Logotipo

Descripción generada automáticamente

Universidad Nacional Autónoma de México

Facultad de ingeniería

Sistemas operativos

**Proyecto 2 de Sistemas Operativos**

**Un día en el aeropuerto Felipe Ángeles**

Profesor: Wolf Iszaevich Gunnar Eyal

Alumnos: Leyva Mercado Christian Alejandro y Velasco Pachuca Bryan

Grupo: 06

**Situación**

Se acaba de inaugurar el nuevo aeropuerto Felipe Ángeles en el que se cuentan con las siguientes zonas de operación:

* Pista de aterrizaje/despegue.
* Torre de control.
* Zona de espera(terrestre).
* Zona de espera (aérea).
* Zona de carga/descarga de pasajeros.
* Zona de descarga/carga de mercancía.

Se sabe que se tiene un flujo continuo de peticiones de la pista de aterrizaje las cuales son gestionadas por la torre de control, la torre de control deberá administrar los aterrizajes y despegues según como lleguen las peticiones de los aviones (sin generar inanición a alguno de los aviones), cada uno de los aviones que aterriza volverá a despegar tras dejar a sus pasajeros o mercancía y ser llenado nuevamente con los pasajeros o la mercancía.

El andén de carga/descarga de pasajeros permite la operación de hasta 10 aviones comerciales al mismo tiempo y el andén de carga/descarga de mercancía permite la operación de hasta 6 aviones de carga también al mismo tiempo.

Cuando un avión no puede aterrizar se deja volando en la zona de espera aérea y cuando un avión no puede despegar se deja esperando en la zona de espera terrestre.

**Consecuencias nocivas de la concurrencia:**

En un aeropuerto la peor situación de concurrencia se da cuando existe un excedente de tráfico aéreo que obliga a los aviones a sobrevolar el aeropuerto o retrasar el despegue más de lo debido, incurriendo en retrasos o inclusive accidentes. Por esta razón una correcta administración de los espacios es primordial para cualquier torre de control.

**Eventos concurrentes para los cuales el ordenamiento no resulte importante:**

Dado que el tiempo de permanencia de cada avión en el aeropuerto es una característica intrínseca de cada cargamento, el tiempo que estos tarden en despegar de nuevo generará un nuevo ordenamiento en los despegues, por ello su orden de salida no resulta relevante.

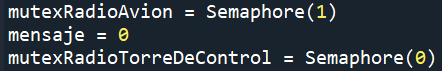
**Documentación**

**Mecanismos de sincronización empleados**

Los mecanismos de sincronización utilizados son los siguientes:

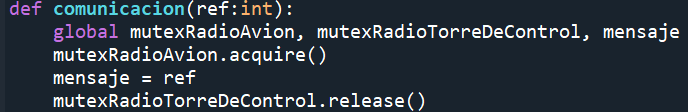
* **Rendezvous**: El rendezvous fue utilizado para establecer una comunicación efectiva entre el avión y la torre de control y de esta manera asignarle la pista al avión.

**Declaración:**

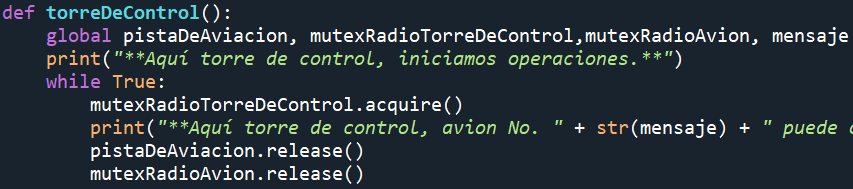
[](https://user-images.githubusercontent.com/86135452/161147861-90406903-097f-4cc0-933d-9d3cae45edf0.png)

**Implementación:**

En método *comunicacion()*:

[](https://user-images.githubusercontent.com/86135452/161148569-a7ce9d36-6c4d-40ca-beb8-84ce9f818189.png)

En método *torreDeControl()*:

[](https://user-images.githubusercontent.com/86135452/161148693-58b0fbe3-9809-44c3-a9b7-a6655466f131.png)

* **Mutex:** El mutex fue utilizado para proteger la pista de aviación y que esta no sea utilizada por más de un avión, evitando colisiones.

**Declaración:**

[image](https://user-images.githubusercontent.com/86135452/161149697-2da6a27f-7c53-4523-8353-bd3168931a43.png)

**Implementación:**

La implementación se realizó prácticamente en todo el código y para no confundir al lector se mostrará su implementación en la descripción algorítmica.

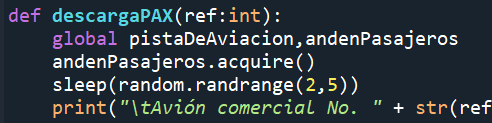
* **Multiplex:** El multiplex fue utilizado para determinar la cantidad de aviones que podían utilizar los andenes de pasajeros o mercancía.

**Declaración:**

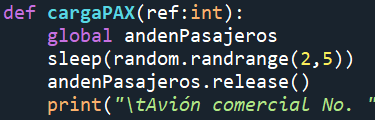
[image](https://user-images.githubusercontent.com/86135452/161150299-f0c42395-b9a1-4cdd-9f51-b74b0feb94a2.png)

**Implementación:**

En método *descarga()*:

[](https://user-images.githubusercontent.com/86135452/161150410-2f6c318f-9555-4f0d-80fc-b4404e59d06a.png)

En método *carga()*:

[](https://user-images.githubusercontent.com/86135452/161150483-ecb92c7a-0964-4db1-a372-5b48a3a86285.png)

**Lógica de operación**

**Identificación del estado compartido:**

**Descripción algorítmica del avance de cada hilo/proceso:**

**Descripción de la interacción entre hilos:**

**Descripción del entorno de desarrollo**

**¿Qué lenguaje emplean? ¿Qué versión?**

El lenguaje utilizado es Python en su versión 3.9.10

**¿Qué bibliotecas más allá del estándar del lenguaje?**

Para el desarrollo de la solución se emplearon las siguientes bibliotecas:

1. **threading:** Esta biblioteca es la más importante del proyecto pues nos permitió manipular diferentes hilos y correrlos cuando fuese necesario, además de proporcionar la estructura de semáforos.



1. **time:** Con esta biblioteca fue posible manipular el tiempo de cada hilo, durmiéndolos cuando fuese necesario y obteniendo el tiempo actual del programa para calcular el desempeño de cada avión/hilo.



1. **random:** Con la biblioteca random se generaron números aleatorios para simular el tiempo de operación de cada avión.



1. **tkinter:** Finalmente, la interfaz de usuario fue desarrollada empleando tkinter la cual proporciona un conjunto de elementos gráficos que nos permitieron darle una mejor vista a la aplicación.



**¿Bajo qué sistema operativo / distribución lo desarrollaron y/o probaron?**

La solución fue desarrollada y probada en el sistema operativo Windows en sus distribuciones 10 y 11.

**Ejemplos o pantallazos de una ejecución exitosa**