

PROGRAMACIÓN SERVICIOS Y PROCESOS

Act 01. Programación Multiproceso

Tomás Pérez Carrillo

Notas previas antes de revisar la tarea:

A lo largo del desarrollo del código pueden aparecer comentarios que no siguen ninguna metodología específica. Se han utilizado a modo de guía personal y no deben ser considerados.

EJERCICIO 1.

Clase 1:

```
package Ejercicio1;

public class MiniCalc {  Ⓜ TomásPérezClaseDAM *

    public static void main(String[] args) {  Ⓜ TomásPérezClaseDAM *

        int numero1 = Integer.parseInt(args[0]);
        int numero2 = Integer.parseInt(args[1]);
        char operador = args[2].charAt(0);

        int resultado = 0;

        switch (operador) {
            case '+':
                resultado = numero1 + numero2;
                break;
            case '-':
                resultado = numero1 - numero2;
                break;
            case '*':
                resultado = numero1 * numero2;
                break;
            case '/':
                resultado = numero1 / numero2;
                break;
        }

        System.out.println("Resultado de minicalc: " + resultado);

    }
}
```

Clase 2:

```
package Ejercicio1;

import ...

public class MiniCalcMain { @TomásPérezClaseDAM *

    public static void main(String[] args) { @TomásPérezClaseDAM *
        Scanner sc = new Scanner(System.in);

        System.out.println("Introduce el valor 1: ");
        int valor1 = sc.nextInt();
        String valor1String = String.valueOf(valor1);

        System.out.println("Introduce el valor 2: ");
        int valor2 = sc.nextInt();
        String valor2String = String.valueOf(valor2);

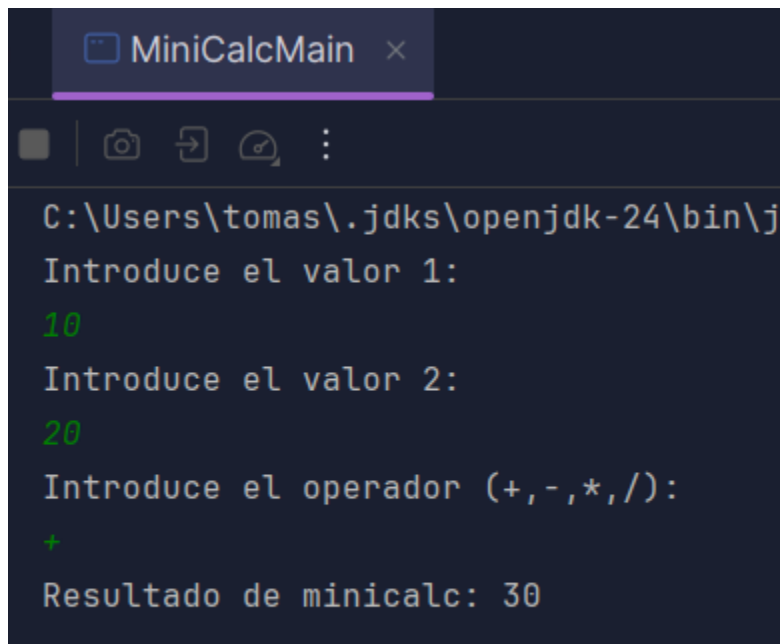
        System.out.println("Introduce el operador (+,-,*,/): ");
        char operador = sc.next().charAt(0);
        String operadorString = String.valueOf(operador);

        String claseEje = "src/Ejercicio1/MiniCalc.java";

        ProcessBuilder pb = new ProcessBuilder(...command: "java", claseEje, valor1String, valor2String, operadorString);

        try {
            pb.inheritIO();
            Process p = pb.start();
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }
}
```

Salida por terminal:



```
C:\Users\tomas\.jdk\openjdk-24\bin\j
Introduce el valor 1:
10
Introduce el valor 2:
20
Introduce el operador (+,-,*,/):
+
Resultado de minicalc: 30
```

```
Introduce el valor 1:  
15  
Introduce el valor 2:  
20  
Introduce el operador (+,-,*,/):  
-  
Resultado de minicalc: -5
```

```
Introduce el valor 1:  
5  
Introduce el valor 2:  
5  
Introduce el operador (+,-,*,/):  
*  
Resultado de minicalc: 25
```

```
Introduce el valor 1:  
5  
Introduce el valor 2:  
5  
Introduce el operador (+,-,*,/):  
/  
Resultado de minicalc: 1
```

Observaciones:

Este ejercicio no merece mucho comentario al respecto sobre su funcionamiento. En clase te pregunté si había que tratar el error de que el usuario ponga letras en vez de números o equivocarse al introducir el operador y dijiste que no por lo que se queda en un ejercicio algo más simple.

Ejercicio 2:

Clase 1:

```
public class App1 {  @TomásPérezClaseDAM *

    public static void main(String[] args) {  @TomásPérezClaseDAM *
        Scanner sc = new Scanner(System.in);

        System.out.println("¿Cuántas palabras quieres introducir?");
        int n = sc.nextInt();
        sc.nextLine();

        StringBuilder palabrasEspacio = new StringBuilder();

        for (int i = 0; i < n; i++) {
            System.out.println("Palabra " + (i + 1) + ":");
            palabrasEspacio.append(sc.nextLine()).append(" ");
        }

        //Borrar último espacio
        palabrasEspacio.deleteCharAt(index: palabrasEspacio.length() - 1);

        String rutaApp2 ="src/Ejercicio2/App2.java";

        ProcessBuilder pb = new ProcessBuilder(...command: "java", rutaApp2, palabrasEspacio.toString());

        try {
            pb.inheritIO();
            Process p = pb.start();
            //p.waitFor();
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }
}
```

Clase 2:

```
package Ejercicio2;

import ...

public class App2 { @ TomásPérezClaseDAM *

    public static void main(String[] args) { @ TomásPérezClaseDAM *

        String[] palabras = args[0].split(regex: " ");

        ArrayList<String> palabras2 = new ArrayList<>(Arrays.asList(palabras));
        Collections.sort(palabras2);
        Collections.reverse(palabras2);

        StringBuilder sb = new StringBuilder();
        for (String palabra : palabras2) {
            sb.append(palabra).append(" ");
        }

        //Borramos último espacio
        sb.deleteCharAt(index: sb.length()-1);

        System.out.println("Palabras ordenadas a la inversa: " +sb);
```

Salida por terminal:

```
C:\Users\tomas\.jdk\openjdk-24\bin\java.exe "-javaage
¿Cuántas palabras quieres introducir?
3
Palabra 1:
Hola
Palabra 2:
Tomas
Palabra 3:
Zaragoza
Palabras ordenadas a la inversa: Zaragoza Tomas Hola

Process finished with exit code 0
```

Observaciones:

Este ejercicio es algo más complejo ya que he tenido que tratar con diversas estructuras de datos y colecciones. Además, tuve problemas a la hora de ordenar el array de Strings de forma inversa porque me faltó primero ordenarlo y después hacer el reverse y yo solo estaba haciendo reverse. No obstante, el ejercicio creo que me ha salido bien.

Ejercicio 3:

Clase 1:

```
public class App1 { new *

    public static void main(String[] args) { new *

        if (args.length < 2) {
            System.out.println("Argumentos incorrectos. Son 2.");
            System.exit(status: 1);
        }

        String nombre = args[0];
        String opcion = args[1];

        String fecha = LocalDate.now().getDayOfMonth() + "/"
            + LocalDate.now().getMonthValue() + "/"
            + LocalDate.now().getYear();

        String horaActual = String.valueOf(LocalDateTime.now().getHour()) + ':'
            + LocalDateTime.now().getMinute() + ':' + LocalDateTime.now().getSecond();

        generarFichero(nombre, opcion, fecha, horaActual);

    }

    public static void generarFichero(String nombre, String op, String fechaActual, String horaActual) { 1 usage new *

        switch (op) {
            case "1":
                try {
                    File archivo = new File(nombre);
                    FileWriter fw = new FileWriter(archivo, append: false);

                    fw.write(str: fechaActual + " a las " + horaActual + System.lineSeparator());
                    fw.close();
                } catch (IOException e) {
                    throw new RuntimeException(e);
                }
                break;

            case "2":
                try {
                    File archivo = new File( nombre);
                    FileWriter fw = new FileWriter(archivo, append: true);

                    fw.write(str: fechaActual + " a las " + horaActual + System.lineSeparator());
                    fw.close();
                } catch (IOException e) {
                    throw new RuntimeException(e);
                }
                break;
        }
    }
}
```

Clase 2:

```
public class App2 {  
    new *  
  
    public static final String NOMBRE_FICHERO = "Ejercicio3.txt";  
    2 usages  
  
    public static void main(String[] args) {  
        new *  
        Random r = new Random();  
        Scanner sc = new Scanner(System.in);  
  
        System.out.println("¿Cuántas veces quieres llamar al hijo?" + System.lineSeparator()  
            + "Escribe un número entre el 4 y el 6(incluidos): ");  
  
        int cantidad = sc.nextInt();  
  
        while (cantidad < 4 || cantidad > 6) {  
            System.out.println("Tiene que ser entre el 4 y el 6 (incluidos): ");  
            cantidad = sc.nextInt();  
        }  
  
        String rutaArchivo = "src/Ejercicio3/App1.java";  
  
        for (int i = 0; i < cantidad; i++) {  
            String modo;  
  
            if (i == 0) {  
                modo = "1";  
            }  
            else {  
                modo = "2";  
            }  
  
            ProcessBuilder pb = new ProcessBuilder(...command: "java", rutaArchivo, NOMBRE_FICHERO, modo);  
  
            try{  
                Process p = pb.start();  
  
                int segundos = r.nextInt( bound: 4) + 2;  
  
                System.out.println("Llamada: " + (i + 1) + ". Ha tardado: " + segundos + " segundos.");  
  
                p.waitFor();  
  
                if(i < cantidad - 1){  
                    Thread.sleep( millis: segundos * 1000L);  
                }  
            } catch (IOException | InterruptedException e) {  
                throw new RuntimeException(e);  
            }  
        }  
    }  
}
```

```

System.out.println("Contenido del fichero: ");
try{
    BufferedReader br = new BufferedReader(new FileReader(NOMBRE_FICHERO));
    String linea = br.readLine();
    while (linea != null){
        System.out.println(linea);
        linea = br.readLine();
    }
    br.close();
}catch (Exception e){
    e.printStackTrace();
}

```

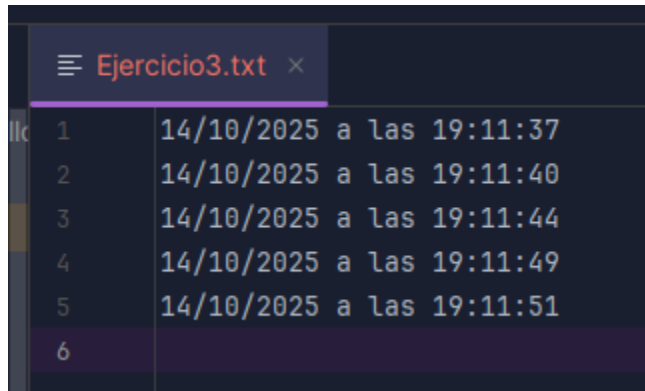
Salida por terminal:

```

¿Cuántas veces quieres llamar al hijo?
Escribe un número entre el 4 y el 6(incluidos):
7
Tiene que ser entre el 4 y el 6 (incluidos):
3
Tiene que ser entre el 4 y el 6 (incluidos):
5
Llamada: 1. Ha tardado: 3 segundos.
Llamada: 2. Ha tardado: 3 segundos.
Llamada: 3. Ha tardado: 5 segundos.
Llamada: 4. Ha tardado: 2 segundos.
Llamada: 5. Ha tardado: 2 segundos.
Contenido del fichero:
14/10/2025 a las 19:11:37
14/10/2025 a las 19:11:40
14/10/2025 a las 19:11:44
14/10/2025 a las 19:11:49
14/10/2025 a las 19:11:51

```


Creación de archivo al ejecutar:



1	14/10/2025 a las 19:11:37
2	14/10/2025 a las 19:11:40
3	14/10/2025 a las 19:11:44
4	14/10/2025 a las 19:11:49
5	14/10/2025 a las 19:11:51
6	

Observaciones:

Este ejercicio se me ha complicado más que los anteriores ya que me ha costado un poco entender el enunciado y le he tenido que dar varias vueltas. Aún así, he entendido el flujo del ejercicio y lo único que he tenido que mirar en internet ha sido lo del tema de dejar entre 2 y 5 segundos de espera entre llamada a la aplicación 1. Para una salida por consola más clara he añadido el tiempo de espera de cada llamada para comprobar que funcionaba correctamente. Además, como se puede ver en la captura de la salida del programa por terminal, he tratado que el usuario no introduzca números fuera del rango.

Ejercicio 4:

Clase 1:

```
public class App1 {  
    new *  
  
    public static void main(String[] args) {  
        new *  
        Random r = new Random();  
  
        LocalDate ahora = LocalDate.now();  
  
        String fecha = ahora.getDayOfMonth() + "/"  
            + ahora.getMonthValue() + "/"  
            + ahora.getYear();  
  
        String horaActual = String.valueOf(LocalDateTime.now().getHour()) + ":"  
            + LocalDateTime.now().getMinute() + ":" + LocalDateTime.now().getSecond();  
  
        int menor;  
        int mayor;  
  
        if (Integer.parseInt(args[0]) < Integer.parseInt(args[1])) {  
            menor = Integer.parseInt(args[0]);  
            mayor = Integer.parseInt(args[1]);  
        } else {  
            menor = Integer.parseInt(args[1]);  
            mayor = Integer.parseInt(args[0]);  
        }  
  
        int aleatorio = r.nextInt(menor, bound: mayor + 1);  
  
        String nombreFichero = args[2];  
  
        File fichero = new File(nombreFichero);  
  
        try {  
            FileWriter fw = new FileWriter(fichero, append: true);  
            fw.write(str: fecha + " " + horaActual + " Temperatura: " + aleatorio + System.lineSeparator());  
            fw.close();  
        } catch (IOException e) {  
            throw new RuntimeException(e);  
        }  
    }  
}
```

Clase 2:

```
public class App2 { new *

    public static void main(String[] args) { new *

        if (args.length < 1) {
            System.err.println("Falta el nombre del fichero.");
            System.exit(status: 1);
        }

        String nombreFichero = args[0];

        try {
            FileReader fr = new FileReader(nombreFichero);
            BufferedReader br = new BufferedReader(fr);

            //Variable auxiliar con el número mínimo posible
            int tempAux = Integer.MIN_VALUE;

            String lineaPrint = null;

            String linea = br.readLine();
            while (linea != null) {
                String[] temp = linea.split(regex: " Temperatura: ");
                int temperatura = Integer.parseInt(temp[1].trim());
                if (temperatura > tempAux) {
                    tempAux = temperatura;
                    lineaPrint = linea;
                }
                linea = br.readLine();
            }

            if (lineaPrint != null) {
                //Salida que se recibe en la app3
                System.out.println(lineaPrint);
            } else {
                System.out.println("El fichero no tiene datos.");
            }
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }
}
```

Clase 3:

```
public class App3 { new *

    public static void main(String[] args) throws IOException, InterruptedException { new *
        Scanner sc = new Scanner(System.in);

        System.out.println("Introduce el nombre del fichero: ");
        String nombreFichero = sc.nextLine();

        System.out.println("Introduce la temperatura 1: ");
        int temperatura1 = sc.nextInt();
        sc.nextLine();

        System.out.println("Introduce la temperatura 2: ");
        int temperatura2 = sc.nextInt();
        sc.nextLine();

        String rutaApp1 = "src/Ejercicio4/App1.java";

        String ficheroConNombre = "ficherosEjercicios/" + nombreFichero ;

        for (int i = 0 ; i < 4 ; i++) {
            try {

                ProcessBuilder pb = new ProcessBuilder(...command: "java", rutaApp1, String.valueOf(temperatura1),
                    String.valueOf(temperatura2), ficheroConNombre);

                Process p = pb.start();

                p.waitFor();

                //Que espere 3 segundos cada vez que corra la app1
                Thread.sleep( millis: 3000);

            } catch (InterruptedException e) {
                throw new RuntimeException(e);
            }
        }

        //Llamada a app2
        String rutaApp2 = "src/Ejercicio4/App2.java";

        ProcessBuilder pb2 = new ProcessBuilder(...command: "java", rutaApp2, ficheroConNombre);

        pb2.inheritIO();
        Process p2 = pb2.start();

        p2.waitFor();
    }
```

Salida por terminal:

```
Introduce el nombre del fichero:
tomi
Introduce la temperatura 1:
12
Introduce la temperatura 2:
20
15/10/2025 17:34:46 Temperatura: 20

Process finished with exit code 0
```

Contenido del fichero creado:

```
15/10/2025 17:34:46 Temperatura: 20
15/10/2025 17:34:49 Temperatura: 19
15/10/2025 17:34:53 Temperatura: 16
15/10/2025 17:34:56 Temperatura: 14
```

Observaciones:

En este ejercicio lo más importante ha sido dejar unos 2-3 segundos entre cada llamada a la app1 ya que al poner menos tiempo a veces me daba error de que no se llegaba a crear el archivo bien. Cambiando eso, y ajustando la ruta de salida del fichero especificando el directorio que he creado para guardar esos ficheros todo ha funcionado correctamente y creo que he realizado lo que se pide en el enunciado: la app3 pide los parámetros y se envían a las clases hijas y corre dichas clases. La app1 genera el fichero y la app2 obtiene la fecha con la temperatura más alta de dicho fichero generado en la app1.

Ejercicio 5

Clase 1:

```
public class App1 {  @ Tomás *

    public static void main(String[] args){  @ Tomás *
        Scanner sc = new Scanner(System.in);

        ArrayList<Integer> lista = new ArrayList<>();
        System.out.println("Introduce un numero: ");
        int numero = sc.nextInt();

        while(numero > 0){
            System.out.println( "(Introduce 0 o negativo si quieres parar)."
                + System.lineSeparator()+ "Introduce un numero: ");
            lista.add(numero);
            numero = sc.nextInt();
        }

        //Paso la lista a una lista de Strings
        String[] listaString = new String[lista.size()];
        for (int i = 0; i < lista.size(); i++) {
            listaString[i] = lista.get(i).toString();
        }

String rutaApp2 = "src/Ejercicio5/App2.java";

ProcessBuilder pb = new ProcessBuilder( ...command: "java", rutaApp2);

try {
    Process p = pb.start();

    BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(p.getOutputStream()));
    for (int i = 0; i < listaString.length; i++) {
        bw.write(listaString[i]);
        bw.newLine();
    }
    bw.flush();
    bw.close();

    BufferedReader br = new BufferedReader(new InputStreamReader(p.getInputStream()));
    String linea = br.readLine();
    while (linea != null) {
        System.out.println(linea);
        linea = br.readLine();
    }
    br.close();

    p.waitFor();

} catch (IOException | InterruptedException e) {
    throw new RuntimeException(e);
}
```

Clase 2:

```
public class App2 {  🧑 Tomás *

    public static void main(String[] args) {  🧑 Tomás *

        Scanner sc = new Scanner(System.in);

        double suma = 0;

        //Para luego revertir y sacar los dos números más grandes y size para la media
        ArrayList<Integer> lista = new ArrayList<>();

        while(sc.hasNextInt()){
            lista.add(sc.nextInt());
        }

        //Añado esta condición para que mínimo la lista tenga dos números y evitar errores
        if (lista.isEmpty() || lista.size() < 2) {
            System.out.println("No hay números suficientes. Tienen que ser mínimo 2.");
            System.exit( status: 1);
        }

        for (int i = 0; i < lista.size(); i++) {
            suma += lista.get(i);
        }
    }
}
```

```

//Media
double media = suma / lista.size();
//Menor
int menor = Collections.min(lista);

//Dos nums top2 de la lista
int[] top2 = new int[2];

Collections.sort(lista);
Collections.reverse(lista);

for (int i = 0; i < top2.length; i++) {
    top2[i] = lista.get(i);
}

//Número más repetido
int moda = lista.get(0);
int mejorConteo = 1;

int numActual = lista.get(0);
int conteoActual = 1;

for (int i = 1; i < lista.size(); i++) {
    if (lista.get(i) == numActual) {
        conteoActual++;
    } else {
        if (conteoActual > mejorConteo) {
            moda = numActual;
            mejorConteo = conteoActual;
        }
        numActual = lista.get(i);
        conteoActual = 1;
    }
}

```

```

if (conteoActual > mejorConteo) {
    moda = numActual;
    mejorConteo = conteoActual;
}

//Salida hacia app1 con los resultados
System.out.println("Dos números mayores: " + Arrays.toString(top2));
System.out.println("Menor número: " + menor);
System.out.println("Media: " + media);
System.out.println("Número más repetido: Número " + moda + " con " + mejorConteo + " apariciones.");

```


Salida por terminal:

```
Introduce un numero:
5
(Introduce 0 o negativo si quieres parar).
Introduce un numero:
10
(Introduce 0 o negativo si quieres parar).
Introduce un numero:
7
(Introduce 0 o negativo si quieres parar).
Introduce un numero:
7
(Introduce 0 o negativo si quieres parar).
Introduce un numero:
0
Dos números mayores: [10, 7]
Menor número: 5
Media: 7.25
Número más repetido: Número 7 con 2 apariciones.
```

Observaciones:

Para este ejercicio he añadido que el usuario tenga que mínimo introducir dos números para así sacar los dos mayores y poder tener mínimo dos números en el array de la app2. Además, en la app2 he ido guardando los números en una ArrayList ya que no necesita especificar su tamaño y una vez obtenidos los números de la entrada pues ya he hecho las operaciones pertinentes para obtener la salida que se envía a la app1.