

# Programación Concurrente y de Tiempo Real<sup>\*</sup>

## Grado en Ingeniería Informática

### Asignación de Prácticas Número 10

En esta asignación aplicará control de exclusión mutua y sincronización, utilizando para ello el API de alto nivel de Java, a diferentes situaciones que se le plantean. Documente todo su código con etiquetas (será sometido a análisis con `javadoc`). Si lo desea, puede también agrupar su código en un paquete de clases, aunque no es obligatorio.

## 1. Enunciados

1. Se desea modelar una carrera de triatlón utilizando un 100 100 tareas `Runnable` para simular a los cien participantes de la prueba. Cada participante modelado por una tarea realizará una secuencia de ejecución compuesto de las siguientes fases:

- Posta de natación
- Posta de carrera ciclista
- Posta de carrera a pie

La duración de cada una de las postas para cada concursante estará modelada por una espera del hilo determinada por un número aleatorio como parámetro del método `sleep(int)`. Cada hilo toma tiempos cuando inicia y acaba cada posta, y acumula los tiempos totales para el triatlón completo en un array de tiempos de 100 ranuras; una por cada hilo. Cada posta tiene una salida neutralizada, de forma que todos los hilos deben llegar a la meta antes de que se inicie la posta siguiente. Modele esta condición mediante los objetos de clase `CyclicBarrier` que sean necesarios. Una vez terminado el triatlón, el programa principal lee el array de tiempos totales, determina el menor de ellos, e imprime cuál es el concursante (hilo) ganador. Guarde el código del programa en `triatlonBarreras.java`.

2. Utilizando cerrojos de clase `ReentrantLock` y colas `Condition`, adapte con estas herramientas el monitor de los lectores/escritores que utilizó en la práctica anterior. Renombre al monitor como `RWMonitorAN.java`. Utilícelo de nuevo para acceder de forma controlada a una fichero externo de disco.

3. Utilice la solución de sockets con servidor multihebrado mediante pool de threads para escribir un servidor que actúe como contador de páginas web.

---

<sup>\*</sup>©Antonio Tomeu

Cada petición de un cliente remoto incrementa en uno el contador que el servidor mantiene. Controle la exclusión cuando sea necesario, y escriba dos versiones de todo el código, que hagan ese control mediante:

- Un objeto de clase `AtomicLong`. Fichero `atomicServer.java`.
- Un cerrojo de clase `ReentrantLock`. Fichero `reentrantServer.java`.

El cliente lo guardará en `clienteContador.java`. Tome tiempo para un número fijo de peticiones de clientes, y muestra una curva con el tiempo total obtenido utilizando ambos tipos de cerrojos vs. número de peticiones. 4. Escriba una solución al problema de los filósofos utilizando un monitor que emplea los recursos del API de alto nivel. Guarde el monitor en `filoApiAn.java`. Escriba una tarea `Runnable` que modele a los cinco filósofos, y actívelos mediante un ejecutor de capacidad fija. Guarde todo esto en `usaFiloApiAn.java`.

## 2. Procedimiento y Plazo de Entrega

Se ha habilitado una tarea de subida en *Moodle* que le permite subir cada fichero que forma parte de los productos de la práctica de forma individual en el formato original. Para ello, suba el primer fichero de la forma habitual, y luego siga la secuencia de etapas que el propio *Moodle* le irá marcando. Recuerde además que:

- No debe hacer intentos de subida de borradores, versiones de prueba o esquemas de las soluciones. *Moodle* únicamente le permitirá la subida de los ficheros por **una sola vez**. utilizará paralelismo de grano grueso, particionando un problema y definiendo el número de hilos necesarios para su resolución en función de ecuaciones de balanceado de carga.
- La detección de plagio (copia) en los ficheros de las prácticas, o la subida de ficheros vacíos de contenido o cuyo contenido no responda a lo pedido con una extensión mínima razonable, invalidará plenamente la asignación, sin perjuicio de otras acciones disciplinarias que pudieran corresponder.
- El plazo de entrega de la práctica se encuentra fijado en la tarea de subida del Campus Virtual.
- Entregas fuera de este plazo adicional no serán admitidas, salvo causa de fuerza mayor debidamente justificadas mediante documento escrito.
- Se recuerda que la entrega de todas las asignaciones de prácticas es recomendable, tanto un para un correcto seguimiento de la asignatura, como para la evaluación final de prácticas, donde puede ayudar a superar esta según lo establecido en la ficha de la asignatura.