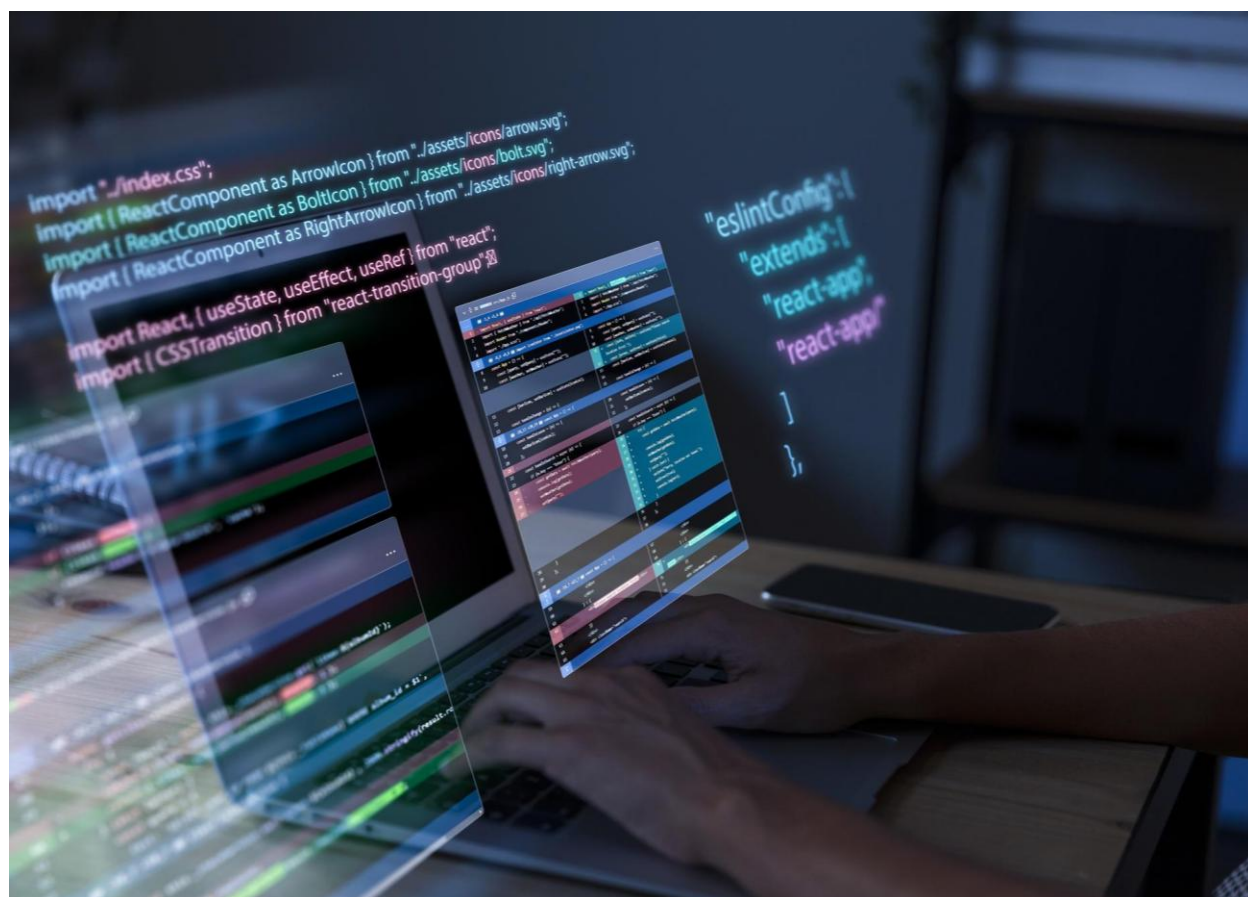


PROGRAMACIÓN DE SERVICIOS Y PROCESOS

# EJERCICIOS DE HILOS

## Programación concurrente en Java

---



## EJERCICIOS DE HILOS — NIVEL 1

### 1. Ejercicio: Diferencia entre `run()` y `start()`

Crea un programa que:

- Tenga un hilo que imprime números del 1 al 5.
- Llama a `run()` primero.
- Luego llama a `start()`.
- Observa el orden de ejecución.
- Explica por qué ocurre.

### 2. Ejercicio: Pausa de un hilo con `sleep()`

Haz un hilo que:

- Imprima "tic" cada 500 ms.
- Cuando haya impreso 5 veces, termine.
- Mientras tanto, el hilo main imprimirá "MAIN sigue..." cada 300 ms.

### 3. Ejercicio: `isAlive()`

Crea un hilo que tarde 2 segundos en terminar.

- Desde el main, antes de `start()`, comprueba `isAlive()`.
- Justo después de `start()`, vuelve a comprobar.
- Después de `join()`, comprueba otra vez.

#### 4. Ejercicio: join() básico

Crea dos hilos:

- Hilo A tarda 3 segundos en terminar.
- Hilo B no puede comenzar hasta que A haya terminado → usa join().

#### 5. Ejercicio: Prioridades

Crea 3 hilos:

- Uno con prioridad mínima.
- Otro con prioridad normal.
- Otro con prioridad máxima.

Cada hilo imprime su nombre 10 veces.

Observa el orden y explica si cambia según ejecuciones.

## EJERCICIOS DE HILOS — NIVEL 2

### 6. Ejercicio: Carrera de Hilos (con prioridades y sleep)

Simula una carrera entre 3 hilos:

- Cada hilo avanza de 0 a 100.
- Cada paso es un número aleatorio entre 1 y 5.
- Cada hilo tiene una prioridad distinta.
- Imprime quién llega primero.

### 7. Ejercicio: Tres tareas que deben ejecutarse en orden

Tienes 3 hilos:

- Hilo 1 debe ejecutar primero.
- Hilo 2 debe esperar a que el 1 termine → join.
- Hilo 3 debe esperar al 2 → join.

Cada hilo imprime un mensaje.

### 8. Ejercicio: Imprimir números pares e impares

Hilo A imprime solo los pares del 0 al 20.

Hilo B imprime solo los impares del 1 al 19.

El main debe esperar a que ambos terminen (join).

Cada hilo debe hacer un sleep(200) entre número y número.

### 9. Ejercicio: Simular un semáforo simple

Crea un hilo que imprima:

- VERDE → espera 2s
- AMARILLO → espera 1s
- ROJO → espera 3s

Repítelo 3 ciclos.

Mientras tanto, el main imprime "Semáforo funcionando..." cada 500 ms.

### 10. Ejercicio: Contador compartido sin sincronización

Crea 2 hilos que incrementen el mismo contador 1000 veces.

Usa sleep(1) en algunas vueltas.

Observa resultados distintos cada ejecución → condición de carrera.

Explica por qué pasa.

## EJERCICIOS DE HILOS — NIVEL 3

### 11. Ejercicio: Timer casero con hilos

Crea un hilo que:

- Cada segundo incremente un contador.
- Imprima el contador como si fuera un cronómetro.  
El main debe esperar 5 segundos y luego interrumpir el hilo.

### 12. Ejercicio: Descargar archivos simulados

Simula 3 descargas:

- Cada descarga es un hilo.
- Cada hilo tarda entre 1 y 5 segundos (sleep).
- El hilo main NO puede continuar hasta que todas las descargas terminen → join() sobre cada una.

### 13. Ejercicio: Prioridades dinámicas

Crea 2 hilos que imprimen su nombre.

Mientras se ejecutan:

- El main cambia su prioridad cada segundo (entre MIN y MAX).
- Observa el efecto.

**14. Ejercicio: Control de proceso con isAlive()**

Un hilo va procesando 10 pasos con sleep(400).

Mientras tanto, el main:

- Cada 300 ms imprime si el hilo sigue vivo o ya ha terminado usando isAlive().
- Cuando termine, imprime "Proceso acabado".

**15. Ejercicio: Dos hilos que deben alternarse**

Hilo A imprime "PING".

Hilo B imprime "PONG".

Deben alternarse con un sleep(500).

No deben imprimir dos PING seguidos ni dos PONG seguidos

(versión simple: usa flags compartidas sin synchronized).

## EJERCICIOS DE HILOS — NIVEL 4

### 16. Ejercicio: Simulador de cocinero y repartidor

Crea dos hilos:

- COCINERO: tarda entre 1 y 3 segundos en “preparar un pedido”.
- REPARTIDOR: tarda entre 2 y 4 segundos en “repartirlo”.

El main debe controlar:

- El cocinero empieza.
- Cuando termine, start del repartidor → join.
- Repite 5 pedidos.

Usa `isAlive()` para mostrar estados.

### 17. Ejercicio: Fabricación de piezas

Tres hilos:

1. Hilo A fabrica la pieza base (tarda 2s).
2. Hilo B pule la pieza (tarda 1s) → solo puede empezar cuando A termine.
3. Hilo C pinta la pieza (tarda 3s) → solo puede empezar cuando B termine.

Usa cadena de `join()`.



## 18. Ejercicio: Sistema de registro

Crea 4 hilos:

- Registrar usuario (2s)
- Validar correo (1.5s)
- Crear carpeta personal (1s)
- Enviar email de bienvenida (1s)

El orden obligatorio:

Registrar → Validar → Carpeta → Email

Usa join() encadenado + sleep().

## 19. Ejercicio: Productor lento y consumidor rápido

(No usar wait/notify todavía: versión con sleep.)

- Productor genera números cada 700 ms.
- Consumidor imprime números cada 300 ms.  
Ambos imprimen sus estados continuamente.  
Usa isAlive() para mostrar si alguno termina antes.

## 20. Ejercicio: Backup simulado

Crea un hilo que simule un backup de 10 pasos:

- Cada paso tarda entre 300 y 700 ms.  
El main:
- Comprueba isAlive() cada 200 ms.

- Si pasan más de 3 segundos sin completar el backup → muestra un aviso.
- Espera a que termine con `join()`.