

# Prácticas Concurrencia y Distribución (15/16)

Arno Formella, Francisco Rodríguez Martínez, David Olivieri

27 de enero de 2016

## Introducción

Según la guía docente las prácticas se evalúan con los siguientes aspectos:

- (P3) **Informes:** Elaboración de informes (según una guía) que recogen los principales desarrollos y resultados obtenidos por el/la estudiante. Partes de los informes se elaborarán en pequeños grupos.
- (P4) **Pruebas prácticas:** Demostración de los desarrollos e implementaciones de las tareas de programación y experimentos de estudio.
- (P5) **Resolución de problemas:** Elaboración de algoritmos y análisis con cierto nivel de formalismo para comprobar la corrección y estudios de rendimiento.
- (P6) **Presentaciones:** Breves presentaciones orales con medios audiovisuales de desarrollo y resultados obtenidos por el/la estudiante.

Eso se traduce en la realidad de este curso a:

### (P3) Informes

Informes de realización de las prácticas a entregar en pequeños grupos de 2 o 3 personas con posibles variaciones/sugerencias como puedan surgir durante las clases presenciales por la tutoría del profesor.

Un informe deberá constar de y cumplir con:

- Una identificación clara de sus autores (nombres completos y DNIs).
- Una sección para cada apartado y subapartado que se deba incluir en el informe, siguiendo la misma numeración que la de los apartados de los anuncios de las prácticas, si procede.
- Un informe no debe superar las tres (3) páginas.
- Los informes se entregan a través de la plataforma TEMA en el plazo que ahí se indica.
- Se debe entregar también un archivo comprimido con todos los ficheros del código fuente que fueron elaborados para realizar las prácticas con su documentación embebida como se genera con una herramienta adecuada (como por ejemplo doxygen).

#### **(P4) Pruebas prácticas**

El profesor mirará durante las prácticas los desarrollos realizados y evalúa las competencias adquiridas en el diálogo con los estudiantes.

#### **(P5) Resolución de problemas**

Se distribuye problemas que se pueden resolver en el plazo fijado con la entrega del informe correspondiente.

#### **(P6) Presentaciones**

Se ofrece la posibilidad de realizar pequeñas presentaciones sobre el contenido y los resultados de las actividades previamente acordados con el profesor en clases prácticas (10 min) y se evalúan las competencias correspondientes.

### **Evaluación y competencias**

Los informes se evaluarán y computarán dentro del apartado de realización de informes y memorias de prácticas de la materia. Para considerar que el informe alcanza los requisitos mínimos para ser considerado adecuado, será necesario que todos los apartados estén resueltos de manera correcta, y que el código elaborado para cada apartado funcione correctamente y realice la tarea pedida. Para la evaluación del informe se tendrán en cuenta la corrección del código elaborado en cada apartado, su claridad, idoneidad, eficiencia y buen diseño, control de errores y excepciones, finalización correcta, comentarios, modularidad; y las respuestas a las preguntas planteadas y mediciones realizadas (completitud y corrección de las respuestas, número e idoneidad de los ejemplos aclaratorios, número y adecuación de las pruebas realizadas, realización de las cuestiones opcionales, número de máquinas utilizadas para hacer pruebas si es el caso, adecuación y completitud de gráficas y datos presentados, conclusiones obtenidas etc.).

# 1. Actividad: Introducción a la concurrencia en Java

**Objetivos:** Adquirir conocimientos básicos sobre la forma como está implementada la concurrencia en Java, las estructuras básicas para crear hilos, y su utilización.

**Metodología:** Esta práctica se realizará en horas de prácticas presenciales y en horas no presenciales. El tiempo de dedicación estimada es de 2 semanas. Al tratarse de la primera práctica de esta asignatura, además de las actividades descritas se llevarán a cabo tareas de familiarización con las herramientas y métodos de trabajo, y la explicación general del enfoque de las prácticas de la asignatura.

**Material adicional:** Son de especial interés los siguientes enlaces

- <http://docs.oracle.com/javase/7/docs/>
- <http://docs.oracle.com/javase/8/docs/>
- <http://docs.oracle.com/javase/tutorial/essential/concurrency/index.html>
- <http://www.stack.nl/~dimitri/doxygen/index.html>
- <http://en.wikipedia.org/wiki/Markdown> y todo lo que ya se presentó en clases de teoría.

Requisito general a todos los programas es: siempre termina el programa con un mensaje como *Program of exercise X has terminated*, es decir, todos los componentes del programa concurrente terminan correctamente su ejecución.

## 1.1. Introducción a los hilos en Java

1. Examina las dos formas que provee Java para crear un hilo: la clase `Thread` y la interfaz `Runnable`.
2. Utiliza ambas formas para crear un programa que cree y ejecute un hilo que imprima en pantalla un mensaje como *Hello world, I'm a java thread*.  
¿Hay alguna diferencia de funcionamiento entre ambas formas? ¿A nivel de diseño, cuál te parece preferible, y por qué?
3. Modifica tu programa para que el hilo tarde aproximadamente un segundo en mostrar el mensaje. ¿Qué método usarás para ello?
4. Cuando arrancas un hilo desde el programa principal, ¿cuántos hilos hay activos? Saca una lista de los hilos activos por pantalla.

## 1.2. Creación de múltiples hilos

1. Escribe un programa en Java que mediante parámetros de línea de commando reciba cuantos hilos se debe crear. El programa creará y ejecutará el número de hilos indicado. Cada hilo debe imprimir en pantalla un mensaje como *Hello, I'm thread number X*, y después de un segundo,

un mensaje como *Bye, this was thread number X*, siendo *X* el valor de un contador que se va incrementando con el número de hilos creados.

2. Queremos ahora medir el tiempo que tardan en ejecutarse todos los hilos que se crean. Copia y modifica el programa anterior para que muestre este tiempo. Asegúrate de quitar cualquier código que implemente esperas en el hilo, que pudieras haber añadido anteriormente, para medir realmente sólo el tiempo de ejecución de los hilos.
3. ¿Cómo debemos hacer para asegurarnos de que la medida de tiempo sea fiable? Es decir, que realmente se mide el tiempo desde que se arranca el primer hilo hasta que todos los hilos hayan finalizado. ¿Qué errores de medida de tiempo pueden ocurrir, si no nos aseguramos de que todos los hilos han acabado?
4. Copia y modifica el programa para que cada hilo mida él mismo el tiempo que tarda en ejecutarse y guarde el valor en un atributo accesible públicamente. Una vez finalizados todos los hilos, haz que el programa principal sume todos estos valores y muestre en pantalla el resultado. ¿Este valor será mayor o menor que el tiempo global que ya estabas midiendo? ¿Por qué? ¿Te sirve esta suma para algo? ¿Qué harías tú con estos valores para tener algo interesante? Compara tus mediciones con la salida del comando `time` en Linux que te ofrece tiempo real, tiempo del *user* y tiempo del *sistema*.
5. Trata de modificar el programa para poder distinguir entre *tiempo de creación de hilos*, *tiempo de ejecución de los hilos* y tiempo de sincronización final de los hilos\*. Si tan pronto como se crea cada hilo ya se intenta ejecutar, ¿se puede distinguir entre estos tiempos? ¿Qué consecuencias tiene esto a la hora de diseñar y evaluar soluciones concurrentes a un problema?
6. Copia y modifica el programa para que los hilos o bien muestren algo por pantalla o bien que hagan por ejemplo unas operaciones matemáticas suficientemente complejas. Distingue el modo de ejecución mediante un parámetro en línea de comando. La ejecución del programa por prueba debe durar unos segundos para obtener mediciones interpretables.

Pruébalo con un número creciente de hilos para conseguir finalmente una función de tiempo de ejecución en relación al número de hilos trabajadores (y modo de ejecución). ¿Cambia mucho el tiempo de ejecución respecto a cuándo se mostraba algo en pantalla? ¿Por qué crees que ocurre esto? ¡Interpreta detenidamente los gráficos que obtienes!

¡Es interesante que antes de realizar las mediciones intentes predecir los resultados por lo menos cualitativamente y averiguar si tu predicción coincide con los resultados medidos!

¡Es interesante realizar las mediciones en diferentes entornos, respecto a hardware, sistema operativo, y máquina virtual de Java!

### 1.3. Control básico de múltiples hilos

1. Estudia en detalle la utilidad del método `interrupt()` de la clase `Thread`. ¿Qué limitaciones tiene? ¿Cómo debe ser utilizado? Aunque existen otros métodos en la clase `Thread` como `stop()` o `suspend()` ¿Por qué crees que el método `interrupt()` es el método recomendado para detener un hilo?

2. ¿Qué diferencia hay entre los métodos `interrupted()` y `isInterrupted()`? Elabora un programa en Java que demuestre claramente cómo funcionan los dos y en qué se diferencian.
3. Utiliza uno de estos dos métodos (`interrupted()` o `isInterrupted()`) para elaborar un programa que lance un hilo que debe:
  - mostrar un mensaje de inicio;
  - luego debe dejar pasar unos 10 segundos, imprimiendo un mensaje aproximadamente cada segundo,
  - y luego mostrar otro mensaje de fin, y terminar.

El usuario debe poder enviar una señal de interrupción al hilo. Si esta señal llega antes de que hayan transcurrido 5 segundos, el hilo debe mostrar un mensaje de que ha recibido la señal, y volver a ponerse a esperar, hasta terminar los 10 segundos (aproximadamente). Si llega una vez pasados 5 segundos, debe mostrar un mensaje de que ha sido interrumpido y acabar de inmediato.