



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

## FACULTAD DE INGENIERÍA

Semestre 2024-1  
**Sistemas Operativos**  
Grupo 06

## UNA SITUACIÓN COTIDIANA PARALELIZABLE

**Ing. Gunnar Eyal Wolf Iszaevich**

Alumnos:

- **Rodríguez Kobeh Santiago**
- **Villaseñor Venegas Carlos Miguel**

### **Situación a modelar:**

1-En el contexto de la aviación comercial, un avión puede aterrizar a la vez por pista de aterrizaje. Un aeropuerto puede tener cuantas pistas de aterrizaje desee. Estas pistas de aterrizaje pueden funcionar de manera paralela siempre y cuando haya una distancia mínima de 1.5 kilómetros entre cada una. En el Aeropuerto Internacional de la Ciudad de México, (AICM), contamos con dos pistas de aterrizaje, por lo que uno pensaría que se pueden utilizar de manera simultánea. El detalle es que estas no tienen 1.5 kilómetros de separación entre ellas, por lo que no es posible utilizarlas de manera paralela.

Para nuestro caso proponemos 2 aeropuertos, uno representa al aeropuerto de la Ciudad de México, y el otro representa a todos los aeropuertos internacionales. Cada uno de los aeropuertos se encargará de mandar aviones (hilos) hacia el aeropuerto opuesto, y tendrá que actualizar la información en la terminal de vuelos (interfaz). Se mantendrá un orden FIFO para administrar la prioridad de los aviones que deseen aterrizar en el mismo país, y se contemplarán únicamente 2 estados, “En camino” para aviones que estén esperando o cumpliendo con sus horas de vuelo y “Llegó” para los que ya aterrizaron.

2-Las consecuencias de una mala concurrencia serían atrasos en los vuelos, información errónea en la terminal de vuelos, o incluso accidentes aéreos.

3-El ordenamiento de los aviones que llegan importa, ya que de haber escogido algún método que se basará en tiempo de vuelo para asignar la prioridad, podría darse el caso de inanición, lo que podría ser grave si tomamos en cuenta que en la vida real los aviones tienen combustible limitado.

Métodos de sincronización:

Para este problema se ocuparon únicamente semáforos inicializados en 1.

### **Lógica de operación:**

1- Las variables compartidas que podrían generar una situación de carrera fueron la hora de despegue (“hora”), el identificador de cada avión (“id”), los contadores usados para asignarle a cada avión un espacio dentro de la terminal (“contadorA1” y “contadorA2”), y las modificaciones de los “Label” de la interfaz.

2-En nuestro caso cada hilo es un avión, entonces una vez presionado el botón de “iniciar” se crea un hilo (para evitar que se congele la interfaz) que generará un número fijo de hilos (aviones) y cada uno de estos hilos ejecutará la función “Volar()” pasando como argumento el número 1 o 2 (se asignan de forma aleatoria) los que entraron con el valor 1 son los aviones del aeropuerto de la Ciudad de México, y los que entraron con el valor 2 son los de el aeropuerto internacional.

Los aviones de la Ciudad de México ejecutarán la función “tipoAvion1()”, dentro lo primero que hacen es que con ayuda de un semáforo se crea un objeto “Avion” con los siguientes

datos: id, hora de despegue, contador, duración, y destino (aleatorio). Después dentro de otro semáforo se modifica la celda correspondiente al contador asignado al avión. Por último cada avión tiene que esperar el tiempo de vuelo determinado por el destino (usando un diccionario) antes de entrar a otro semáforo a modificar su estado de “En camino” a “Llegó”.

El proceso anterior es el mismo para los aviones de el aeropuerto internacional, con la única diferencia siendo los nombres de la función (“tipoAvion2()”), el contador(“contadorA2”), y las celdas que modifica dentro de la terminal.

3-Los hilos del mismo tipo de avión interactúan cuando comparten el mismo destino y uno tiene que esperar a otro (sucede muy rápido), todos los hilos interactúan “formándose” para modificar la terminal. Todo esto se realiza usando “semáforos(1)” para evitar que más de 1 modifique la terminal al mismo tiempo, o aterrice al mismo tiempo.

### Descripción del entorno de desarrollo, suficiente para reproducir una ejecución exitosa:

- Tener Python instalado ( probado en la versión 3.12).
- Importar las librerías Tkinter, threading, time, y random en sus versiones más recientes (el programa las importa).
- Desarrollado y probado en Windows 10.
- Usar la terminal o cualquier compilador para ejecutar el archivo.py.

### Ejecución exitosa:

-Antes de iniciar:



-Al finalizar:

# Departure and Arrivals

**Aeropuerto Int. De la Ciudad de México****Aeropuertos Int.**

Vuelo	Salida	Destino	Estado	Vuelo	Salida	Desde	Estado
1001	1	Francia	Llegó	1002	2	Portugal	Llegó
1004	4	EEUU	Llegó	1003	3	Francia	Llegó
1007	7	Francia	Llegó	1005	5	EEUU	Llegó
1009	9	Japón	Llegó	1006	6	Francia	Llegó
-	-	-	-	1008	8	Portugal	Llegó
Start				Exit			