariumeros.jar**: Contiene la clase principal que lee y ordena los números (el código de la página 2).



/3. Prueba con el Operador Tubería (|)

El operador **tubería** (|) es la clave para esta prueba. En la línea de comandos, redirige la **Salida Estándar (STDOUT)** del programa de la izquierda para que se convierta en la **Entrada Estándar (STDIN)** del programa de la derecha.

/3.1: Ejecutar el Comando con Tubería

Abre la terminal o símbolo del sistema y ejecuta la siguiente instrucción:

java -jar CrearNumeros.jar | java -jar Ordenariumeros.jar

Explicación del Proceso:

1. **java -jar CrearNumeros.jar**: Se inicia el primer programa. Este genera una serie de números aleatorios y los envía a su salida estándar.

```
public class Main {

    public static void main (String[] args) {

        ArrayList<Integer> coleccion = new ArrayList<>();
        InputStreamReader entrada = null;
        BufferedReader br = null;

        try {

            entrada = new InputStreamReader(System.in);
            br = new BufferedReader(entrada);

            String linea;
            while ((linea = br.readLine()) != null) {
                coleccion.add(Integer.parseInt(linea));
            }

        } catch (IOException | NumberFormatException e) {
            System.out.println("Error al procesar los datos\n" +
                    e.getMessage());
        }finally{
            if(entrada != null){
                try {
                    entrada.close();
                } catch (Exception e) {
                    System.out.println("No se ha podido cerrar la entrada\n" +
                            e.getMessage());


                }
            }
        }

        Collections.sort(coleccion);

        for(int i : coleccion){
            System.out.println(" " + i);
        }

    }
}
```

2. **| (Tubería)**: Intercepta la salida de CrearNumeros.jar.
3. **java -jar Ordenariumeros.jar**: Se inicia el segundo programa. Este lee los números interceptados a través de su entrada estándar (la línea de código br.readLine()), los almacena en un ArrayList , los ordena (Collections.sort(coleccion);) y luego debería imprimirlos ordenados.

```java
package org.example;

To Run code, press Mayús F10 or click the ▷ icon in the gutter.
public class Main {
    public static void main(String[] args) {
            int cantidad;
            cantidad = (int) (Math.random() * 100) + 40;

            for (int i = 0; i <= cantidad; i++) {
                int j = (int) (Math.random() * 100);
                System.out.println(j);
            }
    }
}
```

/3.2: Verificar la Salida

La terminal mostrará la lista de números generados, pero **ordenados de forma ascendente**, confirmando que ambos programas se comunicaron correctamente.