

# Practica 4

## Sistemas Distribuidos

### Grado en Ingeniería Telemática

## Concurrencia en GO - Servidores

En esta practica lo que se busca es definir la capacidad de desarrollo cuando la concurrencia sea avanzada, para la comunicación entre los diversos métodos en `GO` y un servidor básico.

### Problema: Modernidad en la República Bananera

Una cantidad `N` de aviones, se aproximan a un aeropuerto donde una torre de control les va asignando el momento en el cual deben aterrizar. La torre de control dispondrá de las `NumPistas` pistas disponibles en el aeropuerto, siempre y cuando no estén siendo usadas por otro avión. A medida que van terminando su proceso de aterrizaje en pista, proceden a desplazarse a las puertas de desembarque donde los pasajeros podrán ir a por su respectivo equipaje. Cada avión posee un `numPass` de pasajeros que permitirá hacer su clasificación.

Se ha designado un canal de información que es gestionado por un servidor que estará enviando una serie de números. Cada número tendrá un valor que designará el tipo de operación que deba hacer el aeropuerto. Los valores que entregara este servidor pueden ser los siguientes:

Numero	Estado	Descripción
0	Aeropuerto Inactivo	No hay Aterrizajes
1	Solo Categoría A	Solo podrán acceder los aviones de dicha categoría
2	Solo Categoría B	Solo podrán acceder los aviones de dicha categoría
3	Solo Categoría C	Solo podrán acceder los aviones de dicha categoría
4	Prioridad Categoría A	La prioridad del aterrizaje lo tienen los aviones de dicha categoría

Numero	Estado	Descripción
5	Prioridad Categoría B	La prioridad del aterrizaje lo tienen los aviones de dicha categoría
6	Prioridad Categoría C	La prioridad del aterrizaje lo tienen los aviones de dicha categoría
7	No definido	Se mantiene el estado anterior
8	No definido	Se mantiene el estado anterior
9	Aeropuerto Cerrado Temporal	No hay Aterrizajes

Las categorías de los aviones están definidas según su número de pasajeros:

- **Categoría A:** Los aviones de más de 100 pasajeros.
- **Categoría B:** Los aviones entre 50 y 100 pasajeros.
- **Categoría C:** Los aviones de menos de 50 pasajeros .

**NOTA:** El hecho que un avión tenga prioridad elevada implica que al estamento con el cual contacte se le atenderá de forma inmediata y los demás deberán esperar a ser atendidos.

## Enunciado

Se requiere un programa en `GO` que modele el comportamiento del sistema anteriormente descrito, de forma que se pueda ejecutar dicha simulación una única vez y en función del programa denominado `enaire` , adjunto en aula virtual a esta práctica, gestione los cambios de estado que deba hacer el aeropuerto.

El programa debería cumplir las siguientes consideraciones:

- La torre de control, las pistas y las puertas tienen un tiempo de uso/utilización que debe ser contemplado.
- La torre de control, las pistas y las puertas tienen una variación de tiempo de uso/utilización que debe ser contemplado.
- La torre de control y las pistas deberían tener una cantidad máxima de aviones esperando a ser atendidos.
- El orden de entrada de los aviones debe ser completamente aleatorio para cada categoría.

- Se debería visualizar en la ejecución del programa un mensaje para cada avión cuando tome contacto con cada una de los estamentos. La salida de este program permitirá hacer procesamiento de la información a posteriori.

Para su correcta implementación, en el aula virtual dispondrá de un conjunto de archivos cuya descripción es la siguiente:

- `servidor.go` : Programa que simula el comportamiento del servidor de información de la entidad aeroportuaria. Es el primero en ejecutarse, no visualiza información. **No se debe modificar**
- `cliente.go` : Programa que simula el cliente que debe existir en el aeropuerto para obtener la información de `servidor.go` . Es el segundo en ejecutarse, y es el **único que se puede modificar**
- `enaire.go` : Programa que simula el comportamiento de la entidad aeroportuaria. Visualiza cada uno de los datos enviados al servidor.**No se debe modificar**

Además de las consideraciones anteriores, el programa debe tener una descripción técnica sobre su funcionamiento usando Diagramas de Secuencia y diagramas UML.

## Tests

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

- Comparativas para cuando se atienden las siguientes cantidades de aviones, que deben aterrizar en un orden aleatorio:

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

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

## Evaluación

Se tendrá en cuenta los siguientes items:

- Que realice las acciones requeridas para cada una de las peticiones de estado que realiza el software `enaire` . (60% de la práctica)

- Memoria técnica de la práctica, con código o repositorio, Diagramas de Secuencia, Diagramas UML. (30% de la práctica)
- El mínimo uso de estructuras de sincronización de procesos ofrecidas por el paquete `sync` de `GO`. (10% de la práctica)

**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](#)

## Indicaciones adicionales

La descripción técnica, las métricas obtenidas en los tests y el código fuente del software (o link en repositorio si se prefiere), deberán estar consagrados en el archivo de formato PDF, que deberá llamarse `Practica_4_nombre_SSOO_dist.pdf`, donde `nombre` sea tu nombre de usuario.

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