

# Prácticas de Programación Concurrente y Distribuida

3º Curso de Grado en Ingeniería Informática

## PRÁCTICA 6

Semáforos en JAVA

El objetivo de la práctica es familiarizarse con el uso de semáforos en Java para la sincronización de hilos.

Un taller de folletos fabrica folletos de 4 páginas. Para crear dichos folletos dispone de cuatro brazos de robot iguales, que cogen cada uno una hoja del montón de la página que procesan y la llevan a una mesa de grapado. Una vez que las cuatro páginas están en la mesa, hay otro robot *grapador* que coge las cuatro páginas de la mesa, las grapa y las pasa a una cinta transportadora.

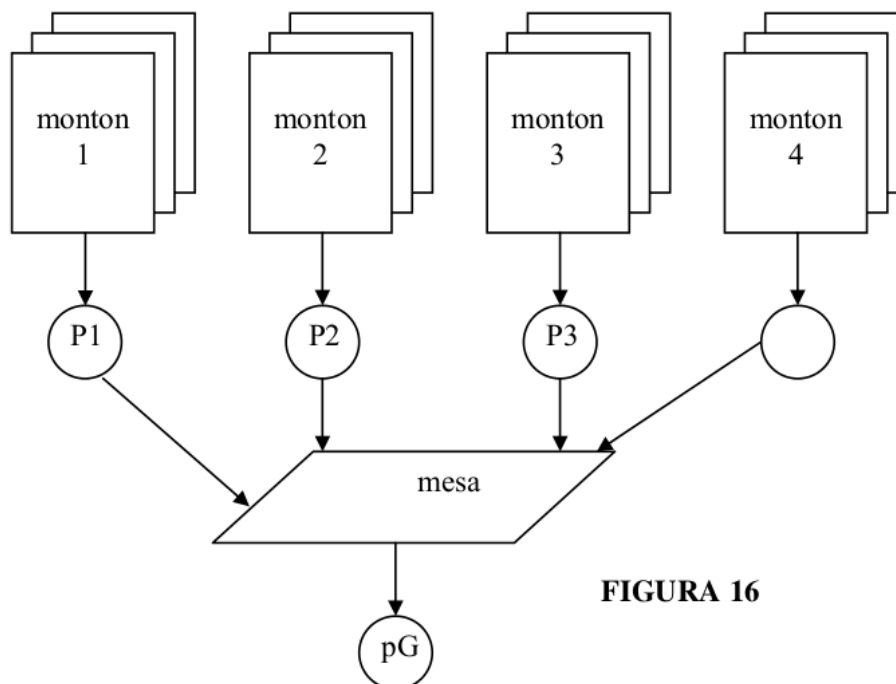


FIGURA 16

Se debe solucionar el problema anterior usando semáforos. Se permite implementar la clase semáforo o usar `java.util.concurrent.semaphore`.

El proyecto base para solucionar el enunciado contendrá las siguientes clases:

- **RobotA.** Representará cada uno de los robots que cogen las hojas mediante un hilo. El hilo se creará heredando de la clase `Thread`. El robot cogerá una hoja de su montón y lo llevará a la mesa de grapado. No podrá llevar otra hoja hasta que la mesa de grapado esté libre. El tiempo que cada robot tarda en recoger una nueva hoja será aleatorio entre 1 y 3 segundos. Cada robot trasladará 10 hojas antes de finalizar.
- **RobotG.** Representa al robot *grapador* mediante un hilo. El hilo se creará implementando el interface `Runnable`. El tiempo empleado por el robot en grapar y depositar el folleto en la cinta será de 2 segundos. Este robot será interrumpido por el generador cuando los otros acaben.
- **Generador.** Contendrá el método `main` y será quién comience la ejecución mediante un `Frame`. Debe lanzar los cuatro robots tipo 'A' y al robot grapador, esperar a que finalicen los robots RA e interrumpir al robot *reponedor*.

Con objeto de visualizar la correcta evolución de los hilos, se deberá realizar la presentación gráfica en pantalla de su estado. Para tal fin, se deberá incluir en el *Frame* un *canvas*. El *canvas* deberá mostrar en cada momento el estado de la mesa de grapado así como la actividad de los robots.

**NOTA: No se permite crear ninguna otra clase aparte de las tres descritas anteriormente y el *canvas*.**