

Practica 3

Sistemas Distribuidos

Grado en Ingeniería Telemática

Concurrencia en GO

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`.

Problema: Gentrificacion 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.

Debido a las restricciones aeroportuarias se deberán clasificar los aviones que lleguen al aeropuerto en 3 categorías diferentes. Cada categoría tendrá una importancia relativa mayor con respecto a la otra en función de la siguiente tabla:

- **Categoría A:** Los aviones de más de 100 pasajeros tendrán prioridad alta.
- **Categoría B:** Los aviones entre 50 y 100 pasajeros tendrán prioridad normal.
- **Categoría C:** Los aviones de menos de 50 pasajeros tendrán prioridad baja.

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 `sim`s veces.

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.
- Se debería visualizar en la ejecución el mensaje de `Desembarco del avión N` , cuando se termine el proceso de cada uno.
- El orden de entrada de los aviones debe ser completamente aleatorio para cada categoría.

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

Tests

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

- Comparativas para cuando se atienden las siguientes cantidades de aviones:

# 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:

- El mínimo uso de estructuras de sincronización de procesos ofrecidas por el paquete *sync* de GO .
- El mínimo tiempo de ejecución de los test al ser replicados de forma local, para esto asegurar de tener las instrucciones para su ejecución disponibles en la entrega.

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 un archivo de formato PDF, que deberá llamarse `Practica_3_nombre_SSOO_dist.pdf` , donde `nombre` sea tu nombre de usuario.

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