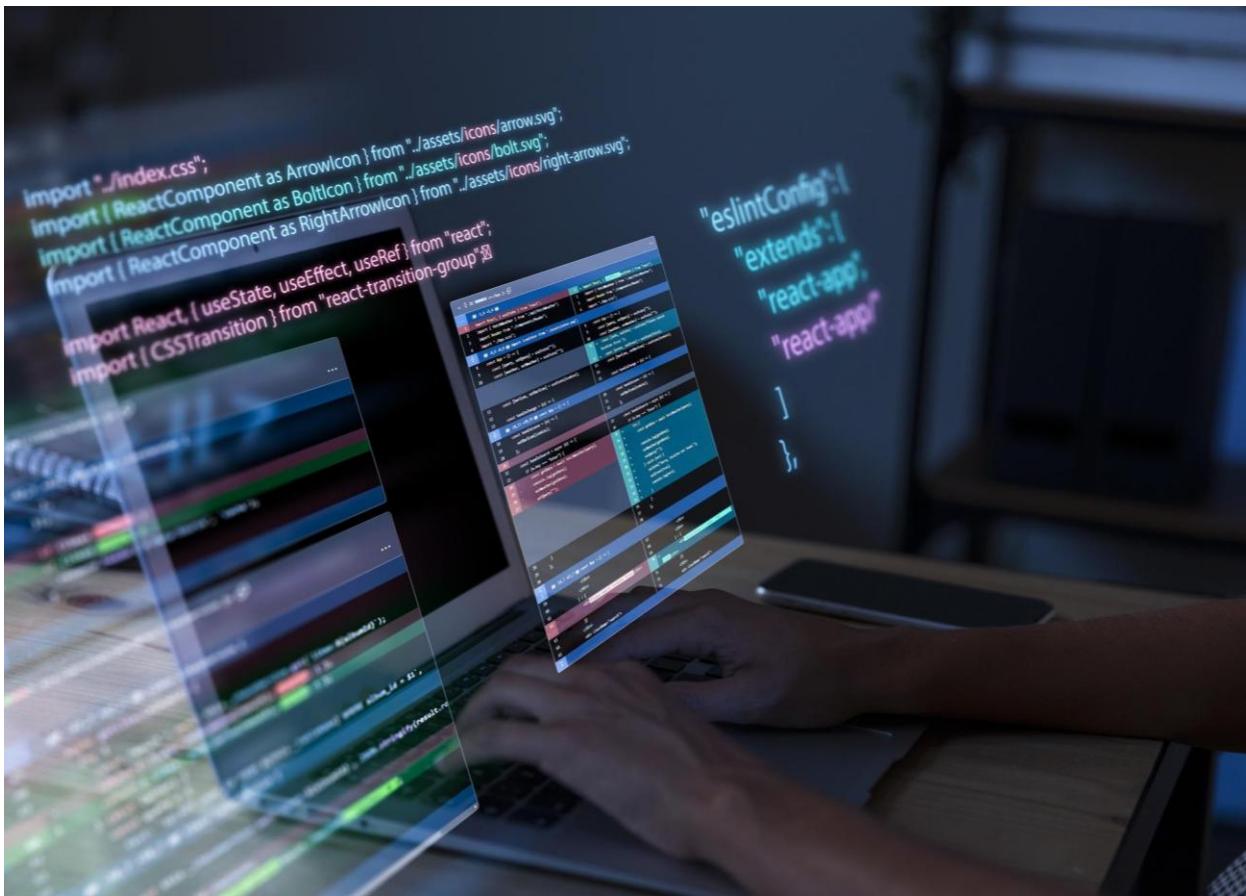


2º Programación de Servicios y Procesos – DAM

# Ejercicios Process y Process Builder



**Ejercicio 1 – Mostrar directorio actual**

Crea un programa que ejecute el comando del sistema operativo que muestra el contenido del directorio actual (dir en Windows o ls en Linux) y muestre la salida por pantalla.

**Ejercicio 2 – Guardar la fecha del sistema**

Crea un programa que ejecute el comando que muestra la fecha (date) y guarde la salida en un fichero llamado fecha.txt.

**Ejercicio 3 – Redirigir errores**

Crea un programa que intente ejecutar un comando inexistente y redirija los mensajes de error a un archivo llamado errores.txt.

**Ejercicio 4 – Preguntar un comando y guardar su salida**

Haz que el programa pida al usuario un comando del sistema, lo ejecute y guarde tanto la salida como los errores en el archivo resultado.txt.

**Ejercicio 5 – Mostrar variables de entorno**

Crea un programa que utilice pb.environment() para listar todas las variables de entorno del sistema.

**Ejercicio 6 – Comprobar conexión**

Haz un programa que ejecute el comando ping a una dirección (por ejemplo, ping google.com) y muestre si la conexión fue exitosa o no en base al código de salida del proceso.

**Ejercicio 7 – Llamar a un archivo externo**

Crea un programa que ejecute un archivo externo (por ejemplo, un .bat o .sh) y muestre el código de salida del proceso una vez terminado.

**Ejercicio 8 – Abrir un programa del sistema**

Escribe un programa que abra una aplicación del sistema (por ejemplo, Bloc de notas, Paint o la Calculadora) usando ProcessBuilder.

**Ejercicio 9 – Ejecutar varios procesos**

Haz un programa que lance dos programas a la vez (por ejemplo, Notepad y Calc) y espere a que ambos terminen antes de mostrar “Todos los procesos finalizados”.

**Ejercicio 10 – Guardar salida y errores en archivos separados**

Ejecuta un comando que genere salida y errores (por ejemplo, dir /xyz) y redirige la salida a salida.txt y los errores a errores.txt.

**Ejercicio 11 – Mostrar solo la primera línea de la salida**

Ejecuta un comando y muestra solo la **primera línea** de su salida (por ejemplo, ipconfig).

### **Ejercicio 12 – Comando según sistema operativo**

Haz que el programa detecte automáticamente si está en Windows o Linux, y ejecute un comando diferente en cada caso (por ejemplo, ipconfig o ifconfig).

### **Ejercicio 13 – Leer desde InputStream y contar líneas**

Ejecuta un comando y muestra cuántas líneas ha devuelto su salida total.

### **Ejercicio 14 – Clase utilitaria**

Crea una clase llamada ComandoUtils con un método estático ejecutar(String comando) que devuelva la salida del comando como un String.

### **Ejercicio 15 – Crear copia de seguridad**

Crea un script .bat o .sh que copie una carpeta a otra y un programa Java que lo ejecute y diga si ha sido exitoso.

### **Ejercicio 16 – Ejecutar un proceso con variable de entorno personalizada**

Haz que el programa establezca una variable de entorno (JAVA\_HOME o MI\_NOMBRE) y luego ejecute un comando que la muestre por pantalla.

**Ejercicio 17 – Repetir ejecución**

Crea un programa que ejecute un mismo comando 3 veces seguidas (por ejemplo, echo Hola mundo) y muestre el número de ejecución.

**Ejercicio 18 – Esperar tiempo antes de lanzar otro proceso**

Ejecuta un proceso, espera 3 segundos (Thread.sleep(3000)), y luego lanza otro.

**Ejercicio 19 – Mostrar procesos en ejecución**

Ejecuta el comando tasklist (Windows) o ps -e (Linux) y guarda la salida en procesos.txt.

**Ejercicio 20 – Menú interactivo**

Haz un programa que muestre un pequeño menú con opciones:

1. Ver IP del equipo
2. Ver variables de entorno
3. Probar conexión con Google

El programa ejecutará los comandos correspondientes según la opción.