



Escuela Técnica Superior de Ingeniería

Prácticas de Programación Concurrente y Distribuida

3º Curso de Grado en Ingeniería Informática

PRÁCTICA 5

Condición de Sincronización en JAVA

El objetivo de la práctica es familiarizarse con los mecanismos de Java para la sincronización de hilos. Para ello, se resolverá el siguiente supuesto:

En una tienda de bicicletas trabajan un vendedor y un técnico. A la tienda llegan clientes para comprar o para hacer reparaciones. Los clientes que compran son atendidos por el vendedor, y los que reparan por el técnico, pero mientras haya más de 2 clientes esperando para comprar, el técnico se dedica también a vender, y no atiende las reparaciones.

PROYECTO 1.

Tiempo estimado: 40 minutos.

Puntos: 4

Será el proyecto base para solucionar el enunciado. Contendrá las siguientes clases:

- **Comprar.** Representará cada uno de los clientes de la simulación que llegan a comprar a la tienda mediante un hilo. El hilo se creará heredando de la clase `Thread`. El hilo pondrá un mensaje de inicio, esperará un tiempo aleatorio entre 1 y 3 segundos y pondrá un mensaje de finalización al acabar.

- **Reparar.** Representará cada uno de los clientes de la simulación que llegan a reparar a la tienda mediante un hilo. El hilo se creará implementando el *interface* `Runnable`. El hilo pondrá un mensaje de inicio, esperará un tiempo aleatorio entre 1 y 3 segundos y pondrá un mensaje de finalización al acabar.
- **Generador.** Contendrá el método `main` y será quién comience la ejecución. Debe lanzar 10 clientes a intervalos de tiempo de entre 1 a 2 segundos. El tipo de cliente generado será aleatorio. La ejecución finalizará cuando todos los hilos hayan finalizado.
- **Tienda.** La clase `Tienda` controlará la atención a los clientes e implementará los siguientes métodos:
 - **EntraComprar.** Que deberá ser invocado por los clientes que quieren comprar al acceder a la tienda.
 - **SaleComprar.** Que deberá ser invocado por los clientes que quieren comprar al terminar de ser atendidos.
 - **EntraReparar.** Que deberá ser invocado por los clientes que quieren reparar al acceder a la tienda.
 - **SaleReparar.** Que deberá ser invocado por los clientes que quieren reparar al terminar de ser atendidos.

El control de la concurrencia y la sincronización se realizará en la clase `Tienda`, mediante las primitivas de Java `wait()`, `notify()` y/o `notifyAll()`

Con objeto de visualizar la correcta evolución de los hilos, se deberá realizar la presentación gráfica en pantalla de su estado. Para tal fin, se deberá crear un *Frame* que contenga un *canvas*. El *canvas* deberá mostrar en cada momento el estado de la tienda y las colas de espera. En la representación, cada cliente será identificado por su "id".