

# Procesos

## Ejercicio 1

Crea un programa Java que lance un proceso para ejecutar el comando ping a una dirección IP.

### Requisitos:

- Mostrar por pantalla que el proceso se ha lanzado
- Esperar a que termine el proceso
- Mostrar el código de salida del proceso

## Ejercicio 2

Modifica el ejercicio anterior para capturar y mostrar la salida del comando ping.

### Requisitos:

- Capturar la salida estándar del proceso
- Mostrar línea por línea la salida del ping
- Controlar posibles errores en la salida de error

## Ejercicio 3:

Crea un programa que lance procesos separados para realizar operaciones matemáticas usando la calculadora del sistema operativo o comandos básicos.

### Requisitos:

- Primera versión que directamente haga la suma de dos números pedidos al usuario
- Segunda versión: crear un menú con opciones: suma, resta, multiplicación, división. Que siga pidiendo operaciones hasta que seleccionemos una opción para salir.
- Cada operación debe ejecutarse en un proceso separado
- Pasar los números como argumentos al proceso
- Mostrar el resultado obtenido del proceso

PowerShell puede ejecutar operaciones directamente como comandos de consola. Para indicar desde código que lo que le pasamos es un comando y no un fichero a ejecutar hay que usar “-Command”

## Ejercicio 4:

Desarrolla un programa que lance varios procesos simultáneamente y monitoree su estado.

### Requisitos:

- Lanzar al menos 3 procesos diferentes (pings con varios intentos a diferentes URLs)
- No esperar a que terminen
- Verificar periódicamente si están vivos usando
- Mostrar el estado de cada proceso cada 2 segundos (Podemos hacer que el programa principal espere con `Thread.sleep(2000)`)
- Terminar cuando todos los procesos hayan finalizado y mostrar el código de salida (Además de con el `waitFor` se puede obtener a partir del objeto `process` con el método `exitValue()`)

## Hilos

## Ejercicio 5

Crea un programa que lance 3 hilos, cada uno contando números en rangos diferentes.

### Requisitos:

- Hilo 1: cuenta del 1 al 5
- Hilo 2: cuenta del 10 al 15
- Hilo 3: cuenta del 100 al 105
- Cada hilo debe identificarse con su nombre
- Pausa de 500ms entre números
- Observar cómo los números aparecen mezclados
- El hilo principal después de lanzarlos hará un par de esperas de 1 segundo imprimiendo mensajes informando de las esperas y de la terminación de su ejecución.

## Ejercicio 6:

Crea una simulación de carrera donde varios hilos compiten por llegar a la meta.

### Requisitos:

- Crear 4 corredores (hilos)
- Cada corredor avanza una distancia en metros aleatoria (1-8 metros) cada 200ms
- Meta: llegar a 50 metros
- Mostrar el progreso de cada corredor
- Cuando un corredor llegue a la meta, que anuncie su llegada
- Los demás corredores pueden continuar hasta llegar

### **Ejercicio 7:**

Crea 4 magos que tienen que luchar contra 4 dragones al mismo tiempo (cada uno con su dragón).

#### **Requisitos:**

- Crear 4 magos
- Cada mago hace un daño aleatorio (5-20 puntos) cada 400ms
- Los dragones tienen 100 puntos de vida cada uno.
- Cuando un mago mata al dragón lo anuncia

### **Ejercicio 8:**

Cuatro heroes entran en mazmorras separadas y luchan contra diferentes monstruos subiendo de nivel hasta que llegan a nivel 5 y las abandonan.

#### **Requisitos:**

- Crear 4 héroes con un nombre. Todos empiezan en nivel 1 y con experiencia 0
- Los monstruos pueden ser de 8 tipos: Goblin, Esqueleto, Orco, Araña gigante, Zombi, Slime, Dragón y Murciélago
- Cada vez que entran a la mazmorra se encuentran un enemigo de forma aleatoria
- Lo matan automáticamente y ganan una cantidad de experiencia aleatoria entre 10 y 50 puntos
- Si la experiencia es igual o mayor a 100 suben un nivel y la experiencia se resetea a 0.
- Cuando suben de nivel estan medio segundo celebrandolo antes de continuar
- Entre aventura y aventura esperan entre 1 y 3 segundos

- Ocasionamente (probabilidad del 30%) muestran en un mensaje dando información de su nivel y de la cantidad de experiencia que tienen
- Informan al salir de la mazmorra una vez alcanzado el nivel 5