

**Escuela Técnica Superior de Ingeniería**

Prácticas de Programación Concurrente y Distribuida

3º Curso de Grado en Ingeniería Informática

PRÁCTICA 7

`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 *Frame* que solucione el problema siguiente:

Una protectora de animales acoge a perros y gatos. La protectora dispone de un comedero donde pueden comer hasta cuatro animales simultáneamente, pero para evitar conflictos no podrán estar simultáneamente tres de un tipo y uno del otro, es decir, no podrán estar tres perros y un gato, o tres gatos y un perro.

Para solucionar el problema se emplearán las siguientes clases:

- **Perro.** Representará, mediante un hilo, cada uno de los perros. El hilo se creará heredando de la clase `Thread`. El tiempo para comer será aleatorio entre 1 y 3 segundos.
- **Gato.** Representará, mediante un hilo, cada uno de los gatos. El hilo se creará implementando el *interface* `Runnable`. El tiempo para comer será aleatorio entre 1 y 3 segundos.

- **Generador.** Contendrá el método `main` y será quién comience la ejecución. Debe lanzar 20 hilos a intervalos de tiempo de entre 1 a 2 segundos. El tipo de hilo generado será aleatorio, con una probabilidad del 50% . La ejecución finalizará cuando todos los hilos hayan finalizado.
- **Comedero.** La clase `Comedero` controlará el acceso de los animales según las restricciones del problema.
- **CanvasComedero.** Representará las colas de espera, así como la ocupación del comedero mediante un canvas.

NOTA: No se permita usar `signalALL` con las variables `Condition`