SSOO

Práctica 4 – Modernidad en el taller del pueblo

Alejandro Tejero de la Morena

1 - Introducción y objetivos

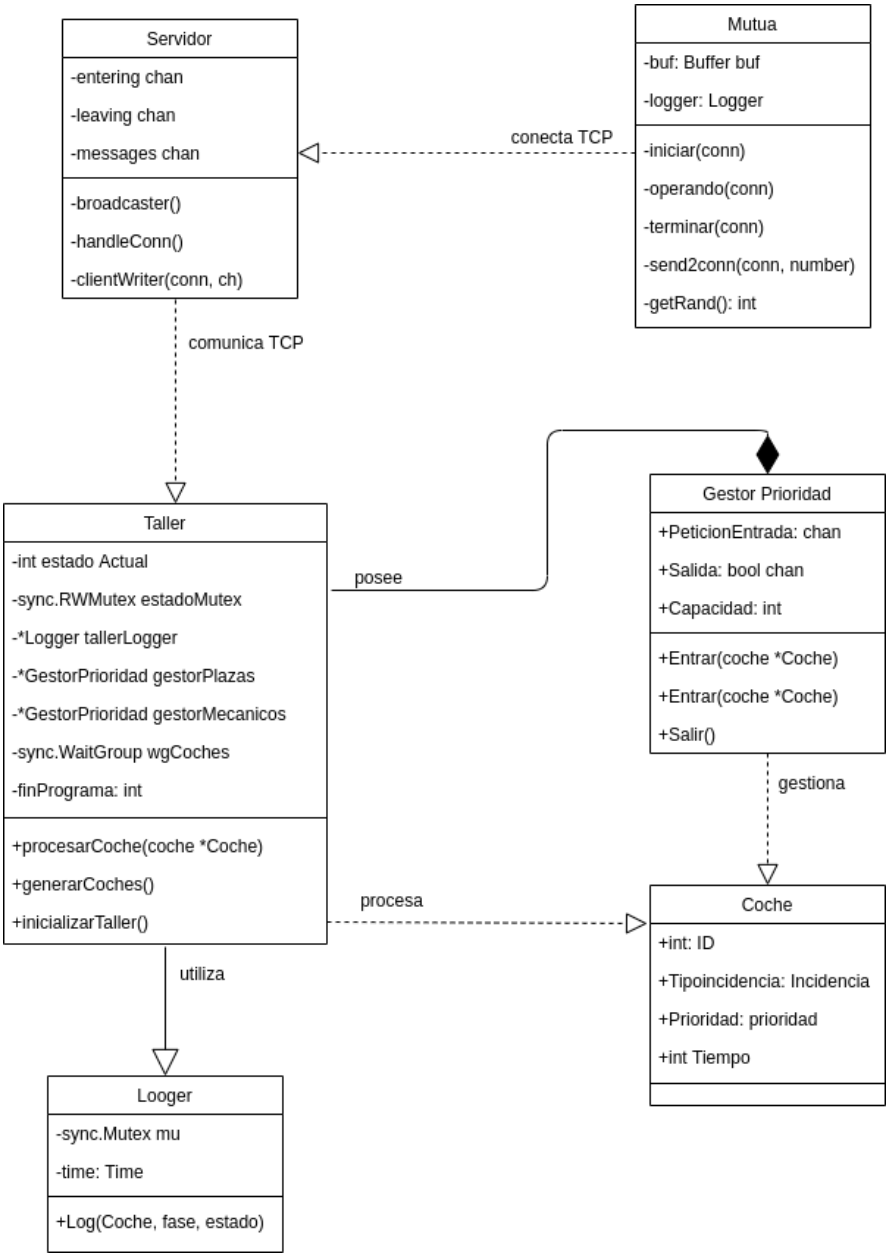
Esta práctica ces una mejora de la Práctica 3, donde se implementó la lógica concurrente de un taller del pueblo con gestión de prioridades y fases secuenciales de reparación. El objetivo actual es migrar dicha lógica hacia un entorno de concurrencia avanzada, integrando la comunicación entre procesos de Go mediante una arquitectura distribuida.

Para ello, el sistema se ha estructurado en torno a tres componentes clave:

- **Mutua (Cliente):** Actúa como el controlador externo, enviando flujos de peticiones y monitorizando el estado del sistema.
- **Servidor (Intermediario):** Gestiona la capa de red, facilitando el intercambio de mensajes entre el cliente y el taller.
- **Taller (Lógica de Negocio):** El módulo desarrollado en esta práctica, que implementa la máquina de estados y gestiona la sincronización de los vehículos en reparación.

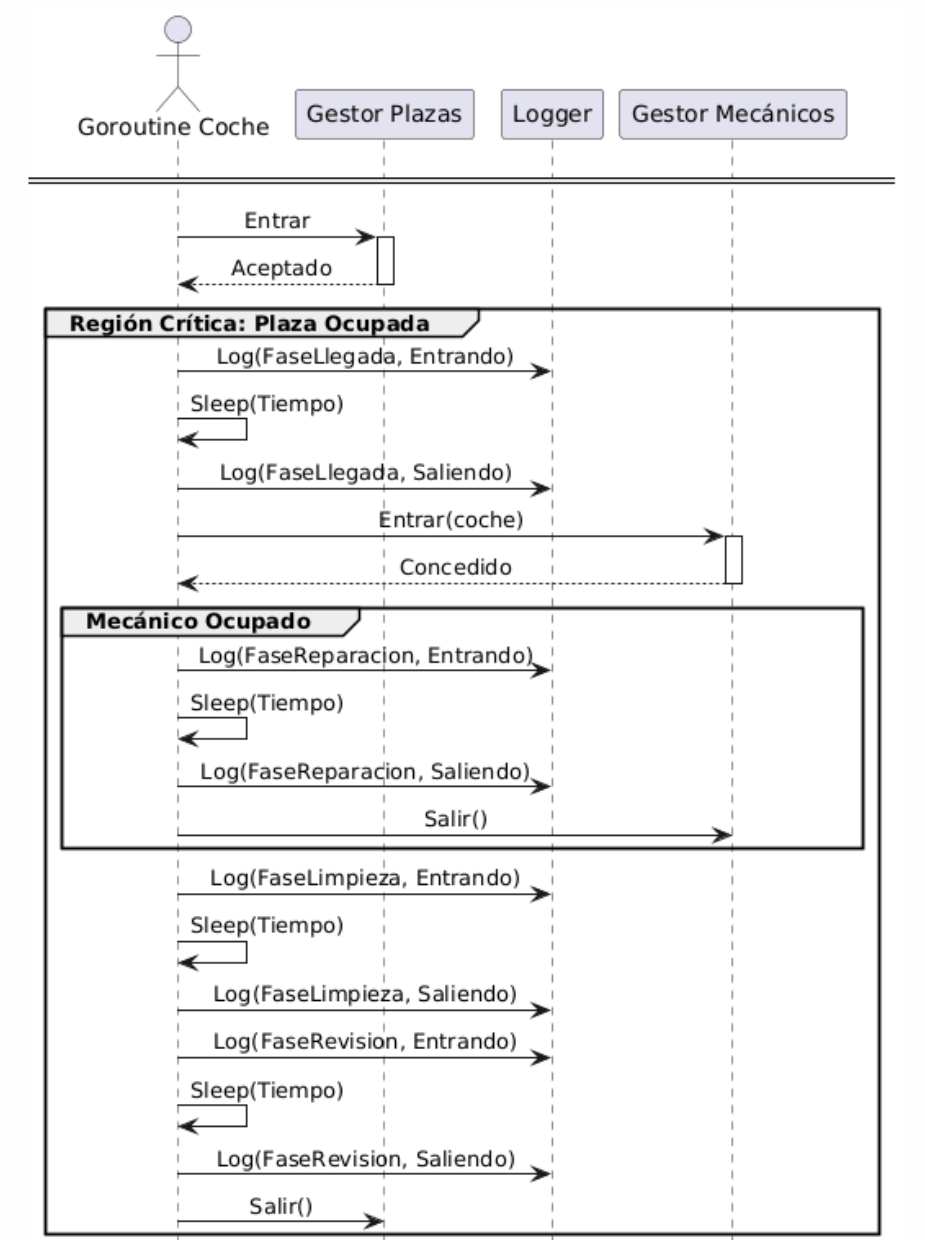
2 - Explicación del diseño del sistema mediante diagramas UML

2.1 - Diagrama de clases



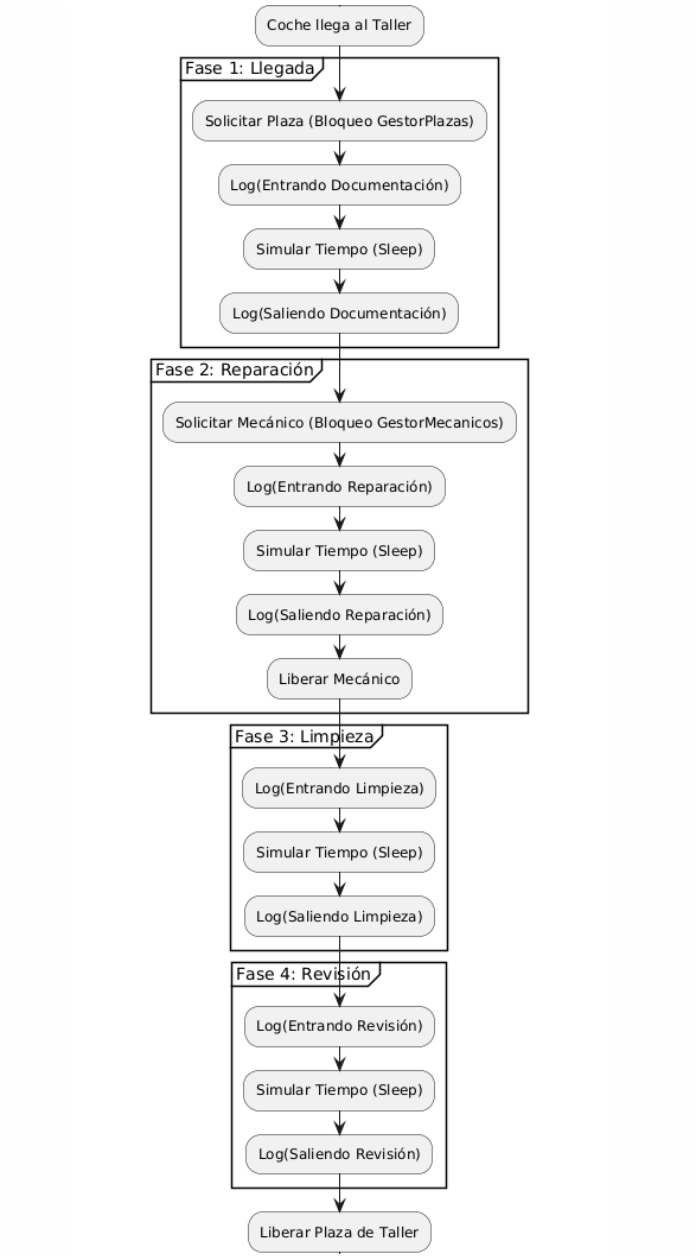
- **Explicación**: El diagrama ilustra la arquitectura del sistema distribuido, donde la clase principal Taller actúa como cliente TCP recibiendo órdenes del Servidor (intermediario con la Mutua). El diseño delega la gestión de recursos en la clase GestorPrioridad, que administra las colas de plazas y mecánicos mediante canales. La entidad Coche representa la unidad de trabajo que fluye por el sistema según su prioridad. Finalmente, se garantiza la seguridad en el acceso concurrente a los estados y la salida por consola mediante el uso de exclusion mutua (sync.RWMutex)

2.2 - Diagrama de secuencia



- **Explicación**: El diagrama muestra el flujo de la goroutine del coche pasando por sus cuatro fases. Primero valida la entrada con el Gestor de Plazas y luego va mandando logs al Logger cuando empieza y termina cada tarea. La parte clave es cuando interactúa con el Gestor de Mecánicos, ya que ahí el coche pide paso y se tiene que esperar hasta que le confirmen que hay uno libre antes de seguir y salir.

2.3 - Diagrama de actividad



- **Explicacion**: Este diagrama muestra como ejecuta la goroutine de cada coche. Se distinguen claramente las cuatro etapas del ciclo de vida (Llegada, Reparación, Limpieza y Revisión), mostrando cómo el vehículo avanza linealmente de una tarea a otra y registra su estado en el Logger, permitiendo visualizar de un vistazo todo el recorrido desde que entra al sistema hasta que finaliza el servicio.

El punto fuerte del diseño es la distinción en la gestión de recursos: mientras que la llegada y la reparación incluyen acciones de "Solicitud" porque dependen de semáforos o monitores (Plazas y Mecánicos) que pueden bloquear la ejecución, las fases de limpieza y revisión se representan como procesos directos. Esto refleja que estas últimas etapas no requieren pelear por un recurso extra, sino que simplemente consumen tiempo de simulación manteniendo la plaza ya asignada.

3 - Análisis de resultados

Para el análisis debemos realizar 3 test de dos formas diferentes, primero debemos hacerlo con 6 plazas y 3 mecánicos, y luego con 4 plazas y 4 mecánicos. Los test que debemos realizar son los siguientes:

# Test	Categoría	Coches	Categoría	Coches	Categoría	Coches
1	A	10	B	10	C	10
2	A	20	B	5	C	5
3	A	5	B	5	C	20

3.1 - Comandos de ejecución

- Ejecucion de los test:

1) cd taller / go test -v

- Ejecucion de la conexión TCP:

1) cd servidor / go run servidor.go

2) cd taller / go run taller.go types.go utils.go

3) cd mutua / go run mutua.go

3.2 - Capturas de la ejecución

Test1: 10A - 10B - 10C – 6 Plazas y 3 Mecánicos

```
Tiempo 72.08 Coche 30 Incidencia Carrocería Fase Revisión Final Estado Saliendo

Resumen Test 1

Duración: 72.08 segundos
Coches procesados: 30/30
Rendimiento: 0.42 coches/seg
```

Test2: 10A - 10B - 10C – 4 Plazas y 4 Mecánicos

```
Tiempo 98.91 Coche 30 Incidencia Carrocería Fase Revisión Final Estado Saliendo

Resumen Test 2

Duración: 98.91 segundos
Coches procesados: 30/30
Rendimiento: 0.30 coches/seg
```

Test3: 20A - 5B - 5C – 6 Plazas y 3 Mecánicos

```
Tiempo 89.66 Coche 30 Incidencia Carrocería Fase Revisión Final Estado Saliendo

Resumen Test 3

Duración: 89.66 segundos
Coches procesados: 30/30
Rendimiento: 0.33 coches/seg
```

Test4: 20A - 5B - 5C – 4 Plazas y 4 Mecánicos

```
Tiempo 130.70 Coche 25 Incidencia Eléctrica Fase Revisión Final Estado Saliendo

Resumen Test 4

Duración: 130.70 segundos
Coches procesados: 30/30
Rendimiento: 0.23 coches/seg
```

Test5: 5A - 5B - 20C – 6 Plazas y 3 Mecánicos

```
Tiempo 49.07 Coche 30 Incidencia Carrocería Fase Revisión Final Estado Saliendo

Resumen Test 5

Duración: 49.07 segundos
Coches procesados: 30/30
Rendimiento: 0.61 coches/seg
```

Test6: 5A - 5B - 20C – 4 Plazas y 4 Mecánicos

```
Tiempo 70.23 Coche 30 Incidencia Carrocería Fase Revisión Final Estado Saliendo

Resumen Test 6

Duración: 70.23 segundos
Coches procesados: 30/30
Rendimiento: 0.43 coches/seg
```

Tabla de resultados:

Test	A	B	C	Plazas	Mecánicos	Duración	Rendimiento (Coches/seg)
1	10	10	10	6	3	72.08 seg	0.42
2	10	10	10	4	4	98.91 seg	0.30
3	20	5	5	6	3	89.66 seg	0.33
4	20	5	5	4	4	130.70 seg	0.23
5	5	5	20	6	3	49.07 seg	0.61
6	5	5	20	4	4	70.23 seg	0.43

3.3 - Análisis de resultados

Test1:

En este primer escenario con una carga equilibrada, la configuración de 6 plazas y 3 mecánicos demostró ser más eficiente, terminando en 72 segundos frente a los 99 segundos de la opción de 4 plazas. La clave aquí ha sido la capacidad del buffer ya que, aunque tengamos un mecánico menos, el disponer de dos plazas extra permite que los coches sigan entrando sin bloquear el sistema. Con solo 4 plazas, el taller se llena enseguida y el hilo principal tiene que detenerse constantemente hasta que queda un hueco libre, lo que acaba penalizando el tiempo total.

Test2:

Este ha sido el caso más lento de todos, ya que los coches de tipo A requieren 5 segundos de reparación, reteniendo los recursos durante más tiempo. La diferencia de rendimiento fue muy notable: la configuración de 6 plazas tardó unos 90 segundos, mientras que la de 4 plazas se disparó hasta los 130 segundos. Al ser reparaciones largas, la rotación de vehículos es baja, por lo que tener solo 4 plazas de espera provoca un cuello de botella constante en la entrada, impidiendo aprovechar la velocidad extra del cuarto mecánico.

Test3:

Finalmente, en el escenario con reparaciones rápidas (tipo C), se consiguió bajar los tiempos a 49 segundos con la configuración de 6 plazas. La configuración de 4 mecánicos fue más lenta con 70 segundos, a pesar de tener más potencia de trabajo. Esto confirma que el factor limitante en este sistema no es la velocidad de los mecánicos, sino el espacio disponible para esperar: al saturarse las 4 plazas rápidamente, se pierden segundos valiosos bloqueando la admisión de nuevos vehículos

4 - Código fuente del sistema (Repositorio del proyecto)

Link: https://github.com/atejero2021/DistribuidosP4