

# SINCRONIZACION DE LOCKS

Por Edward Steven y Alejandro Gimenez





# INTRODUCCIÓN

En esta presentación veremos:


- ReentrantLock
- ReadWriteLock





# ¿QUE ES REENTRANLOCK?

Es una implementación de la interfaz Lock que proporciona un mecanismo de sincronización mas flexible y explicito. Este lock es "reentrante", es decir, puede adquirir el mismo lock varias veces sin causar un bloque, siempre y cuando se libere la misma cantidad de veces



# Utilidad y casos de uso

## Proposito principal

Controlar el acceso a los recursos compartidos en entornos multi-hilo de forma mas precisa teniendo un mayor control gracias a métodos como `lock()` y `unlock()`

## casos de uso

- Cuando se necesita una mayor flexibilidad en la sincronización como establecer un tiempo de espera para adquirir el lock (`tryLock()`)
- En situaciones donde se requiere evitar la inversión de prioridades entre hilos (problema conocido como "priority inversion").
- Para implementar patrones de bloqueo más avanzados, como locks justos o locks condicionales.

# ventajas y desventajas

**Flexibilidad**

**Complejidad**

**Reentrancia**

**Posible Perdida  
Rendimiento**

**Funciones  
Avanzadas**

**Posible Falta de  
visión global**





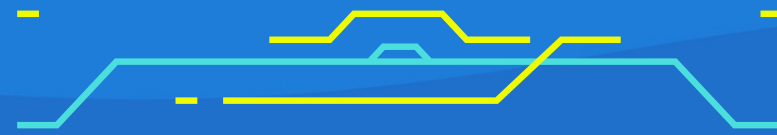


# ¿QUÉ ES READWRITELOCK?

Mecanismo de bloqueo utilizado en programación concurrente para gestionar el acceso a recursos compartidos entre varios hilos de ejecución.

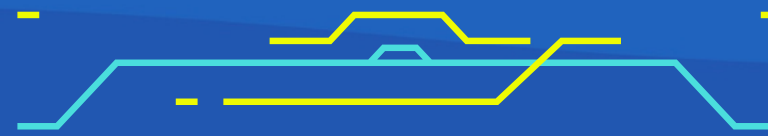


# CASOS DE USO



## LECTURAS FRECUENTES, ESCRITURAS OCASIONALES:

Cuando hay + operaciones de  
lectura que de escritura



## RENDIMIENTO MEJORADO EN LA LECTURA

Permite que haya varios hilos  
leyendo simultaneamente



## ESCRITURAS ATÓMICAS

No deben ser interrumpidas  
por otras operaciones de  
lectura o escritura



# VENTAJAS E INCONVENIENTES

de ReadWriteLock

## VENTAJAS

### 1- MEJORA RENDIMIENTO

### 2- EVITA BLOQUEOS INNECESARIOS

Al usar la herramienta, evita que los hilos de lectura se vean afectados

### 3- ESCRITURAS ATOMICAS



## INCONVENIENTES

### 1- POSIBLE DEBILIDAD EN LAS ESCRITURAS

Al dar prioridad a las operaciones de lectura, los hilos de escritura se pueden debilitar e ir más lentos

### 2- COMPLEJIDAD

Se debe tener cuidado con la gestión de los bloqueos de lectura y escritura

### 3- REQUISITOS DE SINCRONIZACION

No siempre es esta la opción más óptima