



## Programación Concurrente - Examen parcial - 11-11-2016

Apellido y nombre: \_\_\_\_\_ Legajo: \_\_\_\_\_

NOTA: realizar pseudocódigo detallado o java sin considerar las excepciones

### Ejercicio 1

Escriba un monitor en Java que simule a un semáforo general ( llame a la clase MonitorSemaf ); incluya operaciones acquireM(), releaseM(), tryacquireM() , acquireM(n), releaseM(n), y el constructor MonitorSemaf(n).

A partir de una instancia de este monitor, escriba código para el problema del productor-consumidor

### Ejercicio 2

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 ( $t_{Umbral} = 30$ ), 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 una hebra. Además, hay una hebra que mide periódicamente la temperatura de la sala y notifica su valor al sistema. Se pide desarrollar un monitor (GestorSala) que sincronice a las hebras que representan personas y a la hebra que mide la temperatura, de acuerdo con las especificaciones anteriores.

Utilizar monitor o 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.
```

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

### Ejercicio 3

El problema del taxista: Considere los siguientes dos hilos: un taxista y usted. Usted necesita un taxi para llegar a su destino y el taxista necesita que un pasajero tome el taxi para cobrar por su trabajo. Por lo tanto cada uno tiene una tarea. Usted espera tomar el taxi y viajar hasta que el taxista le notifique que ha llegado a su destino. Resultaría molesto para usted y para el preguntar cada dos segundos “¿ya llegamos a destino?”. Entre distintos viajes, el taxista quiere dormir en el taxi hasta que otro pasajero necesite ser conducido hasta algún lugar no desea despertarse de su siesta cada cinco minutos para ver si arriba un pasajero. Por lo tanto, ambos hilos preferirían realizar su trabajo de la manera mas relajada posible. Usted y el taxi necesitan alguna forma de comunicar sus necesidades al otro. Mientras usted esta ocupado caminando por la calle buscando un taxi, el taxista esta durmiendo plácidamente en la cabina. Cuando usted le avisa que quiere tomar el taxi, el se despierta y comienza a conducir. Cuando usted arriba a su destino, el taxista le notifica que ha llegado y continua con su trabajo. El taxista debe ahora esperar y dormir nuevamente una siesta hasta que el próximo pasajero llegue.

Diseñe una solución de comunicación de los hilos utilizando semaforos que modele dicha situación.