



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO**  
**FACULTAD DE INGENIERÍA**



**SISTEMAS OPERATIVOS.**

Profesor: Gunnar Wolf

Grupo: 06

**TAREA 1:**

“Ejercicios de sincronización”

**Alumno:**

Aquino Lozada Gabriela

**Fecha de entrega:**

22/Octubre/2024

Semestre 2025-1

**Ejercicio:** Los alumnos y el asesor.

**Planteamiento:** Un profesor de la facultad asesora a varios estudiantes, y estamos en su horario de atención. Modelar la interacción durante este horario de modo que la espera (para todos) sea tan corta como sea posible.

### **Reglas:**

- Un profesor tiene  $x$  sillas en su cubículo:
  - Cuando no hay alumnos que atender, las sillas sirven como sofá, y el profesor se acuesta a dormir la siesta.
- Los alumnos pueden tocar a su puerta en cualquier momento, pero no pueden entrar más de  $X$  alumnos.
  - $X$  tiene un valor de 5 alumnos.
- Para evitar confundir al profesor, sólo un alumno puede presentar su duda (y esperar a su respuesta) al mismo tiempo.
  - Los demás alumnos sentados deben esperar pacientemente su turno.
  - Cada alumno puede preguntar desde 1 y hasta  $Y$  preguntas (permitiendo que los demás alumnos pregunten entre una y otra).
  - $Y$  tiene un valor random que puede generar entre 1 y 2 dudas por alumno.

### **Lenguaje y entorno de desarrollo:**

Se utilizo el lenguaje Python, versión 3.13.

Para ejecutar el programa, es necesario tener un equipo/herramienta que pueda correr programas en Python. Si utiliza la terminal solo basta ubicarse en la carpeta donde se encuentra el programa y ejecutar la instrucción:

```
$ ./alumno_asesor_AG.py
```

### **Estrategia:**

Se implementaron varias **estrategias de concurrencia y sincronización** para gestionar correctamente el acceso compartido a los recursos y la comunicación entre los hilos (profesor y alumnos).

#### **1. Semáforos:**

- a. **sem\_profesor:** Controla el momento en que el profesor está disponible para resolver dudas. Un alumno libera este semáforo para indicar que está listo para ser atendido, y el profesor lo adquiere para empezar a resolver una duda.
- b. **sem\_alumno:** Gestiona la llegada de alumnos. El profesor está bloqueado en `sem_alumno.acquire()` hasta que un alumno lo libera señalizando que ha llegado al salón
- c. **sem\_duda:** Permite que los alumnos esperen hasta que puedan resolver sus dudas, en caso de que el salón esté lleno o haya más alumnos con dudas pendientes. Este semáforo gestiona que un alumno espere su turno para ser atendido después de que otro termine.

## 2. Mutex:

- a. **mutex:** Protege el acceso a la variable `num_sillas_ocupadas`, que lleva el control del número de alumnos presentes en el salón. Este recurso compartido solo puede ser modificado por un hilo a la vez. Esto garantiza que los alumnos entren y salgan del salón sin que los valores de las sillas ocupadas se corrompan.

## 3. Señalización:

- a. **Entre profesor y alumnos:** Los semáforos `sem_alumno` y `sem_profesor` actúan como mecanismos de señalización entre los hilos del profesor y los alumnos, asegurando que el profesor no intente atender a un alumno hasta que este haya llegado, y que los alumnos esperen su turno para ser atendidos.
- b. **Entre alumnos:** El semáforo `sem_duda` gestiona la espera de los alumnos que tienen dudas pendientes, lo que asegura que los alumnos tomen turnos de manera ordenada para resolver sus dudas con el profesor.

# DUDAS:

Hay una situación con el código, al gestionar las entradas de los alumnos al salón, nos aseguramos de que el profesor resuelva las dudas de manera ordenada, sin exceder la capacidad del salón.

El número de **sillas ocupadas** debería actualizarse correctamente, de modo que nunca haya más de 3 alumnos dentro del salón simultáneamente. Lo cual no está sucediendo. El número de sillas ocupadas a veces se vuelve negativo.

En otras ocasiones, el número de sillas ocupadas llega a cero mientras aún hay alumnos que están esperando para resolver sus dudas.

Si no me equivoco, el problema esta relacionado con la variable `num_silla_ocupadas`, ya que el acceso a esta variable depende de una protección mediante un mutex. Pero no eh podido solucionar el problema, lo cual me vi comprometida por el tiempo a entregar el código con ese error 😞.