

# Examen procesos

## Oscar Melero Camacho

### Ejercicio 1

```
public static void main(String[] args) {  
  
    try (Scanner teclado = new Scanner(System.in)) {  
  
        ProcessBuilder pb = new ProcessBuilder(...command: "find", "\"\\hola\\\"", "/C");  
        //ProcessBuilder pb = new ProcessBuilder("md5sum");  
  
        pb.redirectError(ProcessBuilder.Redirect.INHERIT);  
  
        System.out.print("md5txt> ");  
        String linea = teclado.nextLine();  
  
        while (linea.length() > 0) {  
            Process p = pb.start();  
  
            try (PrintWriter printWriter = new PrintWriter(p.getOutputStream(), autoFlush: true)){  
                printWriter.println(linea);  
            }  
            p.waitFor();  
  
            try (Scanner in = new Scanner(p.getInputStream())){  
                if(in.hasNextLine()) {  
                    String lineaSalida = in.nextLine();  
                    System.out.println((lineaSalida.split(regex: " ")[0]));  
                }else{  
                    System.out.println("No se ha recibido nada del proceso.");  
                }  
            }  
  
            System.out.print("md5txt> ");  
            linea = teclado.nextLine();  
        }  
  
    } catch (IOException ex) {  
        System.err.println("Error de E/S");  
        ex.printStackTrace();  
    } catch (InterruptedException ex) {  
        System.err.println("Proceso interrumpido");  
    }  
}
```

Al no poder utilizar el metodo md5sum (al estar en windows) he realizado la prueba con el metodo find /c "hola", que cuenta cuantas veces tiene la palabra hola en la linea escrita:

Aquí va una captura de una ejecución:

```
md5txt> hola
1
md5txt> que tal
0
md5txt> hola que tal
1
md5txt> no esta la palabra
0
md5txt>
Process finished with exit code 130
```

find /C "hola" escribe 1 si tiene la palabra hola y 0 si no la tiene.

## Ejercicio 2

```
public static void main(String[] args) {

    /*
    Como me encontraba en windows, el ejercicio realizado ha sido contar la cantidad de líneas que tiene la palabra hola
    */
    //Almaceno el comando a utilizar
    final String[] COMANDO = {"find", "\"hola\"", "/C"};

    //Creo el scanner con el cual recogeré el nombre del archivo
    try(Scanner teclado = new Scanner(System.in)){
        System.out.println("Introduce el nombre del archivo a leer: ");
        File archivo = new File(teclado.nextLine());
        //Me aseguro de que el archivo exista
        if(!archivo.exists()){
            System.out.println("El archivo no existe!");
            System.exit(1);
        }
        //Preparo el process builder con su comando y sus redirecciones.
        ProcessBuilder findBuilder = new ProcessBuilder().command(COMANDO)
            .redirectError(ProcessBuilder.Redirect.INHERIT).redirectOutput(ProcessBuilder.Redirect.INHERIT)
            .redirectInput(archivo);
        //Ejecuto el comando
        try{
            Process find = findBuilder.start();
        } catch (IOException e) {
            System.out.println("No se ha podido crear el proceso");
            e.printStackTrace();
        }
    }
}
```

Adjunto diferentes ejecuciones:

```
Introduce el nombre del archivo a leer:
asdasda
El archivo no existe!

Process finished with exit code 1
|
```

Si el archivo no existe.

```
Introduce el nombre del archivo a leer:
archivo.txt
4

Process finished with exit code 0
```

Con un archivo valido, contenido del archivo:

hola que tal tengo 22 años y estoy en 2 de dam en 2022

me encuentro diciendo hola ya que el programa lee

cuantas lineas tiene hola

deberia decir que hay 4 lineas

find /C cuenta el numero de lineas que tiene una palabra

en este caso hola

## Ejercicio 3

Una ejecución:

```
El argumento es: hola
Introduce línea:
hola que tal
hola que tal contiene hola
Introduce línea:
como estas
como estas no contiene hola
Introduce línea:
aquí estamos
aquí estamos no contiene hola
Introduce línea:
en clase de procesos
en clase de procesos no contiene hola
Introduce línea:
todo el mundo, hola
todo el mundo, hola contiene hola
Introduce línea:

Process finished with exit code 0
|
```

```

public static void main(String[] args) {
    //Preparo el comando
    final String[] COMANDO = {"find", ""};
    //Miro que se me haya enviado un argumento
    if(args.length != 1){
        System.out.println("Necesito un argumento que buscar");
        System.exit(status: 1);
    }
    //Imprimo el argumento y lo formateo como lo necesito
    System.out.printf("El argumento es: %s\n",args[0]);
    //find requiere que lo que buscar este entre comillas
    String argumento = args[0];
    if(!argumento.startsWith("\""))
        argumento = "\"" + argumento;
    if(!argumento.endsWith("\""))
        argumento = argumento + "\"";
    COMANDO[1] = argumento;

    //Preparo el process builder
    ProcessBuilder processBuilder = new ProcessBuilder().command(COMANDO);
    String input;
    //Creo el scanner que leera de mi teclado
    Scanner teclado = new Scanner(System.in);
    do{
        System.out.println("Introduce linea: ");
        input = teclado.nextLine();
        //Si no es una linea vacia:
        if(!input.equals("")){
            //Creo el proceso
            try {
                Process p = processBuilder.start();
                //Escribo la linea al proceso
                try(PrintWriter printWriter = new PrintWriter(p.getOutputStream(), autoFlush: true)){
                    printWriter.println(input); //No necesito un flush ya que puse autoFlush al crear el printWriter
                }

                //Espero a que el proceso termine su ejecución
                try {
                    p.waitFor();
                } catch (InterruptedException e) {
                    System.out.println("Error al esperar al proceso");
                }

                //Leo del proceso
                try(Scanner inFind = new Scanner(p.getInputStream())){
                    //Si el proceso ha escrito algo, ha reescrito la linea porque tiene la expresión buscada
                    if(inFind.hasNextLine()){
                        System.out.printf("%s contiene %s\n",inFind.nextLine(),args[0]);
                    }else{ //si no, es que no la contenia
                        System.out.printf("%s no contiene %s\n",input,args[0]);
                    }
                }
            } catch (IOException e) {
                System.out.println("Error al crear el proceso");
            }
        }
    }while(!input.equals(""));
}

```