
Prácticas de Programación Concurrente y Distribuida

3º Curso de Grado en Ingeniería Informática

Curso 2018-19

EXAMEN

Enero de 2019

CONSIDERACIONES PREVIAS:

- No se permite el uso de ningún tipo de documentación.
- El acceso a Internet está desactivado conscientemente.
- Apague el teléfono móvil.

ANTES DE COMENZAR EL EXAMEN:

- Cree una carpeta con su nombre y primer apellido en el **Escritorio** separados por un guión bajo (ejemplo: `Pedro_Abad`).
- En dicha carpeta aparecerá un proyecto por cada una de las preguntas del examen. Dichos proyectos se denominarán `Proyecto1`, `Proyecto2`, ..., `Proyecto4`.

ENUNCIADO:

Una piscina dispone de dos calles para usuarios de un club de natación y una calle para nado libre. En cada calle del club puede haber un máximo de dos usuarios, y en la calle libre un solo usuario.

A la piscina acceden usuarios del club, y otros para nadar por libre. Los usuarios por libre siempre usarán su calle para nadar, pero los usuarios del club podrá usar la calle libre si está vacía y no hay usuarios por libre esperando.

NOTA: No es necesario identificar la calle del club que usan los usuarios del club.

PROYECTO 1.

Tiempo estimado: 40 minutos.

Puntos: 4

Será el proyecto base para solucionar el enunciado. Contendrá las siguientes clases:

- **Piscina.** La clase `Piscina` mantendrá el estado de ocupación del piscina e implementará los siguientes métodos:
 - **EntraLibre.** Que deberá ser invocado por los usuarios libres cuando quieren acceder a la piscina .
 - **SaleLibre.** Que deberá ser invocado por los usuarios libres al salir.
 - **EntraClub.** Que deberá ser invocado por los usuarios del club cuando quieren entrar en la piscina.
 - **SaleClub.** Que deberá ser invocado por los usuarios del club al salir.
- **Libre.** Representará cada uno de los usuarios libres mediante un hilo. El hilo se creará heredando de la clase `Thread`. El hilo pondrá un mensaje de inicio indicando su identificador, intentará acceder a la piscina usando la clase `Piscina`, permanecerá en la piscina una cantidad de tiempo aleatoria de entre 2 y 4 segundos y saldrá de la piscina.
- **club.** Representará cada uno de los usuarios del club mediante un hilo. El hilo se creará implementando el *interface* `Runnable`. El hilo pondrá un mensaje de inicio indicando su identificador, intentará acceder a la piscina usando la clase

`Piscina`, permanecerá en la piscina una cantidad de tiempo aleatoria de entre 3 y 6 segundos y saldrá de la piscina.

- **Generador.** Contendrá el método `main` y será quién comience la ejecución. Debe lanzar, de forma aleatoria, usuarios del club y libres a intervalos de tiempo de entre 1 a 2 segundos. La frecuencia de llegada de usuarios del club será del 80% y la de libres del 20%. Deberá esperar a que finalicen todos los hilos para finalizar.

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

PROYECTO 2.

Tiempo estimado: 20 minutos.

Puntos: 3

Se modificará el *Proyecto1* para que la clase `Piscina` controle la concurrencia mediante `ReentrantLocks` y `Conditions`.

No podrá usarse el método `signalAll()` de las `Conditions`.

PROYECTO 3.

Tiempo estimado: 15 minutos.

Puntos: 2

Tomará como base el *Proyecto1* y se modificará en la forma necesaria, de manera que **Generador** permita el acceso de los usuarios de forma remota, haciendo uso de R.M.I. Es decir, generador funcionará como servidor y aceptará conexiones remotas a los métodos de la clase `Piscina`.

PROYECTO 4.

Tiempo estimado: Depende de la implementación que se pretenda

Puntos: 1

Se creará un *Applet* que visualice de forma gráfica, mediante un *Canvas*, la situación del `Piscina` y las colas de espera del *Proyecto 1*.