

**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 un centro de fisioterapia trabajan un masajista deportivo y un fisioterapeuta. A dicho centro llegan clientes para darse masajes o para hacer rehabilitación. Los clientes que quieren masajes lo pueden hacer con cualquiera de los empleados, pero los que necesitan rehabilitación solo lo pueden hacer con el fisioterapeuta. Tras finalizar su sesión todos los clientes deben acceder a un único vestuario antes de marcharse. Mientras no consigan acceder al vestuario el empleado que los atendió no podrá atender al siguiente cliente.

Además, los clientes que llegan para rehabilitación tendrán prioridad de acceso al fisioterapeuta.

El proyecto para solucionar el enunciado contendrá las siguientes clases:

- **Masaje.** Representará, mediante un hilo, cada uno de los clientes que quieren masaje. El hilo se creará heredando de la clase `Thread`. El tiempo de masaje será aleatorio entre 2 y 4 segundos, y el tiempo en el vestuario será de 2 segundos.

- **Rehabilita.** Representará, mediante un hilo, cada uno de los clientes que quieren rehabilitación. El hilo se creará implementando el *interface* `Runnable`. El tiempo de rehabilitación será aleatorio entre 2 y 4 segundos, y el tiempo en el vestuario será de 2 segundos.
- **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, con una probabilidad del 40% de que necesite rehabilitación. La ejecución finalizará cuando todos los hilos hayan finalizado.
- **Centro.** La clase `Centro` controlará el paso de los clientes e implementará los siguientes métodos:
 - **EntraMasaje.** Que deberá ser invocado por los clientes que quieren masaje antes de entrar.
 - **SaleMasaje.** Que deberá ser invocado por los clientes que se dan masaje cuando terminan el masaje.
 - **EntraRehabilitacion.** Que deberá ser invocado por los clientes que quieren rehabilitación antes de entrar.
 - **SaleRehabilitacion.** Que deberá ser invocado por los clientes que hacen rehabilitación cuando terminan la rehabilitación.
 - **Termina.** Que será invocado por todos los clientes al salir del vestuario.

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

Además, se deberá crear un *Frame* que visualice de forma gráfica, mediante un *Canvas*, la ocupación de los empleados, el vestuario y las colas de espera. El nivel de representación gráfica es libre, pero al menos, se deberán identificar los dos tipos de cliente, la ocupación del centro y las colas de espera.