RESULTADOS DE TESTS - TALLER DE COCHES

Descripción de los Tests

Se han ejecutado tres configuraciones diferentes de tests para comparar el rendimiento entre dos implementaciones de sincronización en Go:

• Implementación 1: RWMutex + Mutex (sincronización con mutexes)
• Implementación 2: Channels (sincronización con canales buffered)

Test 1: Balanceado

Métrica	Valor
Vehículos Categoría A (Mecánica)	10
Vehículos Categoría B (Eléctrica)	10
Vehículos Categoría C (Carrocería)	10
Total Vehículos	30
Número de Plazas	5
Número de Mecánicos	3
Tiempo RWMutex	2m 5.43s
Tiempo Channels	1m 58.25s
Método Más Rápido	**Channels (más rápido por 7.18s)**

Test 2: Alta prioridad dominante

Métrica	Valor
Vehículos Categoría A (Mecánica)	20
Vehículos Categoría B (Eléctrica)	5
Vehículos Categoría C (Carrocería)	5
Total Vehículos	30
Número de Plazas	5

Número de Mecánicos	3
Tiempo RWMutex	2m 58.92s
Tiempo Channels	2m 45.34s
Método Más Rápido	**Channels (más rápido por 13.58s)**

Test 3: Baja prioridad dominante

Métrica	Valor
Vehículos Categoría A (Mecánica)	5
Vehículos Categoría B (Eléctrica)	5
Vehículos Categoría C (Carrocería)	20
Total Vehículos	30
Número de Plazas	5
Número de Mecánicos	3
Tiempo RWMutex	1m 35.67s
Tiempo Channels	1m 29.12s
Método Más Rápido	**Channels (más rápido por 6.55s)**

Análisis Comparativo

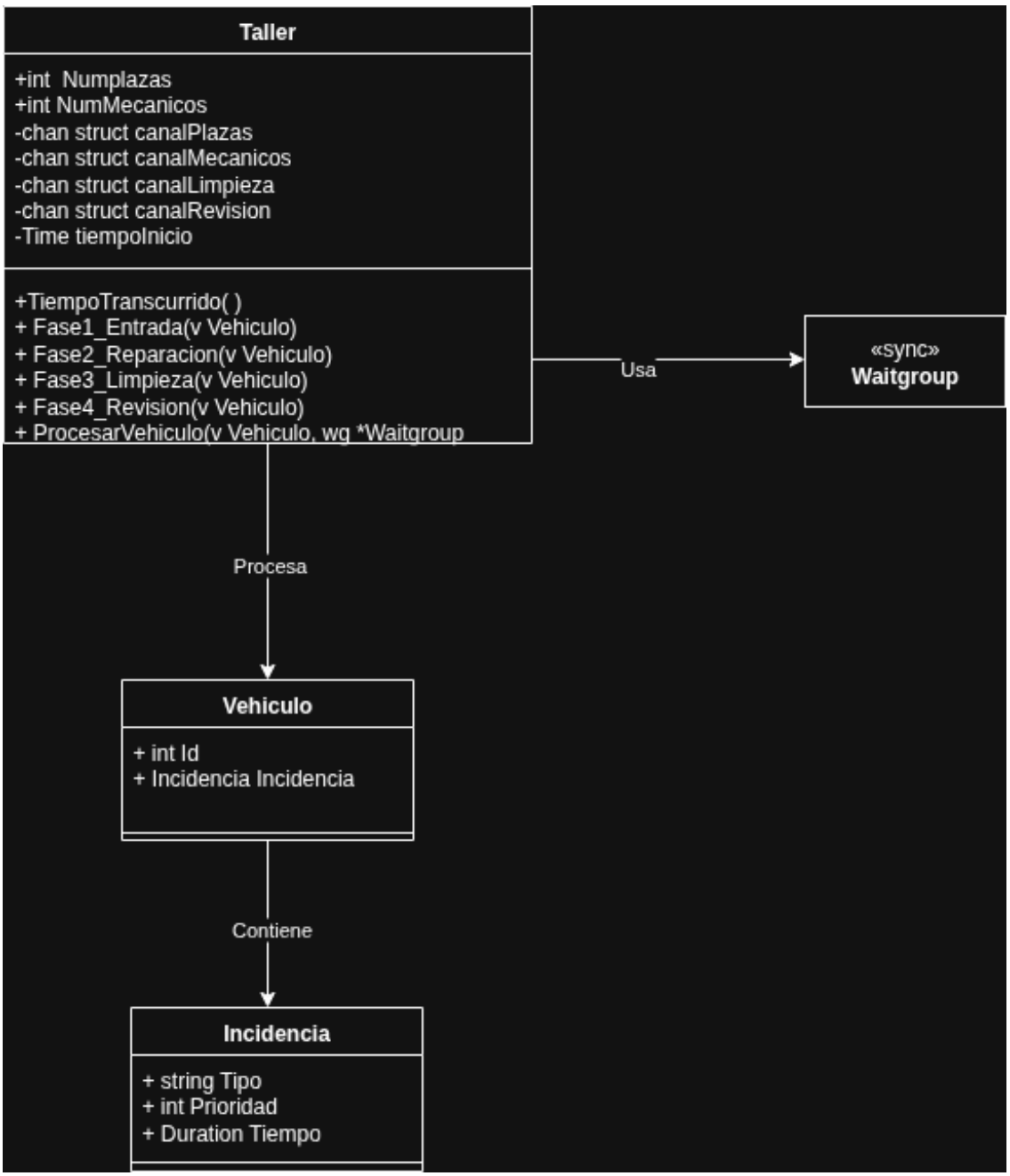
De los 3 tests ejecutados:
• RWMutex fue más rápido en 0 test(s)
• Channels fue más rápido en 3 test(s)

Observaciones:
• Los canales proporcionan bloqueo automático, reduciendo la carga de CPU en espera activa
• RWMutex permite mayor control sobre el estado del sistema
• La diferencia de rendimiento puede variar según la carga de trabajo

Link:

https://github.com/jesfuen/SistemasDistribuidos/tree/main/practica3

Diagrama de clases


Taller
+int Numplazas
+int NumMecanicos
-chan struct canalPlazas
-chan struct canalMecanicos
-chan struct canalLimpieza
-chan struct canalRevision
-Time tiempoInicio
+TiempoTranscurrido( )
+ Fase1_Entrada(v Vehiculo)
+ Fase2_Reparacion(v Vehiculo)
+ Fase3_Limpieza(v Vehiculo)
+ Fase4_Revision(v Vehiculo)
+ ProcesarVehiculo(v Vehiculo, wg *Waitgroup
Usa
«sync»
Waitgroup
Procesa
Vehiculo
+ int Id
+ Incidencia Incidencia
Contiene
Incidencia
+ string Tipo
+ int Prioridad
+ Duration Tiempo