



# Trabajo práctico Nº 6

## Mecanismos de sincronización: Monitores - Locks y Variables de condición

### **1. Control de Acceso a una Sala de Estudio**

Imagina una sala de estudio en una biblioteca universitaria con un número limitado de mesas disponibles para los estudiantes. Para asegurarse de que los estudiantes tengan un lugar para estudiar y evitar la sobrecarga de la sala, se implementará un sistema de control de acceso utilizando monitores. La sala de estudio tiene un límite máximo de mesas y se representa como un recurso compartido. Los estudiantes llegan a la sala y desean ocupar una mesa para estudiar. El monitor de control se encarga de regular el acceso a la sala de la siguiente manera: a. Si hay mesas disponibles en la sala, un estudiante puede ocupar una mesa. b. Si todas las mesas están ocupadas, el estudiante debe esperar hasta que una mesa esté disponible. c. Cuando un estudiante sale de la sala, su mesa se vuelve disponible para otro estudiante que está esperando afuera. Diseña un programa en el que los estudiantes lleguen a la sala de estudio y deseen ocupar una mesa. Si hay mesas disponibles, el estudiante ocupa una mesa y estudia. Si todas las mesas están ocupadas, el estudiante debe esperar afuera hasta que haya una mesa disponible. Utiliza monitores para garantizar un acceso ordenado y sincronizado a la sala de estudio.

### **2. Barco de Pasajeros y Automóviles en un Ferry**

Imagina un ferry que puede transportar tanto pasajeros como automóviles. El ferry tiene una capacidad limitada para pasajeros y automóviles, y se debe garantizar un proceso de embarque y desembarque seguro y eficiente. Utiliza monitores para implementar este sistema de manejo de ferry. Diseña un programa en Java que modele la interacción entre el ferry, los pasajeros y los automóviles de manera que los pasajeros llegan al puerto y desean abordar el ferry. Cada pasajero ocupa un espacio en el ferry, los automóviles llegan al puerto y desean subir al ferry. Cada automóvil ocupa un número determinado de espacios en el ferry. Si no hay suficiente espacio en el ferry para acomodar a un pasajero o un automóvil, deben esperar hasta que haya espacio disponible. Los pasajeros y los automóviles pueden desembarcar del ferry SOLO cuando llega a su destino. Utiliza un monitor para coordinar la carga y descarga de pasajeros y automóviles, garantizando la exclusión mutua y la sincronización adecuada. a. Implementa el programa de manera que los pasajeros y los automóviles puedan abordar y desembarcar del ferry de manera segura y eficiente.



### 3. Observatorio.

Se realiza una visita al observatorio de la ciudad. El mismo permite el ingreso de visitantes, personal de mantenimiento e investigadores.

- los visitantes, que van a estudiar las estrellas,
- las personas de mantenimiento, que se ocupan de controlar que todo esté aseado,
- Los investigadores analizan estrellas y realizan su trabajo de investigación.

La sala tiene capacidad para 50 personas, pero si entra algún visitante en silla de ruedas, la capacidad baja a 30 personas hasta que se retire. Cuando se retira, la capacidad vuelve a ser de 50 personas. Puede ocurrir que el visitante quiera ingresar cuando la capacidad sea mayor a 30, en ese caso debe esperar a que la capacidad disminuya.

Existen cierta restricciones impuestas por el director del observatorio:

- Los visitantes pueden ingresar cuando haya otros visitantes, siempre que no superen la capacidad.
- Las personas de mantenimiento ingresan periódicamente, siempre que no haya visitantes, aunque sí puede haber otras personas de mantenimiento.
- Los investigadores tienen exclusividad, es decir deben estar solos en el recinto ya que se tienen que concentrar para realizar sus observaciones. Las observaciones deben ser guardadas en el libro de observaciones diarias.
- Si un investigador quiere entrar y hay visitantes y/o personal de mantenimiento, debe esperar a que se retiren.

Identificar los roles activos y los recursos compartidos en el escenario presentado.

Dar una solución que modele el comportamiento explicado, utilizando mecanismos de sincronización.

Identificar problemas de inanición que puedan surgir y cómo podría solucionarlo.

### 4. El puente estrecho.

Coches que vienen del norte y del sur pretenden cruzar un puente sobre un río. Solo existe un carril sobre dicho puente (de una sola vía, es decir que los coches en la misma dirección no pueden adelantarse). Por lo tanto, en un momento dado, sólo puede ser cruzado por uno o más coches en la misma dirección (pero no en direcciones opuestas).

a. Implementar la solución con semáforos.

b. Implementar teniendo en cuenta el código del siguiente monitor, que resuelva el problema del acceso al puente.

```
MONITOR GestionaTrafico
  Operación EntrarCocheDelNorte
  Operacion SalirCocheDelNorte
  Operacion EntrarCocheDelSur
  Operacion SalirCocheDelSur
```

c. Mejorar la implementación del monitor anterior, de forma que la dirección del tráfico a través del puente cambie cada vez que lo hayan cruzado 10 coches en una dirección, mientras 1 o más coches estuviesen esperando cruzar el puente en dirección opuesta.



**IMPORTANTE:** • Debe considerar que la implementación asegure que los coches crucen el puente en el orden en que llegaron, es decir, si C1 llegó al puente antes que C2, en la misma dirección, C1 debe poder empezar y terminar de cruzar el puente antes que C2

## 5. Pastelería

Una pastelería tiene tres hornos que producen tres tipos diferentes de pasteles: A, B y C, con pesos diferentes: *pesoA*, *pesoB* y *pesoC*. Procedentes de los hornos, los pasteles se van situando en un mostrador común.

Los pasteles son empaquetados en cajas. Uno o varios robots Empaqueadores toman pasteles del mostrador y los introducen en la caja. Cada caja puede contener un número diferente de pasteles siempre y cuando no se sobrepase un peso límite, denominado *PesoMaximo*. Por este motivo, antes de incluir un pastel en la caja, cada empaqueador debe asegurarse de que con su inclusión no se sobrepasa el peso máximo. Si no se sobrepasa el peso, se incluye el pastel en la caja; en otro caso, un **brazo** mecánico se encarga de retirar la caja que se estaba llenando y posteriormente la sustituye por una caja vacía.

Tener en cuenta de que se trata de llenar cada caja lo más posible, lo cual puede ser conseguido por uno cualquiera de los robots que intentan depositar simultáneamente algún pastel, de pesos que pueden variar. Se considera que no hay interferencia física entre robots que intentan soltar ubicar pasteles al mismo tiempo en la caja.

Se asume que inicialmente hay una caja vacía junto al mostrador.

Considere las siguientes operaciones:

Retirar Caja: *hace que el proceso que lo invoca quede bloqueado hasta que la caja que estaba siendo llenada es retirada por el brazo auxiliar*. Requiere que haya una caja en la zona de relleno.

Reponer Caja: *hace que el proceso que lo invoca quede bloqueado hasta que el brazo auxiliar coloque una caja vacía en el área de relleno. No debe haber ninguna caja en la zona de relleno.*

Tomar Pastel, retorna Peso: *provoca que el proceso que lo invoca quede bloqueado hasta que el empaqueador toma el pastel más cercano al mostrador, indicando Peso el peso del mismo.*

Soltar Pastel: *provoca que el proceso que lo invoca quede bloqueado hasta que el empaqueador suelta el pastel que acaba de tomar en la caja del área de relleno. Es necesario que haya una caja en el área de relleno, que el robot empaqueador en cuestión tenga un pastel y que la inclusión de ese pastel en la caja no haga sobrepasar el peso máximo permitido. Físicamente, no hay interferencia entre dos robots que intentan depositar pasteles simultáneamente en una misma caja.*

Implementar utilizando los mecanismos de sincronización que crea más conveniente.



**Facultad de Informática**  
**Programación Concurrente - Departamento de Programación**  
**2025**

