

# Documentación Práctica 1

Jaime Frías Funes

50625384S

## Problema de productor-consumidor

Para resolver este problema, he utilizado tres semáforos:

- **Semáforo para producir** (*puede\_producir*). Este permite que se produzcan datos desde la función productora. Inicializado al tamaño del vector.
- **Semáforo para consumir** (*puede\_consumir*). Permite que se consuman. Inicializado a 0, inicialmente no se puede consumir.
- **Semáforo de exclusión mutua** (*mutex*). No permite la superposición de datos a la vez durante la ejecución. Inicializado a 1.

### Función productor

Esta función produce datos *int* usando la función dada *producir\_dato()*, y los inserta en el búfer *buf[]* para que la función consumidora los vaya consumiendo, se espera al semáforo productor y cuando finaliza llama al semáforo consumidor.

### Función consumidor

La función consume datos que han sido previamente producidos por la función productora y esta usando los semáforos explicados arriba de modo que a la vez que utilizando el búfer *buf[]* se permita consumir y producir a la vez de forma controlada. Al comienzo espera al semáforo consumidor y cuando finaliza llama al semáforo productor.

### Main

Se inician tanto los semáforos como las hebras y posteriormente cuando finalizan se destruyen, como en cualquier problema de este tipo.

## Problema de los fumadores

Para resolver este problema, he creído conveniente crear **4 semáforos**:

- **3 para los fumadores**. Se crea un array de semáforos de tamaño 3 para simplificar el código y también por si en un futuro se desea añadir más al problema.
  - o Se inician a 0, comienzan fumando.
- **1 para el estancero**.
  - o Se inician a 1, ya que el estancero debe esperar la primera vez a los fumadores.

He asignado cada ingrediente a un número en la variable *int ingrediente*, cada ingrediente asociado a un fumador que lo necesita para fumar.

### Función fumar

Ya que las hebras no permiten el paso de argumentos de tipos de datos normales usamos un puntero y luego se convierte a tipo de dato (*unsigned long*) para utilizarlo en la función. Esta consiste en un bucle infinito que espera a cada fumador y llama al estancero cuando finaliza.

### Función producir

Esta función tiene otro bucle infinito que espera al estancero de la función anterior y genera un ingrediente aleatorio con la función *rand()* y llama a cada fumador dependiendo de su ingrediente demandado.

### Main

Se inician las hebras y los semáforos, y en la creación de hebras, al necesitar como argumento un puntero, se convierte el *unsigned long* a puntero, posteriormente se destruyen los semáforos como en cualquier problema de semáforos.