

# Ejercicios Multihilo (Sincronización II)

En el presente documento se adjuntan 2 ejercicios para practicar la programación en multihilo con sincronización.

## Ejercicio 1

En este ejercicio vamos a trabajar los semáforos (`from threading import Semaphore`)

Para declarar un semáforo: `sem = Semaphore(WIDTH)` donde `WIDTH` indica la cantidad de hilos que pueden entrar en la sección crítica (1..N)

Realizar un programa que genere 20 hilos a partir del programa principal.

Cada hilo ejecutará una función en la que simula el acceso a un tunel de lavado de coches. Al tunel se accede de uno en uno, y se necesitan **5 segundos** para finalizar el lavado del coche.

Debemos **cambiar el nombre del hilo** por `Coche1`, `Coche2`,... desde el 1 hasta el 20.

La función que ejecuta cada hilo deberá:

1. Imprimir el mensaje: **"El CocheX en espera"**.
2. Espera **2 segundos** para adquirir el semáforo (`.acquire()`).
3. Cuando adquiera el semáforo mostrará el mensaje: **"Iniciando lavado del CocheX"**.
4. Tardará **5 segundos** en el lavado.
5. Libera el semaforo (`.release()`).
6. Imprime el mensaje: **"El CocheX ya está limpio"**.

Finalmente, desde el programa principal, esperamos a que finalice todo el lavado de coches y mostrará un mensaje final diciendo **"El programa de lavado ha finalizado"**.

## Ejercicio 2:

En este ejercicio probamos a trabajar con Conditions (`from threading import Condition`).

Realizar un programa que genere 6 hilos simulando 6 alumnos/as de DAM2 comprando entradas de un espectáculo. Selecciona 6 nombres de la clase (diferentes para cada uno) y **asignale ese nombre al hilo**.

Para simplificar nuestro espectáculo tendrá **5 butacas** lo que significa que cada alumno/a se sentará en una butaca, excepto uno que se quedará sin asiento. Nuestras butacas puedes representarlas con una lista de cinco valores vacíos (""), que significan que están las butacas vacías.

Inicialmente cada alumno esperará un número aleatorio de segundos [1..5] simulando la carga de la aplicación de reserva. Posteriormente genera un número aleatorio de 0 a 4 que representa el asiento que quiere reservar. **El proceso de reserva debe ser en exclusión mutua para evitar que dos personas reserven a la vez.**

- **Si el asiento está vacío** significa que puede reservarlo. Este proceso será:
  1. Esperar 1 segundo simulando el trámite de reserva del asiento
  2. Reservar el asiento (poner su nombre en la posición de la lista).
  3. Esperar 1 segundo simulando la confirmación.
  4. Mostrar un mensaje que diga: **<NombreDelAlumno>** ha reservado el **asiento Y** (0..4).
- **Si el asiento no está vacío** significa que NO puede reservarlo. Este proceso hará:
  1. Mostrar un mensaje diciendo que **<NombreDelAlumno>** no puede reservar el **asiento Y** (0..4) por que está ocupado.
  2. Esperar 2 segundo y volver a intentar reservar.

Cada uno de los hilos de reserva finaliza cuando consiga hacer la reserva con éxito o cuando no queden asientos.

El proceso principal esperará a que se hayan reservado todas las butacas y mostrará una lista de las personas y qué asiento han reservado. También dirá quien se ha quedado sin asiento. Por ejemplo:

Reservadas: [Adrián, Ivan, Mario, Laura, Luca]  
Cesar no tiene asiento