

# Practica 2

## Sistemas Distribuidos

### Grado en Ingeniería Telemática

## Enunciado

Con los mismos requisitos del Taller de coches de la práctica 1, se deberá hacer la implementación usando `goroutines` y `channels` de GO. A diferencia de la práctica 1, aquí se podrá usar más de un fichero `.go`.

Para esta práctica se deberán implementar las siguientes funcionalidades:

- Cada coche que llega al taller deberá ser atendido por un mecánico. Si no hay mecánicos libres, el coche esperará en una cola de espera.
- La cola de espera no tiene límite de tamaño.
- Cada mecánico atenderá a un coche durante un tiempo (Use el paquete `Time` de GO) según la incidencia y se visualiza a continuación:
  - mecánica: Tiempo de atención medio de 5 segundos.
  - eléctrica: Tiempo de atención medio de 7 segundos.
  - carrocería: Tiempo de atención medio de 11 segundos.
- Cada coche acumulará el tiempo en que un mecánico lo atiende.
- Si un coche acumula más de 15 segundos de atención, se le dará prioridad asignando otro mecánico de cualquiera de las especialidades que tengan asignadas, de no haber más mecánicos se deberá contratar a otro mecánico de la especialidad demandada de inmediato.

## Consideraciones

- Solo se puede hacer con las instrucciones que hemos dado en clase hasta el ejercicio 5.
- Se puede usar más de un archivo fuente de `GO`.
- Se debe usar `goroutines` y `channels` para la implementación.
- No es necesario implementar persistencia de datos.

# Tests

Para esta ocasión se deberá hacer una comparación entre los siguientes casos:

- Comparativas para cuando la cantidad máxima de coches con una sola incidencia del mismo tipo se duplica.
- Comparativas para cuando duplicamos la plantilla de 3 mecánicos (uno de cada especialidad) a 6.
- Comparativas para cuando se tiene 3 mecánicos de especialidad mecánica por cada uno de eléctrica y uno de carrocería, y para cuando se tienen 1 mecánicos de especialidad mecánica por cada tres de eléctrica y tres de carrocería.

Para realizar estas comparativas es altamente recomendable que se use el paquete `testing` de `GO`.

**NOTA:** Es **altamente recomendable** que se use un proyecto de referencia como el que se expone en el capítulo 8 del libro *The Go Programming Language*, proyecto que están disponibles en este [link](#)

## Entregables

Un documento en formato **PDF** que contenga:

1. Explicación del diseño del sistema (estructuras de datos, funciones principales, funcionamiento de la aplicación, etc.) haciendo uso explícito de diagramas de secuencia de definidos por UML. (40%)
2. Resultados de los tests realizados, con las métricas obtenidas y su análisis. (40%)
3. Código fuente del software desarrollado o link a repositorio donde se encuentre el código. (20%)

## Indicaciones adicionales

Guarda el archivo con un nombre `Practica_2_nombre_SS00_dist.pdf`, donde `nombre` sea tu nombre de usuario.

Luego este archivo se subirá a la actividad de aula virtual.