

**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 *Frame* que solucione el problema siguiente:

En una revisión de exámenes hay tres profesores que revisan simultáneamente. Todos los profesores pueden revisar la teoría, pero sólo dos profesores han llevado su portátil, por tanto, solo ellos puede revisar prácticas (los portátiles no son intercambiables, siempre los tienen los mismos). A la revisión llegarán estudiantes que desearán revisar sólo la teoría o sólo las prácticas. Cuando un estudiante termina la revisión (del tipo que sea) con un profesor que tiene portátil, priorizará la entrada de un estudiante para revisar prácticas.

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

- Cada estudiante para revisar teoría será un hilo “thread” y cada estudiante para revisar prácticas será un hilo “runnable”.

- Los estudiantes llegarán a intervalos de tiempo aleatorio. El hilo correspondiente a cada estudiante se creará con la llegada del mismo. No se permite que todos los hilos se creen al iniciar la práctica.
- El tipo de estudiante generado será aleatorio, con una probabilidad del 60% para teoría y del 40% para prácticas.
- La ejecución finalizará cuando todos los hilos hayan finalizado.
- Se debe representar gráficamente la situación de las colas y de la revisión.

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