

# PM\_Tarea1

1) Crea un programa que ejecute un comando en Linux o Windows (dependiendo del S.O. donde se esté programando) utilizando la clase Runtime. El comando a ejecutar se proporcionará en los parámetros de línea de comandos. Verificar que el comando se ejecuta, aunque no muestre su salida. Probar a ejecutar un comando que no existe.

## Código:

```
private static void ejecutarComando() {
    try {
        String comando = Inputs.input("Indique el comando: ");
        Process proceso = Runtime.getRuntime().exec(new String[]{"cmd.exe", "/c", comando});
        int codigoSalida = proceso.waitFor();
        System.out.println("El proceso terminó con código: " + codigoSalida);
    } catch (Exception e) {
        System.out.println("Error: " + e.getMessage());
    }
    System.out.println("\n--- Fin de la opción ---\n");
}
```

**Tengo una clase externa en el código, Inputs, en la que armo un Scanner con un String como parámetro de entrada y salida, Input, y otro con un int de salida, InputInt.**

## Imágenes con resultados de pruebas:

```
5 Ejecutar dir en distintos directorios.
6 Mostrar entorno del proceso.
0 Salir.

1
Indique el comando:
dir
El proceso terminó con código: 0

--- Fin de la opción ---

#####

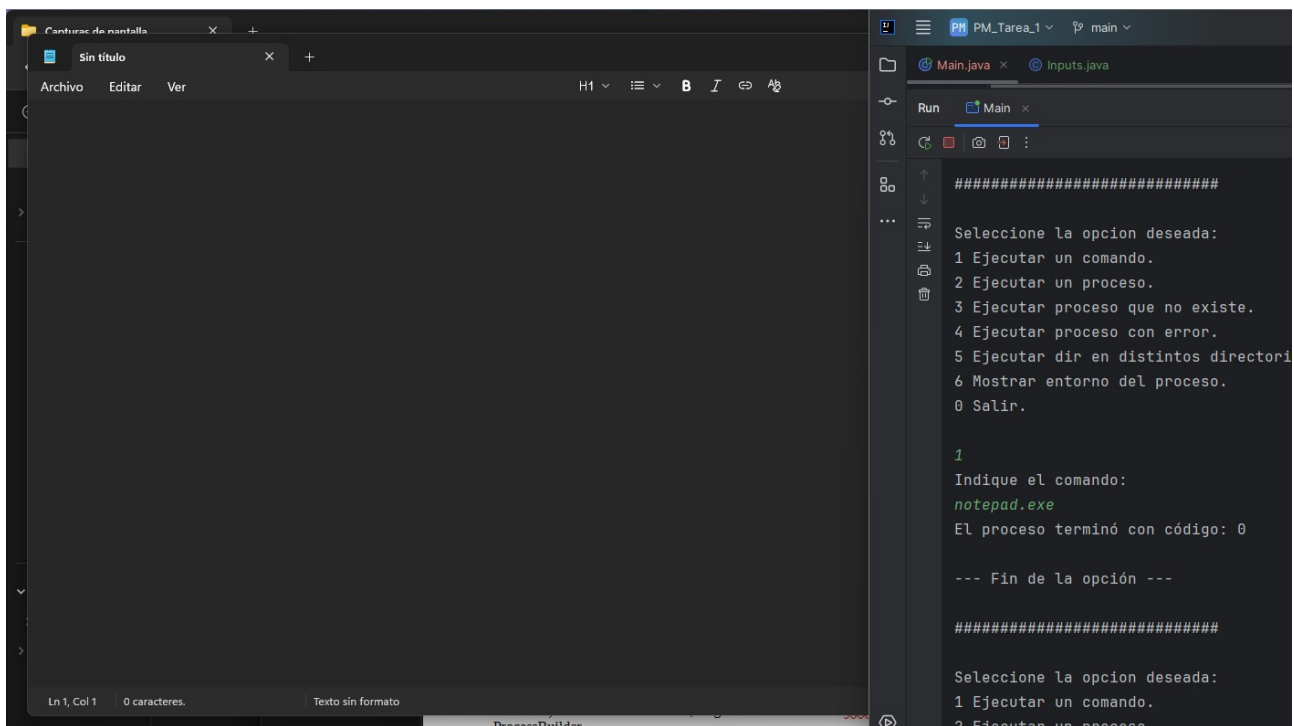
Seleccione la opcion deseada:
1 Ejecutar un comando.
2 Ejecutar un proceso.
3 Ejecutar proceso que no existe.
4 Ejecutar proceso con error.
5 Ejecutar dir en distintos directorios.
6 Mostrar entorno del proceso.
0 Salir.

1
Indique el comando:
ls
El proceso terminó con código: 1

--- Fin de la opción ---|
```

Se ejecuta el comando “dir”, obteniendo código 0, se realizó con éxito.

Se ejecuta comando “ls”, obteniendo código 1, ya que no es un comando válido en Windows.



Se prueba el comando “notepad.exe”, obteniendo código 0 y abriéndose el notepad.

2)

- Ejecución de un proceso que se ejecuta sin errores. Por ejemplo, ls /etc en Linux o dir en Windows (puede utilizarse otro proceso)
- Ejecución de un programa que no existe. Por ejemplo, programa\_que\_no\_existe u otro.
- Ejecución de un proceso que termina con un código de error. Por ejemplo, ls /et en Linux o dir /e en Windows u otros

## Código:

```
private static void ejecutarYReportar(String... cmdArray) {
    try {
        Process p = Runtime.getRuntime().exec(cmdArray);
        try (Scanner sc = new Scanner(p.getInputStream())) {
            while (sc.hasNextLine()) {
                System.out.println(sc.nextLine());
            }
        }
        int code = p.waitFor();

        if (code != 0) {
            try (Scanner err = new Scanner(p.getErrorStream())) {
                while (err.hasNextLine()) {
```

```
        System.out.println("ERR> " + err.nextLine());
    }
}

System.out.println("Código de salida: " + code);

}catch (Exception e) {
    System.out.println("Error: " + e.getMessage());
}

System.out.println("\n--- Fin de la opción ---\n");
}
```

## Imágenes con resultados de pruebas:

```
Seleccione la opcion deseada:
1 Ejecutar un comando.
2 Ejecutar un proceso.
3 Ejecutar proceso que no existe.
4 Ejecutar proceso con error.
5 Ejecutar dir en distintos directorios.
6 Mostrar entorno del proceso.
0 Salir.

2
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: E849-670C

Directorio de C:\Users\Nico\Desktop\Programacion\Segundo\Procesos\PM_Tarea_1

31/10/2025  13:36    <DIR>        .
31/10/2025  15:31    <DIR>        ..|
31/10/2025  13:07                344 .gitignore
31/10/2025  13:38    <DIR>        .idea
31/10/2025  13:36    <DIR>        out
31/10/2025  13:07                433 PM_Tarea_1.iml
31/10/2025  15:29    <DIR>        src
                2 archivos                777 bytes
                5 dirs 382.422.384.640 bytes libres
Código de salida: 0

--- Fin de la opción ---

#####
```

Seleccione la opcion deseada:

- 1 Ejecutar un comando.
- 2 Ejecutar un proceso.
- 3 Ejecutar proceso que no existe.
- 4 Ejecutar proceso con error.
- 5 Ejecutar dir en distintos directorios.
- 6 Mostrar entorno del proceso.
- 0 Salir.

3

ERR> "programa\_que\_no\_existe" no se reconoce como un comando interno o externo,  
ERR> programa o archivo por lotes ejecutable.

Código de salida: 1

--- Fin de la opción ---

#####

Seleccione la opcion deseada:

- 1 Ejecutar un comando.
- 2 Ejecutar un proceso.

Seleccione la opcion deseada:

- 1 Ejecutar un comando.
- 2 Ejecutar un proceso.
- 3 Ejecutar proceso que no existe.
- 4 Ejecutar proceso con error.
- 5 Ejecutar dir en distintos directorios.
- 6 Mostrar entorno del proceso.
- 0 Salir.

|

4

ERR> El modificador no es válido: "e".

Código de salida: 1

--- Fin de la opción ---

#####

3) Crear un programa que muestre cuál es el directorio por defecto de ejecución de un proceso, con el método `directory()` de `ProcessBuilder`, y que ejecute el mismo comando (le en Linux, `dir` en Windows) en distintos directorios, asignados con el método `directory(File directorio)` de `ProcessBuilder`.

## Código:

```
private static void ejecutarDirEnDistintosDirectorios() {
    ProcessBuilder pb = new ProcessBuilder("cmd.exe", "/c", "dir");
    File dir = pb.directory();
    if (dir == null) {
        System.out.println("Hereda el directorio actual de tu programa");
    } else System.out.println(dir);
    File[] directorios = {
        new File("C:\\"),
        new File(System.getProperty("user.home")),
        new File("C:\\Windows")
    };
    try {
        for (File d : directorios) {
            pb.directory(d);
            System.out.println("\n=== Ejecutando en: " + d.getAbsolutePath() + " ===");
            Process p = pb.start();

            try (Scanner sc = new Scanner(p.getInputStream())) {
                while (sc.hasNextLine()) {
                    System.out.println(sc.nextLine());
                }
            }
            p.waitFor();
        }
    } catch (Exception e) {
        System.out.println("Error: " + e.getMessage());
    }
}
```

## Imágenes con resultados de pruebas:

```
0 Salir.

5
Hereda el directorio actual de tu programa

=== Ejecutando en: C:\ ===
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: E849-670C

Directorio de C:\

07/10/2025  13:39  <DIR>          Drivers|
07/10/2025  11:28  <DIR>          inetpub
31/10/2025  12:51  <DIR>          Intel
07/05/2022  06:24  <DIR>          PerfLogs
29/10/2025  20:02  <DIR>          Program Files
24/10/2025  08:57  <DIR>          Program Files (x86)
08/10/2025  10:37  <DIR>          ProgramData
05/10/2025  06:42  <DIR>          Users
29/10/2025  20:02  <DIR>          Windows
               0 archivos                0 bytes
               9 dirs 382.413.336.576 bytes libres

=== Ejecutando en: C:\Users\Nico ===
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: E849-670C

Directorio de C:\Users\Nico

25/10/2025  19:56  <DIR>          .
05/10/2025  06:42  <DIR>          ..
```

Se verifica que dir este en null, imprimiendo "Hereda el directorio actual de tu programa", después ejecuta los otros comandos tirando las listas respectivas de direcciones(bastante largo como para mostrar imagen de todo).



4) Mostrar el entorno de ejecución de un proceso con Map environment() de ProcessBuilder. Hay que iterar sobre el Map devuelto para mostrar cada entrada

## Código:

```
private static void mostrarEntornoProceso() {
    ProcessBuilder pb = new ProcessBuilder("cmd.exe", "/c", "echo Entorno de ejecución");
    Map<String, String> env = pb.environment();

    System.out.println("=== VARIABLES DE ENTORNO DEL PROCESO ===");
    for (String clave : env.keySet()) {
        String valor = env.get(clave);
        System.out.println(clave + " = " + valor);
    }
    env.put("MI_VAR", "Hola desde Java!");
    pb.command("cmd.exe", "/c", "echo %MI_VAR%");

    try {
        Process proceso = pb.start();

        try (Scanner sc = new Scanner(proceso.getInputStream())) {
            while (sc.hasNextLine()) {
                System.out.println(sc.nextLine());
            }
        }

        int codigo = proceso.waitFor();
        System.out.println("Código de salida: " + codigo);
    } catch (Exception e) {
        System.out.println("Error al mostrar el entorno: " + e.getMessage());
    }
}
```

## Imágenes con resultados de pruebas:

0 Salir.

6

=== VARIABLES DE ENTORNO DEL PROCESO ===

USERDOMAIN\_ROAMINGPROFILE = DESKTOP-I3RFV10

LOCALAPPDATA = C:\Users\Nico\AppData\Local

PROCESSOR\_LEVEL = 6

USERDOMAIN = DESKTOP-I3RFV10

FPS\_BROWSER\_APP\_PROFILE\_STRING = Internet Explorer

LOGONSERVER = \\DESKTOP-I3RFV10

JAVA\_HOME = C:\Program Files\Java\jdk-25

SESSIONNAME = Console

ALLUSERSPROFILE = C:\ProgramData

PROCESSOR\_ARCHITECTURE = AMD64

PSModulePath = C:\Program Files\WindowsPowerShell\Modules;C:\Windows\system32\WindowsPowerShell\v1.0\Modules

SystemDrive = C:

EFC\_10004 = 1

OneDrive = C:\Users\Nico\OneDrive

APPDATA = C:\Users\Nico\AppData\Roaming

USERNAME = Nico

ProgramFiles(x86) = C:\Program Files (x86)

CommonProgramFiles = C:\Program Files\Common Files

Path = C:\Program Files\Common Files\Oracle\Java\javapath;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32

FPS\_BROWSER\_APP\_PROFILE\_STRING = Internet Explorer

Se imprimen toda la lista de variables(largo como para sacar un screen de todo)