

# Repaso Parcial

🕒 Created time	@17 de octubre de 2024 10:44
📅 Date	@17 de octubre de 2024
⚙️ Estad	Revisar
☰ Materia	SISOP
☰ Tipo	

## 1. Lock (bloqueo)

### Cuándo usarlo:

Usa un **lock** cuando quieres asegurarte de que solo **una persona** a la vez pueda acceder o modificar un recurso específico. Piensa en esto como una **llave** para una puerta: si alguien tiene la llave, nadie más puede entrar hasta que esa persona salga y devuelva la llave.

### Ejemplo cotidiano:

Imagina que tienes **una única computadora** en casa y varias personas quieren usarla. Para evitar que dos personas intenten usarla al mismo tiempo, la computadora tiene una llave. Cuando una persona la usa, cierra la puerta con llave. Cuando termina, abre la puerta y otra persona puede usarla.

```
import threading

lock = threading.Lock()

def usar_computadora():
    lock.acquire()
    try:
```

```
print("Usando la computadora...")
# Solo una persona puede acceder a este recurso a la vez
finally:
    lock.release()
```

## 2. Semaphore (semaforo)

### Cuándo usarlo:

Usa un **semaforo** cuando quieres permitir que un **número limitado de personas** acceda a un recurso compartido simultáneamente. Es como tener **varias llaves**, pero solo un número específico de llaves disponible.

### Ejemplo cotidiano:

Imagina que en una cafetería hay **solo 3 mesas**. Si 3 personas (o grupos) están ocupando las mesas, las siguientes personas tienen que esperar hasta que se libere una mesa. No pueden entrar más de 3 grupos a la vez

```
import threading

semaforo = threading.Semaphore(3) # Solo 3 personas pueden usar

def usar_mesa():
    semaforo.acquire()
    try:
        print("Usando una mesa en la cafetería...")
        # Solo 3 personas pueden estar aquí a la vez
    finally:
        semaforo.release()
```

## 3. Event (evento)

## Cuándo usarlo:

Usa un **evento** cuando quieres que varios hilos esperen hasta que **una señal** les indique que pueden continuar. Es como esperar una **luz verde** en un semáforo para poder avanzar.

## Ejemplo cotidiano:

Imagina que en una fiesta, todos están esperando que **el anfitrión dé una señal** (como encender las luces) para comenzar a comer. Hasta que no reciban la señal, nadie empieza.

```
import threading

evento = threading.Event()

def comenzar_fiesta():
    print("Esperando que el anfitrión encienda las luces...")
    evento.wait() # Los invitados esperan hasta recibir la señal
    print("¡La fiesta ha comenzado!")

def encender_luces():
    print("El anfitrión encendió las luces.")
    evento.set() # Envía la señal de que la fiesta puede comenzar

# Simula la fiesta
threading.Thread(target=comenzar_fiesta).start()
threading.Thread(target=encender_luces).start()
```

## 4. Condition (condición)

### Cuándo usarlo:

### Ejemplo cotidiano:

Usa una **condición** cuando varios hilos necesitan esperar a que ocurra una **condición específica** antes de continuar. Es similar a un **evento**, pero con más control, ya que puedes usar varias señales o condiciones.

Imagina que varias personas están esperando en una **fila para entrar a un cine**, pero la entrada solo se abre cuando las entradas están listas. Hasta que no estén listas, todos esperan.

```
import threading

condicion = threading.Condition()
entradas_listas = False

def entrar_al_cine():
    with condicion:
        print("Esperando que las entradas estén listas...")
        condicion.wait() # Esperan hasta que las entradas estén
        print("Entrando al cine...")

def preparar_entradas():
    global entradas_listas
    with condicion:
        print("Las entradas están listas.")
        entradas_listas = True
        condicion.notify_all() # Avisa a todos que las entradas

# Simula la situación
threading.Thread(target=entrar_al_cine).start()
threading.Thread(target=preparar_entradas).start()
```

## 5. Barrier (barrera)

**Cuándo usarlo:**

**Ejemplo cotidiano:**

Usa una **barrera** cuando varios hilos deben esperar a que **todos** lleguen a un punto antes de seguir adelante. Es como esperar que todos **estén listos** para comenzar algo al mismo tiempo.

Imagina que estás organizando una **carrera de autos**. Todos los autos deben alinearse en la línea de salida y nadie puede comenzar hasta que **todos** estén listos.

```
import threading

barrera = threading.Barrier(3) # Necesitamos 3 autos listos para

def auto():
    print("Auto listo en la línea de salida.")
    barrera.wait() # Espera a que todos los autos estén listos
    print("¡La carrera ha comenzado!")

# Simula 3 autos esperando
for i in range(3):
    threading.Thread(target=auto).start()
```

## 6. RLock (bloqueo reentrante)

### Cuándo usarlo:

Usa un **RLock** cuando un hilo necesita adquirir el mismo **lock varias veces** sin bloquearse a sí mismo. Un **RLock** permite que un hilo vuelva a adquirir el bloqueo sin quedar atrapado si ya lo tiene.

### Ejemplo cotidiano:

Imagina que tienes una **puerta que necesitas abrir varias veces** en el mismo proceso (entrar y salir de una habitación varias veces). Si usas una llave común (lock), podrías quedarte bloqueado si intentas abrir la puerta desde adentro. Un **RLock** es como una llave maestra que te permite abrir y

cerrar la puerta sin problemas aunque ya la tengas abierta.

```
import threading

rlock = threading.RLock()

def entrar_y_salir():
    rlock.acquire()
    print("Entrando a la habitación...")
    rlock.acquire() # Puedes adquirir el mismo lock otra vez
    try:
        print("Dentro de la habitación, haciendo algo...")
    finally:
        rlock.release() # Debes liberar tantas veces como lo has
        rlock.release()

# Simula entrar y salir
threading.Thread(target=entrar_y_salir).start()
```

## Resumen:

- **Lock:** Uso exclusivo de un recurso (una única llave para una puerta).
- **Semaphore:** Control de acceso limitado a un recurso (varias llaves pero con límite).
- **Event:** Esperar una señal o luz verde para avanzar (esperar a que enciendan las luces para empezar la fiesta).
- **Condition:** Esperar una condición específica (esperar a que estén listas las entradas para entrar al cine).
- **Barrier:** Todos deben llegar a un punto antes de continuar (esperar que todos los autos estén listos para comenzar la carrera).

- **RLock:** Un mismo hilo necesita adquirir el mismo lock varias veces sin bloquearse (puerta que puedes abrir y cerrar varias veces sin quedarte bloqueado).