

# CI0117 - Laboratorio 4

---

## Instrucciones generales

**Fecha de entrega:** Lunes 21 de Setiembre 11:50pm.

Este laboratorio se podrá realizar en grupos de máximo 2 personas. Por favor, apuntar el nombre de los dos integrantes de su grupo [en este documento](#) en cualquier slot a más tardar el **Martes 15 de Setiembre a las 11:50pm**. Si no tiene grupo, apunte solamente su nombre dentro cualquier slot o coordine con otro de los estudiantes que se haya apuntado sin pareja.

Las soluciones a los ejercicios de este laboratorio deberán ser subidas a su repositorio *CI0117-2020-S2* creado para este curso en Github. Dentro de dicho repositorio se deberá crear un folder con el nombre *laboratorios*. La solución a cada laboratorio deberá estar contenida dentro de un folder con el respectivo número dentro del folder *laboratorios*. A su vez, la solución al ejercicio deberá subirse en un folder con el nombre indicado entre paréntesis cuadrados.

Para los ejercicios que, además de código, también requieran responder preguntas sobre la solución, se deberá crear un archivo *README.md* con las respuestas dentro del folder del ejercicio respectivo. Agregue un archivo *Makefile* para cada uno de los ejercicios con los comandos necesarios para compilar el código fuente, de forma que el ejecutable tenga el mismo nombre dado en el ejercicio. Asegúrese de subir al repositorio únicamente archivos *.c*, *.md* y *Makefile*. Cree un archivo *.gitignore* para evitar subir archivos ejecutables.

Únicamente uno de los integrantes del grupo requiere subir la solución.

## Ejercicio 1 [Nombre a definir por el profesor]

---

Verifique el problema o patrón de sincronización asignado por el profesor a su grupo [en este documento](#). Utilice el nombre entre paréntesis cuadrado dado en ese documento como nombre del folder y archivo de código de su solución. Deberá investigar acerca del problema o patrón asignado y realizar lo siguiente:

- Descripción del problema / patrón de sincronización.
- Conteste la pregunta: ¿Se puede presentar alguna situación de *deadlock* o *starvation*? Explique.
- Agregue alguna ilustración o pseudocódigo que permita visualizar la solución al problema.
- Implemente un programa en C utilizando Pthreads donde se haga una simulación de la solución al problema o patrón asignado. Asegúrese de que al ejecutar su programa se pueda observar de forma clara lo que ocurre en cada hilo (Puede hacer uso de funciones de *sleep* aleatorios en hilos).

Su grupo debe estar preparado para explicar durante la clase la solución y realizar una demostración del programa implementado.

### 1) Descripción del problema / Patrón de sincronización :

Se asume que un cigarro está compuesto por 3 ingredientes: tabaco, papel y cerillos . Hay 3 fumadores alrededor de una mesa, cada uno tiene una cantidad infinita de uno de los ingredientes y limitada de los otros 2. También hay un agente que no es fumador que habilita a los fumadores para que creen sus cigarrillos de forma arbitraria escogiendo 2 de los ingredientes para poner en la mesa. El fumador que tiene el tercer ingrediente debe remover los 2 ingredientes de la mesa, usándolos con su ingrediente infinito para hacer el cigarro y luego fumarlo. Una vez que este termina de fumar, el agente pone 2 nuevos ingredientes en la mesa y se repite el proceso.

Los semáforos son utilizados para representar los ingredientes en la mesa, el agente incrementa el semáforo apropiado para señalar que el ingrediente se puso en la mesa, los fumadores decrementan el semáforo cuando los quitan de la mesa. Además cada fumador tiene asociado un semáforo que usan para avisar al agente que un fumador en particular terminó de fumar, el agente tiene un proceso que espera en cada fumador para hacer ver al agente si puede poner más ingredientes en la mesa

### 2) Deadlock:

Ejemplo: si un agente pone tabaco y papel, entonces como el fumador con cerillos espera por tabaco, puede ser desbloqueado, pero el fumador con tabaco está esperando por papel también puede ser desbloqueado, así que el primer hilo va a ser bloqueado en papel, además del segundo que será bloqueado en cerillos y terminaría en deadlock.

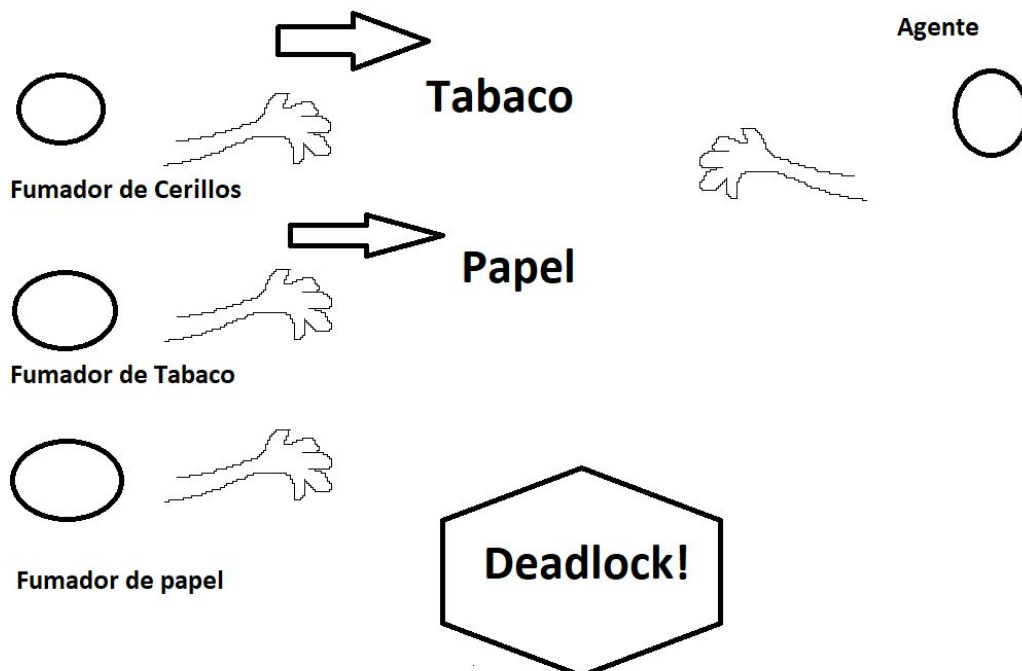


Imagen descriptiva del problema:

