# Programación Avanzada(TC2025)

## Tema 5. Programación concurrente

Instituto Tecnológico y de Estudios Superiores de Monterrey. Campus Santa Fe
Departamento de Tecnologías de Información y Electrónica
Dr. Vicente Cubells (vcubells@itesm.mx)

# Temario

- Los hilos de POSIX
  - Mutexes y regiones críticas
    - Problema de la suma y resta de números
  - Problemas con el orden de los mutexes
    - Ver ejemplo práctico
  - Uso de variables de condición
    - Solución al problema del productor-consumidor

# Ejemplo 1

- Problema de operaciones inversas sobre una variable global sin protección
  - Ejemplo de suma y resta de la misma cantidad de números
  - Simular que un proceso es más rápido que otro

# Ejercicio 2

- Solucionar el problema anterior mediante la protección a la región crítica (variable global) usando mutexes
  - pthread_mutex_lock()
  - pthread_mutex_unlock()
  - Inicialización estática:
    - pthread_mutex_t variable = PTHREAD_MUTEX_INITIALIZER

# Ejercicio 3

- Demostrar el problema de bloqueos potenciales cuando se utilizan dos mutexes para proteger dos variables globales diferentes y se invierte el orden de los bloqueos en cada hilo

# Ejercicio 4

- Solucionar el problema anterior utilizando la función pthread_mutex_trylock() y liberando bloqueos
  - Ocasiona espera activa
  - Mala solución porque desperdicia tiempo de procesador

# Ejercicio 5

- Solución al problema anterior utilizando variables de condición
  - Se elimina la espera activa
  - Más eficiente
  - Uso de funciones
    - pthread_cond_wait()
    - pthread_cond_signal()
    - Inicialización estática:
      - pthread_cond_t variable = PTHREAD_COND_INITIALIZER

# Ejercicio 6

- Solución al problema del productor-consumidor utilizando variables de condición
  - Modelación para un solo productor y un solo consumidor

# Ejercicio 7

- Extensión de la solución anterior para N productores y M consumidores
  - Uso de la función pthread_cond_broadcast()

# Ejercicio 8

- Resolver el problema del productor-consumidor con semáforos en lugar de utilizar variables de condición

# Resumiendo

- Las regiones críticas deben protegerse mediante mutexes
- El orden en que se establecen los bloqueos es determinante para no producir deadlocks
- Los mutexes no se utilizan para sincronización entre procesos (a quien le corresponde el turno)
- Para lo anterior se utilizan variables de condición o semáforos
- Las variables de condición no ocasionan “espera activa”