

# Concurrencia y Paralelismo

Grado en Informática 2026

## Práctica 1 – Contadores

Vamos a implementar un sistema donde varios threads trabajan con dos contadores: `a` y `b`, que empiezan con valor 0. Habrá dos tipos de threads: El primero suma uno en `a` y resta uno en `b`, y el segundo resta uno en `a` y suma uno en `b`. El número de threads de cada tipo, así como el número de iteraciones que hace cada thread, y una espera en `ns` que cada thread hace después de hacer una iteración, se configura por constantes declaradas con `#define` al principio del código.

Implemente:

**Ejercicio 1 (Threads de tipo a)** Implemente el lanzamiento de los threads de tipo a. Los threads de tipo a deberían recibir sus parámetros a través de una estructura que se crea en el heap. El thread principal debería esperar a que todos los threads de tipo a terminen.

Los contadores no pueden declararse como variables globales. Deberán pasarse a los threads a través de la estructura. Declarelos como variables `volatile`, para evitar que el compilador optimice las lecturas y escrituras.

**Ejercicio 2 (Threads de tipo b)** Implemente el lanzamiento de los threads de tipo b. Los threads de tipo b deberían recibir sus parámetros a través de una estructura que se crea en el stack del thread principal. El thread principal debería esperar a que todos los threads de tipo b terminen, pero los threads de tipo a y b deberían ejecutarse concurrentemente.

La función donde se ejecutan los threads de tipo b debería ser la misma que los de tipo a, sin ningún tipo de comprobación de en qué tipo de thread estamos. La estructura que se pasa a los threads debería incluir un puntero al contador en el que sumar, y un puntero al contador en el que restar.

**Ejercicio 3 (Thread de monitorización)** Implemente un thread de monitorización que se lance desde el thread principal justo antes de lanzar los threads de tipo a y b, y que termine justo después de que todos los threads de tipo a y b hayan terminado. El thread de monitorización debería imprimir el valor de los contadores y su suma de forma periódica.

**Ejercicio 4 (Proteja los contadores con un mutex)** Proteja las operaciones sobre los contadores con un mutex. El thread de monitorización debería utilizar el mutex, y si está implementado de forma correcta la suma debería ser siempre 0. Ese mutex debería pasarse a los threads a través de la estructura de argumentos.

**Ejercicio 5 (Separe el mutex de cada contador)** En vez de utilizar un único mutex para proteger ambos contadores, utilice un mutex para cada contador. Esos mutex deberán pasarse a los threads a través de la estructura de argumentos en dos campos, uno para el mutex que protege el contador en el que se suma, y otro que protege el contador en el que se resta.

## Entrega

La fecha límite de entrega es el 13 de febrero, en la tarea habilitada en el aula virtual.