
Prácticas de Programación Concurrente y Distribuida

3º Curso de Grado en Ingeniería Informática

Curso 2019-20

EXAMEN

Enero de 2020

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:

En una agencia de viajes trabajan dos empleados (Empleado A y Empleado B) atendiendo al público. Los clientes acuden a la agencia a concertar viajes o a comprar entradas. Ambos empleados atienden a los dos tipos de cliente, pero los clientes de viajes son prioritarios para el empleado A

PROYECTO 1.

Tiempo estimado: 40 minutos.

Puntos: 4

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

- **Agencia.** La clase Agencia mantendrá el estado de ocupación de la Agencia e implementará los siguientes métodos:
 - **EntraViaje.** Que deberá ser invocado al llegar a la agencia por las clientes que quieren viajes.
 - **SaleViaje.** Que deberá ser invocado por las clientes que quieren viajes al salir.
 - **EntraEntradas.** Que deberá ser invocado al llegar a la agencia por las clientes que quieren entradas.
 - **SaleEntradas.** Que deberá ser invocado por las clientes que quieren entradas al salir.
- **Viajes.** Representará cada uno de los clientes que quieren viajes mediante un hilo. El hilo se creará heredando de la clase Thread. El hilo pondrá un mensaje de inicio indicando su identificador, intentará ser atendido usando la clase Agencia, será atendido durante un intervalo aleatorio entre 3 y 5 segundos y saldrá.
- **Entradas.** Representará cada uno de las clientes que quieren entradas mediante un hilo. El hilo se creará implementando el *interface* Runnable. El hilo pondrá un mensaje de inicio indicando su identificador, intentará ser atendido usando la clase Agencia, será atendido durante un intervalo aleatorio entre 2 y 3 segundos y saldrá.

- **Generador.** Contendrá el método `main` y será quién comience la ejecución. Debe lanzar, de forma aleatoria, clientes para entradas y clientes para viajes a intervalos de tiempo entre 1 y 3 segundos. La frecuencia de llegada de los clientes para entradas será del 70% y la de clientes para viajes del 30%. Deberá esperar a que acaben todos los hilos para finalizar.

El control de la concurrencia y la sincronización se realizará en la clase `Agencia`, 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 `Agencia` 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 *Proyecto 1* y se modificará en la forma necesaria, de manera que **generador** permita el acceso de los clientes a la `Agencia` de forma remota, haciendo uso de R.M.I. Es decir, generador hará de servidor del objeto de la clase `Agencia` para los clientes remotos.

PROYECTO 4.

Tiempo estimado: Depende de la implementación que se pretenda

Puntos: 1

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