



Prácticas de Programación Concurrente y Distribuida

3º Curso de Grado en Ingeniería Informática

Curso 2014-15

EXAMEN

Enero de 2015

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**, ..., **Proyecto5**.

ENUNCIADO:

Un almacén tiene capacidad para recoger M kilos de fruta. Dicho almacén recibe la visita de proveedores y clientes. Cada proveedor deposita una determinada cantidad de kilos en el almacén, mientras que los clientes compran (retiran) del almacén los kilos que desean. Inicialmente, el almacén contiene N kilos de fruta.

Se probarán las soluciones con $M=10$ y $N=2$.

PROYECTO1.

Tiempo estimado: 15 minutos.

Puntos: 1,5

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

- **Cliente.** Representará cada uno de los clientes mediante un hilo. El hilo se creará heredando de la clase `Thread`. El cliente pondrá un mensaje de inicio indicando su identificador y la cantidad de fruta a comprar, y un mensaje final cuando haya conseguido toda la cantidad. La cantidad será aleatoria entre 2 y 5 kilos.
- **Proveedor.** Representará cada uno de los proveedores mediante un hilo. El hilo se creará implementando el *interface* `Runnable`. El proveedor pondrá un mensaje de inicio indicando su identificador y la cantidad de fruta a depositar, y un mensaje final cuando haya conseguido dejar toda la cantidad. La cantidad será aleatoria entre 2 y 5 kilos.
- **Generador.** Contendrá el método `main` y será quién comience la ejecución. Debe lanzar, de forma aleatoria, proveedores o clientes a intervalos de tiempo de entre 1 a 2 segundos. Cuando la suma de hilos lanzados sea 10, esperará 2 segundos y abortará todos los hilos que no hayan finalizado.

Dado que es el proyecto base, no se controlará la concurrencia ni la capacidad del almacén. Se supone que siempre hay sitio en el almacén y fruta disponibles.

PROYECTO2.

Tiempo estimado: 15 minutos.

Puntos: 2

Se modificará el *Proyecto1* para controlar la compra de fruta en el almacén mediante un **semáforo general**.

Los clientes podrán retirar la fruta del almacén de forma incremental, es decir, podrán ir retirando uno a uno los kilos de fruta en función de su disponibilidad. Cada vez que un cliente consiga un kilo, pondrá un mensaje indicándolo. Los proveedores siempre podrán depositar la fruta, se considera que la capacidad del almacén es ilimitada para este proyecto.

No puede crearse ninguna nueva clase. El código deberá ser añadido a las clases ya existentes en el *Proyecto1*.

PROYECTO3.

Tiempo estimado: 25 minutos.

Puntos: 2,5

Se añadirá al *Proyecto1* una nueva clase llamada **Almacen**. La clase `Almacen` mantendrá el estado de ocupación del almacén e implementará los siguientes métodos:

- **Deposita.** Que deberá ser invocado por los proveedores cuando traen fruta. Los proveedores dejarán toda la fruta en el almacén de una vez, y si no la pueden dejar toda esperarán a que haya sitio para hacerlo
- **Compra.** Que deberá ser invocado por los clientes cuando quieren comprar fruta. Los clientes comprarán toda la cantidad disponible de una vez. Si el almacén no tiene la cantidad que requieren no cogerán nada hasta que haya cantidad suficiente.

Cada vez que se produzca una entrega o una compra, se visualizará un mensaje indicando la cantidad de fruta disponible en el almacén.

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

PROYECTO4.

Tiempo estimado: 30 minutos.

Puntos: 3

Se modificará el *Proyecto3* para que la clase `Almacen` controle la concurrencia mediante `ReentrantLocks` y `Conditions`.

Se modificará también la forma de proceder de los clientes para que actúen de la siguiente manera: si cuando un cliente está esperando, porque no hay fruta suficiente, dos proveedores dejan fruta y dicho cliente no consigue comprar, desistirá de hacerlo y se marchará sin hacer la compra.

PROYECTO5.

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 almacén y las colas de espera del *Proyecto4*.

El nivel de representación gráfica es libre, pero al menos, se deberán identificar la ocupación del almacén y las colas de clientes y proveedores con sus respectivas necesidades.