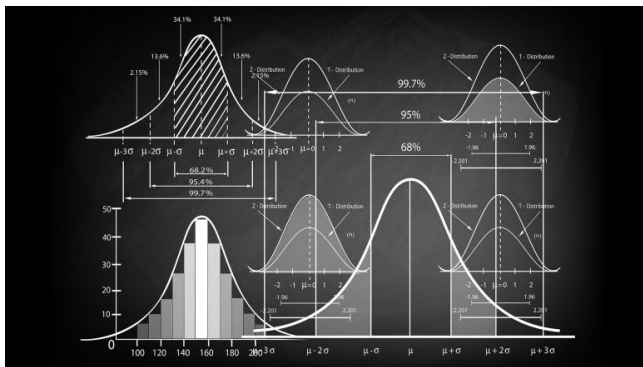




UNIVERSIDAD
POLITÉCNICA
DE MADRID



Métodos de Simulación.

Máster Universitario en Inteligencia Artificial
Universidad Politécnica de Madrid

Memoria Enunciado 7.

Ejercicio 2: Simulación de sucesos discretos: aeropuerto

Javier Cardeñosa Alabau.

Adrián Michelena Sanz.

Alberto Miño Calero.

Índice

1.	Descripción del problema de simulación.....	2
1.1	Modelo de aeropuerto	2
1.2	Objetivos de la simulación	3
2.	Sucesos discretos del modelo a estudiar	3
2.1	Validación de las distribuciones de las muestras	4
2.1.1	Distribución del tiempo de la maniobra de aterrizaje.....	4
2.1.2	Distribución del tiempo de desembarque	6
3.	La simulación como red de colas	7
4.	Implementación del sistema.....	9
4.1	Distribuciones de eventos para la generación de tiempos.....	9
4.2	Descripción de las rutinas de la simulación.....	11
4.2.1	Rutina de llegadas de aviones al aeropuerto	11
4.2.2	Rutina de aterrizaje: salida del nodo 1	13
4.2.3	Rutina de guiado: salida del nodo 2	14
4.2.4	Rutina de desembarque: salida del nodo 3	15
4.2.5	Rutina de despegue: salida del sistema.....	16
4.2.6	Sobre los contrastes.....	17
5.	Resultados.....	17
5.1	Aviones Aterrizados	18
5.2	Aviones despegados	18
5.3	Tiempo medio de la cola de aterrizajes	18
5.4	Tiempo medio de la cola de despegues	19
5.5	Número medio de aviones en el aeropuerto	19
5.6	Porcentaje de tiempo de pistas ocupadas.....	19
5.7	Conclusiones de los resultados.....	19

1. Descripción del problema de simulación

1.1 Modelo de aeropuerto

El problema de simulación se encuentra dentro del marco de funcionamiento de un aeropuerto. Para modelarlo, se establecen una serie de criterios que rigen su funcionamiento mediante una serie de mecanismos, que consisten lo siguiente:

- Los aviones llegan al aeropuerto según un proceso de Poisson no homogéneo de tasa expresada según la Figura 1.1.

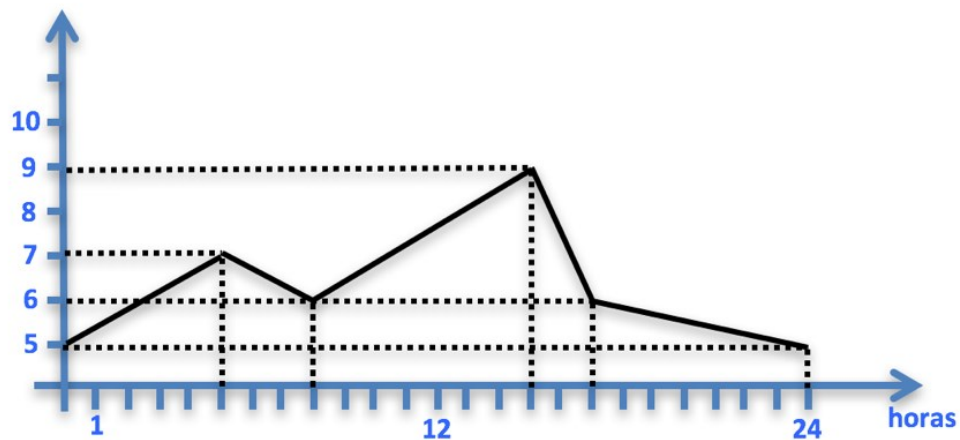


Figura 1.1 Tasa de llegada de los aviones

- Una vez llegan, los aviones esperan hasta que se les asigna pista de aterrizaje. Una vez asignada, inician la maniobra de aterrizaje, que les lleva un tiempo aleatorio de distribución la misma que en una muestra de tiempos proporcionada en *aterrizajes.txt*. Esta distribución, desconocida a priori, puede ser normal, uniforme o exponencial.
- Una vez aterriza, un avión espera a que un vehículo guía lo lleve al puesto de desembarco. El tiempo de traslado sigue una distribución exponencial de tasa 30 aviones por hora. Los tiempos de traslado del coche sin avión se consideran despreciables.
- La programación de la salida de los aviones es independiente del aterrizaje. Se tarda un tiempo en desembarcar pasajeros y equipaje y embarcar nuevos pasajeros y desplazarse a la zona donde se puede iniciar el despegue. Este tiempo es probabilístico según la distribución desconocida de la muestra *desembarques.txt*. Esta distribución, desconocida a priori, puede ser normal, weibull o exponencial.
- Una vez llegan a esta zona, los aviones esperan a que se les asigne pista para hacer la maniobra de despegue, cuya duración varía entre 10 y 15 minutos según una distribución uniforme. En este punto, pueden converger los aviones que esperan para aterrizar y los que esperan para despegar.

- El aterrizaje tiene prioridad sobre el despegue.
- El aeropuerto cuenta con 3 pistas de aterrizaje/despegue.
- Hay 20 vehículos guía.
- Una pista queda libre cuando un avión ha terminado su maniobra de aterrizaje.
- El aeropuerto cuenta con 50 puertas de embarque/desembarque.
- El aeropuerto está vacío en el instante inicial.

1.2 Objetivos de la simulación

El objetivo principal es implementar una simulación del comportamiento de un aeropuerto durante un período de tiempo de un mes, estimando varias medidas de rendimiento que servirán para evaluarlo. Las medidas de rendimiento son las siguientes:

- Tiempos medios de espera de los aviones, tanto para aterrizaje como para despegue.
- Tiempos máximos de espera de los aviones, tanto para aterrizaje como para despegue.
- Número medio de aviones en el aeropuerto.
- Porcentaje de tiempo que las pistas están ocupadas.

Una vez se cuente con estas medidas, se modificará el sistema de dos maneras diferentes:

- Construir una nueva pista de aterrizaje/despegue.
- Contratar 5 coches guías adicionales.

La simulación se verá modificada y se comprobarán los cambios en las medidas de rendimiento que cada modificación del aeropuerto provoca. En función de las nuevas medidas, se podrá evaluar cual de todas las configuraciones mejora el desempeño del sistema.

2. Sucesos discretos del modelo a estudiar

En esta simulación existen varios tipos de sucesos de carácter discreto, comentados en el modelo del aeropuerto, que ocurren de manera independiente. Junto a los mecanismos de funcionamiento del aeropuerto, la ocurrencia de estos sucesos dará lugar a la generación de los tiempos que rijan el flujo de la simulación.

- A. Llegada de un avión al aeropuerto.
- B. Salida de la pista tras la finalización de la maniobra de aterrizaje.
- C. Finalización del guiado del avión desde la pista de aterrizaje hasta la terminal por parte de un vehículo.

- D. Salida de un avión de la puerta de embarque, tras hacer el desembarque de pasajeros, equipaje, embarque de nuevos pasajeros y llegar a la zona donde puede iniciar la maniobra de despegue.

Cada uno de estos sucesos da lugar a un proceso que dura un tiempo y que debe modelarse para poder incluirse en la simulación. En el caso del tiempo de la maniobra de aterrizaje y el tiempo de desembarque, se proporcionan muestras de apoyo. Estas muestras deben estudiarse para comprobar que distribución siguen. El resto de tiempos siguen distribuciones preestablecidas, siendo la primera un proceso de Poisson no homogéneo tal y como se muestran en la Figura 1.1, la segunda una exponencial de parámetro 2 (este parámetro indica que un coche tiene una tasa de desplazamiento de aviones de 2 minutos por avión), y la tercera una uniforme entre 10 y 15 minutos.

2.1 Validación de las distribuciones de las muestras

Para averiguar la distribución que siguen los aterrizajes y los desembarques, hay que estudiar las muestras que se proporcionan. Para hacer este estudio, se ha seguido un proceso que consiste en lo siguiente:

1. En primer lugar, se va a representar el histograma de la muestra, ya que puede ofrecer una idea visual aproximada de la distribución que sigue. En función del aspecto de este histograma, podrán ocurrir dos cosas:
 - 1.1. El histograma muestra una distribución identificable.
 - 1.2. El histograma no aporta información sobre la misma.
2. En el caso 1.1, entonces se realizará un contraste de Kolmogorov-Smirnov (K-S) que nos permitirá confirmar la hipótesis de que la muestra sigue la distribución visualizada en el histograma.
3. En el caso 1.2, se tendrán que realizar contrastes K-S para cada una de las distribuciones que la muestra puede seguir, de acuerdo con las restricciones establecidas en el apartado del modelo del aeropuerto.
4. Una vez se confirme qué distribución sigue la muestra, se estimarán sus parámetros.

Para la validación de las hipótesis sobre la distribución teórica que siguen las muestras, se ha escogido el contraste K-S frente al de la χ^2 debido a que su potencia es superior y para distribuciones continuas, como es el caso, permite contrastar la hipótesis. Para hacer este contraste, vamos a aprovechar el paquete estadístico SPSS, y en concreto su versión de prueba gratuita. Con este paquete, se puede realizar el contraste de forma rápida y sencilla, incluyendo los datos de las muestras proporcionadas y realizando la prueba no paramétrica correspondiente a la K-S para una muestra.

2.1.1 Distribución del tiempo de la maniobra de aterrizaje

Siguiendo el proceso descrito, primero vamos a representar el histograma, por intervalos, de la muestra.

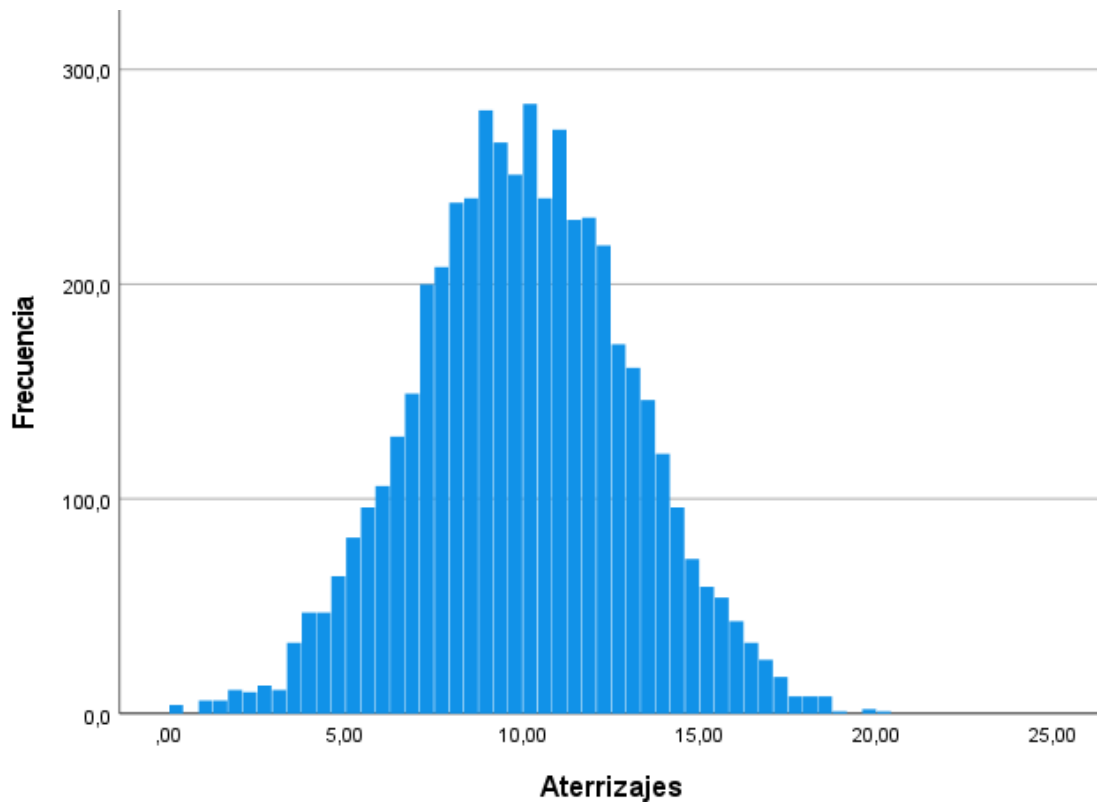


Figura 2.1 Histograma de las muestras de tiempos de aterrizaje

En este primer caso, se puede observar a simple vista que el histograma tiene una forma que encaja claramente con una distribución normal de media aproximadamente 10, por lo que podemos establecer la hipótesis de que los tiempos que tardan los aviones en ejecutar sus maniobras de aterrizaje siguen una distribución normal.

Con la información del histograma de la Figura 2.1, vamos a realizar un contraste K-S sobre la muestra. Con este contraste se podrá verificar si la hipótesis que hemos planteado puede validarse o no. En la Tabla 2.1 podemos observar los resultados. Seleccionada la normal como distribución teórica o hipótesis, observamos el valor *Sig. Asintot. (bilateral)*. Al tener un valor superior a 0,05, no se rechaza la hipótesis nula y se puede concluir que la muestra presenta evidencias suficientes como para confirmar la validez de nuestra hipótesis inicial con un nivel de significación del 5%. Además, el sistema también nos proporciona el valor de sus parámetros.

Aterrizajes			
N		5000	
Parámetros normales ^{a,b}	Media	10,0076	
	Desv. Desviación	3,03770	
Máximas diferencias extremas	Absoluta	,010	
	Positivo	,010	
	Negativo	-,006	
Estadístico de prueba		,010	
Sig. asin. (bilateral) ^c		,200 ^d	
Sig. Monte Carlo (bilateral) ^e	Sig.	,232	
	Intervalo de confianza al 99%	Límite inferior	,221
		Límite superior	,243

a. La distribución de prueba es normal.

b. Se calcula a partir de datos.

Tabla 2.1 Contraste de K-S de la muestra de los tiempos de aterrizaje en el sistema SPSS

Podemos concluir tras el contraste que el tiempo que tardan los aviones en realizar las maniobras de aterrizaje sigue una distribución normal de media 10,0076 y desviación típica 3,03770.

$$N(\mu = 10,0076, \sigma = 3,03770)$$

2.1.2 Distribución del tiempo de desembarque

Primero, representamos el histograma de la muestra por intervalos.

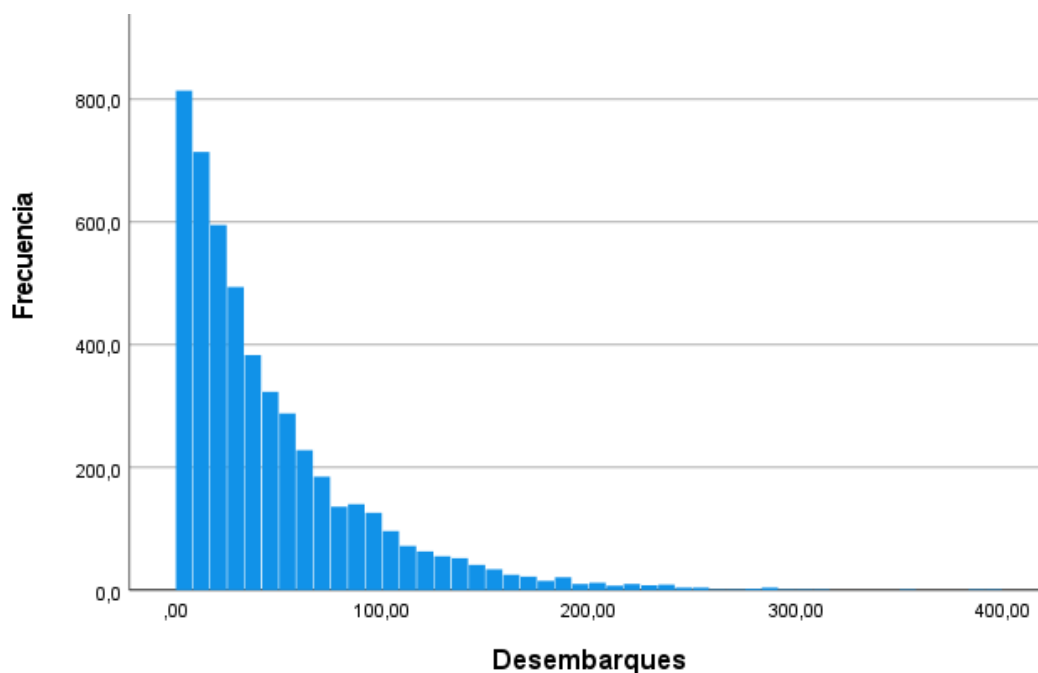


Figura 2.2 Histograma de las muestras de tiempos de desembarque

En el histograma de esta muestra representado en la Figura 2.2, puede verse de nuevo la forma de una distribución conocida incluida en las restricciones impuestas en el modelo, la de una exponencial. Se considerará, por tanto, siguiendo el mismo proceso anterior, la hipótesis de que esta muestra sigue una distribución exponencial.

Para comprobar su validez, volvemos a utilizar SPSS para hacer un contraste K-S, esta vez con una distribución teórica exponencial. En la Tabla 2.2 se pueden revisar los resultados. Atendiendo al parámetro de *Sig. Monte Carlo (bilateral)*, se puede observar que tiene un valor superior a 0,05. Por tanto, se puede decir con un nivel de significación del 5% que hay evidencias suficientes para considerar válida la hipótesis de que nuestra muestra sigue una distribución exponencial.

		Desembarques
N		5000
Parámetro exponencial. a. b	Media	45,9012
Máximas diferencias extremas	Absoluta	,009
	Positivo	,009
	Negativo	-,007
Estadístico de prueba		,009
Sig. Monte Carlo (bilateral) c	Sig.	,566
	Intervalo de confianza al 99%	Límite inferior
		Límite superior

a. La distribución de prueba es exponencial.

b. Se calcula a partir de datos.

Tabla 2.2 Contraste de K-S de los tiempos de desembarque en el sistema SPSS

Podemos concluir tras el contraste que los tiempos de desembarque, siguen una distribución exponencial de parámetro 45,9012.

$$\exp(\lambda = 45,9012)$$

3. La simulación como red de colas

En este trabajo, siendo una simulación de sucesos discretos en un sistema de espera que podría considerarse complejo, se va a seguir la estructura de una red de colas. En base a esto, la premisa es que en la simulación de este modelo de aeropuerto van a existir una serie de nodos en los que se va a prestar un servicio a unos clientes, que serán los aviones, y en donde existirán unos tiempos de espera o de cola que variarán dependiendo de en qué nodo se encuentre el avión y de las características del servicio que dicho nodo preste.

Siguiendo la estructura del aeropuerto y su flujo de funcionamiento, se van a ir produciendo una serie de eventos que harán avanzar la simulación en el tiempo. Estos eventos, partiendo del evento inicial “abstracto” de inicio de simulación, son los que se han descrito anteriormente.

Vamos a describir ahora los nodos en los que se puede descomponer nuestro sistema. En base a los servicios que se “ofrecen” a los aviones, y de acuerdo con la estructura del aeropuerto y la distribución de eventos, se pueden distinguir 3 nodos diferentes: el nodo de las pistas de aterrizaje/despegue, el nodo de guiado de los aviones, y el nodo de la terminal del aeropuerto (puertas de embarque). Conceptualmente, estos son los 3 nodos que componen nuestro sistema.

Durante la construcción del sistema, sin embargo, se ha tomado una decisión de implementación de separar el nodo de pista en las colas que lo componen, una para aterrizajes y otra para despegues, de manera que la gestión de estas quede separada y bien diferenciada. El motivo de esta separación se debe a que aportaba ventajas y facilidades a la hora de implementar esta gestión de colas respecto a haber tenido que gestionarlas en la misma rutina. No obstante, esta partición en la implementación no se contradice con que, conceptualmente, solo haya un nodo pista, puesto que la separación no es más que una división de la gestión de colas en rutinas, y se trabaja con los mismos elementos, los mismos parámetros datos que si se gestionase todo en una única rutina.

Tenemos, por tanto, la siguiente división en nodos:

Nodo 1. Pistas de aterrizaje y despegue.

- 1.a. Se corresponde con la entrada y salida de los aviones al sistema. Al ser un nodo de entrada y salida, y al ser necesario un tiempo de servicio en ambos procesos, necesita repartir su capacidad de servicio entre los aviones que salen y que entran, con la restricción de que los que entran tienen prioridad.
- 1.b. Tiene dos colas de espera, una para los aviones que han llegado al sistema y esperan para aterrizar y otra para los que esperan para despegar y salir del aeropuerto. Para su gestión, las dos colas quedarán separadas en rutinas diferentes.

Nodo 2. Guiado de aviones.

- 2.a. Se corresponde con el servicio de guiado de los aviones realizado por los coches, que ocurre una vez terminan la maniobra de aterrizaje (nodo 1) y dura hasta que llegan a una puerta de desembarque libre (nodo 3).
- 2.b. Tiene una única cola, en la que los aviones que acaban de aterrizar tendrán que esperar en el caso de que no haya ningún coche guía libre.

Nodo 3. Terminal del aeropuerto.

- 3.a. Se corresponde, una vez el avión ha sido guiado (nodo 2) con el servicio de desembarque de pasajeros y equipaje de un avión, el embarque de

los nuevos, y el desplazamiento hasta un lugar del aeropuerto desde el que puedan iniciar la maniobra de despegue (nodo 1) para dejar el sistema.

- 3.b. Tiene una sola cola en la que los aviones, tras haber sido guiados por un coche, tendrán que esperar en el caso de que no haya ninguna puerta de embarque libre en la terminal.

La llegada de un avión al aeropuerto es el evento que activa el proceso, llevando un cliente a la cola del primer nodo, que tendrá que pasar por todos los nodos hasta el 3. Una vez termine el servicio en el tercer nodo, se considerará que el avión entra en la cola de despegue del nodo 1. En la Figura 3.1, se puede ver un esquema que ilustra el proceso que siguen los aviones desde que entran hasta que salen del aeropuerto.

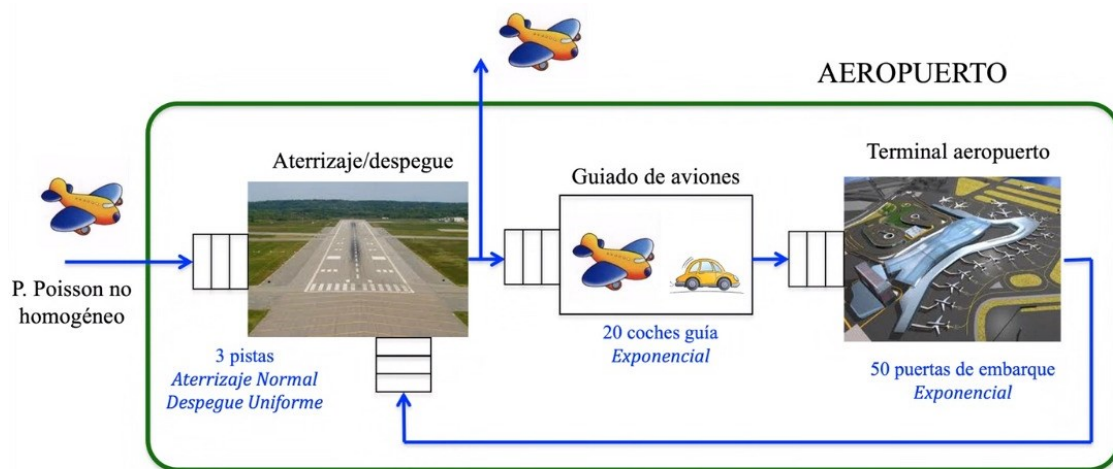


Figura 3.1 Esquema de nodos y colas de espera del aeropuerto

4. Implementación del sistema

Una vez conocemos los nodos y las colas del sistema, lo único que nos queda para poder montar la simulación son los tiempos de servicio. Según el modelo del aeropuerto, estos tiempos de servicio se corresponderán con los procesos que deben realizarse entre que un avión termina la cola de espera de un nodo, entra en el mismo y desarrolla la actividad que corresponda, y sale del nodo para entrar en otra cola o salir del sistema. Ya se han descrito cuales son estos, y ahora se va a describir cómo se simulan estos procesos de manera que se generen valores realistas que representen lo que, según el modelo, los aviones y demás elementos del sistema podrían tardar en desempeñar sus tareas en los nodos.

4.1 Distribuciones de eventos para la generación de tiempos

A partir de los parámetros de funcionamiento del sistema, se conocen las distribuciones que siguen los tiempos de las llegadas de los aviones y el guiado de los mismos. Por otro lado, en base a lo que se ha estudiado en el apartado anterior,

se han averiguado las distribuciones que siguen los tiempos de aterrizaje y de desembarque de los aviones:

- A. Llegadas: sigue una distribución de **Poisson no homogénea** según la Figura 1.1.
- B. Aterrizajes: sigue una distribución **Normal de media 10,0076 y desviación típica 3,0377**.
- C. Guiado: sigue una distribución **Exponencial de tasa 2**.
- D. Desembarques: sigue una distribución **Exponencial de tasa 45,9012**.
- E. Despegues: sigue una distribución **Uniforme entre 10 y 15 minutos**.

A continuación, se puede encontrar el pseudocódigo que sería el programa principal (main) del simulador, el cual gestionara los distintos eventos y colas.

```
def simul_main():  
  
    X = GenerarExponencial(lambdaRespectoT(t)) # Generamos el tiempo de la primera llegada  
  
    # El tiempo generado es mayor que el tiempo de simulación  
  
    if X>T:  
  
        tMedColaAterrizaje = tMedColaDespegue = tMaxColaAterrizaje = tMaxColaDespegue =  
nMedAviones = porcentajeTPistasOcupadas = 0  
  
        # El tiempo generado es menor que el tiempo de simulación  
  
    else:  
  
        llegada_avion(X,0) # Llamamos a la función de llegada  
  
  
        # Mientras no hayamos llegado a tiempo de simulación y haya eventos pendientes  
seguimos con esta  
  
        while (t<=T and (len(LISTATLL1)>0 or len(LISTATS1)>0 or len(LISTATS2)>0 or  
len(LISTATS3)>0 or len(LISTATS4)>0)):  
  
            # Recorremos las listas y nos quedamos con el evento mas cercano, el avion y el  
tipo  
  
            eventInstante,eventoIdAvion,eventoTipo = RecorrerListas()  
  
            # Lanzamos el evento  
  
            if eventType == "Llegada": llegada_avion(eventTmin,eventplaneId)  
  
            elif eventType == "Aterrizaje":  
rutina_nodo_pista_aterrizaje(eventTmin,eventplaneId)  
  
            elif eventType == "Guiado": rutina_nodo_vehiculo(eventTmin,eventplaneId)  
  
            elif eventType == "Embarcado": rutina_nodo_terminal(eventTmin,eventplaneId)  
  
            elif eventType == "Despegado": rutina_nodo_pista_despegue  
(eventTmin,eventplaneId)
```

```

#Calcular número medio de aviones en el aeropuerto

nMedAviones = (nmed_n1/t)+(nmed_n2/t)+(nmed_n3/t)+(nmed_n4/t)

#Calcular porcentaje de tiempo que las pistas estan ocupada

porcentajeTPistasOcupadas = nmed_pistasOcupadas/(t*numPistas)

#Calcular tiempos medios y maximos de cola de aterrizaje y despegue

tMedColaAterrizaje,MedColaDespegue,tMaxColaAterrizaje,tMaxColaDespegue      =
calcularTiemposMedMaxCola()

```

Frag. código 1 Pseudocódigo principal de la simulación

4.2 Descripción de las rutinas de la simulación

4.2.1 Rutina de llegadas de aviones al aeropuerto

Esta rutina se da cuando llega un nuevo avión al sistema. Si hay pistas disponibles, se le dará paso para aterrizar y se calculará su tiempo de aterrizaje, que se describirá en el siguiente apartado. Los aviones seguirán llegando hasta que se alcance el tiempo horizonte de simulación.

La llegada de los aviones sigue una distribución de **Poisson no homogénea**, tal y como muestra la Figura 1.1. Para generar los tiempos en los que ocurre, lo que se ha hecho es descomponer el proceso de Poisson en intervalos de tiempo, ya que dependiendo de la hora del día llegan más o menos aviones. Cada uno de estos intervalos se puede expresar como una ecuación lineal en función del tiempo, permitiendo calcularlo entre una llegada y la siguiente en cada una de estas rectas dependiendo de en qué instante (a qué hora del día) se encuentre la simulación cada vez que se genera un valor. Las ecuaciones para cada uno de los intervalos de tiempo se pueden ver en la siguiente ecuación Ec 1.

$$\lambda(t) = \begin{cases} \frac{2}{5}t + 5 & \text{si } 0 < t \leq 5 \\ -\frac{1}{3}t + \frac{26}{3} & \text{si } 5 < t \leq 8 \\ \frac{3}{7}t + \frac{18}{7} & \text{si } 8 < t \leq 15 \\ -\frac{2}{3}t + \frac{63}{2} & \text{si } 15 < t \leq 17 \\ -\frac{1}{7}t + \frac{59}{7} & \text{si } 17 < t \leq 24 \end{cases}$$

Ec 1

Por tanto, se tiene el día dividido en 5 intervalos, y para aproximar el proceso de Poisson lo que se hace es comprobar el instante en el que se va a generar una llegada. Sabiendo a qué franja del día pertenece dicho instante, se puede ubicar en uno de esos intervalos para luego generar un valor con una distribución exponencial

cuyo parámetro vendrá dado por la resolución de la ecuación que corresponda a ese intervalo con el valor de tiempo correspondiente al instante en el día.

Dado un instante de tiempo de la simulación en el que se necesita averiguar cuándo falta para que llegue un nuevo avión, el cálculo se hace de acuerdo con los intervalos de la Ec 1, tal y como se puede ver en el siguiente fragmento de la implementación:

```
def lambdaRespectoT(t):  
    tiempo = t/60  
    tiempo = tiempo%24  
    if(tiempo<5):  
        lambdaInT = (2*tiempo/5)+5  
    elif(tiempo<8):  
        lambdaInT = (-1*tiempo/3)+(26/3)  
    elif(tiempo<15):  
        lambdaInT = (3*tiempo/7)+(18/7)  
    elif(tiempo<17):  
        lambdaInT = (-3*tiempo/2)+(63/2)  
    elif(tiempo<=24):  
        lambdaInT = (-1*tiempo/7)+(59/7)  
    return lambdaInT
```

Frag. código 2 Generación de parámetro para exponencial a partir de intervalos de Poisson no homogéneo

Una vez se conoce el valor del parámetro, se genera el tiempo mediante una exponencial con dicho parámetro, y se devuelve ese tiempo a la rutina de la simulación que necesitare generarlo. En caso de haber pistas libres, también se calcula el tiempo de servicio del nodo 1. Como se ya se ha validado en un apartado anterior, se conoce que este tiempo de servicio del primer nodo, que se refiere a los aterrizajes, sigue una distribución Normal de media 10,0076 y desviación típica 3,0377. Por tanto, su generación es bastante sencilla, y basta con hacer uso de alguna función que pueda devolver valores de esta distribución en base a sus parámetros.

```
def llegada_avion(tsuc, idAvion):  
  
    A esta parte en gris la llamaremos InicioRutina() en el resto de pseudocodigos  
  
    #Sumo las ocupaciones de los nodos así como las de las pistas.  
  
    nmed_n1 += n1 * (tsuc-t); #mismo para n2,n3,n4,pistasOcupadas  
  
    #Actualizo el tiempo de simulación al instante actual  
  
    t=tsuc  
  
    #Sumo 1 al n° de aviones en el nodo 1, nodo de aterrizaje.  
  
    n1 += 1  
  
    #Añado el instante al que llega el avión idAvion a la lista de llegadas (LL1)
```

```

    LL1[idAvion] = tsuc

    #Genero siguiente llegada con exponencial tasa lambda para el t actual

    Y = GenerarExponencial(lambdaRespectoT(t))

    #Si el tiempo programado es menor que T, programo siguiente llegada con el
    tiempo generado. Si es mayor se acaban las llegadas.

    if (t+Y)<T:

        NLL1 +=1 #Aumentamos el numero de llegadas, usaremos como idAvion

        LISTATLL1.append([(t+Y), (NLL1)])

        #Si hay pistas disponibles programar aterrizaje. Genero el tiempo de aterrizaje,
        guardo el instante en que será y ocupó la pista

        if pistasOcupadas < numPistas:

            A esta parte en gris la llamaremos ProgramarAterrizaje() en el resto de
            pseudocodigos

            Z = GenerarNormal(mediaAterrizajes, desviacionTipicaAterrizajes)

            LISTATS1.append([(t+Z), idAvion])

            ColaAterrizaje.append([0, idAvion])

            pistasOcupadas+=1

        #Si no hay pistas disponibles añadir a la cola aterrizaje.

    else:

        EnColaAterrizaje.append(idAvion)

```

Frag. código 3 Pseudocódigo rutina de llegada de aviones al aeropuerto

4.2.2 Rutina de aterrizaje: salida del nodo 1

Ocurre cada vez que un avión termina su aterrizaje y sale del nodo 1. Esta rutina calcula el tiempo de servicio del nodo 1 para el siguiente avión en aterrizar o despegar en caso de que haya alguno en cola, y dando prioridad al aterrizaje.

El tiempo de servicio de este nodo referente al aterrizaje ya se ha explicado en la rutina anterior, y la parte de despegue se comentará más tarde, siguiendo el orden del flujo de los aviones por el aeropuerto. En esta rutina, también se genera el tiempo de guiado de los aviones. En este caso, queda establecido en el modelo del aeropuerto que los guiados siguen una distribución Exponencial de tasa 30 aviones por hora. Haciendo una conversión, se puede establecer que la tasa para representar el tiempo que tarda cada coche en guiar un avión es de 2 minutos, quedándose en una **Exponencial de tasa 2**.

```

def rutina_nodo_pista_aterrizaje(tsuc, idAvion):

    InicioRutina()

```

```

pistasOcupadas-=1 #Vaciamos la pista

n1 -=1; n2 += 1 #Decrementamos el n° de aviones del nodo 1 e incrementamos el del
nodo 2

S1[idAvion] = tsuc #Registramos el instante de salida del avión del nodo 1.

#Si hay aviones en cola de aterrizaje(programo aterrizaje). Si no hay, si hay aviones
en cola de despegue(programo despegue).

if len(EnColaAterrizaje)>0:

    #Programamos aterrizaje del primer avion en cola, calculamos el tiempo que ha estado
    en cola[t-LL1]. Lo quitamos de la cola

    ProgramarAterrizaje()

elif len(EnColaDespegue)>0:

    #Programamos despegue del primer avion en cola, calculamos el tiempo que ha estado
    en cola[t-S3]. Lo quitamos de la cola

    ProgramarDespegue()

#Si hay vehículos disponibles programar guiado. Genero el tiempo de guiado,
guardo el instante en que será y ocupó el vehículo

if vehiculosOcupados < numVehiculos :

    A esta parte en gris la llamaremos ProgramarGuiadoVehiculo() en el resto de
    pseudocodigos

    Z = GenerarExponencial(TasaDistribucionVehiculos) #Genero el tiempo que tarda en
    llevar el coche al terminal

    LISTATS2.append([(t+Z),(idAvion)]) #Actualizo en la lista el instante en el que
    saldrá del nodo 2.

    ColaVehiculos.append([0,idAvion])

    vehiculosOcupados+=1

#Si no hay vehículos disponibles añadir a la cola de vehiculos.

else:

    EnColaVehiculos.append(idAvion)

```

Frag. código 4 Pseudocódigo rutina de aterrizajes de aviones al aeropuerto

4.2.3 Rutina de guiado: salida del nodo 2

Ocurre cada vez que un avión termina su guiado y sale del nodo 2. Esta rutina calcula el tiempo de servicio del nodo 2 para el siguiente avión que vaya a ser guiado. En caso de que haya aviones en la cola de guiado, se calculará su tiempo de servicio.

El tiempo de servicio de este nodo, referente al guiado, se ha descrito anteriormente. En esta rutina, también se genera el tiempo de desembarque de los aviones. Al igual que en el caso de los aterrizajes, ya se ha validado en un apartado anterior la distribución de los tiempos que duran este tipo de procesos, que es un **Exponencial de parámetro 4,9012**.

```

def rutina_nodo_vehiculo(tsuc, idAvion):

    InicioRutina()

    vehiculosOcupados-=1 #Vaciamos el vehiculo

    n2 -=1; n3 += 1 #Decrementamos el nº de aviones del nodo 2 e incrementamos el del
nodo 3

    S2[idAvion] = tsuc #Registramos el instante de salida del avión del nodo 2.

    #Si hay aviones en cola de guiado(programo guiado)

    if len(EnColaVehiculos)>0:

        #Programamos guiado del primer avion en cola, calculamos el tiempo que ha estado
en cola[t-S1]. Lo quitamos de la cola

        ProgramarGuiadoVehiculo()

        #Si hay puertas disponibles programar desembarque. Genero el tiempo de guiado,
guardo el instante en que será y ocupó el vehiculo

        if puertasOcupadas < numPuertas :

            A esta parte en gris la llamaremos ProgramarDesembarque() en el resto de
pseudocodigos

            Z = GenerarExponencial(TasaDistribucionPuertas) #Genero el tiempo que tarda en
preparar desembarque y embarque

            LISTATS3.append([(t+Z),(idAvion)]) #Actualizo en la lista el instante en el que
saldrá del nodo 3.

            ColaPuertas.append([0,idAvion])

            puertasOcupadas+=1

        #Si no hay puertas disponibles añadir a la cola de puertas de desembarque.

    else:

        EnColaPuertas.append(idAvion)

```

Frag. código 5 Pseudocódigo rutina de guiado a los aviones

4.2.4 Rutina de desembarque: salida del nodo 3

Ocurre cada vez que acaba el desembarque de un avión y sale del nodo 3. Esta rutina calcula el tiempo de servicio del nodo 3 para el siguiente avión que vaya a desembarcar.

El tiempo de servicio de este nodo, referente al desembarque, se ha descrito en el apartado anterior. En esta rutina, también se genera el tiempo de despegue de los aviones. En caso de que haya pistas libres, se genera este tiempo para el siguiente avión. En este caso, la distribución de tiempo viene fijada en el modelo, y es una **Uniforme entre 10 y 15 minutos**.


```

def rutina_nodo_terminal(tsuc, idAvion):

    InicioRutina()

    puertasOcupadas-=1 #Vaciamos la puerta

    n3 -=1; n4 += 1 #Decrementamos el n° de aviones del nodo 3 e incrementamos el del
nodo 1 en la parte de despegues

    S3[idAvion] = tsuc #Registramos el instante de salida del avión del nodo 3.

    #Si hay aviones en cola de puerta de desembarque(programo desembarque)

    if len(EnColaPuertas)>0:

        #Programamos puerta de desembarque del primer avion en cola, calculamos el tiempo
que ha estado en cola[t-S2]. Lo quitamos de la cola

        ProgramarDesembarque()

        #Si hay pista disponibles programar despegue. Genero el tiempo de despegue,
guardo el instante en que será y ocupó la pista

        if pistasOcupadas<numPistas:

            A esta parte en gris la llamaremos ProgramarDespegue() en el resto de
pseudocodigos

            Z = GenerarUniforme(minDespegues,maxDespegues) #Genero el tiempo que tarda en
despegar

            LISTATS4.append([(t+Z),(idAvion)]) #Actualizo en la lista el instante en el que
saldrá del nodo 1 en la parte de despegues.

            pistasOcupadas+=1

            ColaDespegue.append([0,idAvion])

        else:

            EnColaDespegue.append(idAvion)

```

Frag. código 6 Pseudocódigo rutina de desembarque y embarque en la terminal

4.2.5 Rutina de despegue: salida del sistema

Esta rutina se refiere a la salida de los aviones del sistema, que se hace a través del nodo 1. Su funcionamiento es igual al de la rutina de aterrizaje en cuanto a la programación de siguientes aterrizajes o despegues, pero en este caso como los aviones están saliendo del aeropuerto, ya no ocurre el guiado. Aquí es donde residía la principal problemática en la gestión de los aviones, ya que vuelven al nodo 1 pero no se les puede tratar como los aviones que aterrizan. Cabe recordar que la cola de despegue tiene menor prioridad que la de aterrizaje, lo que significa que un avión no podrá despegar hasta que no haya una pista libre y no haya ningún avión esperando para aterrizar.

```

def rutina_nodo_pista_despegue(tsuc, idAvion):

    InicioRutina()

    pistasOcupadas-=1 #Vaciamos la pista

```

```

n4 -=1 #Decrementamos el n° de aviones del nodo 1 en la parte de despegues

S4[idAvion] = tsuc #Registramos el instante de salida del avión del nodo 1 en la
parte de despegues.

#Si hay aviones en cola de aterrizaje(programo aterrizaje). Si no hay, si hay aviones
en cola de despegue(programo despegue).

if len(EnColaAterrizaje)>0:

    #Programamos aterrizaje del primer avion en cola, calculamos el tiempo que ha estado
    en cola[t-LL1]. Lo quitamos de la cola

    ProgramarAterrizaje()

elif len(EnColaDespegue)>0:

    #Programamos despegue del primer avion en cola, calculamos el tiempo que ha estado
    en cola[t-S3]. Lo quitamos de la cola

    ProgramarDespegue()

```

Frag. código 7 Pseudocódigo rutina de despegue de aviones

4.2.6 Sobre los contrastes

Cabe mencionar que, en el caso de tratarse de una simulación crítica, lo correcto es comprobar que cada una de las funciones utilizadas para generar valores de las que se necesitan lo hacen correctamente. Es decir, habría que generar muestras de las distribuciones con las funciones seleccionadas, y aplicar un contraste de distribución para validar si realmente siguen esa distribución.

En este trabajo, por cuestiones de tiempo y por la dificultad que ha supuesto el problema de programación, se ha confiado en las funciones de las librerías *random* y *numpy* de *Python*, utilizando como justificación, aunque solo sea hasta cierto punto, que se tratan de unas librería populares y muy utilizadas de un lenguaje con mucha tirada en cuestiones estadísticas.

5. Resultados

A continuación, mostraremos los resultados obtenidos. Se mostrarán los resultados obtenidos en la simulación original, además de los obtenidos usando 4 pistas, así como 5 vehículos adicionales.

Las principales medidas de rendimiento que analizaremos son: tiempos medios y máximos de espera de los aviones para aterrizar y despegar, número medio de aviones en el aeropuerto y el porcentaje de tiempo que las pistas están ocupadas.

Además, se añadirán algunos datos que nos resultan interesante a la hora de apreciar las diferencias cuando se varían los parámetros del número de pistas y de vehículos guía.

	Simulación Original	Simulación 4 Pistas	Simulación 25 Coches
Nº aviones aterrizados	6759	6860	6844
Nº aviones despegados	4978	6826	4852
% de aviones despegados	73.65%	99.50%	70.89%
Tº Med Cola Aterrizaje(min)	5.30	2.73	5.23
Tº Max Cola Aterrizaje(min)	30.44	20.05	29.53
Tº Med Cola Despegue(min)	5816.82	48.98	5995.80
Tº Max Cola Despegue(min)	11592.80	274.14	12375.93
Nº medio aviones aeropuerto	926.08	19.47	962.89
Tº pistas ocupadas (%)	99.95%	89.16%	99.98%

Tabla 5.1 Resultados de las medidas de rendimiento de las simulaciones

5.1 Aviones Aterrizados

Observamos que el número de aviones aterrizados es bastante parecido en las 3 simulaciones, aunque es algo mayor en la simulación con 4 pistas, seguido de la de 5 vehículos extra.

5.2 Aviones despegados

Al igual que en el apartado anterior, cabe esperar que el número de aviones despegados aumente al tener una pista más. En este caso vemos que el número de aviones despegados en la simulación normal es bastante parecido a la simulación con 5 vehículos adicionales.

Si analizamos el porcentaje de aviones despegados con respecto al total de aviones, vemos que despegan prácticamente el 100% de los aviones que llegan en el caso de las 4 pistas. Esto nos puede hacer pensar que el principal problema se encuentra en el nodo de las pistas.

5.3 Tiempo medio de la cola de aterrizajes

En cuanto al tiempo medio en la cola de aterrizajes se reduce al añadir una pista. Esto es bastante lógico. Añadimos un servidor más en el nodo y permitimos

que se dé servicio a un avión más en el nodo de pistas. Con esto reducimos considerablemente el tiempo que los aviones deben esperar para aterrizar.

Al igual, vemos que el tiempo máximo que se encuentran los aviones en dicha cola se ve reducido en la simulación con 4 pistas.

5.4 Tiempo medio de la cola de despegues

En esta parte es donde se observa el principal cambio entre las diferentes simulaciones y encontramos el gran problema. Vemos que el tiempo medio de la cola de despegues se reduce considerablemente al añadir una pista. Esto nos hace pensar que el nodo de pistas es un cuello de botella teniendo 3 pistas. Esto, como vemos, se resuelve añadiendo una pista más.

Confirmamos que la cola de despegues se reduce en la simulación de 4 pistas. Esto nos permite que los aviones que llegan al aeropuerto puedan despegar y no se queden atascados en la cola de despegues dándole preferencia a los aviones que llegan.

5.5 Número medio de aviones en el aeropuerto

Esta parte nos permite seguir confirmando que, al tener una pista más, no se crea un cuello de botella en el nodo pista y los aviones no se quedan atascados en la cola de despegues. Esto permite que los aviones puedan abandonar el sistema y el número medio de aviones en el sistema (aeropuerto) se reduzca en gran cantidad.

5.6 Porcentaje de tiempo de pistas ocupadas

Observamos que al tener una pista más, hay un mayor tiempo en el que hay al menos una pista libre. Esto es precisamente el motivo por el que los aviones en la cola de despegue (que no tienen preferencia sobre los que aterrizan) puedan despegar.

5.7 Conclusiones de los resultados

Como conclusión, podemos ver que en la simulación original se crea un cuello de botella en el nodo de las pistas. Esto se debe a que, al tener 3 pistas y que los aviones que aterrizan tengan preferencia sobre los que despegan, no existan instantes en los que los aviones en la cola de despegue tengan pistas libres para salir. Esto vemos que se resuelve al añadir una pista más. Vemos que el porcentaje de aviones que despega se incrementa mucho al añadir una pista solucionando ese colapso en el nodo.

En cuanto a la simulación con los 5 vehículos adicionales, vemos que no existen diferencias significativas con respecto a la simulación original. Esto se debe a que nuestro problema no se encuentra en el nodo de los vehículos ni en la terminal, sino que está en el nodo de las pistas, y aunque se aumente su número los tiempos de

cola en el nodo pista no se reduce. De hecho, para el despegue llega a aumentar, lo cual puede deberse a que los aviones pasan por el resto sistema más rápido desde el aterrizaje hasta el despegue, en donde quedan varados al estar el aeropuerto colapsado.