***Imagen que contiene Logotipo

Descripción generada automáticamente***

Alejandro Ortega Martínez

Grado en Ingeniería Informática

01/03/2021

Practica 1

Programacion Concurrente

Contenido

**No se encontraron entradas de tabla de contenido.**

# Caso de estudio 1: Concurrencia básica

**Ejecuta el programa varias veces y analiza el resultado de cada ejecución. ¿Son las trazas de ejecución siempre iguales?**

No, no son iguales. Al ser programación concurrente, depende del resto de procesos que haya en ejecución, cuantos soliciten tiempo de CPU, y como gestione cada hilo el gestor de hilos de Eclipse.

**Describe el funcionamiento del planificador de hilos.**

El planificador de hilos se encarga de decidir qué hilo de todos los que están activos recibe tiempo de ejecución por parte de la CPU.

**El método yield de la clase Thread se encarga de dejar paso a otro hilo para que se ejecute, ¿ha cambiado algo en la ejecución?**

Si. Cuando se ejecuta el yield, el thread “abandona” la CPU dejando paso a otro hilo o proceso. Si por ejemplo, ponemos que se ejecute un yield cada 2 impresiones, nunca imprimirá el thread mas de 2 cosas seguidas.

# Caso de estudio 2: Hilos en espera

**¿Qué ocurre si todas los hilos tienen el mismo tiempo de espera?**

Si todos los tiempos son iguales, los hilos “despertaran” al mismo tiempo, y el planificador de hilos tendrá que estar eligiendo todo el rato a que hilo darle tiempo de CPU.

**¿Y si estos tiempos de espera son distintos?**

En este caso, es difícil que coincidan los hilos, por lo que el planificador no tendrá que elegir tantas veces como cuando tiene los sleeps iguales.

**Haz que el programa principal espere la terminación de todos los hilos usando el métodoThread.join().**

Con este método, hasta que no termina el Join del hilo del que se ha dicho, no se ejecuta lo siguiente.

# Caso de estudio 3: Constructores parametrizados

**¿Qué se puede decir del orden en el que se activan los hilos?**

Que la prioridad solo resuelve los empates.