# COMPUTACIÓN CONCURRENTE

PRÁCTICA 2

1. **Objetivo**

El objetivo de esta práctica es resolver un problema de concurrencia de recursos teniendo evitando el problema ibre de hambruna y exclusión mutua.

1. **Desarrollo**

El programa debe ser desarrollado en Java, particularmente en maven como se encuentra en la carpeta del proyecto. El programa debe estar debidamente comentado, modularizado y con buen tiempo de ejecución.

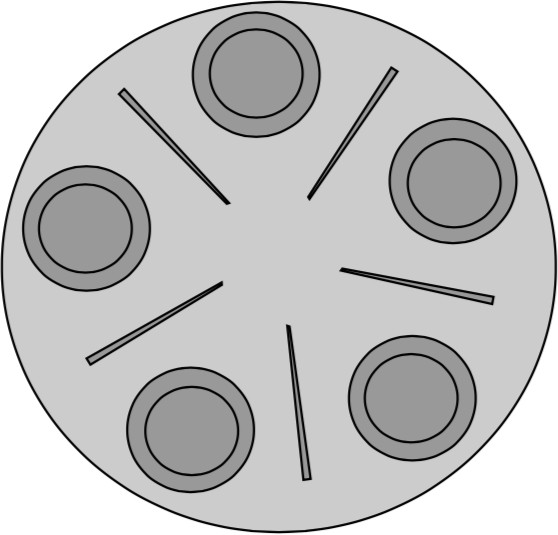
## Evaluación

Deberás realizar la implementación del problema de los filósofos en la carpeta adjunta de la práctica 2. En está práctica no habrá pruebas unitarias. La implementación del código valdrá el 80% de la calificación y las preguntas el otro 20 %.

## Problema de los Filósofos

El problema de los filósofos fue inventado por E. W. Dijkstra, un pionero de la concurrencia, para clarificar la noción de algoritmos libres de ***abrazos mortales*** y de ***hambruna*** . Imagina cinco filósofos que solo pasan su vida pensando y comiendo.

Ellos se sientan alrededor de una mesa redonda con cinco sillas. La mesa tiene cinco platos grandes de arroz, sin embargo, solamente hay cinco palillos disponibles (un poco insalubre), como se muestra en la figura. Cada filósofo piensa, luego le da hambre, por lo que se sienta y toma los dos palillos que se encuentran a su lado. Si el filósofo pueda tomar ambos palillos, entonces procede a comer por un rato. Una vez que el filósofo termina de comer, éste regresa los palillos al lugar de donde los tomó y de nuevo se pone a pensar.



Tu programa debe resolver el problema de los filósofos, tomando en cuenta el problema de abrazos mortales y libre de hambrunas. Recibirá un entero mayor o igual a 5 que representaran a los filósofos, mandar un mensaje el estado de cada filósofo y cuando entra a su sección crítica y cuando todos hallan comido por lo menos una vez mandar un mensaje de que todos han comido.

Puedes utilizar las librerías de hilos y semáforos de java, pero si implementas el TDA obtendrás puntos extras. No olvides documentar ampliamente tu programa y si quieres utilizar una librería poco común agregala al pomp.

2 puntos extras.

Si implementas TDA de hilos y de Semáforos obtendrás dos puntos extras.

## Preguntas.

## Deberán detallar a profundidad las respuestas y en caso de ser necesario hacer diagramas para ejemplificar tu respuesta.

## ¿Qué es un proceso? 1 punto

## ¿Qué es la sección crítica de un proceso? 1 punto

## ¿Qué es el problema libre de hambruna? 2 puntos

## ¿Qué es el problema de abrazos mortales? 2 puntos

## Haz un TDA de Hilos y un TDA de Semáforos de 4 puntos