

Proyecto de Simulación basada en Eventos Discretos

David Guaty Domínguez C412
[Link en Github](#)

Aeropuerto de Barajas

En el Aeropuerto de Barajas, se desea conocer cuánto tiempo se encuentran vacías las pistas de aterrizaje. Se conoce que el aeropuerto cuenta con un máximo de 5 pistas de aterrizaje dedicadas a aviones de carga y que se considera que una pista está ocupada cuando hay un avión aterrizando, despegando o cuando se encuentra cargando o descargando mercancía o el abordaje o aterrizaje de cada pasajero.

Se conoce que el tiempo cada avión que arriba al aeropuerto distribuye, mediante una función de distribución exponencial con $\lambda = 20$ minutos.

Si un avión arriba al aeropuerto y no existen pistas vacías, se mantiene esperando hasta que se vacíe una de ellas (en caso de que existan varios aviones en esta situación, pues se establece una suerte de cola para su aterrizaje).

Se conoce además que el tiempo de carga y descarga de un avión distribuye mediante una función de distribución exponencial con $\lambda = 30$ minutos. Se considera además que el tiempo de aterrizaje y despegue de un avión distribuye normal ($N(10,5)$) y la probabilidad de que un avión cargue y/o descargue en cada viaje corresponde a una distribución uniforme.

Además de esto se conoce que los aviones tiene una probabilidad de tener una rotura de 0.1. Así, cuando un avión posee alguna rotura debe ser reparado en un tiempo que distribuye exponencial con $\lambda = 15$ minutos. Las roturas se identifican justo antes del despegue de cada avión.

Igualmente cada avión, durante el tiempo que está en la pista debe recargar combustible y se conoce que el tiempo de recarga de combustible distribuye exponencial $\lambda = 30$ minutos y se comienza justamente cuando el avión aterriza.

Se asume además que los aviones pueden aterrizar en cada pista sin ninguna preferencia o requerimiento.

Simule el comportamiento del aeropuerto por una semana para estimar el tiempo total en que se encuentran vacía cada una de las pistas del aeropuerto.

Análisis

Primero que todo se analiza las variables aleatorias que hay que generar.

Para generar una variable uniforme $U(0,1)$ se utilizó el módulo de python random, la función random.

Para generar una variable aleatoria que siguiera la distribución exponencial se utilizó el método de la transformada inversa.

Listing 1: Variable con distribucion exponencial

```
def exp(l: float):
    U = random.random()
    return -1/l* math.log(U)
```

Para generar una variable aleatoria con distribución normal se usó el método de los rechazos.

Listing 2: Variable con distribucion normal

```
def normal(miu:float, var:float ):
    sd = math.sqrt(var)
    return std_normal()*sd + miu
def std_normal():
    Y1 = 1
    Y2 = 0
    while( Y2 - (Y1 - 1)**2/2 <=0):
        Y1 = exp(1)
        Y2 = exp(1)

    U = random.random()
    Z = Y1
    if(U > 1/2):
        Z*=-1
    return Z
```

Se genera una variable aleatoria con distribución normal estándar y se transforma a la distribución normal requerida.

Ahora se pasa a analizar la situación a simular.

Del problema se concluye que cada avión está en alguno de los estados:

- Aterrizando
- Cargando combustible
- Descargando
- Cargando
- Reparando
- Despegando

En el Aeropuerto de Barajas, se desea conocer cuánto tiempo se encuentran vacías las pistas de aterrizaje. Se conoce que el aeropuerto cuenta con un máximo de 5 pistas de aterrizaje dedicadas a aviones de carga y que se considera que una pista está ocupada cuando hay un avión aterrizando, despegando o cuando se encuentra cargando o descargando mercancía o el abordaje o aterrizaje de cada pasajero.

Por lo que se asume que la pista está ocupada desde que el avión esta aterrizando hasta que termina de despegar.

Para conocer el tiempo total que se encuentra vacía una pista, se tendrá una lista de intervalos de la forma $[t_a, t_d]$ que significa que desde t_a hasta t_b la pista estará ocupada. Para obtener el tiempo que se quiere se restará al total de tiempo transcurrido de la simulación la suma de las longitudes de los intervalos.

Se conoce que el tiempo cada avión que arriba al aeropuerto distribuye, mediante una función de distribución exponencial con $\lambda = 20$ minutos.

Se asume que cada evento de arribo al aeropuerto será independiente. Como todos los arribos siguen una distribución exponencial con la misma frecuencia y son independientes entonces se tiene un proceso de Poisson homogéneo. Por lo que para obtener el tiempo en el que el siguiente avión

arribó al aeropuerto se sigue el procedimiento $t_{a+1} = t_a + Y$ donde Y es una variable aleatoria obtenida mediante el listado 1 y t_a es el tiempo en que arribó el avión anterior.

Si un avión arriba al aeropuerto y no existen pistas vacías, se mantiene esperando hasta que se vacíe una de ellas (en caso de que existan varios aviones en esta situación, pues se establece una suerte de cola para su aterrizaje).

Esta parte del problema se modeló solamente con una variable contadora cnt_q que dice cuántos aviones están en cola. No fue necesario implementar una cola explícitamente debido a que se pueden considerar indistintos los aviones .

Se conoce además que el tiempo de carga y descarga de un avión distribuye mediante una función de distribución exponencial con $\lambda = 30$ minutos. Se considera además que el tiempo de aterrizaje y despegue de un avión distribuye normal ($N(10,5)$) y la probabilidad de que un avión cargue y/o descargue en cada viaje corresponde a una distribución uniforme. Además de esto se conoce que los aviones tiene una probabilidad de tener una rotura de 0.1. Así, cuando un avión posee alguna rotura debe ser reparado en un tiempo que distribuye exponencial con $\lambda = 15$ minutos. Las roturas se identifican justo antes del despegue de cada avión. Igualmente cada avión, durante el tiempo que está en la pista debe recargar combustible y se conoce que el tiempo de recarga de combustible distribuye exponencial $\lambda = 30$ minutos y se comienza justamente cuando el avión aterriza.

Del texto anterior se asume que la probabilidad de que un avión cargue(descargue) es de $1/2$. Como solo importa el tiempo en que el avión está en la pista solo se tienen dos eventos: el arribo de un avión a la pista(cuando el avión comienza el aterrizaje en la pista) y el despegue del avión(cuando el avión termina el despegue). Cuando se da el evento de que un avión comienza su aterrizaje en el tiempo t se calcula un tiempo total en el cual el avión estará en la pista(se asume que ninguno de los eventos del párrafo anterior transcurren en paralelo, por ejemplo, no se carga combustible al mismo tiempo en que se esta descargando mercancía) simulando cada una de las variables aleatorias y sumándolas. El código siguiente muestra el procedimiento:

Listing 3: Cálculo del tiempo total que toma un avión en la pista

```
def calculate_time_in_airport(self):
    total_time = 0
    #Landing Time
    total_time += rand.normal(10,5)
    #Fueling Time
    total_time += rand.exp(1/30)
    if rand.uniform() > 1/2:
        #Unloading Time
        total_time += rand.exp(1/30)
    if rand.uniform() > 1/2:
        #Loading Time
        total_time += rand.exp(1/30)
    if rand.uniform() <= 0.1:
        #Repair Time
        total_time += rand.exp(1/15)
    #Takeoff Time
    total_time += rand.normal(10,5)
    return total_time
```

Se asume además que los aviones pueden aterrizar en cada pista sin ninguna preferencia o requerimiento.

Por lo que cuando un avión se dispone a aterrizar lo hará en alguna pista aleatoria de las disponibles. Para modelar esta parte se tiene una variable de estado del sistema *tracks* que es una lista de tiempos en los cuales los aviones presentes en cada una de las pistas despegarán, o sea, $tracks[i] = t_i$ significa que en el tiempo t_i se dará el evento de que el avión que está en la pista i finalizó el despegue. $tracks[i] = inf$ si no existe en la pista un avión en el momento t , por lo que un avión que va a aterrizar en una pista, selecciona una aleatoria de entre todas las pistas que contengan valor *inf*.

Modelo de Simulación de Eventos Discretos

1. Variable de tiempo:

- (a) t : cantidad de tiempo transcurrido en la simulación.

- (b) *time_limit* : tiempo máximo de simulación.
- (c) *t_a*: Tiempo del próximo arribo de un avión.

2. Variables contadoras:

- (a) *cnt_q*: Cantidad de aviones esperando en la cola para poder aterrizar.

3. Variables de estado:

- (a) *tracks*: Lista de tiempos en los cuales los aviones presentes en cada una de las pistas finalizarán el despegue.
- (b) *track_tp*: *track_tp[i]* es una lista de intervalos de la forma $[t_a, t_d]$ los cuales significan que la pista *i* estuvo ocupada desde el tiempo t_a hasta el tiempo t_d , la parte *tp* del nombre *track_tp* significa timestamp.

La variable de salida del sistema será *track_tp*. Solo existen dos eventos: arribo de un avión, despegue de un avión.

Funcionamiento

Todo el código de simulación está en la clase Airport.

Luego de la inicialización de las variables, a continuación se explica una iteración de la simulación.

Si el tiempo de arribo de un nuevo avión *t_a* es menor que todos los demás tiempos entonces se adelanta el tiempo de simulación *t* hasta *t_a* y se trata de hacer aterrizar el avión en alguna pista, de ninguna estar disponible se incrementa el contador *cnt_q* poniendo el avión en cola. Si el avión puede aterrizar en la pista *i* lo que se hace es calcular el tiempo total en que el avión estará en la pista mediante el código en el listado 3 y se añade a la lista *track_tp[i]* el intervalo $[t, t + td]$ y se setea *tracks[i] = t + td*, donde *td* es el tiempo total calculado. Note que *track_tp* es una lista de listas, una lista que posee 5 listas(las pistas).

Si el tiempo en el cual el avión en la pista *i* termina de despegar es el menor de entre todos los tiempos de los eventos futuros se avanza el tiempo *t* hasta el tiempo *tracks[i]*, el cual dice el tiempo en que el avión en la pista *i* terminó de despegar. De haber algún avión en cola se selecciona la pista *i* para que aterrice el próximo avión y se sigue el mismo procedimiento que en el párrafo anterior. De no haber ningún avión en cola se setea *tracks[i] = inf*.

Consideraciones

Debido a que los aviones pueden aterrizar en cada pista sin ninguna preferencia o requerimiento, los tiempos en que se encuentran vacías cada pista son bastantes uniformes. Si se siguiera otra estrategia, por ejemplo seleccionar la pista con menor índice disponible para el aterrizaje, este patrón no se vería.

El tiempo máximo registrado entre todas las simulaciones por una pista fue de 2200 minutos aproximadamente, un día y medio. El mínimo fue de 1000 minutos aproximadamente, 16 horas.

Al parecer no hay diferencias significativas en los tiempos de aterrizar, rellenar combustible, cargar, descargar y despegar. Ningún tiempo influye mucho más que los demás en el tiempo total que se va a quedar el avión en la pista.