

## Práctica 9. Fumadores en Java

Implementar por equipo de prácticas el siguiente enunciado de procesos cooperativos en Java.

### Enunciado del problema

Considere una organización con tres fumadores y un agente. Cada fumador continuamente enrolla un cigarrillo y luego lo fuma. Pero para enrollar y fumar un cigarrillo, el fumador necesita 3 ingredientes: tabaco, papel y cerillos.

Cada fumador tiene asignado un ingrediente de por vida, es decir, un fumador tiene papel, otro tabaco y el último tiene cerillos. Por otro lado, el agente tiene un suministro infinito de los tres materiales o ingredientes.

El agente coloca dos de los tres ingredientes en una mesa de manera aleatoria.

El fumador que tiene el ingrediente restante toma el ingrediente y hace entonces un cigarrillo y lo fuma, indicándole al agente cuando termina.

Una vez de que el agente se entera que el fumador terminó su cigarrillo pone otros dos de los tres ingredientes, y el ciclo se repite de manera infinita.

Mensajes **mínimos** de impresión:

Imprimir cuando se crean los hilos de los fumadores indicando el **nombre** del ingrediente asignado.

Imprimir los nombres de los ingredientes que genera el agente.

Imprimir que ingredientes genera el agente y cuando los pone en la mesa.

Imprimir cuando el fumador intenta fumar indicando el ingrediente que tiene.

Imprimir cuando el fumador fuma.

Imprimir cuando el fumador termina de fumar.

Imprimir si el fumador no puede fumar y por qué.

### NOTAS de implementación:

1. Tener su código con comentarios.
2. Imprimir en pantalla cada acción que esté realizando en su programa.
3. Cada clase de Java debe estar en un archivo.
4. Los programas solo serán revisados en Java bajo el sistema operativo de Linux, modo terminal.
5. El proceso padre solo debe de crear los recursos compartidos y los hilos. Además de iniciar la ejecución de los hilos.
6. Cuando el padre crea a cada fumador le asigna un ingrediente el cual va a tener durante toda su ejecución y no va a cambiar. Es decir a un fumador le va asignar papel, a otro cerillos y al último tabaco.
7. Un hilo no puede mandarse como parámetro a otro hilo.
8. Debe de implementar la clase enlace que será la encargada de realizar la sincronización y comunicación entre los hilos.
9. Los hilos deben de ejecutarse de manera infinita y no deden tener código que realice sincronización. El manejo de los recursos compartidos deberá ser a través de la invocación de los métodos de la clase enlace.