

# Trabajo práctico N° 7

## Mecanismos de sincronización: Locks - Locks con Condiciones

Consideraciones para la realización del práctico:

- Debe tener en cuenta que los valores que se presentan en los enunciados son a modo de ejemplo. Deberán probar con distintos valores para asegurar el correcto funcionamiento del programa.
- Recuerde utilizar variables en vez de números fijos.

### 1. Sala de museo.

Se quiere desarrollar un sistema para controlar la temperatura y el número de personas que se encuentran en una sala de un museo. En condiciones normales, se permiten **50** personas en la sala. Si la temperatura sube por encima de un umbral ( $t_{\text{Umbral}} = 30$ ), se limita el número de personas a **35**. Si cuando se detecta este suceso el número de personas en la sala es mayor que 35, no es necesario desalojarlas. Si una persona jubilada intenta entrar, tendrá prioridad frente al resto de personas que estén esperando. Cada persona se representa mediante un hilo. Además, hay un hilo que mide periódicamente la temperatura de la sala y notifica su valor al sistema. Se pide desarrollar un monitor (*GestorSala*) que sincronice los hilos que representan personas y al hilo que mide la temperatura, de acuerdo con las especificaciones anteriores.

- a. Utilizar locks, proporcionando los siguientes métodos:

```
... void entrarSala()
// se invoca cuando una persona quiere entrar en la sala.

... void entrarSalaJubilado()
// se invoca cuando una persona jubilada quiere entrar en la sala.

... void salirSala()
// se invoca cuando una persona, jubilada o no, quiere salir de la sala.

... void notificarTemperatura(int temperatura)
// lo invoca la hebra que mide la temperatura de la sala para indicar el último valor medido.
```

Observación: No es necesario garantizar que el orden de acceso a la sala coincide con el orden de llegada a la puerta de entrada.



## **2. Los programadores y los recursos**

Imagina un grupo de programadores que trabajan juntos en un proyecto de software. Cada programador necesita acceder a dos recursos: una computadora portátil para escribir código y un libro de referencia para consultar información técnica. Los programadores pasan su tiempo escribiendo código y consultando el libro de referencia para resolver problemas y mejorar el proyecto. Sin embargo, el acceso a los recursos es limitado, y los programadores deben coordinarse para evitar conflictos. Diseña un programa en Java que modele la interacción entre los programadores y los recursos (computadoras portátiles y libros de referencia) utilizando Locks y Condiciones teniendo en cuenta que cada programador necesita tomar una computadora portátil y un libro de referencia para trabajar en el proyecto. Los programadores deben esperar si no hay suficientes recursos disponibles.

Implementa el programa de manera que los programadores puedan acceder a las computadoras portátiles y los libros de referencia de manera segura y eficiente para trabajar en el proyecto de software.

## **3. Cocineros y Recetas**

Imagina una cocina donde hay tres tipos de ingredientes: carne, vegetales y pasta. También hay tres tipos de cocineros, cada uno especializado en cocinar un tipo particular de receta: carne a la parrilla, ensalada de vegetales y pasta al pesto. Cada cocinero necesita los ingredientes correspondientes para preparar su receta. Diseña un programa en Java que modele la interacción entre los cocineros y los ingredientes utilizando Locks.

Debe tener en cuenta que los ingredientes se encuentran disponibles en cantidades limitadas y deben ser compartidos entre los cocineros, los ingredientes se reponen automáticamente después de que un cocinero haya utilizado una cantidad determinada.

- a. Implementa tres cocineros que trabajan de manera concurrente, cada uno tomando los ingredientes necesarios y preparando su receta.
- b. Implementa el programa de manera que los cocineros puedan obtener los ingredientes adecuados y preparar sus recetas de manera segura y eficiente.

## **4. ¿Cómo resolvería el ejercicio anterior usando semáforos?**

## **5. Toma 1 ejercicio del práctico de Monitores y uno del práctico de Semáforos e intenta hacerlo con Locks.**

- a. ¿Encuentra dificultades? ¿Cuáles?
- b. ¿Considera que un mecanismo de sincronización para ese ejercicio es mejor que otro?