

**Escuela Técnica Superior de Ingeniería**

Prácticas de Programación Concurrente y Distribuida

3º Curso de Grado en Ingeniería Informática

PRÁCTICA 8

java.util.concurrent.locks y conditions

El objetivo de la práctica es familiarizarse con el mecanismo de control de la concurrencia mediante `ReentrantLock` y `Conditions`.

Se debe implementar un *applet* que solucione el problema del aparcamiento de vehículos en un centro comercial. El problema es el siguiente:

El parking de un famoso centro comercial tiene tres plazas de aparcamiento para coches y una plaza de aparcamiento para autobuses. En la plaza designada para los autobuses caben dos coches. Los coches al llegar intentarán aparcar en las plazas designadas para coches, pero si no hay sitio y en la plaza de autobuses hay sitio libre y no hay ningún autobús esperando, entonces podrán usar un hueco de la plaza de autobuses. Los autobuses solo podrán usar la plaza designada para ellos, y si al llegar encuentran que hay coches aparcados, esperará a que salgan para poder aparcar. Mientras el autobús espera a que salgan los coches aparcados en su plaza, no podrá aparcar ningún coche, a pesar de que haya un hueco libre para coches.

Se debe implementar la solución haciendo uso de `ReentrantLock` y `Condition`.

En esta práctica el diseño es libre, pudiéndose emplear la estructura de clases que se crea más conveniente. Las únicas restricciones son:

- Cada coche y cada autobús deberá ser un hilo.
- Los coches y los autobuses llegarán a intervalos de tiempo aleatorio. El hilo correspondiente a cada coche y cada autobús se creará con la llegada del mismo. No se permite que todos los hilos se creen al iniciar la práctica. La probabilidad de que llegue un coche deberá ser del 90% y de que llegue un autobús del 10%
- Se debe representar gráficamente la situación de las colas y del parking.