



# COMUNICACIÓN ENTRE PROCESOS: SUMADORES CON FICHEROS

**MARIO JIMÉNEZ MARSET**

**ÍNDICE**

1. ENUNCIADO - OBJETIVOS.....	3
2. DESARROLLO .....	3

## 1. ENUNCIADO - OBJETIVOS

En esta práctica se pedía, a partir del código adjuntado, entregar este mismo comentado, con la realización de pruebas que prueben la comprensión de la programación multiproceso.

El código está compuesto de una clase Sumador y una clase Main.

## 2. DESARROLLO

En primer lugar, se adjunta el código de las dos clases anteriormente mencionadas.

```
package multiproceso;

public class Sumador {

    public static int sumar(int n1, int n2) {

        int suma = 0;
        if (n1 > n2) {
            int aux = n1;
            n1 = n2;
            n2 = aux;
        }
        for (int i = n1; i <= n2; i++) {
            suma = suma + i;
        }
        return suma;
    }

    public static void main(String[] args) {
        int n1 = Integer.parseInt(args[0]);
        int n2 = Integer.parseInt(args[1]);
        int suma = sumar(n1, n2);
        System.out.println(suma);
        System.out.flush();
    }
}
```

Se muestra la clase Sumador. En este programa se realiza un método sumar, el cual pasa por parámetros dos enteros, a los cuales se les hace una condición para que, si el primer número es mayor que el segundo, estos se intercambien y el primer número sea el menor de los dos.

Se hace un for para realizar la suma y, finalmente, el método retorna esta suma.

En el main simplemente se llama por argumentos a los dos valores de la suma, se crea una variable para llamar al método anterior y se imprime la suma (el resultado).

```
package multiproceso;

import java.io.*;

public class Main {

    static final int NUM_PROCESOS = 4;
    static final String PREFIJO_FICHEROS ="fich";

    public static void lanzarSumador(int n1, int n2, String fichResultados) throws
    IOException {
        String comando;
        comando = "multiproceso.Sumador";
        File directorioSumador;
        directorioSumador = new File("bin\\");
        File fichResultado = new File(fichResultados);
        ProcessBuilder pb;
        pb = new ProcessBuilder("java", comando, String.valueOf(n1),
        String.valueOf(n2));
        pb.redirectError(new File ("error.txt"));
        pb.directory(directorioSumador);
        pb.redirectOutput(fichResultado);
        pb.start();
    }

    public static int getResultadoFichero(String nombreFichero) {
        int suma = 0;
        try {
            FileInputStream fichero = new FileInputStream(nombreFichero);
            InputStreamReader fir = new InputStreamReader(fichero);
            BufferedReader br = new BufferedReader(fir);
            String linea = br.readLine();
            br.close();
            suma = Integer.valueOf(linea);
            return suma;
        } catch (FileNotFoundException e) {
            System.out.println("No se pudo abrir " + nombreFichero);
        } catch (IOException e) {
            System.out.println("No hay nada en " + nombreFichero);
        }
        return suma;
    }

    public static int getSumaTotal(int numFicheros) {
        int sumaTotal = 0;
        for (int i = 1; i <= NUM_PROCESOS; i++) {
            sumaTotal += getResultadoFichero("Ficheros\\"+PREFIJO_FICHEROS
            + String.valueOf(i));
        }
        return sumaTotal;
    }
}
```

```

public static void main(String[] args) throws IOException, InterruptedException {
    int n1 = Integer.parseInt(args[0]);
    int n2 = Integer.parseInt(args[1]);
    int salto = (n2 / NUM_PROCESOS) - 1;
    for (int i = 1; i <= NUM_PROCESOS; i++) {
        System.out.println("n1:" + n1);
        int resultadoSumaConSalto = n1 + salto;
        System.out.println("n2:" + resultadoSumaConSalto);
        lanzarSumador(n1, n1 + salto, "Ficheros\\"+PREFIJO_FICHEROS +
            String.valueOf(i));
        n1 = n1 + salto + 1;
        System.out.println("Suma lanzada...");
    }
    Thread.sleep(200);
    int sumaTotal = getSumaTotal(NUM_PROCESOS);
    System.out.println("La suma total es: " + sumaTotal);

    try {
        File f = new File("C:\\Users\\jimen\\OneDrive\\Documentos\\programa.txt");
        if (f.createNewFile()) {
            System.out.println("File created");
        }
        else {
            System.out.println("File already exists");
        }
        f.delete();
    }
    catch (Exception e) {
        System.err.println(e);
    }
}

```

Este programa tiene como objetivo lanzar el proceso de sumar a partir de otra clase primero, se declaran dos variables estáticas: la primera, recogerá el número de procesos; la segunda, qué nombre común tendrán los ficheros.

Se crea un método que lanza el sumador: se pasa por parámetros los dos números de la suma en cuestión y un String que será un file que recoja los resultados.

Se crea un String comando que llamará a la clase Sumador a través del package (que es el mismo para el Sumador y el Main)

Se crea un directorioSumador, de tipo File, que llamará a la carpeta bin (donde están los .class)

Se crea el fichResultado, de tipo File, que coge por parámetro el String anterior que recogía los resultados

Se crea un objeto del tipo ProcessBuilder, pasando "java", el comando y los dos números de la suma (pasándolos como String)

Se redirige el contenido del fichero y comienza el proceso.

Se crea un método que consigue introducir el resultado en el fichero: se pasa por parámetro un String nombreFichero

Se crea un FileInputStream, pasando por parámetro nombreFichero

Lo mismo ocurre con el `InputStreamReader` (para leerlo) y el `BufferedReader`, linealmente.

Lo que haya dentro, se almacena en el `String` `linea`, el cual se convertirá a `Integer` con la variable `suma`.

Este método devuelve la suma; también se manejan excepciones, por si algún paso está mal hecho.

Este método devuelve la suma total hecha entre todos los ficheros que intervengan en el programa.

Se declara una variable `sumaTotal` que, con un bucle `for`, será igual a lo que devuelva el método anterior `getResultadoFichero`.

El `main` del programa recibirá por argumentos dos valores y, a través de un algoritmo, se realiza la suma de los valores comprendidos entre estos dos parámetros.

Con un `for`, por cada fichero, se realiza la suma con salto, la cual se consigue llamando al `n1` y sumándole el salto.

Se lanza el sumador con este último resultado, además de la ruta y de llamar a los ficheros a los que hacerles la suma total.

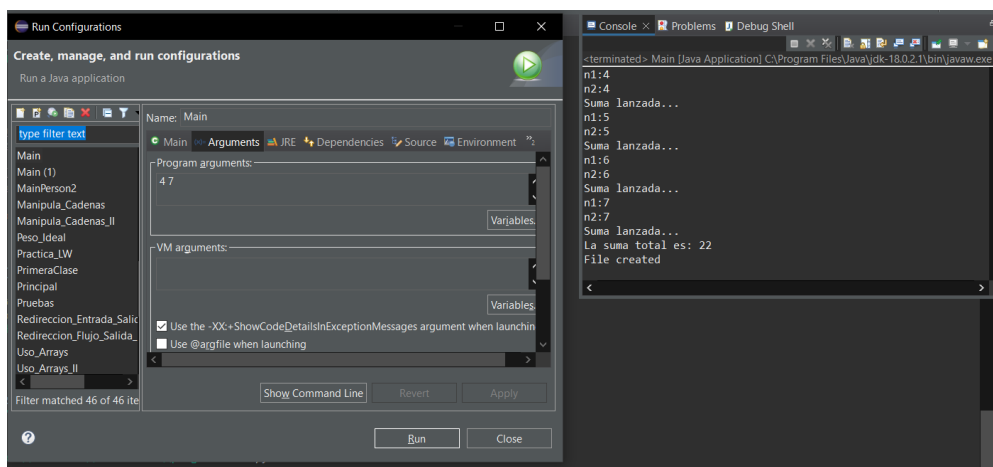
Se termina imprimiendo la `sumaTotal`, la cual se consigue finalmente llamando al método `getSumaTotal`.

Finalmente, se hace un `try catch` donde se crea un fichero de tipo `File` y un condicional que controla si se ha creado o ya existe el fichero.

Para poder sobrescribir los ficheros creados anteriormente, a este último fichero se le aplica el método `delete`, el cual borra este fichero.

Cada vez que se ejecute el `main`, se crea el fichero y automáticamente se borra, para poder dar paso a una nueva ejecución con argumentos diferentes.

Explicado el programa, lo que hace, sus métodos... se adjunta una imagen del resultado por consola y, ese resultado, influenciado por los argumentos puestos para su ejecución.



El directorio Ficheros para almacenar los ficheros se ha creado dentro del workspace, del Java Project correspondiente a la creación de las clases, pero se podría haber creado en cualquier otro lugar.

Organizar

Nuevo

Abrir

Seleccionar

(C:) > Usuarios > jimen > eclipse-workspace > ServiciosProcesos > Ficheros

Buscar en Ficheros

Nombre	Fecha de modificación	Tipo	Tamaño
fich1	13/10/2022 10:38	Archivo	1 KB
fich2	13/10/2022 10:38	Archivo	1 KB
fich3	13/10/2022 10:38	Archivo	1 KB
fich4	13/10/2022 10:38	Archivo	1 KB