

## Prácticas Concurrency y Distribución (22/23)

Arno Formella, Alba Nogueira Rodríguez, David Ruano Ordás

semana 27 febrero – 3 marzo

### Práctica 3: Gestión de interrupciones de hilos

**Objetivos:** Gestión de interrupciones de hilos.

**Material adicional:**

- <https://docs.oracle.com/javase/tutorial/essential/concurrency/interrupt.html>

Para experimentar con las interrupciones e interrumpir algunos hilos desde `main` sigue los siguientes pasos:

1. Cambia la tarea de los hilos de la práctica pasada (código con `atan()`) por un simple `sleep` de algunos segundos. Elige un valor suficientemente grande para poder actuar.

Desde el hilo principal pide un índice al usuario e interrumpe el hilo correspondiente (recuerda que seguimos teniendo las referencias a los hilos almacenados en un array).

2. Cambia la tarea de los hilos de nuevo y usa una de las fórmulas conocidas para calcular la constante  $\pi$  (pi). Una muy sencilla de implementar es la siguiente:

$$\pi = 4 \cdot \left( 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots \right)$$

```
for(int i=1; i<10000000; i+=2) {  
    if(negative) pi -= 1.0/i;  
    else pi += 1.0/i;  
    negative = !negative;  
}  
pi*=4.0;
```

Elige como constante máximo del bucle `for` un valor suficientemente grande para que el bucle dure algunos segundos en tu ordenador.

A continuación, escribe el código apropiado en el programa principal para que interrumpa uno o varios hilos (empieza con la interrupción de un único hilo trabajador) después de que el hilo *main* duerma por un tiempo aleatorio adecuadamente elegido.

¿Se interrumpe el hilo trabajador?

3. Consulta de nuevo el tutorial mencionado arriba y cambia la forma como captar la interrupción lanzado desde el hilo principal al hilo trabajador. Familízate especialmente con los métodos `interrupt()`, `interrupted()` y `isInterrupted()`.
4. ¿Es posible captar una interrupción producida por un hilo trabajador en el hilo principal?