# GUIÓN DE ACTIVIDAD

# PSP/Tema 2-Act 3

## Título de la Actividad

Actividad 3: Secciones críticas

## Objetivos

###### Identificar de manera correcta los conceptos desarrollados en la parte teórica a través de ejercicios prácticos.

## Temporalización

###### La duración de esta actividad está prevista en 3 horas

## Proceso de desarrollo

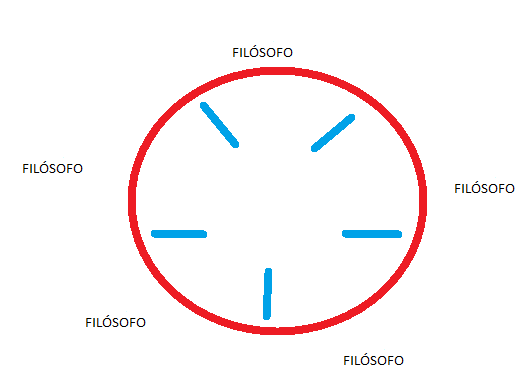
###### El alumno deberá repasar los conceptos expuestos en los recursos .

**Problema de los filósofos**

En una mesa hay varios filósofos que comen y piensan. Cada filósofo tiene un plato para él, pero sólo un cubierto a su izquierda y otro a su derecha y para poder comer tiene que conseguir los dos. Los filósofos siempre tienen el mismo comportamiento:

* Intentan conseguir los dos cubiertos para comer. Si los consiguen, tardan una cantidad aleatoria de tiempo entre 1000 y 5000 milisegundos en acabar el plato. Una vez acaban de comer, sueltan los cubiertos y se ponen a pensar una cantidad aleatoria de tiempo entre 1000 y 5000 milisegundos.
* Si no consiguen los cubiertos, se ponen a pensar una cantidad aleatoria de tiempo entre 1000 y 5000 milisegundos.
* Después de pensar, los filósofos vuelven a intentar comer.

La siguiente imagen muestra una mesa de ejemplo con 5 filósofos y 5 cubiertos.



**Desarrollo**

1. En primer lugar, crea la clase GestorCubiertos. Esta clase tendrá un array de n cubiertos dónde cada cubierto deberá estar marcado como disponible o no disponible. El número de cubiertos n será especificado en su constructor. Debe ofrecer, como mínimo los siguientes métodos:
   1. cogerCubiertos(int pos1, pos2): si están disponibles los palillos en las posiciones pos1 y pos2 los marcará como no disponibles y devolverá true. Si no, devolverá false.
   2. liberarCubiertos(int pos1, int pos2): marcará los cubiertos en las posiciones pos1 y pos2 como libres.
2. En segundo lugar, crea la clase filósofo. Un filósofo necesita conocer al gestor de cubiertos y la posición del cubierto a su derecha y del cubierto a su izquierda. La clase debe implementar, como mínimo, los siguientes métodos:
   1. Su método principal que será un bucle sin fin en el que realizarán las acciones explicadas anteriormente.
   2. comer(): este método esperará una cantidad aleatoria entre 1000 y 5000ms y mostrará por pantalla el siguiente texto: Filósofo <nombreFilósofo> comiendo nyam nyam.
   3. pensar(): este método esperará una cantidad aleatoria entre 1000 y 5000ms y mostrará por pantalla el siguiente texto: Filósofo <nombreFilósofo> pensando mmh mmh.
3. Por último, crea el programa principal. Define un valor constante N\_FILOSOFOS que será como mínimo 5. El programa principal creara N\_FILOSOFOS, un gestor de cubiertos y dará “vida” a los filósofos.

## Evaluación

###### La valoración de esta actividad será de un% de la nota reservada para entregables de clase:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Insuf | Suf | Bien | Notable | sobresaliente |
| Realización de código pero no compila | Compila pero no se ejecuta correctamente | Compila pero no se ejecutan las prioridades correctamente | Resolución correcta pero sin aportes de comentarios | Realización del problema correctamente **y el código está comentado** |

## Recursos

###### Vídeos y ejemplos de la zona de recursos didácticos, y recursos complementarios.

###### Herramientas:

* *Eclipse.*
* *JDK*