



Universitat  
Autònoma  
de Barcelona



# 5558,5557: Desarrollo de un modelo probabilístico y aplicación en Java para el cálculo de tiempos de Taxi Out.

Memòria del Treball Fi de Grau  
Gestió Aeronàutica  
realitzat per  
Oriol Sánchez Ramírez  
Daniel Pascual Sánchez  
i dirigit per  
Miquel Àngel Piera Eroles  
Sabadell, a 11 de Juliol de 2013



El sotasingnat, .....

Professor/a de l'Escola d'Enginyeria de la UAB,

**CERTIFICA:**

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en

I per tal que consti firma la present.

Signat: .....

Sabadell, .....de.....de 200.....

## **RESUMEN**

### **CASTELLANO**

En este proyecto se ha desarrollado un nuevo modelo para el cálculo de tiempos de Taxi-Out así como una aplicación escrita en el lenguaje de programación Java que hace uso de una base de datos SQL. La aplicación integra y utiliza el modelo presentado en combinación con la simulación orientada a eventos discretos para intentar calcular de forma más fidedigna los tiempos de recorrido de una aeronave, desde que sale de parking hasta su llegada a cabecera de pista.

Palabras clave: Taxi Out, taxi, simulación, SED, modelo probabilístico.

### **CATALÀ**

En aquest projecte s'ha desenvolupat un nou model pel càlcul de temps de Taxi-Out així com una aplicació escrita en el llenguatge de Java que fa ús d'una base de dades SQL. La aplicació integra i utilitza el model presentat en combinació amb la simulació orientada a esdeveniments discrets per intentar calcular de forma més fidedigna els temps de recorregut dels aeronaus, des de que surt del pàrquing fins la seva arribada a capçalera de pista.

Paraules clau: Taxi Out, simulació, SED, model probabilístic.

### **ENGLISH**

In this paper we have developed a new model to calculate aircrafts taxi-out times. Also we have designed an application in Java code which uses a SQL database. The application integrates the developed model in combination with a discrete event simulation. This makes more accurate the calculation of the time between parking to takeoff queue.

Key words: Taxi-Out, simulation SED, probabilistic model.

## CONTENIDO DE LA MEMORIA DEL PROYECTO

Resumen.....	2
Castellano.....	2
Català .....	2
English.....	2
Listado de gráficas, ilustraciones, diagramas tablas.....	5
Graficas .....	5
Ilustraciones.....	5
Diagramas .....	5
Tablas .....	6
Introducción .....	7
Motivación .....	7
Objetivos .....	8
Objetivos de la Aplicación .....	8
Metodología .....	9
Conceptos.....	10
Aeropuerto de Barcelona, contexto.....	10
Taxi .....	12
Pushback .....	13
Simulación orientada a eventos discretos .....	14
Revisión de literatura científica.....	16
Estructura del modelo .....	20
Parametrización del modelo .....	21
Datos de Campo.....	26
Desarrollo de la aplicación .....	30
Introducción .....	30
El lenguaje Java .....	30
JavaFX.....	31
Características.....	31
Razones para el uso de Java y JavaFx.....	31
Java .....	31
JavaFX .....	31
Uso de la base de datos .....	32
Elección de MySQL y phpMyAdmin .....	32
Diseño de la BBDD .....	32
Clases Principales.....	36

taxiOut.java.....	36
ttoutController.java .....	45
Clases secundarias .....	52
Clases específicas para mostrar información en tablas.....	52
Clases que actúan como nodos de las estructuras .....	53
Clases para la conexión a la base de datos.....	54
Algoritmo probabilista propuesto.....	55
Pseudocódigo .....	55
Características principales de la aplicación.....	56
Flexibilidad.....	56
Programación concurrente .....	56
Uso de la librería Parallel Colt.....	56
Interfaz de usuario .....	56
Sección Superior .....	57
Sección Inferior .....	58
Conclusiones.....	63
Conclusiones del modelo .....	63
Conclusiones del desarrollo de la aplicación .....	63
Resultados Obtenidos .....	63
Bibliografía.....	68

# LISTADO DE GRÁFICAS, ILUSTRACIONES, DIAGRAMAS TABLAS

## GRAFICAS

Gráfica 1. Selección de evento .....	14
Gráfica 2. Función de Densidad propuesta por Khadilkar .....	23
Gráfica 3. Función de densidad de probabilidad para las velocidades.....	23

## ILUSTRACIONES

Ilustración 1. Taxi Chart - configuración oeste .....	13
Ilustración 2. Simulación en Simio del arco R4-R10 .....	24
Ilustración 3. Resultado en Simio del arco R4-R10.....	24
Ilustración 4. Red de flujo de movimientos de aeronaves propuesta.....	25
Ilustración 5. Composición de la Interfaz de usuario .....	57
Ilustración 6. Sección superior de la interfaz de usuario.....	57
Ilustración 7 . Sección inferior de la interfaz de usuario .....	58
Ilustración 8. Pestaña Map Network .....	58
Ilustración 9. Pestaña "Aircraft Results".....	59
Ilustración 10. Pestaña "Detailed Aircrafts Results" .....	60
Ilustración 11. Pestaña "Arc Results" .....	60
Ilustración 12. Pestaña "Zone Results".....	61
Ilustración 13. Pestaña "Database Configuration" .....	61
Ilustración 14. Pestaña "About" .....	62
Ilustración 15. Tiempos de recorrido medio por aeronave y arco .....	66
Ilustración 16. Tiempos medio de aeronave por arco .....	66
Ilustración 17. Tiempos de taxi medios.....	67

## DIAGRAMAS

Diagrama de flujo 1. copiarListaAeronaves .....	37
Diagrama de flujo 2. copiarHasmapArcos .....	38
Diagrama de flujo 3. ordear.....	38
Diagrama de flujo 4. copiarNodoAeronave .....	39
Diagrama de flujo 5. copiarNodoArco .....	39
Diagrama de flujo 6. SimPushback .....	39
Diagrama de flujo 7. SimStops.....	40
Diagrama de flujo 8. simStopDuration .....	40
Diagrama de flujo 9. simTimeArc .....	41
Diagrama de flujo 10. Run .....	42
Diagrama de flujo 11. StdDev.....	42
Diagrama de flujo 12. startAlg.....	44

Diagrama de flujo 13. detailAir.....	45
Diagrama de flujo 14. graficarDetalles .....	46
Diagrama de flujo 15. EntrarDatosAeronave.....	46
Diagrama de flujo 16. EntrarDatosArco.....	47
Diagrama de flujo 17. EntrarDatosArcoRes.....	48
Diagrama de flujo 18. EntrarDatosZona .....	48
Diagrama de flujo 19. procesarResultados.....	49
Diagrama de flujo 20. addDatabase .....	50
Diagrama de flujo 21. initialize.....	51
Diagrama de flujo 22. startAlgorithm.....	52

## TABLAS

Tabla 1. Pistas del Aeropuerto de Barcelona.....	11
Tabla 2. Evolución del tráfico de pasajeros .....	12
Tabla 3. COMPONENTES de un SED .....	15
Tabla 4. Arco R4-R10 .....	24
Tabla 5. Parametrización de los arcos .....	26
Tabla 6. Tabla de tiempo de Pushback .....	27
Tabla 7. Tabla de velocidades de arcos .....	28
Tabla 8. Velocidad media de arcos .....	29
Tabla 9. Duración de tiempo de paradas.....	29
Tabla 10. Tabla Aeronaves.....	33
Tabla 11. Tabla arcos .....	34
Tabla 12. Tablas usadas en la base de datos .....	35
Tabla 13. Resultados obtenidos para las aeronaves .....	65
Tabla 14. Resultados obtenidos para los arcos .....	65

## INTRODUCCIÓN

El retraso de vuelos ha sido y sigue siendo uno de los problemas más críticos, no sólo para los aeropuertos, sino también para la sociedad en su conjunto. Para reducir el retraso de un vuelo, existen dos soluciones por parte de un aeropuerto, aumentar la capacidad de los aeropuertos o utilizar la capacidad del aeropuerto con mayor eficiencia. La primera opción, con todas sus limitaciones sociales reales, no es viable. Requiere de una fuerte inversión económica para ampliar la infraestructura, con el consecuente impacto económico y social que conlleva. La alternativa real será hacer más eficiente la gestión del aeropuerto en cuanto a su capacidad operativa y toma de decisiones, dotar de herramientas que faciliten estas decisiones es por tanto un claro objetivo en la actualidad por parte de los gestores aeronáuticos. Herramientas que proporcionen previsiones fiables ayudaran en la adopción de maniobras tácticas que favorezcan la eficiencia de un aeropuerto para extraer el máximo rendimiento de los recursos limitados de este. Debe entenderse que las limitaciones físicas de estas grandes infraestructuras son varias; capacidad de absorción de pasajeros de las terminales, capacidad de operativa de aviones (aterrizajes y despegues), capacidad de parking para emplazar aviones, capacidad de las pistas de rodaje para absorber las colas de tráfico, limitación de las bandas de radiofrecuencia de comunicaciones, medios de asistencia en tierra, puertas de embarque, fingers y otros muchos recursos.

Los tiempos de Taxi-Out son dependientes principalmente de 6 factores<sup>1</sup>:

1. Pista Activa.
2. Distancia entre la pista y la puerta de embarque.
3. Condiciones meteorológicas (visibilidad).
4. Nivel de congestión del aeropuerto.
5. Configuración de pista.
6. Tipo de Aeronave.

## MOTIVACIÓN

La previsión del crecimiento de tráfico aéreo en los próximos años, estará claramente relacionada con los recursos de un aeropuerto. Las operaciones en tierra provocan y provocarán efectos directos en los retrasos de los vuelos. Efectos que se traducen en coste para las compañías por el gasto de combustible, en la calidad del servicio ofrecido hacia los clientes, por el coste de tiempo perdido y en los efectos negativos medioambientales por las emisiones. Además la complejidad innata en las conexiones del transporte aéreo, significa una propagación de los problemas a gran velocidad y escala.

---

<sup>1</sup>Eurocontrol,[http://www.eurocontrol.int/sites/default/files/content/documents/official-documents/facts-and-figures/coda%20reports/taxi\\_time\\_planning.pdf](http://www.eurocontrol.int/sites/default/files/content/documents/official-documents/facts-and-figures/coda%20reports/taxi_time_planning.pdf)

La finalidad del presente trabajo, es aportar una posible herramienta que mejore la capacidad de gestión de un aeropuerto. Para ello se ha recopilado información sobre la problemática tratada y se elegido como foco de atención la predicción de tiempos de rodaje de las aeronaves salientes de un aeropuerto (Taxi-Out). La predicción del tiempo en que una aeronave puede estar en rodaje es de gran interés para la correcta gestión de un aeropuerto. Una mejora en la precisión de predicciones de tiempos de Taxi-Out es el objetivo principal de este estudio. Para facilitar la solución a este problema se propone una herramienta en forma de aplicación informática que ayude a pronosticar los tiempos de rodaje de una aeronave de la forma más precisa posible y que se pueda integrar como una ayuda en el sistema de soporte de decisiones de un aeropuerto.

## OBJETIVOS

La propuesta planteada requiere de tres objetivos básicos.

1. El primero consistirá en desarrollar un modelo de trabajo, que permita la conversión del sistema complejo real (Aeropuerto de Barcelona) a un modelo de red teórico apto para el desarrollo de una herramienta informática. El desarrollo de este modelo se basara en una de las posibles configuraciones del aeropuerto, la configuración Oeste, al ser esta configuración la más utilizada durante el año. Esta configuración Oeste determina que las aeronaves salientes utilizaran la pista 25L para despegar y la 07L para aterrizar.
2. Como segundo objetivo será necesario desarrollar un algoritmo probabilista que proporcione una mejora en la predicción de tiempos. Para ellos se intentara definir aquellos factores que definan los tiempos de una entidad (avión) durante su transición en la red.
3. El tercer objetivo radicara en el desarrollo del programa informático mediante lenguaje Java. Esta aplicación simulará el paso de las entidades por la red y mostrara la predicción de tiempos de Taxi-Out.

## OBJETIVOS DE LA APLICACIÓN

El principal objetivo de la aplicación desarrollada en Java es poder predecir tiempos de Taxi Out del aeropuerto de Barcelona basado en los parámetros establecidos por Eurocontrol y en aquellos que afecten los rodajes de aeronaves. Debido a la falta de información tiempos de Taxi, se aproximarán las distribuciones de tiempo así como las funciones de parada de aeronaves, por lo que los datos no serán reales sino aproximados.

Adicionalmente se pretende mostrar gráficos de los tiempos de Taxi-Out de las aeronaves así como otros datos relevantes como los tiempos que han estado paradas, el número de paradas durante el proceso de Taxi-Out y los tiempos de espera de cada una de las paradas.

## METODOLOGÍA

Para la predicción de los tiempos de Taxi nos basaremos en la metodología propuesta por Harshad Khadilkar en su tesis *Analysis and Modeling of Airport Surface Operations*.

Para ello se desarrollará un modelo de transito del Aeropuerto de Barcelona. Las tareas para el desarrollo de este serán las siguientes.

- Establecer los nodos de partida, tránsito y terminales del grafo que representará la red del aeropuerto de Barcelona.
- Establecer el tipo de distribución a utilizar para realizar el cálculo de tiempos de desplazamiento entre nodos así como parametrizarlas (K-Erlang).
- Determinar las funciones de probabilidad de número de paradas de aeronaves durante el proceso de TaxiOut que serán dependientes del tránsito. (Geometrica)
- Determinar el tiempo de las paradas efectuadas (Exponencial).

## CONCEPTOS

1. Departure roll: Comprende el inicio del despegue (Take off) iniciado en cabecera de pista hasta que las ruedas pierden el contacto con la pista por la sustentación del avión (wheels off).
2. Departure hold: Período en que las aeronaves que salen se mantienen inmóviles en la pista, a la espera del despegue (Take off).
3. Approach: Etapa en que la aeronave está en corto de pista, dentro de 2,5 millas náuticas del umbral de pista en la senda de descenso para la toma de tierra o aterrizaje.
4. Arrival: Intervalo de la toma de contacto de las ruedas con la superficie hasta el instante que abandona la pista de aterrizaje por una de sus calles de rodaje.
5. Crossings/Taxi: Lapso en que las aeronaves cruzan una pista activa en su rodaje por superficie.
6. Taxi-In: Rodaje por superficie de las aeronaves entrantes al aeropuerto, es decir aquellas que acaban de aterrizar, han abandonado la pista y se dirigen a cualquier punto del aeropuerto, gate o remoto.
7. Taxi-Out: Rodaje por superficie de las aeronaves salientes del aeropuerto, que se dirigen al umbral de la pista activa de despegues.
8. De-icing: Eliminación de hielo, nieve o escarcha de la superficie de la aeronave mediante la aplicación de productos químicos que evitan su formación.
9. Handling: Asistencia y servicios provistos en tierra a una aeronave desde el momento en que aterriza, hasta su posterior marcha. Comprende actividades como la carga y descarga de equipajes, de-icing, Pushback o remolque.
10. Categoría de aeronaves: Heavy Medium Light.

## AEROPUERTO DE BARCELONA, CONTEXTO

El aeropuerto de Barcelona es el segundo aeropuerto más importante de la red AENA, dispone de 674.759 m<sup>2</sup> de superficie, tres terminales, la terminal Corporativa y las terminales T1 y T2 y tres pistas. Su localización estratégica le proporciona gran proyección en el futuro para su completa integración como red de transporte intermodal. Está situado en la proximidad del puerto de Barcelona, de la red ferroviaria y futuro eje mediterráneo además de sus múltiples conexiones por carretera. El aeropuerto tiene una capacidad actual teórica de 55 millones de pasajeros, las próximas modificaciones contempladas en plan director, prevén aumentar su capacidad hasta los 70 millones de pasajeros anuales.

La terminal corporativa está destinada a la aviación privada y es la terminal más pequeña de las tres con 2100m<sup>2</sup>. Es operativa 24h y es gestionada por Executive Airlines y la UTE Gestair-Assistair. Dispone de una capacidad de 26 aeronaves, salas VIP, salas de reuniones para los usuarios y servicios exclusivos. La Terminal T2, es la antigua terminal del aeropuerto que actualmente está siendo objeto de un plan de modernización para atender las actuales necesidades de los pasajeros y adaptarla a la altura de las nuevas instalaciones del aeropuerto. Dispone de 149.359 m<sup>2</sup> de superficie total y 81.000 m<sup>2</sup> de zonas públicas, 152 mostradores de facturación, 48 puertas de embarque con 30 fingers, 18 cintas de recogida de

equipaje, lo que proporciona una capacidad operativa de 25 millones de pasajeros operando un servicio de calidad. La nueva terminal T1 con 525.400 m<sup>2</sup> de superficie total y 155.200 m<sup>2</sup> de zonas públicas absorbe una media de 100.000 pasajeros diarios. Dispone de 15.000 trabajadores, 168 mostradores de facturación, 101 puertas de embarque, 43 fingers ampliables a 50, un sofisticado sistema de procesamiento de equipajes con 14 cintas de recogida de equipaje y una capacidad operativa anual de 30 millones de pasajeros, con una calidad de servicio alta. La combinación de estas tres terminales más la nueva Torre de control, proporciona al aeropuerto de Barcelona una capacidad operativa de 90 operaciones/hora gracias a las tres pistas que dispone el aeropuerto.

Pistas			
Nombre	Longitud(m)	Ancho(m)	Ayudas de aproximación
07L – 25R	3352	60	CAT II / III
07R – 25L	2660	45	CAT II / III
02 – 20	2540	45	CAT I

TABLA 1. PISTAS DEL AEROPUERTO DE BARCELONA

Año	Pasajeros	Incremento anual (%)
1990	9.205.000	-
1991	9.145.000	-0,7
1992	10.196.000	11,5
1993	9.999.000	-2
1994	10.647.285	6,5
1995	11.727.814	10,1
1996	13.434.679	14,6
1997	15.065.724	12,1
1998	16.194.805	7,3
1999	17.421.938	7,6
2000	19.809.567	13,8
2001	20.745.536	4,7
2002	21.348.211	2,9
2003	22.752.667	6,6
2004	24.558.138	7,9
2005	27.152.745	14,6
2006	30.008.152	10,5
2007	32.898.249	9,6
2008	30.272.084	-8
2009	27.421.682	-9,4
2010	29.209.595	6,5

2011	34.399.180	<b>17,8</b>
2012	35.145.176	<b>2,2</b>

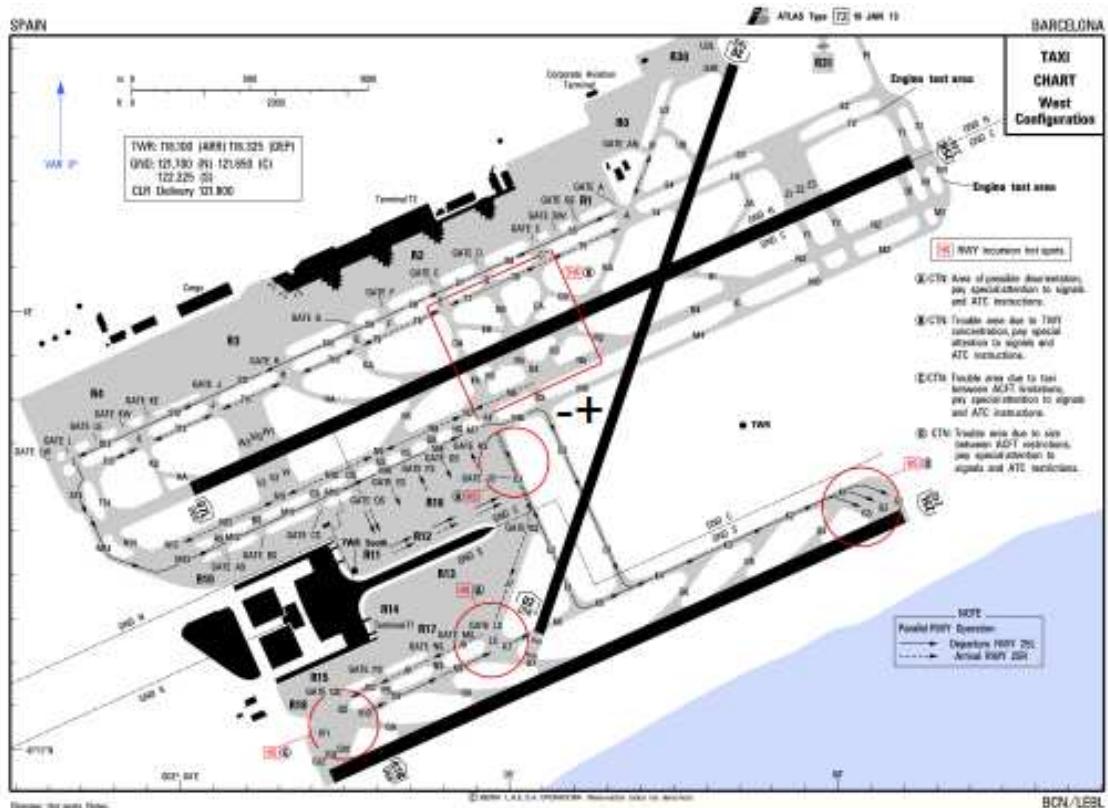
**TABLA 2. EVOLUCIÓN DEL TRÁFICO DE PASAJEROS**

## TAXI

Los aeropuertos se sirven de las calles de rodaje para poder comunicar las pistas de aterrizaje y despegue de otras zonas de operaciones en superficie, como los fingers, rampas o remotos donde se pueden realizar operaciones de carga y descarga, de-icing, abastecimiento de combustible y cualquier otra actividad propia del aeropuerto. De tal modo que la red formada por las conexiones de las zonas de un aeropuerto y las calles de rodaje tendrá repercusión directa en el tiempo de desplazamiento de una aeronave de un punto a otro del aeropuerto. Debido a que las velocidades en superficie de un aeronave están limitadas por motivos de seguridad de tráfico, para evitar colisiones con objetos estáticos u otros vehículos que circulen en superficie, teniendo en cuenta que su capacidad de maniobra al ser vehículos de gran envergadura, tonelaje y gran inercia de movimiento, que requieren de evidente antelación y predicción de movimientos en su conducción, así como por la limitación estructural de la aeronave que vendrá marcada por cada constructor y aún más limitada por las compañías que operen la aeronave. Las distancias que recorran las aeronaves por la superficie, el número de intervenciones por parte del control que obliguen a detener la aeronave antes de llegar a su objetivo por motivos de tráfico y la velocidad en la que se desplaza, concluyen en un tiempo de rodaje o Taxi.

Los rodajes en superficie realizados para abandonar la pista de aterrizaje y dirigirse a un estacionamiento se definen como Taxi-In, aquellos rodajes en superficie de aeronaves que se dirigen a cabecera de pista para despegar y abandonar el aeropuerto se denominan Taxi-Out. La limitación de estos intervalos comprende en el primer caso el instante en que la aeronave abandona la pista activa en la que ha aterrizado hasta que esta estaciona y se le colocan los calzos. En el segundo caso Taxi-Out, sujeto de interés de este trabajo, comprende el lapso de tiempo comprendido entre la retirada de calzos y la entrada a cabecera de pista activa con disposición de despegar (Departure Hold).

Las calles de rodaje, son un recurso limitado del aeropuerto que además de la utilidad mencionada para facilitar la movilidad en la superficie del aeropuerto también son utilizados para absorber las congestiones de tráfico saliente y entrante de un aeropuerto a modo de buffer.



## ILUSTRACIÓN 1. TAXI CHART - CONFIGURACIÓN OESTE

## PUSHBACK

Es el procedimiento por el cual se desplaza el avión mediante ayuda externa de otro vehículo que le permite maniobra en dimensiones reducidas. Normalmente esta maniobra se realiza en fingers donde la aeronave queda estacionada para el embarque de pasajeros de tal forma que solo es posible la salida de la aeronave en dirección opuesta al morro del avión. Algunos aviones pueden generar autopropulsión en este sentido pero es contraproducente para la vida útil de los motores y genera cierto riesgo en la proyección de objetos hacia el edificio o terminal de embarque además de la reducida visibilidad del piloto para realizar dicha maniobra. Las características del aeropuerto y la situación de la aeronave en él, determinarán si es o no necesario la realización del pushback. En el aeropuerto de Barcelona las terminales T1 y T2 requieren este tipo operación, sin embargo existen otros emplazamientos donde se estacionan las aeronaves que no requieren de esta operativa como es el caso de todos los estacionamientos en remoto situados en las rampas R3, R4, R16, R17 y R18.

Los vehículos de pushback son otro recurso limitado de los aeropuertos. Por lo que su distribución en el aeropuerto así como la demanda de su servicio, pueden influir en retrasos de las aeronaves salientes. En la realización de este trabajo se considera que está garantizada la cobertura este servicio en las demandas pico. El estudio de los factores que pueden influir en los retrasos de un aeropuerto es demasiado extenso

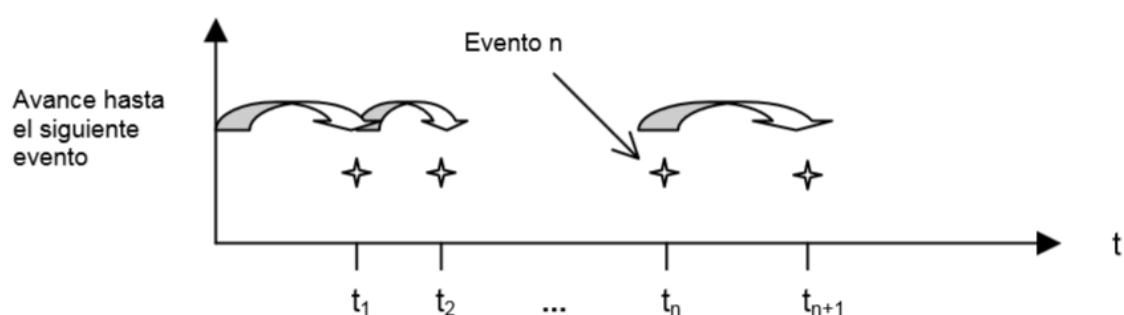
para tratarlo en su globalidad, queda como posible desarrollo futuro la implementación de un módulo encargado de aplicar limitaciones en los servicios de Pushback además de otros servicios de tierra, que generen perturbaciones así como el complejo estudio de sus efectos.

La categoría de las aeronaves no determina el tiempo de pushback, es decir no sufren una penalización en mayor tiempo las aeronaves Heavy que las aeronaves de categoría Medium o Light, los tiempos de remolque son iguales para todo tipo de aeronave. Sí que se requiere de mayor preparación de medios pero esta preparación siempre será previa a la retirada de calzos.

Las condiciones meteorológicas también influyen en la capacidad operativa del aeropuerto, tanto en las operaciones de aterrizaje como en las operaciones de rodaje y despegue. Unas condiciones meteorológicas marginales, pueden dificultar la visibilidad, comprometiendo la seguridad, en estos casos los aterrizajes como es lógico se verán afectados, pero también los rodajes en tierra sufrirán de dificultades y serias interrupciones con visibilidad marginal. En condiciones de visibilidad reducida es necesario proporcionar un servicio de Follow-me mucho más extenso y mantener una separación de aeronaves en tierra más dilatada. El presente estudio se realiza considerando unas condiciones meteorológicas favorables, la implementación futura de un módulo meteorológico en el diseño de programa está contemplada, por lo que no supondrá cambios estructurales notables.

### SIMULACIÓN ORIENTADA A EVENTOS DISCRETOS

La simulación desarrollada está orientada a eventos discretos. La simulación orientada a eventos discretos o SED es una técnica de simulación que permite avanzar el reloj de simulación a intervalos variables de tiempo ya que se asume que entre eventos consecutivos no se produce un cambio en el sistema.



GRÁFICA 1. SELECCIÓN DE EVENTO<sup>2</sup>

Las SED suelen caracterizarse por tener una serie de elementos comunes, en concreto aquellos que son los encargados de avanzar el reloj de simulación hasta el próximo evento. La tabla XX muestra dichos elementos y qué elementos dentro del programa los hacen referencia.

<sup>2</sup> *Simulación de Sistemas Discretos: Aspectos y Tendencias*.

<b>Componentes de un SED</b>	<b>Componente definido en la aplicación</b>
Variables de estado del sistema	Son aquellas definidas en la clase nAeronave y arco.
Variable reloj de simulación	La variable reloj de simulación es la variable nEvent de la primera aeronave de la lista de aeronaves.
Lista de eventos	La lista de eventos es la lista de aeronaves, está siempre está ordenada por siguiente evento a tratar.
Variables estadísticas	Las variables estadísticas están definidas en la clase nAeronave y arco (tiempos medios, paradas medias, desviación estándar, etc.)
Función principal	Método startAlg() de la clase taxiOut.java.
Función de inicialización	Métodos run() de la clase taxiOut.java y initialize de la clase ttourController.java.
Librería de funciones generadoras de valores aleatorios	Librería Parallel Colt
Función de búsqueda del siguiente evento	Puesto que la lista de eventos se ordena por tiempos de siguiente evento, no se requiere función de búsqueda.
Funciones de eventos	No se crean eventos nuevos a partir de eventos dentro de la simulación
Librerías de funciones estadísticas	Librería Parallel Colt
Funciones generadoras de informes	Método procesarResultados de la clase ttourController.java

**TABLA 3. COMPONENTES DE UN SED**

## REVISIÓN DE LITERATURA CIENTÍFICA

Para la realización de este trabajo se inició con un proceso de documentación del que se obtuvo como principales fuentes de inspiración los siguientes artículos brevemente resumidos y adjuntados en su totalidad en los anexos.

- **Título:** Taxi-Out Time Estimationtaxiout,

**Autor:** Husni Idris, John-Paul Clarke, Rani Bhuva and Laura Kang.

**Resumen del artículo:** Se expone que un gran porcentaje de los retrasos son debidos al proceso de salida (Taxi-Out) entre su hora prevista de salida de la puerta y el despegue. Debido a las grandes incertidumbres asociadas, estos retrasos son difíciles de predecir y explicar, lo que dificulta la capacidad de gestionar de forma eficaz el control del tráfico aéreo (ATC). El documento es una recopilación de esfuerzos para la mejorar la precisión de la estimación del tiempo de rodaje de salida (Taxi-Out). El método consiste en identificar los principales factores que han afectado el tiempo de rodaje y construir un modelo de estimación que toma las más importantes en cuenta. Un análisis realizado en el Boston Logan International Airport identificado la configuración de la pista, la línea aérea / terminal y el tamaño de la cola de despegue como los factores causales principales que afectan el tiempo de Taxi-Out. De estos factores el tamaño de la cola de despegue fue el más importante, cuando se midió el tamaño de la cola que una aeronave experimenta como el número de despegues que tuvieron lugar entre el tiempo de pusback y su tiempo de despegue. Por consiguiente, un modelo de cola fue construido para estimar el tiempo de rodaje de salida en el aeropuerto Logan basado en la estimación del tamaño de cola.

- **Título:** Improving Departure Taxi Time Predictions Using ASDE-X Surveillance Data

**Autor:** Husni Idris, John-Paul Clarke, Rani Bhuva and Laura Kang.

**Resumen:** Gran porcentaje de los retrasos de los vuelos se producen en el proceso de salida, sin embargo, predecir el tiempo de rodaje de salida es difícil debido a las incertidumbres asociadas a los factores que influyen en ella, como el tráfico de superficie del aeropuerto, las restricciones posteriores de tráfico, el clima configuración de la pista, y las causas humanas. Los Equipo de Detección/Vigilancia de Datos de Superficie del Aeropuerto (ASDE-X) proporcionan una cobertura de alta resolución del movimiento de aeronaves en superficie que se puede aprovechar para hacer frente a este problema. Este artículo presenta un nuevo enfoque que construye una adaptación del modelo de predicción de Taxi-Out, basado en una base de datos de flujo de tráfico histórico generado con los datos de ASDE-X. El modelo correlaciona el tiempo de Taxi-Out y los retrasos de Taxi-Out a un conjunto de variables explicativas, tales como la posición del avión en cola, la distancia a la pista de aterrizaje, tasas de llegada, las tasas de salida y el tiempo. Se desarrollan dos modelos de predicción. Uno trata el movimiento de aviones de

punto de partida hasta la llegada a pista de manera uniforme, mientras que el otro modelo secciona el tiempo en que las aeronaves tardan en llegar a cola de pista estudiando los tiempos de espera en cola. Los modelos se evaluaron utilizando los datos de John Kennedy de Nueva York F (JFK) Aeropuerto durante el verano de 2010.

- **Título:** Departure Taxi Time predictions using ASDE-X surveillance data

**Autor:** Jeffrey Legge, Benjamin Levy.

**Keywords:** ASDE-X, ATL, Departure, Taxi Time, Prediction.

**Resumen:** Afirma que la predicción precisa de los tiempos de salida de taxis ayudará a las aerolíneas a gestionar de forma proactiva los tiempos de pushback, para asignar de manera óptima los escasos recursos de rampa, y para propagar la información inmediatamente a los aeropuertos de destino de una manera más oportuna. El Control de Tráfico Aéreo (ATC) se beneficiará de la mejor previsión de la demanda en el área del terminal. Un beneficio complementario a tales predicciones es la capacidad de discernir los factores que contribuyen a tiempos más largos de taxi. Para facilitar predicciones exactas, analiza los datos proporcionados por la aplicación Airport Surface Detection Equipment, Model X (ASDE-X). Presenta los resultados basados en los datos de marzo de 2008 a Hartsfield - Jackson Atlanta International Airport (ATL). Se afirma que todos los análisis se podrían aplicar de igual modo a cualquier aeropuerto con sistema ASDE-X.

- **Título:** A stochastic dynamic programming approach to taxi-out prediction using reinforcement learning

**Autor:** Rajesh Ganesan, Lance Sherry.

**Keywords:** Taxi Time, Prediction.

**Resumen:** Investigación impulsada por la necesidad crítica de un avance tecnológico en la predicción de Taxi-Out y en las capacidades de toma de decisiones para el sistema operativo aeroportuario. Con llegada de nuevas tecnologías, el uso de la información basada en sistema inteligente de soporte de decisiones (IIDSS) para controlar las operaciones de servicio del aeropuerto se han convertido en una necesidad para garantizar la eficiencia y el rendimiento. Sin embargo, las compañías aéreas, los aeropuertos, y el controlador de tránsito aéreo (ATC), todavía carecen del uso de sistemas inteligentes que les pueden ayudar en la predicción de los retrasos, los ajustes de horarios y la óptima toma de decisiones frente a la incertidumbre. Según Gobierno de EE.UU, se necesitan nuevas tecnologías para predecir con precisión los retrasos y usar de forma eficiente la existente capacidad para dar soporte a la nueva generación del sistema de transporte aéreo. Por lo tanto, una nueva investigación es necesaria para predecir con precisión los tiempos de rodaje que a su vez puede ayudar a hacer ajustes de horario para reducir

las congestiones y retrasos, y proporcionar un medio para una mejor utilización del personal de tierra de las compañías aéreas. Propone un nuevo aprendizaje por refuerzo (RL) basado en la aproximación estocástica para predecir Taxi-Out que se puso a prueba en los datos de Detroit Metropolitan Wayne County International Airport (DTW). Los resultados iniciales muestran que el error medio de predicción para > 80% de los vuelos son < 3 min.

- **Título:** Analysis and Modeling of Airport Surface Operations

**Autor:** Harshad Khadilkar.

**Keywords:** Taxi Time, Prediction.

**Resumen:** El foco de la investigación en el control de tráfico aéreo ha sido tradicionalmente en la fase de vuelo en el aire. Recientemente se ha reconocido que el aumento de la eficiencia de las operaciones de superficie tienen beneficios significativos en términos de ahorro de combustible y emisiones. Para ganar eficiencia identificando las oportunidades de mejora y la cuantificación de los beneficios consiguientes, es necesario caracterizar las prácticas actuales de operación. Esta tesis describe métodos para el análisis de las operaciones de superficie de aeropuertos y propone indicadores para cuantificar el rendimiento operativo. Estas métricas son evaluadas para Boston Logan International Airport utilizando datos reales de vigilancia de superficie. Se describe un modelo probabilístico para la predicción de tiempos de Taxi-Out en tiempo real que mejora la precisión de modelos anteriores basados en teoría de colas y regresiones. Por último, se describe un modelo de regresión para estimar el consumo de combustible de los aviones que realizan el proceso de Taxi-Out. Juntos, los módulos descritos aquí, constituyen la base para una herramienta de optimización de las operaciones de superficie que actualmente están en producción.

- **Título:** Determination of Minimum Push-Back Time Predictability Needed for Near-Term Departure Scheduling using DEPARTS

**Autor:** Wayne W. Cooper, Jr., Dr. Ellen A. Cherniavsky, James S. DeArmon, J. Glenn Foster, Dr. Michael J. Mills, Dr. Satish C. Mohleji and Frank Z. Zhu

**Resumen:** En este trabajo se describe la funcionalidad del (DEPATRS) “Departure Enhanced Planning And Runway/Taxiway-Assignment System” y cómo esta funcionalidad podría ayudar a reducir los tiempos de Taxi-Out en el aeropuerto internacional Hartsfield Atlanta (ATL). DEPATRS genera recomendaciones sobre la asignación optimizada de pista, la secuencia de salida y corrección de salidas cargadas a torre de control del tráfico aéreo para reducir los tiempos de espera de taxis. DEPATRS inicialmente ha sido desarrollado como un concepto de exploración, prototipo para Atlanta.

Este trabajo analiza el efecto de la mejora de la previsibilidad de los tiempos listos para push-back y de eventos en superficie en tiempo real (es decir, de push-back y la duración de taxi de los despachos reales), potencial beneficio operativo de la planificación previa a la salida. Los resultados provisionales se completan posteriormente con la realización de experimentos de simulación adicionales. Se comentan aspectos operativos de la mejora de la predictibilidad para tiempos de push-back.

- **Título:** The Distribution Function of Airport Taxi-Out Times and Selected Applications

**Autor:** Thuan V.Truong

**Resumen:** Excepto en las condiciones climáticas adversas, la congestión en las grandes plataformas aeroportuarias parece ser predecible. Este trabajo intenta traducir esta previsibilidad en una distribución de los tiempos de espera de taxis, un componente clave de la congestión de los aeropuertos. Cuando los vuelos regulares son elegidos para definir el conjunto de datos, los tiempos de espera de taxis siguen una distribución uniforme, una distribución que puede ser estimada mediante regresión lineal simple que conduce a pronósticos muy precisos. Pero sobre todo, se trata de una función de distribución invertible que puede ayudar a resolver una amplia clase de problemas de optimización estocástica. La Herramienta de Evaluación de Políticas detallada (DPAT) muestra la reducción de la capacidad operativa de un aeropuerto debido a la IMC, causa local de retraso del vuelo y los modelos de la propagación de retrasos en todo el sistema de aeropuertos y sectores. Entre todos los retrasos mencionados anteriormente, los datos históricos indican que los tiempos de Taxi-Out contribuyen con más de 60% del total (Balakrishna 2008). Por lo tanto minimiza el retardo de taxi de salida es muy importante. Y para minimizar el retardo de taxi de salida, es necesario predecir con precisión. La principal contribución de este trabajo es una metodología novedosa y de bajo costo para la predicción de taxi por la construcción de su distribución de probabilidad.

Como breve síntesis de los artículos documentados, se deduce que el Taxi-Out tiene un gran impacto en la operativa de un aeropuerto así como en los retrasos producidos en el mismo aeropuerto y los aeropuertos que estén en la red de conexión de este. La predicción de los tiempos de rodaje de salida y su precisión son de gran importancia para la toma de decisiones que favorezca una óptima operativa, que en consecuencia se traducirá en una mejora de la eficiencia.

Predicción Precisa de Taxi-Out → Mejor DSS → Aeropuerto más Eficiente

## ESTRUCTURA DEL MODELO

La problemática de este trabajo se basa sobre una configuración de salida concreta del Aeropuerto de Barcelona. Esta es la configuración Oeste, que se sirve de la pista 25L para la salida de aviones y la pista 25R para los vuelos entrantes (Configuración de uso más frecuente para este aeropuerto). Para trabajar sobre esta configuración concreta es necesario extraer la topología que forman las posibles rutas de salida del aeropuerto de Barcelona. Esta labor pretende facilitar una red de conexiones compatible con su importación a una base de datos, formada por nodos y arcos que conecten estos nodos. No se pretende realizar un estudio profundo del espacio topológico a modo de ejercicio matemático, sino extraer los conjuntos conexos para diseñar la red de forma adecuada, para poder aplicar una parametrización estadística a cada espacio topológico conexo. El objetivo de los espacios conexos elegidos intenta satisfacer por una parte; un ajuste de límites que ayude a mejorar la precisión de predicción de tiempos del modelo, y en segundo lugar, facilitara la labor en futuros desarrollos sobre el modelo que busquen perfeccionarlo, sin tener que implementar grandes cambios en el modelo.

La configuración Oeste del aeropuerto de Barcelona está estipulada por las autoridades, como se aprecia en las cartas publicadas en la AIP. Sobre estas especificaciones publicadas se desarrolla la red. Compuesta por cuatro elementos básicos, nodos fuente, nodos unión, arcos y nodo final. Los nodos fuente comprenden las posibles localizaciones donde se puede generar una salida de un avión, por lo que corresponden a todos los estacionamientos posibles existentes en el aeropuerto, incluyendo los situados en fingers o en rampas remotas. Los arcos unión no son generadores de aviones pero son necesarios para realizar conexiones entre más de dos arcos. Los arcos son los tramos comprendidos entre nodos que definen la conexión entre ellos, limitando los caminos de la red. Son unidireccionales y tienen unas propiedades derivadas de su longitud y posibles velocidad de rodaje que se traducirán en un tiempo aproximado de desplazamiento determinado por una función K-Erlang además de unos atributos que proporcionarán probabilidad de número de paradas y duración de estas. El nodo final será el extremo meta o fin de todos los aviones.

Los nodos fuente no están basados en un modelo probabilista, sino que disparan entidades según los tiempos cargados en la base de datos preestablecida, de forma que se pueda recuperar datos con frecuencia de una fuente oficial pensando en su posible implementación real. Es decir establecer unas ventanas de tiempo de trabajo para cargar datos desde la fuente oficial deseada, en este caso pensando en Eurocontrol y el CDM del aeropuerto. De modo que se trabajara con una previsión de salidas próxima a la realidad que serán alteradas solo por perturbaciones operativas imprevisibles y de última hora. Los datos que definen la salida de aviones se recuperan de la tabla de la base de datos aeronave, el campo eobt, determina en qué momento se iniciará la retirada de calzos del avión y su inicio de rodaje o Taxi-Out. Los campos parking y rampa determinarán que nodo fuente de la red genera la entidad, es decir donde inicia un avión su recorrido en la simulación.

Los nodos unión no contienen ninguna propiedad o parámetro que modifique el comportamiento de las entidades, están diseñados para delimitar los espacios conexos de la red y la unión de múltiples nodos y arcos en un punto convergente de la red.

Los arcos son elementos de la red que además de definir la conexión entre nodos, proporcionan la parametrización que alterara el comportamiento de las entidades que se desplacen por ellos. Cada arco aportara unos parámetros estadísticos para ejecutar en la simulación un tiempo de desplazamiento que dependerá de tres funciones estadísticas. En cada iteración y por cada aeronave que se desplace por un arco, se lanzaran los valores para extraer este tiempo de desplazamiento. Para determinar el tiempo de desplazamiento nos basaremos en el resultado de las siguientes funciones. Estas funciones están basadas en la metodología propuesta por Harshad Khadilkar en su tesis Analysis and Modeling of Airport Surface Operations.

## PARAMETRIZACIÓN DEL MODELO

### K-ERLANG

Esta distribución determinara el tiempo de transito de las entidades por cada arco en que se desplace dicha entidad. Por lo tanto deberá ser siempre positiva. Esta distribución de probabilidad será la utilizada basándose en las conclusiones del estudio empírico de datos de Harshad Khadilkar. Consiste en una distribución de probabilidad continua con dos parámetros  $k$  y  $\lambda$ , equivalente a una distribución gamma de parámetro  $k$  con valor entero positivo. La función de densidad para valores  $x > 0$  es:

$$f(x) = \lambda e^{\lambda x} \frac{(\lambda x)^{(k-1)}}{(k-1)!} \text{ para } x, \lambda \geq 0$$

### ECUACIÓN 1. FUNCIÓN DE DENSIDAD DE K-ERLANG

Con función generadora de momentos:

$$(1 - t/\lambda)^{-k}$$

### ECUACIÓN 2. FUNCIÓN GENERADORA DE MOMENTOS DE K-ERLANG

Esperanza:

$$E(X) = k/\lambda$$

### ECUACIÓN 3. ESPERANZA DE K-ERLANG

Variancia:

$$V(X) = k/\lambda^2$$

### ECUACIÓN 4. VARIACIA DE K-ERLANG

### *GEOMÉTRICA*

La función geométrica carece de memoria, es decir que si se repite el experimento hasta llegar al primer éxito, este se considera que todavía no ha ocurrido por lo que el número de fallos que se haya observado no viene condicionado por el número de experimentos observados. La distribución geométrica es la utilizada en este trabajo para pronosticar el número de interrupciones de una entidad durante el recorrido por la red, es decir el número de paradas que ara la aeronave en su rodaje Taxi-Out.

La probabilidad de interrupción se incrementa uniformemente según las aeronaves en zonas. En la base de datos se especifica los parámetros que recibe la función de distribución geométrica para dos aeronaves en zona. Este parámetro será incrementado uniformemente +0.1 (parámetro modifiable) por cada aeronave en zona. La probabilidad de parada para una aeronave en zona es 0. El número máximo de interrupciones en un arco definido es 3 independientemente del número de aeronaves en zona.

### *EXPONENCIAL*

La distribución exponencial se puede interpretar como un caso particular de la función Weibull con parámetro de forma igual a 1. La función exponencial completará los tiempos estimado de Taxi-Out al aportar la duración de las paradas o interrupciones de la aeronave durante el rodaje hacia cabecera de pista. Así pues, si la función de distribución geométrica explicada brevemente en el punto anterior, proporciona la cantidad de paradas de una aeronave, la función exponencial determinara la duración de cada una de estas paradas.

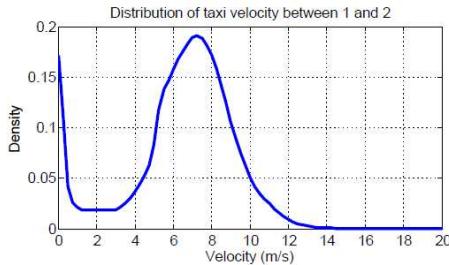
### *TIEMPO DE TAXI-OUT*

La combinación de estas tres distribuciones aportara el tiempo de Taxi-Out de una aeronave determina en un arco concreto. El sumatorio de los tiempos de arcos recorridos por la entidad será el tiempo total de Taxi-Out.

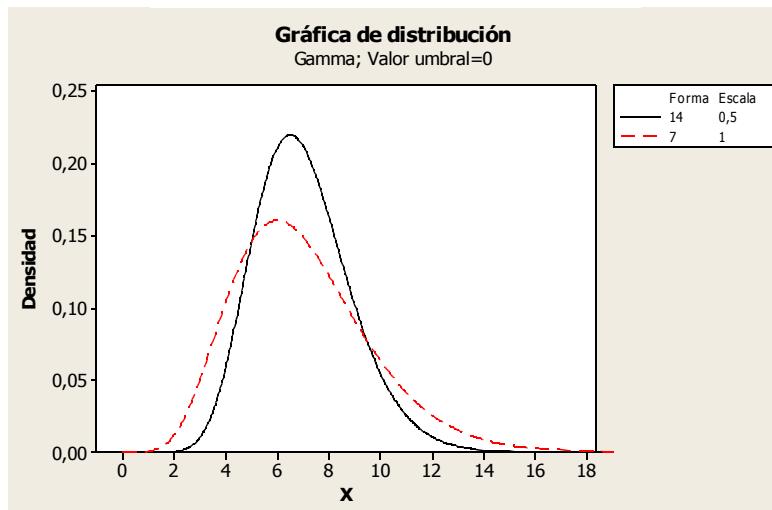
$$TiempoArco\ i = Erlang(\lambda_i, \eta_i) + \sum_{m=Geometrica(a_i)}^{m=0} Exponencial(\mu_i)$$

### **ECUACIÓN 5. TIEMPO DE TRAYECTO DEL ARCO I**

La dificultad para obtener datos de campo reales ha sido muy frecuente durante la realización del trabajo, para solventar esta escasez de fuentes verídicas, se recurre a los datos publicados por Harshad Khadilkar en su tesis Analysis and Modeling of Airport Surface Operations además de la información proporcionada por Francisco Javier Tordaz, instructor de mantenimiento de Iberia y capacitado para la realizar rodajes en de aeronaves. Para extraer una función relativa a la velocidad de desplazamiento de un aeronave durante un rodaje se estipulo una velocidad aproximada de 7m/s en un rodaje normal, con un límite superior de 10,29m/s o 20 nudos y considerándose un límite inferior para rodajes lentos de 5-10 nudos que equivalen a 2,57m/s y 5,14m/s. Obteniendo como resultado mediante MiniTab una Gamma con parámetros de forma y escala [14,0.5] cuya sobre-posición se aproxima al resultado de Harshad Khadilkar.



**GRÁFICA 2. FUNCIÓN DE DENSIDAD PROPUESTA POR KHADILKAR**



**GRÁFICA 3. FUNCIÓN DE DENSIDAD DE PROBABILIDAD PARA LAS VELOCIDADES**

Mediante la ayuda del software de simulación SIMIO y las herramientas de las funciones estadísticas del programa MiniTab, se ha buscado una coherencia entre los datos recopilados y la simulación propuesta en el presente trabajo. Analizando los datos de campo y los conocimientos teóricos, se han ajustado los parámetros que modelaran el comportamiento de las entidades en la simulación. Conociendo de forma aproximada la pauta aproximada de las aeronaves en determinados arcos de la red se ha verificado el correcto comportamiento de las entidades ejecutadas en la simulación. Sin posibilidad de acceder a más datos de campo y validar el modelo actual, este ha sido el método más asequible para orientarnos en el adecuado desarrollo de la aplicación. En los anexos del presente trabajo se adjuntan unas simples simulaciones realizadas en SIMIO que ejecutan las funciones definidas para parametrizar el modelo. Estos cinco ejemplos pueden compararse con la tabla 3 que muestra los datos obtenidos en campo con un patrón más fiable al ser los registros más repetidos. Esta verificación de comportamiento no debe entenderse como una validación global del modelo, sino como una orientación a nivel elemental de los elementos constructores del modelo. Para poder garantizar con los escasos medios disponibles, la aproximación más precisa al comportamiento real.

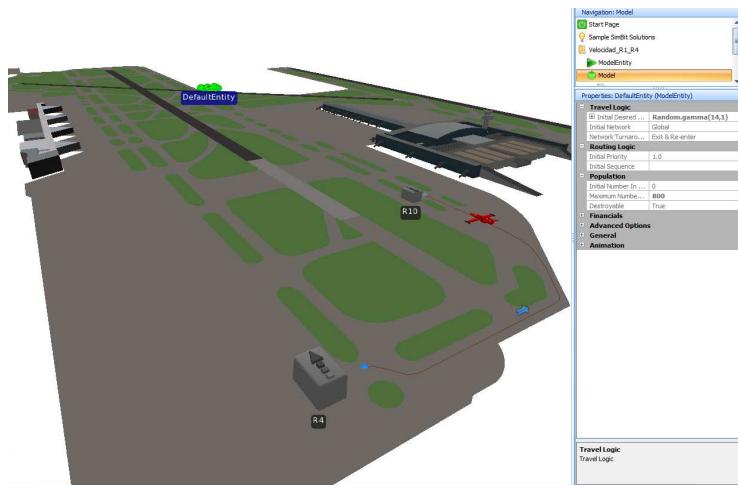


ILUSTRACIÓN 2. SIMULACIÓN EN SIMIO DEL ARCO R4-R10

Statistic	Average...
Average	0,0402
Maximum	4,0000
Average (Seconds)	70,8716
Maximum (Seconds)	275,0129
Minimum (Seconds)	46,0000
Total	17.159,0000
Total	17.159,0000

ILUSTRACIÓN 3. RESULTADO EN SIMIO DEL ARCO R4-R10

Arco	Numero de muestras	Media de Tiempos	Distancia media (m)	Velocidad (m/s)
R4-R10	5	70,2	920	13,11

TABLA 4. ARCO R4-R10

Las longitudes que determinan los arcos se han extrapolado de la posición de los planos de estacionamiento publicados en la AIP y de la herramienta Google Earth. De los nodos fuente se ha contemplado como límite inferior de tiempo simulaciones generadas en SIMIO del estacionamiento más próximo al nodo contiguo y una velocidad de 20 nudos, como límite superior de tiempo se ha contemplado el estacionamiento más alejado del nodo contiguo y una velocidad de 5 nudos. Como tiempos normales una posición de estacionamiento intermedio y velocidad determinada por la función. Los datos obtenidos de SIMIO son coherentes con los datos esperados, tanto por parte del asesoramiento del Javier Sanz Tordable como por parte de los datos obtenidos in situ en el registro de datos de campo. En los anexos se adjuntan simulaciones en SIMIO de los arcos.

#### PARAMETRIZACIÓN FINAL DE LOS ARCOS

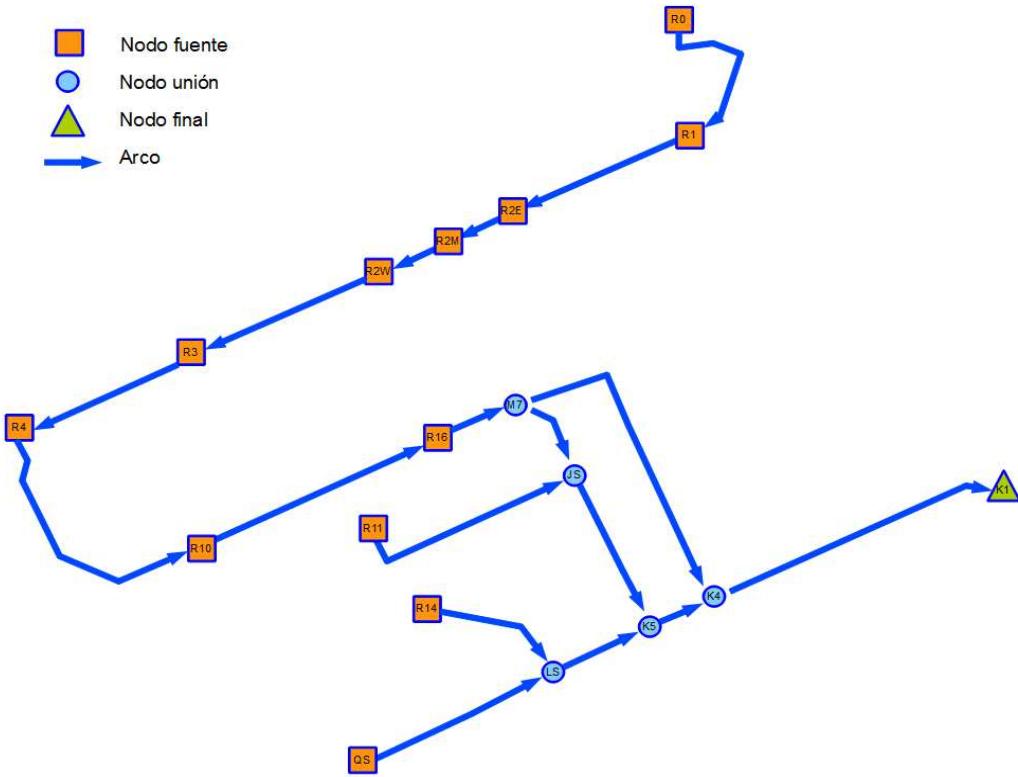


ILUSTRACIÓN 4. RED DE FLUJO DE MOVIMIENTOS DE AERONAVES PROPUESTA

En el grafo anterior representa la red propuesta para el desarrollo de la simulación. El nodo R0 (rampa 0) representa el origen de las aeronaves de aviación general. La conexión de este nodo hasta el nodo M7 es en serie y abarca tres zonas de interés; la primera consiste en el grupo de nodos R2E, R2M, R2W, simbolizan las salidas de la terminal T2, comprendiendo los estacionamientos A, B, C, D y E. Estos nodos de la Terminal T2 tendrán una alta frecuencia y salida de entidades. La segunda zona de interés consiste en la plataforma de aviación corporativa desde donde saldrán principalmente Jets privados de categoría L o M con la peculiaridad de no requerir pushback por su tipo de estacionamiento que engloba los estacionamientos V, W, Y, Z . Los nodos R3 y R4 dispone de los estacionamientos remotos comprendidos entre la plaza 60 a 121 utilizados fundamentalmente por compañías de cargo, no será habitual el servicio de pushback. Los nodos R10, R11, R14 y QS se atribuyen a las salidas de la terminal T1, donde operan compañías comerciales asignando una alta frecuencia de entradas y salidas de aeronaves de categoría M y H. Estos aviones requieren de pushback al estar encarados al terminal.

<i>idArco</i>	<i>nArco</i>	<i>siguiente</i>	<i>nZona</i>	<i>HE a</i>	<i>HE b</i>	<i>ME a</i>	<i>ME b</i>	<i>LEa</i>	<i>LEb</i>	<i>AG</i>	<i>HExp</i>	<i>MEx</i>	<i>LExp</i>
1	R0-R1	R1-R2E	1	21	4	21	4	21	4	0.81	0.018	0.018	0.018
2	R1-R2E	R2E-R2M	2	19	5.5	19	5.5	19	5.5	0.86	0.016	0.016	0.016
3	R2E-R2M	R2M-R2W	3	21	3.3	21	3.3	21	3.3	0.81	0.014	0.014	0.014
4	R2M-R2W	R2W-R3	4	21	2.7	21	2.7	21	2.7	0.81	0.012	0.012	0.012

5	R2W-R3	R3-R4	5	22	3.5	22	3.5	22	3.5	0.81	0.01 6	0.016	0.01 6
6	R3-R4	R4-R10	6	21	5	21	5	21	5	0.86	0.02	0.02	0.02
7	R4-R10	R10-R16	7	21	9	21	9	21	9	0.77	0.02 2	0.022	0.02 2
8	R10-R16	R16-M7	8	21	5.1	21	5.1	21	5.1	0.86	0.02	0.02	0.02
9	R16-M7	M7-K4	9	10	5.2	10	5.2	10	5.2	0.77	0.01 5	0.015	0.01 5
10	M7-JS	JS-K5	10	64	0.6	64	0.6	64	0.6	0.77	0.02 5	0.025	0.02 5
11	R11-JS	JS-K5	11	30	3	38	2	70	1	0.77	0.01 2	0.012	0.01 2
12	JS-K5	K5-K4	12	24	3.8	24	3.8	24	3.8	0.81	0.02 2	0.022	0.02 2
13	K5-K4	K4-K1	13	100	0.5	100	0.5	10 0	0.5	0.81	0.01 8	0.018	0.01 8
14	K4-K1	FINAL	14	20	4	20	4	20	4	0	0.01 2	0.012	0.01 2
15	R14-LS	LS-K5	15	10	9	10	9	10	9	0.81	0.01 5	0.015	0.01 5
16	LS-K5	K5-K4	16	30	1	30	1	30	1	0.81	0.01 6	0.016	0.01 6
17	QS-LS	LS-K5	17	25	5.4	25	5.4	25	5.4	0.81	0.01 2	0.012	0.01 2
18	M7-K4	K4-K1	18	330	0.5	330	0.5	33 0	0.5	0.81	0.01 8	0.018	0.01 8

TABLA 5. PARAMETRIZACIÓN DE LOS ARCOS

## DATOS DE CAMPO

Por la ausencia de datos extensos será imposible validar el modelo de forma precisa, pero sí que se ha comprobado su correcto funcionamiento para valores aproximados, gracias a una recolección de datos de campo que se desarrolló en puntos estratégicos del aeropuerto, para controlar los tiempos de ejecución de distintos procesos de interés del estudio. Las principales acciones a registrar son los tiempos de Pushbacks, la duración de las paradas y cronometrar algunos de los tramos que componen el modelo de nuestra red. La recolección de datos se generó durante dos días recopilando un total de 47 tiempos aplicables al modelo de 22 aeronaves distintas. Para poder recoger los datos se requirió de un emisora VHF, diferentes planos del aeropuerto y acceder a puntos estratégicos de visualización comprendidos entre terminales y la Torre de Spotters del Prat. Los parámetros de entrada de esta base de datos de campo son los siguientes:

- Zona: Definir la zona de partida de la aeronave.
- Hora Local: Registrar la hora local en la que se registran los datos
- Modelo: Modelo de avión, compañía y código de vuelo.
- Parking: Registrar el emplazamiento exacto para ganar precisión en la longitud de arco.
- Time\_Pushback: Registrar los tiempos de remolque.
- Time\_Stop: Cronometrar las interrupciones en el rodaje de la aeronave.
- Arco: Registrar la transición en la que se desplaza la aeronave.
- Tieme\_Trans: Cronometrar el tiempo por arco recorrido.
- CAT: Definir la categoría heavy, medium o light según modelo de avión.

A continuación se muestran unas tablas con los datos más relevantes obtenidos, la totalidad de los datos en crudo recogidos en campo se muestra en los anexos (Datos de Campo). La primera Tabla 4 muestra los tiempos en segundos de Pushback recopilados, según modelo de avión y estacionamiento inicial. La segunda tabla 2 refleja los registros de aeronaves en desplazamiento concretos y conocidos de la red, mostrando el nodo inicio y nodo final del tramo registrado observado, definido en el campo Arco, TiempoTrans como su tiempo de transito por el arco, la longitud del arco y un promedio de la velocidad. La tabla 3 refleja los tiempos y longitudes medias de los arcos con mayor frecuencia de muestras y más fiables para el estudio. La tabla 4 del grupo de datos de campo muestra los tiempos obtenidos en las interrupciones del rodaje de las aeronaves. Este conjunto de datos obtenidos no pretende ser un estudio estadístico exhaustivo sino una pequeña ayuda orientadora para comprobar un funcionamiento coherente de la aplicación desarrollada. Sería necesario muchos más medios materiales e invertir gran cantidad de horas para extraer unos datos suficientemente aceptables para validar la aplicación.

Hora Local	Modelo	Parking	Tiempo_Pusback
13:13:00	A320_VY2472	238	222
13:32:00	A320_VY1814	232	170
15:25:00	A320_VY2491	234	173
19:00:00	A320_VY3736	228	183
09:50:00	737_RYR4585	B5	188
10:10:00	A320_VY3528	232	205
11:30:00	A320_VY6500	234	237
11:55:00	A320_VY1820	228	192
12:10:00	737_RYR6386	B1	146

TABLA 6. TABLA DE TIEMPO DE PUSHBACK

Arco	TiempoTrans	CAT	Longitud	m/s
236->JS	48	M	410	8,54
R1->R3	120	M	1930	16,08
R3->R4	78	M	630	8,08
R4->R10	71	M	920	12,96
R10-R16	115	M	700	6,09
R16-M7	48	M	650	13,54
M7-JS	53	M	300	5,66
238->JS	43	M	350	8,14
232->JS	93	M	500	5,38
R0->R4	276	M	2620	9,49

R1->R4	290	M	2270	7,83
R4->R10	75	M	920	12,27
R10->M7	165	M	1330	8,06
R0->R4	185	L	2620	14,16
R0->R4	220	L	2620	11,91
R0->R4	265	M	2620	9,89
R1->R4	273	M	2270	8,32
R4->R10	72	M	920	12,78
R10->M7	172	M	1330	7,73
R3->R4	78	M	630	8,08
234->JS	76	M	460	6,05
R0->R4	201	M	2620	13,03
228->JS	68	M	600	8,82
R2E->R4	252	M	1910	7,58
R1->R4	229	M	2270	9,91
R4->R10	68	M	920	13,53
R10->M7	175	M	1330	7,60
R0->R4	197	L	2620	13,30
234->JS	98	M	460	4,69
228->JS	105	M	600	5,71
R1->R4	222	M	2270	10,23
R4->R10	65	M	920	14,15
R10->M7	169	M	1330	7,87
R2E->R4	247	M	1910	7,73
			<b>MEAN</b>	<b>9,56</b>

TABLA 7. TABLA DE VELOCIDADES DE ARCOS

	Nº	Med_Tiempos	Med_Dist	Velocidad
<b>R11-JS</b>	7	75,9	482,9	6,37
<b>R0-R4</b>	6	224	2620	11,70
<b>R1-R4</b>	4	253,5	2270	8,95
<b>R4-R10</b>	5	70,2	920	13,11
<b>R10-M7</b>	4	170,25	1330	7,81
			<b>MEAN</b>	<b>9,59</b>

**TABLA 8. VELOCIDAD MEDIA DE ARCOS**

Zona	Hora Local	Modelo	Tiempo de parada(s)
3	13:13	A320_VY2472	108
3	13:32	A320_VY1814	85
3	11:30	A320_VY6500	95

**TABLA 9. DURACIÓN DE TIEMPO DE PARADAS**

# DESARROLLO DE LA APLICACIÓN

## INTRODUCCIÓN

Uno de los pilares de este proyecto es realizar una aplicación partiendo desde cero que permita realizar una simulación de eventos discretos tomando como base el modelo probabilístico desarrollado. El lenguaje de programación elegido ha sido Java y finalmente se ha optado por el uso de una base de datos tipo SQL donde se recogerá los datos necesarios para poder hacerlo funcionar. El programa se ha desarrollado pensando que cualquier usuario que haga uso de él pueda volcar la información referente al estado actual del aeropuerto. El programa simulará todas las aeronaves que se encuentren en la base de datos a la hora de ejecutarlo. Éste integra una consola que muestra eventos relevantes durante el uso del software así como errores posibles a la hora de cargar los datos o los tiempos de cálculo obtenidos. Finalmente la aplicación mostrará los gráficos y tablas referentes a la siguiente información.

- Aeronave:
  - Tiempos de taxi: Tiempos de taxi medio, desviación estándar, tiempo máximo y tiempo mínimo obtenidos en la simulación.
  - Paradas: Paradas medias, desviación estándar, paradas máximas y paradas mínimas obtenidas en la simulación.
- Arcos:
  - Tiempos: Tiempo medio, máximo y mínimo por aeronave en recorrer el arco obtenidas en la simulación.
  - Paradas: Paradas medias, máximas y mínimas por aeronave en el arco obtenidas en la simulación.
- Zonas:
  - Tiempos: Tiempo medio, máximo y mínimo por aeronave en recorrer la zona obtenidos en la simulación.
  - Paradas: Paradas medias, máximas y mínimas por aeronave en la zona obtenidas en la simulación.

## EL LENGUAJE JAVA

Para el desarrollo de la aplicación se ha optado por el lenguaje de programación Java. Java es una plataforma virtual de software desarrollada por Sun Microsystems, en la cual los programas creados en ella puedan ejecutarse sin cambios en diferentes tipos de plataformas. Es por eso que el mismo código compilado es totalmente funcional en cualquier plataforma.

La plataforma Java consta de las siguientes partes:

1. El lenguaje de programación, mismo.
2. La máquina virtual de Java o JRE, que permite la portabilidad en ejecución.
3. El API Java, una biblioteca estándar para el lenguaje.

Java fue diseñado para correr en computadoras incrustadas. Sin embargo, en 1995, dada la atención que estaba produciendo la Web, Sun Microsystems la distribuyó para sistemas operativos tales como

Microsoft Windows. El lenguaje se inspira en la sintaxis de C++, pero su funcionamiento es más similar al de Smalltalk que a éste. Incorpora sincronización y manejo de tareas en el lenguaje mismo (similar a Ada) e incorpora interfaces como un mecanismo alternativo a la herencia múltiple de C++. Los programas en Java generalmente son compilados a un lenguaje intermedio llamado bytecode, que luego son interpretados por una máquina virtual (JVM). Esta última sirve como una plataforma de abstracción entre la máquina y el lenguaje. También existen compiladores nativos de Java, tanto software libre como no libre. La versión libre se denomina OpenJDK que está catalogada dentro la de licencia GPL de GNU con algunas excepciones. El software desarrollado compila perfectamente con OpenJDK en versión 7.

## JAVAFX

JavaFX es su versión 2.0 está implementada como una librería nativa de Java por lo tanto cualquier aplicación que hace uso de JavaFx está escrita nativamente en Java. JavaFx está diseñado para proveer de una plataforma para la creación de interfaces gráficas que sea ligera y aproveche las ventajas de la aceleración por hardware.

### CARACTERÍSTICAS

1. Aceleración por Hardware
2. FXML - sintaxis XML para la definición de las interfaces de usuario.
3. Posibilidad de renderizar componentes Web (WebView) dentro de las aplicaciones Java.
4. Soporta Swing y SWT dentro de las aplicaciones.
5. Soporte para interfaces táctiles.
6. Motor para reproducir contenido multimedia.

## RAZONES PARA EL USO DE JAVA Y JAVAFX

### JAVA

1. Extensa documentación para el desarrollo de aplicaciones.
2. Rapidez de codificación mediante el IDE NetBeans.
3. Fácil implementación de la librería Parallel Colt para el uso de funciones de distribución deseadas para llevar a cabo la simulación necesaria dentro del algoritmo principal.
4. Fácil implementación del algoritmo multihilo desarrollado.

### JAVAFX

Para el desarrollo de la interfaz gráfica de usuario se optó por JavaFX en lugar de Swing. El motivo principal que lo ha motivado es que JavaFx tiene una curva de aprendizaje superior a Swing y mediante su API permite programar interfaces más complejas y en menor tiempo. JavaFx viene con un gran catálogo de componentes implementados como gráficos, lo que evita hacer uso de librerías externas como JPlot para la visualización de los resultados obtenidos.

Finalmente, en el futuro JavaFx sustituirá a Swing como nueva librería de interfaces gráficas para java puesto que es capaz de aprovechar las nuevas tecnologías de aceleración gráfica.

## USO DE LA BASE DE DATOS

### ELECCIÓN DE MySQL Y PHPMYADMIN

MySQL es un sistema de gestión de bases de datos relacional, con un conjunto completo de herramientas que permite el almacenamiento, modificación y extracción de la información de una base de datos. Es multihilo por lo que permite la ejecución de varias funciones simultáneamente además de multiusuario. Se ofrece bajo licencia pública general (GNU GPL).

Al ser MySQL un sistema de administración relacional de base de datos, archiva los datos en distintas tablas de forma que favorece la velocidad y flexibilidad de su proceso computacional. Se basa en el modelo relacional introducido por Edgar Fran Codd.

PhpMyadmin ofrece un micro entorno de administración con el uso de un navegador. Una de las posibilidades barajadas para la realización del proyecto fue la utilización de un sistema noSQL, pero requería de cierto coste en aprendizaje, por lo que se optó por un sistema SQL, ya familiar por los conocimientos impartidos durante el grado de Gestión Aeronáutica en la asignatura de Informática Avanzada.

Otras características apreciables para la elección de MySQL consisten en la seguridad proporcionada por su sistema de privilegios y contraseñas que es muy flexible y segura, permitiendo la verificación basada en host y el tráfico de contraseñas cifrado.

La escalabilidad y límites altos que permiten trabajar con 50 millones de registros. Permite hasta 64 índices por tabla de hasta 16 columnas, con una capacidad de 1000 bytes.

MySQL permite a los clientes una conexión con el servidor mediante sockets TCP/IP en cualquier plataforma. Es de fácil integración y compatibilidad con la mayoría de sistemas existentes, al desarrollar el trabajo en diferentes sistemas operativos y desde distintos lugares es significante esta cualidad.

### DISEÑO DE LA BBDD

La base de datos que utiliza la aplicación desarrollada está compuesta por cuatro tablas principales. De esta forma se garantiza una buena flexibilidad para modificar los datos de entrada de esta aplicación, y se su edición sin generar acoplamientos o interdependencias no deseadas que dificulten los ajustes y edición de parámetros.

#### TABLA AERONAVE

La tabla aeronaves está formada por 9 campos (columnas) que proporcionan la información relativa a las características de la aeronave como el identificador, categoría y tiempo previsto de salida y su posición inicial. La tabla creada dispone de 41 registros o filas que equivalen a un total de 41 aeronaves que realizan

sus operaciones en Barcelona. A continuación se observa cada uno de los campos y el tipo de variable con el que se ha declarado.

<i>idAeronave</i>	<i>int</i>
<i>matricula</i>	<i>varchar</i>
<i>nVuelo</i>	<i>varchar</i>
<i>eobt</i>	<i>datetime</i>
<i>ctot</i>	<i>datetime</i>
<i>parking</i>	<i>varchar</i>
<i>rampa</i>	<i>varchar</i>
<i>tipo</i>	<i>varchar</i>
<i>pushback</i>	<i>int</i>

TABLA 10. TABLA AERONAVES

El *idAeronave* es un identificador interno propio del sistema y no tiene ninguna relación con los identificadores externos también recogidos en la base de datos. Este solo nos proporciona información sobre el orden de ingreso de una aeronave en la base de datos y el número de aeronaves. Los identificadores externos de cada aeronave son la *matricula* que corresponde al identificador alfanumérico unívoco de cada aeronave según la conformidad de la OACI y *nVuelo* que nos proporciona información de la compañía operadora y ruta. Los dos campos siguientes están declarados como *datetime* al proporcionar información relativa a tiempos. El primero de estos dos campos es el *EOBT* que recoge la hora prevista de calzos fuera o Estimated Off Block Time. El segundo campo, *CTOT* (Calculated Takeoff Time) es el tiempo calculado en que la aeronave realizará la maniobra de despegue. La columna *parking* proporciona la posición inicial de la aeronave en el momento de la retirada de calzos. La columna *rampa* designa la posición de la aeronave dentro de la red propuesta, de forma que indique el arco comprendido entre dos nodos donde está posicionada la aeronave. El campo *tipo* indica la categoría de la aeronave, este campo se contempla para modificar las propiedades del elemento en los tiempos de desplazamiento entre nodos o la aplicación de otras propiedades inherentes a la categoría de aeronave. De este modo se identificando la categoría de aeronave y se le podrá aplicar por ejemplo, restricciones de velocidad. El último campo denominado *pushback* indica si la aeronave necesita realizar la acción de *pushback* y simular la penalización temporal según la siguiente condición.

*if pusback = 1 → realizar simulación de pushback*

*if pusback=0 → no realizar simulación pushback*

#### TABLA ARCOS

La tabla *arcos* está diseñada para definir las conexiones de la red, es decir los nodos que la componen y los arcos que comunican dichos nodos. Esta tabla la componen 14 campos, de los cuales los 4 primeros proporcionan información de identificación y las 10 siguientes columnas aportan parámetros estadísticos que definen el comportamiento del modelo.

<i>idArco</i>	<i>int</i>
<i>nArco</i>	<i>varchar</i>
<i>siguiente</i>	<i>varchar</i>
<i>nZona</i>	<i>int</i>
<i>HEa</i>	<i>double</i>
<i>HEb</i>	<i>double</i>
<i>MEa</i>	<i>double</i>
<i>MEb</i>	<i>double</i>
<i>LEa</i>	<i>double</i>
<i>LEb</i>	<i>double</i>
<i>AG</i>	<i>double</i>
<i>HExp</i>	<i>double</i>
<i>MExp</i>	<i>double</i>
<i>LExp</i>	<i>double</i>

TABLA 11. TABLA ARCOS

En primer lugar encontramos *idArco* que identifica cada arco o segmento entre nodos de forma única. Como segundo elemento tenemos *nArco*, denoma el nombre del arco dentro de la red seguido del campo *siguiente* que indica el arco que sigue al arco predecesor *nArco*. De este modo cuando una entidad (aeronave) se encuentra en un arco, puede recuperar la información del arco que posteriormente recorrerá. El *nZona* proporciona información referente a la pertenencia de un arco a una zona, posteriormente se explicara la definición de zona.

Los diez campos restantes son útiles para la modelización estadística del prototipo y siguen la siguiente nomenclatura. La primera letra indicara el tipo o categoría de la aeronave (H=Heavy,M=Medium,L=Light) a excepción de la A, que define el campo AG y hace referencia al parámetro de la función geométrica que indica el número de posibles paradas que pueda realizar la aeronave en cada arco. Las mayúsculas indicativas de categoría (H, M, L) seguidas de “\_Exp” indican que son parámetros de funciones exponenciales para determinar la duración de cada parada que realiza el avión, según su categoría de aeronave (H, M, L). Los campos \_Ea y \_Eb según categoría (H, M, L) indican los dos parámetros que formaran la función K-Erlang (Gamma) para determinar el tiempo de desplazamiento de la aeronave por el arco.

#### TABLA PUSHBACK

La tabla pushback está diseñada para modelizar los tiempos de pushback según categoría de la aeronave relacionando el campo tipo de la tabla aeronave con el campo tipo de la tabla pushback. Está compuesta de tres campos o columnas, el primero identifica el tipo de aeronave (tipo). Los dos campos restantes:

PEa (Pushback Erlang parameter'a') y Peb(Pushback Erlang parameter'b'), son los parámetros de la función K-Erlang o Gamma que determinara los tiempos de pushback. Esta tabla se diseñó con la intención de separar las categorías entendiendo que los tiempos de remolque entre aeronaves de diferentes categorías serían diferentes, pero como se ha comentado anteriormente en la definición de pushback, la categoría de la aeronave no influye en el tiempo de pushback, si en el de preparación de recursos, pero no en el de remolque de la aeronave desde la retirada de calzos al posicionamiento de la aeronave a una situación favorable para su posterior rodaje en pista de forma autónoma. La conservación de los tres registros de esta tabla ha sido consecuencia de su utilidad en las pruebas internas de validación para observar el correcto funcionamiento del software y su posible utilidad en aplicaciones futuras, así como la posibilidad de rectificar y separar tiempos de pushback si estudios posteriores demuestran una independencia de tiempos.

#### TABLA ZONAS

La tabla zonas está compuesta por tres columnas, id como identificador único interno del sistema, zona define la cantidad y precisión de la división de la red, y el campo terminal que define desde qué terminal parte el avión. Las tres versiones difieren en la columna zona ya que es este campo el que define la precisión con la que se dividen las zonas para el posterior cálculo de outputs.

#### LAS TRES VERSIONES DE LA BASE DE DATOS

La base de datos está compuesta por cuatro tablas principales. Pero existen de ella tres versiones diferentes según la definición de zona. Esta definición proporcionará resultados distintos según las zonas definidas en la tabla zona, de modo que en el modelo planteado se han definido tres tipos de zonas distintos. La visión más macro de los resultados estará proporcionada por la tabla zonas propia de la base de datos \_WEST\_Terminal, que divide la red del modelo en dos únicas zonas correspondientes a las terminales T1 y T2. Una versión intermedia consiste en cuatro divisiones de la red, propia la base de datos \_WEST\_Default y por último la versión con mayor definición, se considerarán los arcos definidos como zonas. Esta tabla corresponde a la versión \_WEST\_Arcos.



TABLA 12. TABLAS USADAS EN LA BASE DE DATOS

## CLASES PRINCIPALES

### TAXIOUT.JAVA

La clase taxiOut.java es una de las dos clases que entrarían en la categoría de principales dentro del software desarrollado. Esta clase extiende la clase thread lo que permite heredar los métodos propios de esta última. TaxiOut.java contiene el algoritmo principal para realizar la simulación de los tiempos de taxi. Los objetos taxiOut.java después de llamar al método run, depositan las soluciones en las listas resultArcos, DetalleAeronaveLista y resultArcos. Estas serán recogidas por el objeto ttoutController para procesarlas y calcular los tiempos de medios, máximos, mínimos, número de paradas medias, máximas y mínimas para las tres listas de soluciones.

#### *CONSTRUCTOR*

El constructor de clase recibe nueve parámetros. Ejemplo de construcción del objeto:

```
taxis = new taxiOut (IAeronaves, hArcos, hZonas, listaSolucion, nZonas, Iteraciones,  
resultadoAeronavesADD, resultadoArcosADD,hPushback,detalleAeronaveLista);
```

Estos son:

- `HashMap<String, arco> listaArcos`: HashMap de objetos del tipo arco. Estructura que contiene la información de los arcos.
- `HashMap<Integer, zona> listaZonas`: Hasmap de objetos del tipo zona. Estructura que contiene la información de las zonas.
- `List<List<nAeronave>> listaSoluciones`: Lista de Lista de objetos del tipo nAeronave.
- `int gZonas`: Numero de zonas. Lista donde se añadirán las soluciones aportadas por el algoritmo en referencia a las aeronaves. Solo se usa en caso de querer guardar el resultado de cada iteración del algoritmo. Esta deshabilitada por defecto por el gran consumo de memoria que requiere querer almacenar todos los resultados obtenidos por cada iteración de cada thread.
- `int gIteraciones`: Numero de iteraciones a realizar por cada thread.
- `List<HashMap<String, nAeronave>> resultAeronaves`: Lista de HashMap de objetos del tipo nAeronave. En esta lista se depositará el HashMap obtenido como resultado de ejecutar el algoritmo principal (método startAlg).
- `List<HashMap<String, arco>> resultArcos`: Lista de HashMap de objetos del tipo arco. En esta lista se depositará el HashMap obtenido como resultado de ejecutar el algoritmo principal (método startAlg).
- `HashMap<String, pushback> vPush`: HashMap de objetos del tipo pushback. Estructura de datos que contiene la información necesaria para simular el pushback.
- `List<HashMap<String, HashMap<String,arco>>> gDetalleAeronaveLista`: Lista de HasMaps de Hashmaps de objetos del tipo arco. El método startAlg crea un hashmap de hashmaps de objetos tipo arco. Esta estructura contiene los tiempos medios de las aeronaves por cada arco. Ejemplo

de uso para recuperar el tiempo medio de una aeronave con identificador de vuelo “DLH1137” para el arco con identificador “K4-K1”.

```
tiempoMedio=detalleAeronave.get("DLH1137").get("K4-K1").MedTime
```

## MÉTODOS

### MÉTODO COPIARLISTAAERONAVES.

Este método retorna una copia de una lista pasada por argumento. En el programa desarrollado, cuando se crea un objeto a partir de la clase taxiOut, en el constructor se ha de pasar la lista de aeronaves leída de la base de datos por argumento. Cada thread crea su propio objeto, y necesita su propia lista de aeronaves para calcular sus métricas. Es por eso que siempre utiliza una copia de la lista original, en caso contrario todos los threads trabajarían sobre una misma lista imposibilitando realizar los cálculos.

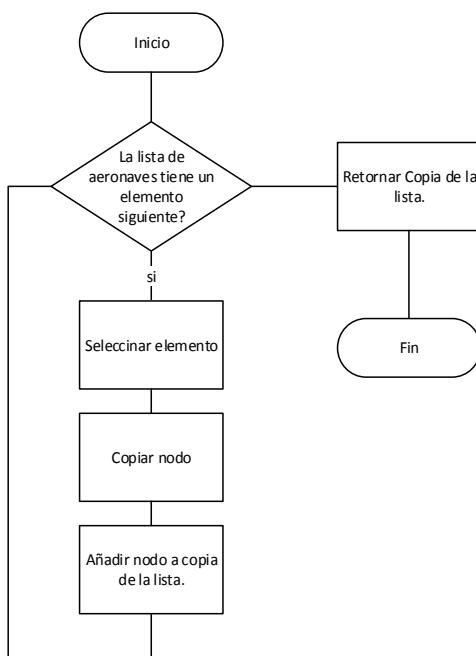


DIAGRAMA DE FLUJO 1. COPIARLISTAAERONAVES

### MÉTODO COPIARHASHMAPARCOS

Por los mismos motivos que el método anterior, es necesario crear una copia del HashMap que contiene la información de los arcos. Cada thread debe trabajar con una copia propia de las estructuras de datos que reciben como argumento en el constructor del objeto.

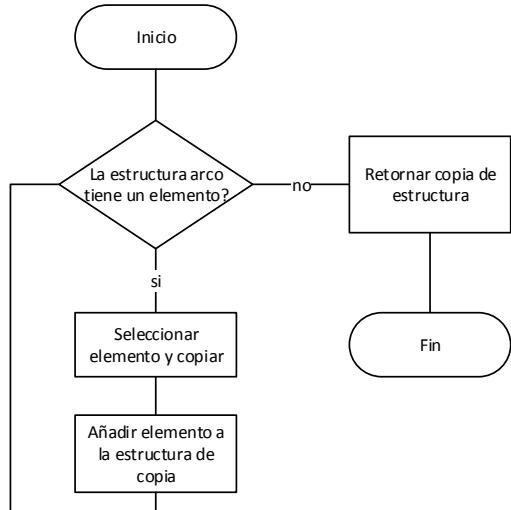


DIAGRAMA DE FLUJO 2. COPIARHASMAPARCOS

#### MÉTODO ORDENAR

El método `ordenar` recibe como único parámetro la lista de aeronaves y retorna la lista ordenada siendo el primer elemento el elemento con tiempo de siguiente evento mínimo. Para realizar esta tarea se utiliza la estructura `collection` y el método `sort` para ordenar la lista.

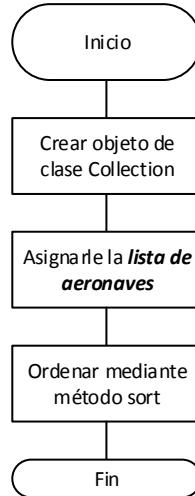


DIAGRAMA DE FLUJO 3. ORDEAR

#### MÉTODO COPIARNODOAERONAVE

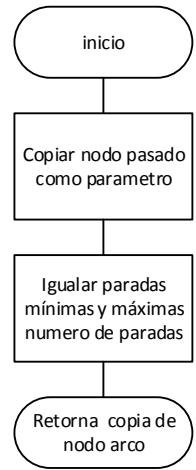
Recibe como parámetro el nodo a copiar y retorna una copia de este mismo nodo.



**DIAGRAMA DE FLUJO 4. COPIARNODOAERONAVE**

#### MÉTODO COPIARNODOARCO

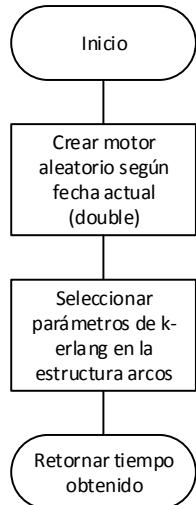
Copia el objeto del tipo arco pasado por argumento e iguala los valores de número de paradas mínimas y máximas al número de paradas.



**DIAGRAMA DE FLUJO 5. COPIARNODOARCO**

#### MÉTODO SIMPUSHBACK

Hace uso de la librería Parallel Colt para simular tiempo de taxi según la tipología de la aeronave recibida por parámetro. La función de distribución usada es una k-erlang de parámetros definidos en la base de datos y cargados en la estructura HashMap `arcos`.



**DIAGRAMA DE FLUJO 6. SIMPUSHBACK**

### MÉTODO SIMSTOP

Determina el número de paradas esperadas según la congestión en zona para un arco pasado por argumento. Este número de paradas se calcula mediante una función binomial negativa de parámetros dados por el arco. Por cada aeronave en zona se incrementa la probabilidad de tener un mayor número de paradas hasta un máximo de 3 por arco.

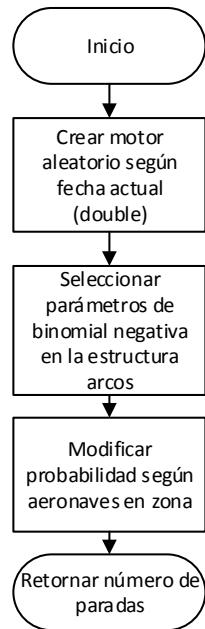


DIAGRAMA DE FLUJO 7. SIMSTOP

### MÉTODO SIMSTOPDURATION

Determina el tiempo de penalización sufrido por una aeronave al realizar un stop. Este tiempo está determinado por una función exponencial de parámetros determinados por el arco y la tipología de la aeronave.

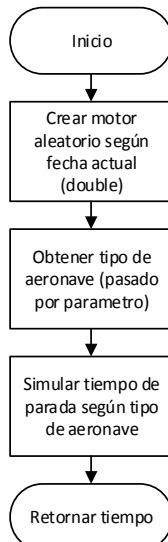


DIAGRAMA DE FLUJO 8. SIMSTOPDURATION

### MÉTODO SIMTIMEARC

Determina el tiempo de duración del trayecto entre dos nodos según la tipología de la aeronave. El tiempo se calcula mediante una distribución k-erlang con parámetros definidos por el arco en cuestión.

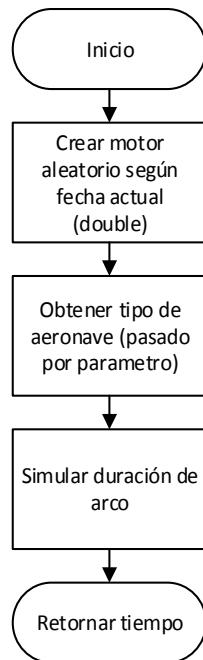


DIAGRAMA DE FLUJO 9. SIMTIMEARC

### MÉTODO RUN

El método Run es un método heredado de la clase thread. Este es el método a llamar para la ejecución del thread. Run ejecuta el método startAlg tantas veces como iteraciones haya y añade las tres estructuras de datos obtenidas al finalizar dichas ejecuciones a las tres listas de soluciones.

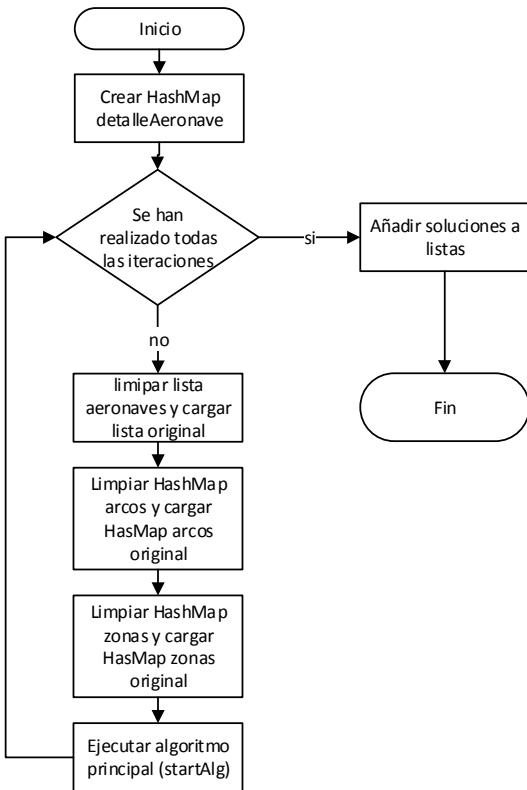


DIAGRAMA DE FLUJO 10. RUN

### MÉTODO STDDEV.

Este método calcula la desviación estándar de una lista de número pasada. Recibe como parámetros la lista, la media y el número total de iteraciones.

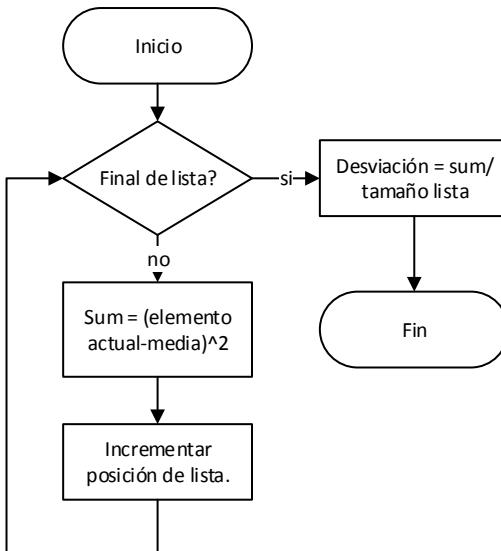
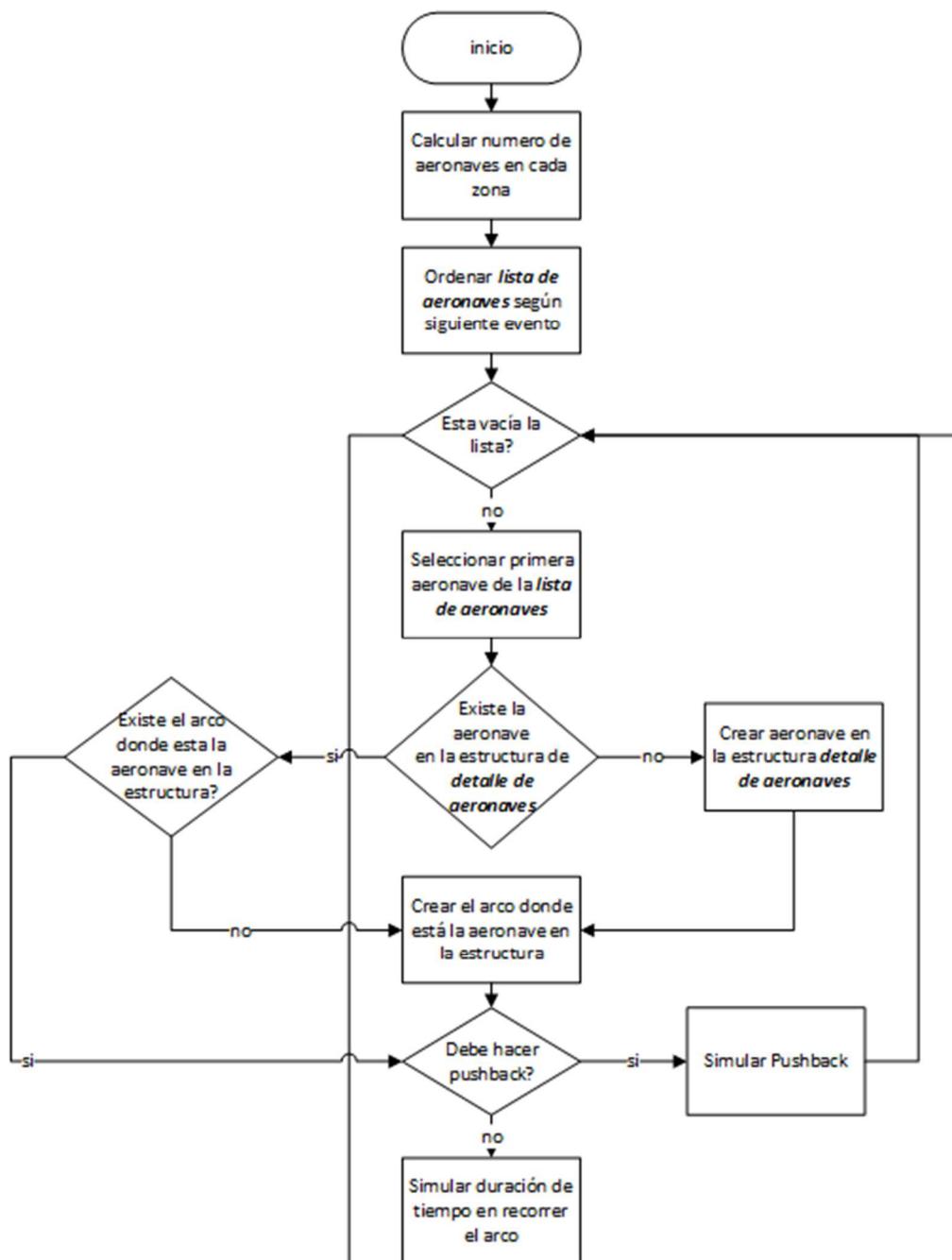


DIAGRAMA DE FLUJO 11. STDDEV

## MÉTODO STARTALG

El método startAlg incluye el algoritmo principal del programa. Este simula los movimientos y tiempos de las aeronaves en pista y crea tres estructuras de datos con las soluciones de las aeronaves, aeronaves en detalle y arcos. Estas estructuras serán procesadas más adelante al finalizar todos los threads y así calcular las métricas globales.



\*Continúa en la siguiente página

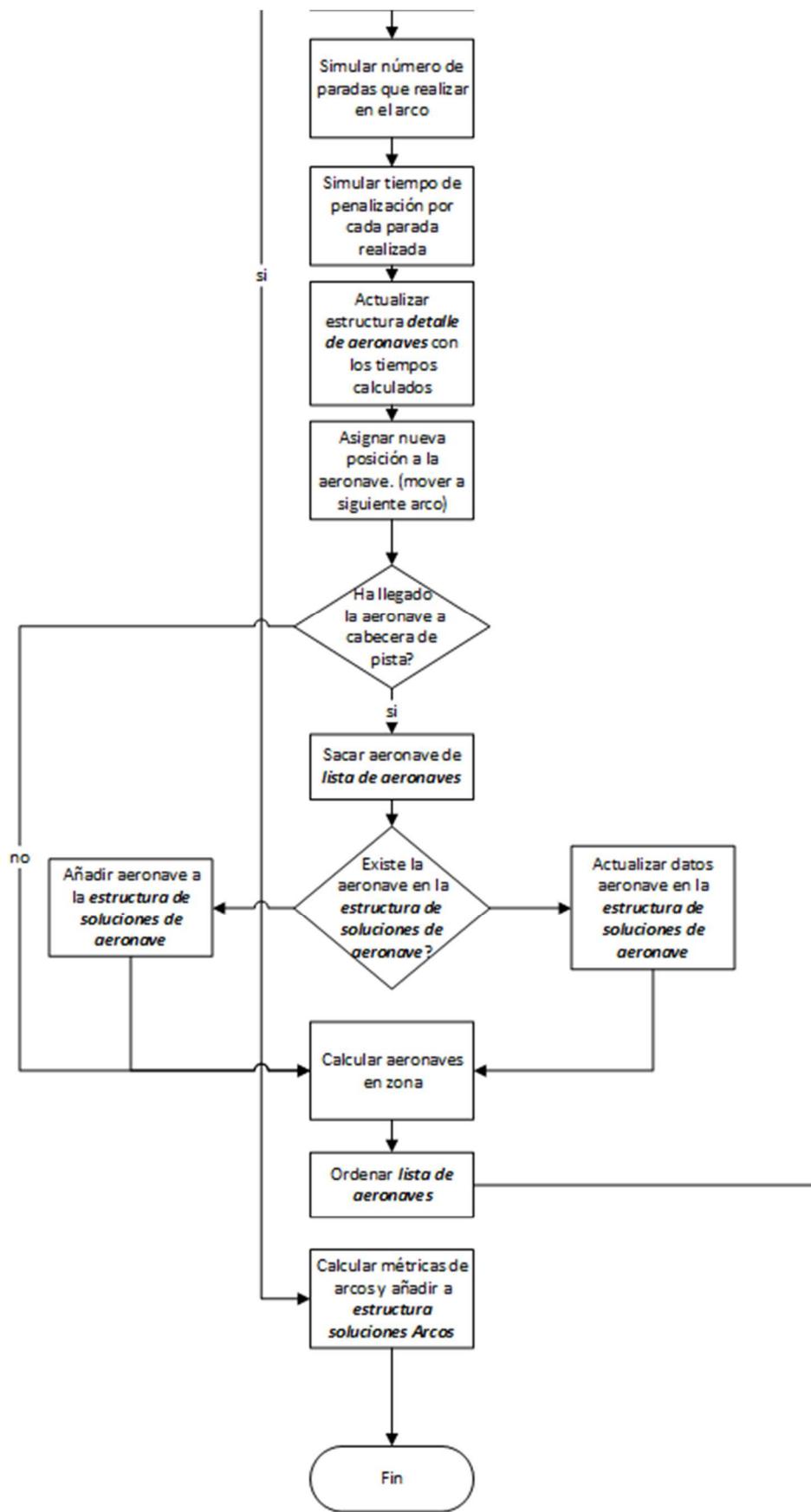


DIAGRAMA DE FLUJO 12. STARTALG

## TTOUTCONTROLLER.JAVA

Conjuntamente con la clase taxiOut.java, la clase ttoutController es la clase principal de la aplicación ya que contiene el resto de métodos principales que se utilizan. Esta clase es un controlador del archivo FXML de JavaFX, por lo tanto dentro de ttoutController están definidos todos los controladores que se utilizan ya sean gráficos, botones, tablas, etc. así como las funciones (o métodos) que se ejecutarán al realizar una acción. Ejemplo: Método que carga la base de datos en memoria cuando se pulsa el botón “Load Data”. TtoutController.java implementa la clase abstracta Initializable, lo que nos permite redefinir el método initialize. En este método se ejecuta al arrancar la aplicación, por lo que aquí podremos inicializar las estructuras de datos, tablas, etc.

## MÉTODOS

### MÉTODO DETAILAIR

El método detailAir es el método que se llama al oprimir el botón “Show Taxiout” en la pestaña “Detailed Aircrafts Results”. Este comprueba que haya resultados para mostrar, si los hay, llamará al método graficarDetalles.

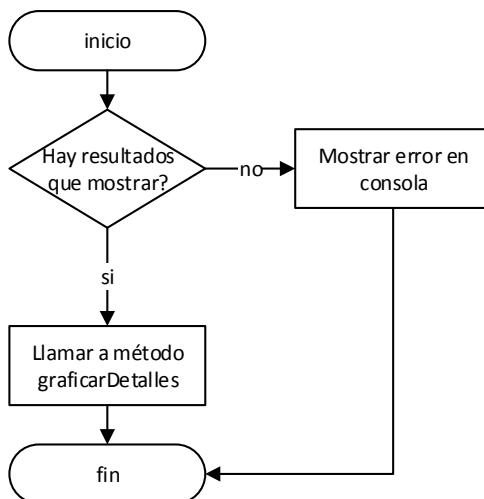


DIAGRAMA DE FLUJO 13. DETAILAIR

### MÉTODO GRAFICARDETALLES

GraficarDetalles es el método que se encarga de graficar los resultados detallados de las aeronaves en la pestaña “Detailed Aircrafts Results” así como mostrar en el área de texto de la pestaña los resultados del tiempo de llegada a cada nodo y número de paradas de forma acumulativa y parcial (tiempo o paradas desde ultimo nodo). Recibe como parámetros un HashMap de objeto tipo arco para una aeronave concreta (el vuelo se recoge del selector de vuelos), el arco de origen de ese vuelo y el número de vuelo. El método graficarZonas hace la misma función pero para las zonas previamente procesadas a partir de los arcos.

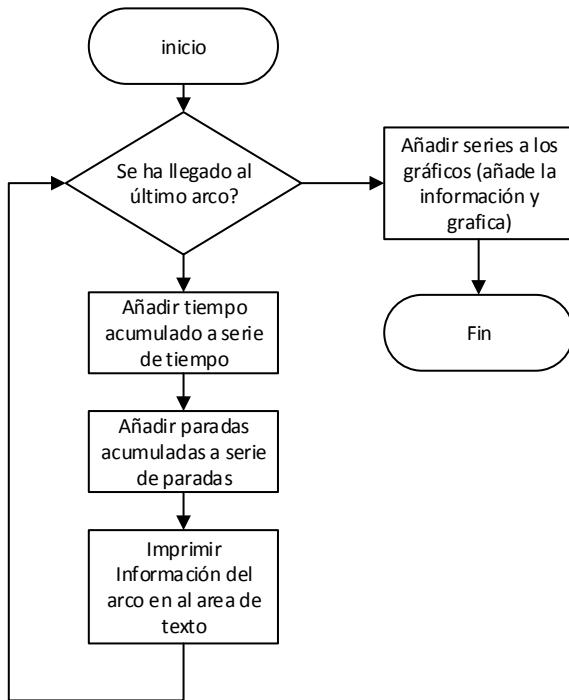


DIAGRAMA DE FLUJO 14. GRAFICARDETALLES

#### MÉTODO ENTRARDATOSAERONAVE

Para poder mostrar información los datos de aeronave en la tabla de la pestaña “Aircrafts Results” es necesario una clase especial con métodos particulares. Este método crea objetos de clase (Aeronave) y los llena con la información pasada por parámetros (número de vuelo, eobt, parking, rampa, etc.). Al inicializar la tabla se realiza un link a una estructura del tipo ObservableList <Aeroanve> y se indica qué variable corresponde a cada celda, por lo que cada vez que añadimos un objeto del tipo Aeronave a la estructura, esta se muestra en la tabla.

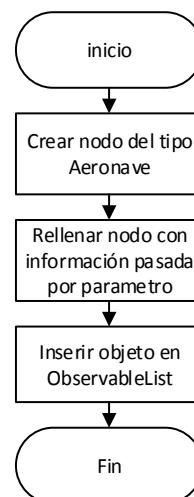


DIAGRAMA DE FLUJO 15. ENTRARDATOSAERONAVE

### MÉTODO ENTRARDATOSARCO

Crea un objeto del tipo tArco y lo complementa con la información obtenida del objeto de tipo arco pasado. Al igual que la clase Aeronave, tArco es una clase preparada para mostrar la información de los arcos en la tabla de la pestaña “Map Network”.

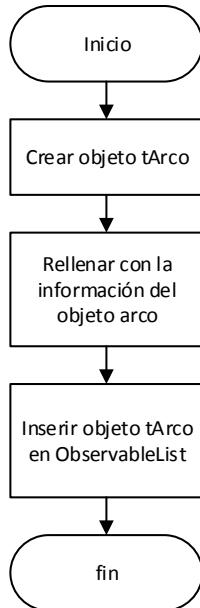
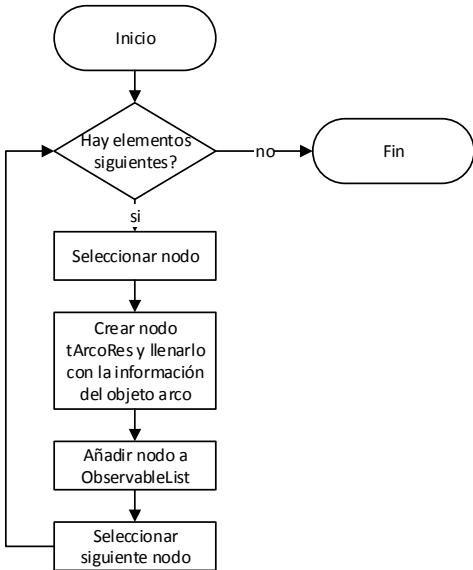


DIAGRAMA DE FLUJO 16. ENTRARDATOSARCO

### MÉTODO ENTRARDATOSARCORES

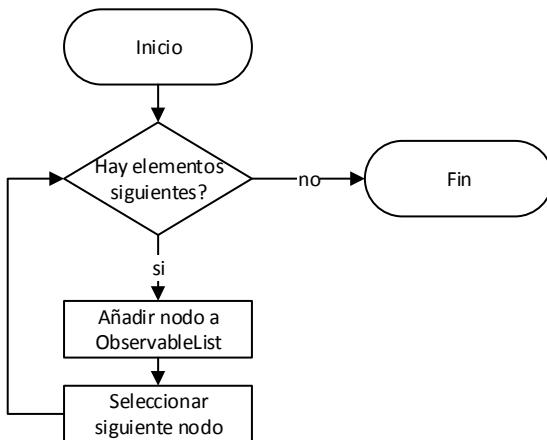
Recibe como parámetro el HashMap con el resultado de los arcos. El método recorre cada nodo de esta estructura de datos y por cada nodo que recorre crea un objeto del tipo tArcoRes y lo llena con la información del nodo actual. Finalmente añade el objeto a una ObservableList ligada a una tabla para poder mostrarlos. En esta estructura de datos se almacena la media de tiempo que pasa una aeronave en un arco.



**DIAGRAMA DE FLUJO 17. ENTRARDATOSARCORES**

#### MÉTODO ENTRARDATOSZONA

La clase zona se ha pensado como un objeto para mostrar en la tabla y en los gráficos de zona. No se pasa ninguna estructura que contenga objetos del tipo zona al algoritmo principal sino que los resultados de zona se obtienen procesando los resultados de los objetos del tipo arcos obtenidos como resultado del algoritmo. Es por eso que únicamente los objetos zona se extraen de una estructura HashMap anteriormente procesada y se añaden a una estructura del tipo ObservableList. EntrarDatosZona contiene las medias de tiempo que pasa una aeronave en una zona concreta.



**DIAGRAMA DE FLUJO 18. ENTRARDATOSZONA**

#### MÉTODO PROCESARRESULTADOS

El método procesarResultados procesa todas las estructuras de datos que los Threads (clase taxiOut.java) añaden a las listas. El programa tiene tres listas compartidas por los threads, estas listas son:

1. Lista que donde cada thread añade la solución de las aeronaves (general).

- a. Da tiempo máximo, tiempo mínimo, tiempo medio, paradas máximas, paradas mínimas y paradas medias por cada aeronave. (Tamaño de la estructura = # de aeronaves).
2. Lista donde cada thread añade la solución de las aeronaves por arco (Detallado).
    - a. Da tiempo máximo, tiempo mínimo, tiempo medio, paradas máximas, paradas mínimas y paradas medias por cada aeronave y arco. (Tamaño de la estructura = número de aeronaves \* número de arcos).
  3. Lista donde cada thread añade la solución de los arcos.
    - a. Da tiempo máximo, tiempo mínimo, tiempo medio, paradas máximas, paradas mínimas y paradas medias por cada arco. (Tamaño de la estructura = # de arcos).

El método crea tres estructuras de datos para almacenar los resultados de arco, aeronave y aeronave detallada más una para almacenar los resultados de zona a partir de la información de arcos. Este método calcula las medias y las desviaciones para número las paradas y el tiempo.

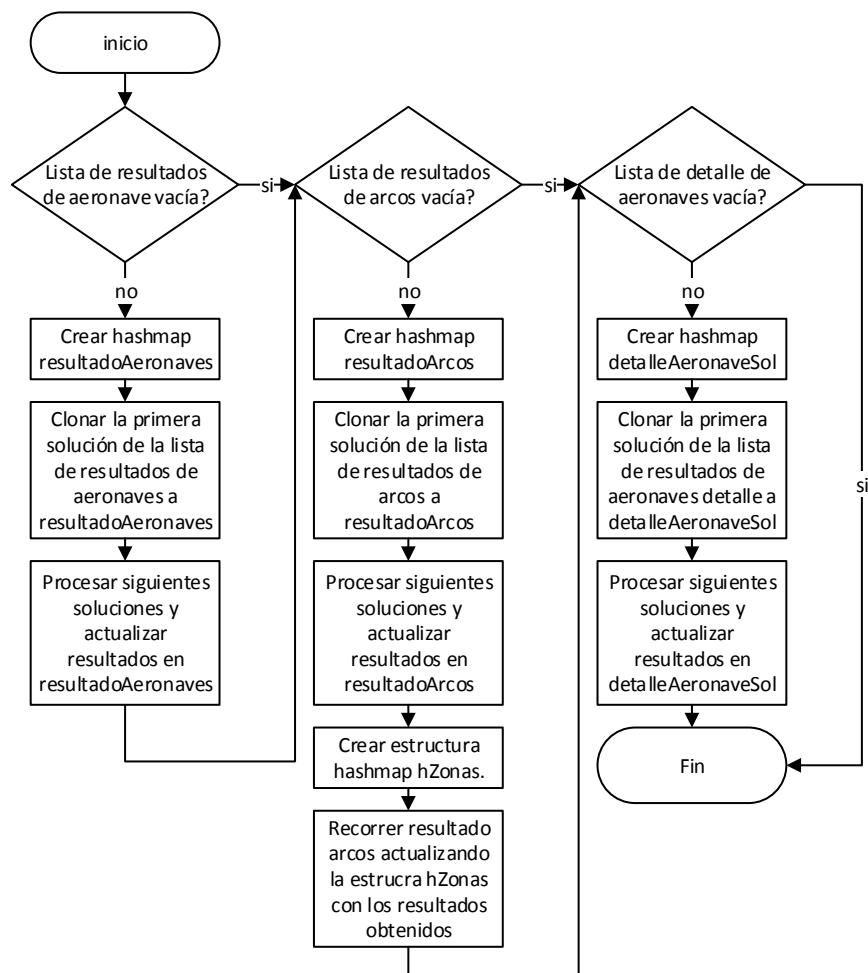


DIAGRAMA DE FLUJO 19. PROCESARRESULTADOS

## MÉTODO ADDDATABASE

El método se ejecuta cuando el usuario pulsa el botón “Load Data”. AddDataBase lee toda la información de la base de datos y la carga en las estructuras correspondientes. Los parámetros necesarios son:

- IP de la base de datos.
- Puerto.
- Nombre de la base de datos.
- Tipología de zona a cargar
  - 4 Zones: Carga 4 zonas tal y como muestra el mapa de la pestaña “Map Network”
  - Terminals: Cada terminal se considera una zona (2 zonas totales).
  - Zones = Arcs. Cada arco se considera una zona independiente.
- Usuario de la base de datos.
- Password de la base de datos.

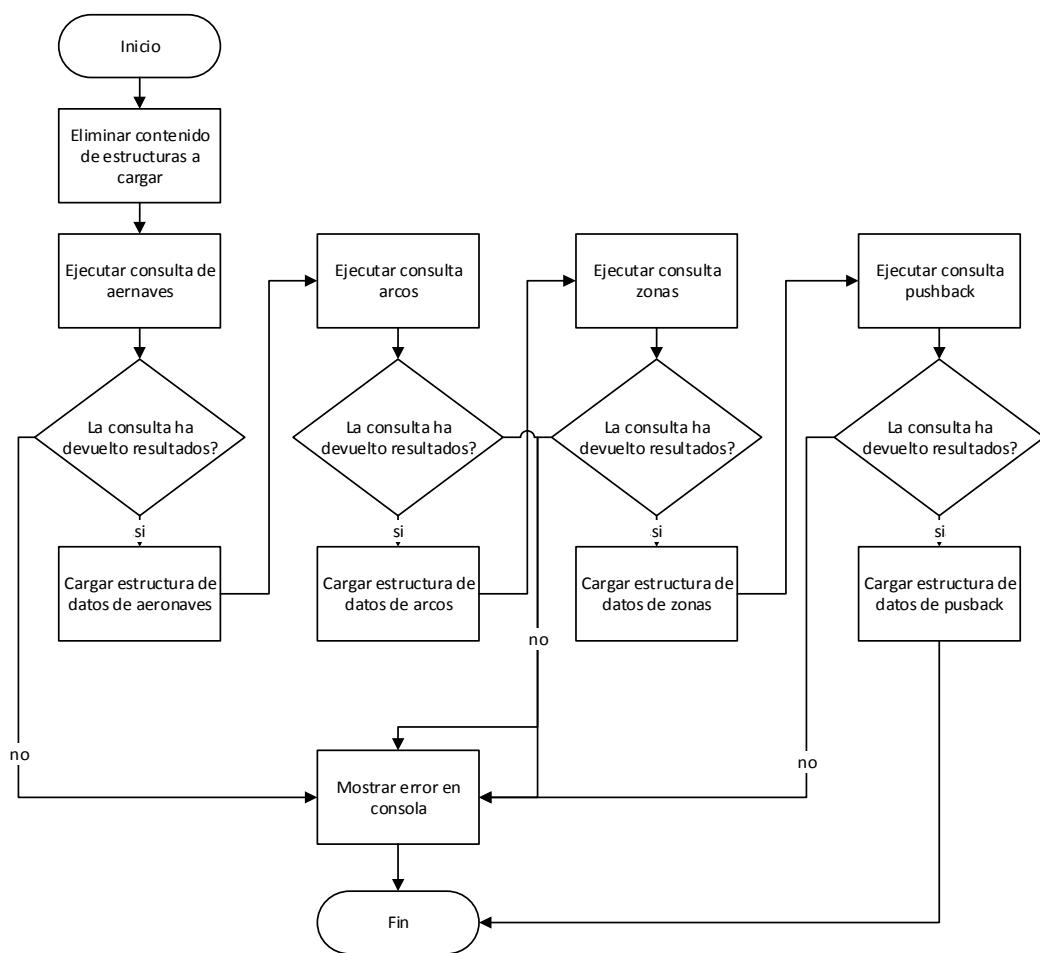


DIAGRAMA DE FLUJO 20. ADDDATABASE

## MÉTODO INITIALIZE

El método initialize es heredado por la implementación de la clase abstracta Initializable. Este método se llama cada vez que se crea el objeto definido por la clase ttoutController.java. Esto sucede cada vez que la interfaz arranca satisfactoriamente. Initialize permite inicializar todas las estructuras y variables que son necesarias.

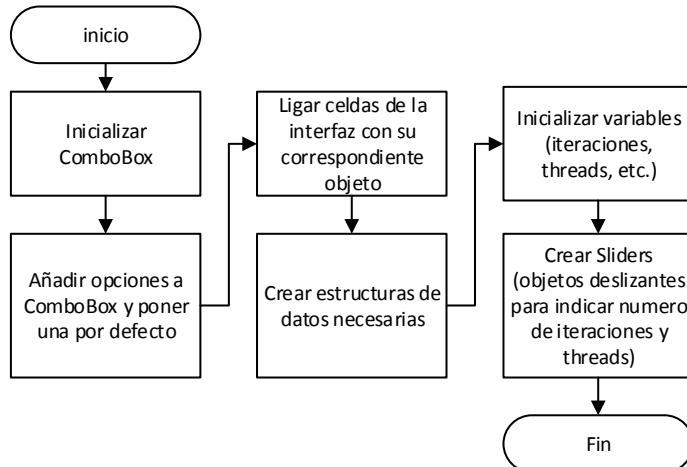
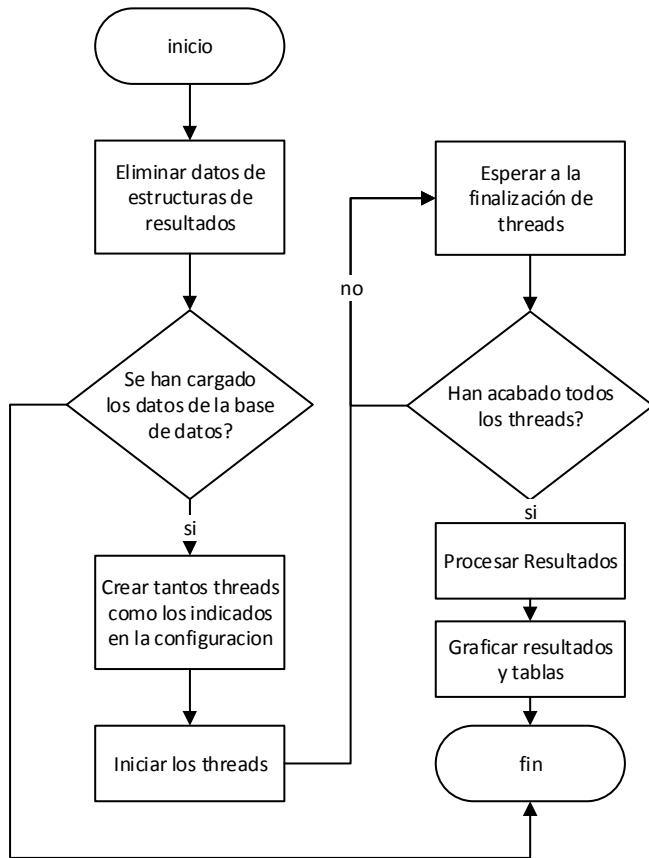


DIAGRAMA DE FLUJO 21. INITIALIZE

#### MÉTODO STARTALGORITHM

El método startAlgorithm es el encargado de crear los objetos del tipo taxiOut. Después de crearlos invoca al método start y los encola en una lista. Finalmente espera a que todos los threads de la lista acaben para llamar al método de ttoutController.java que procesa los datos y los muestra en pantalla.



**DIAGRAMA DE FLUJO 22. STARTALGORITHM**

## CLASES SECUNDARIAS

A continuación se incluye una lista con las clases secundarias. Estas clases son aquellas que se utilizan casi exclusivamente para como nodos de las estructuras de datos, clases específicas para poder mostrar la información de las tablas y la clase que define el objeto para posibilitar la conexión a una base de datos MySQL o MirandaDB.

### CLASES ESPECÍFICAS PARA MOSTRAR INFORMACIÓN EN TABLAS

#### AERONAVE.JAVA

Aeronave.java es una clase utilizada para mostrar la información de las aeronaves en la tabla situada en la pestaña “Aircrafts Results”.

#### TARCO.JPG

Utilizada para mostrar datos referentes a los arcos en la tabla de la pestaña “Map Network”.

#### TARCORES.JPG

Clase creada para mostrar los resultados de los arcos en la tabla de la pestaña “Arc Results”.

#### ZONA.JPG

Clase creada para mostrar la información referente a los resultados de zona en la pestaña “Zone Results”.

## CLASES QUE ACTÚAN COMO NODOS DE LAS ESTRUCTURAS

### AERONAVE.JAVA

La clase aeronave contiene toda la información de la base de datos para una aeronave particular así como más variables que se usan en el algoritmo principal. Esta clase implementa la clase abstracta Cloneable para poder duplicar los objetos con facilidad.

#### VARIABLES DE CLASE

- String matricula; Matricula de la aeronave.
- String nVuelo; Número de vuelo.
- String type; Categoría de aeronave (Heavy, Light, Medium)
- String arcoActual; Arco en que se encuentra la aeronave durante la simulación
- String origen; Arco en que está la aeronave al principio de la simulación.
- double eobt; Tiempo estimado de retirada de calzos (OFF BLOCK TIME)
- double ctot; Tiempo calculado de despegue.
- double nEvent; Tiempo actual de la aeronave en la simulación.
- double arrivalQ; Tiempo de llegada a cabecera de pista.
- double maxTime; Tiempo máximo de llegada a cabecera de pista.
- double minTime; Tiempo mínimo de llegada a cabecera de pista.
- double minStops; Paradas mínimas registradas en las simulaciones.
- double maxStops; Paradas máximas registradas en las simulaciones.
- boolean pushback; Indica si el aeronave ha de realizar pushback.
- boolean pushback\_original; indica si el aeronave realiza pushback (no se modifica en la simulación).
- double pTime; Tiempo de pushback simulado (cambia en cada iteración realizada)
- double nStops; Número de paradas realizadas en cada iteración de la simulacion.
- int iteracion; Número de iteraciones realizadas.
- String Parking; Parking del que sale la aeronave.
- String Rampa; Rampa del que sale la aeronave.
- Double deviation: Desviación estándar del tiempo de taxiout.
- Double stopDeviation: Desviación estándar del número de paradas.
- List<Double> timelist; Lista con tiempos de recorrido de aeronave.
- List<Double> stoplist; Lista con número de paradas de aeronave

#### **ARCO.JAVA**

La clase arco contiene toda la información de la base de datos para un arco particular así como más variables que se usan en el algoritmo principal. Esta clase implementa la clase abstracta Cloneable para poder duplicar los objetos del tipo arco con facilidad.

#### **VARIABLES DE CLASE**

- String id; Identificador de arco.
- String Siguiente; Identificador de nodo siguiente.
- int Zona; Número de zonas definidas.
- double HEa; Primer parámetro de función k-erlang para aeronaves de categoría H.
- double HEb; Segundo parámetro de función k-erlang para aeronaves de categoría H.
- double LEa; Primer parámetro de función k-erlang para aeronaves de categoría L.
- double LEb; Segundo parámetro de función k-erlang para aeronaves de categoría L.
- double MEa; Primer parámetro de función k-erlang para aeronaves de categoría M.
- double MEB; Segundo parámetro de función k-erlang para aeronaves de categoría M.
- double AG; Parámetro para la función geométrica.
- double HExp; Parámetro para la función exponencial para aeronaves de categoría H.
- double LExp; Parámetro para la función exponencial para aeronaves de categoría L.
- double MExp; Parámetro para la función exponencial para aeronaves de categoría M.
- double MedTime; Tiempo medio en arco de todas las aeronaves.
- double MaxTime; Tiempo máximo en arco de todas las aeronaves.
- double MinTime; Tiempo mínimo en arco de todas las aeronaves.
- double nStops; Número de paradas para la simulación actual en el arco.
- double MaxStops; Número máximo de paradas para la simulación actual en el arco
- double MinStops; Número mínimo de paradas para la simulación actual en el arco.
- int numAeronaves; Número de aeronaves que han pasado por el arco.

#### **CLASES PARA LA CONEXIÓN A LA BASE DE DATOS**

#### **DBCONNECT.JAVA**

Para la conexión a la base de datos la clase hace uso de la API JDBC de java así como el controlador JDBC oficial de MySQL que es el que implementa las funcionalidades de las clases de acceso a la base de datos y proporciona la comunicación entre el API JDBC y la base de datos real.

#### **VARIABLES DE CLASE**

- private static Connection conn; Variable de conexión.
- private static String url; IP o url de la base de datos.
- private static String user; Usuario de la base de datos.
- private static String pass; Password del usuario

## ALGORITMO PROBABILISTA PROPUESTO

Un algoritmo probabilista es aquel que emplea el uso de la aleatoriedad como parte de su lógica. Los comportamientos de estos algoritmos pueden variar incluso con inputs iguales. El algoritmo descrito a continuación predice los tiempos de taxi-out basándose en un modelo de la superficie del aeropuerto desarrollo en durante el proyecto. Éste tiene en cuenta el tráfico presente en el aeropuerto para determinar el número de paradas que se esperan que las aeronaves realicen con el fin de determinar el tiempo de taxi esperado para las aeronaves recibidas como input.

### PSEUDOCÓDIGO

```
00: inicio
01: ordenar lista de aeronaves por tiempo de evento
02: calcular número de aeronaves en zona
03: mientras haya aeronaves en lista de aeronaves
04:     seleccionar primera aeronave de la lista
05:     si aeronave seleccionada debe realizar pushbak
06:         simular pushback de aeronave
07:         tiempo evento <- tiempo de evento + pushback
08:     si no
09:         simular tiempo de recorrido para arco actual
10:         tiempo evento <- tiempo de evento + tiempo de recorrido simulado
11:         número de paradas <- simular paradas
12:         para cada número de paradas simulas
13:             simular duración de parada
14:             tiempo de evento <- tiempo de evento + duración de parada
15:         fin para
16:         calcular métricas de arco
17:         calcular métricas detalladas de aeronave
18:         incrementar posición de la aeronave
19:         si posición de la aeronave es cabecera de pista
20:             eliminar aeronave de la lista de aeronave
21:             calcular métricas de la aeronave
22:         fin si
23:     fin si
24: fin si
25: ordenar lista de aeronaves por tiempo de evento
26: calcular número de aeronaves en zona
27: fin mientras
28: agregar soluciones parciales a la lista de soluciones
30: fin
```

## CARACTERÍSTICAS PRINCIPALES DE LA APLICACIÓN

### FLEXIBILIDAD

La aplicación se ha desarrollado con el objetivo que toda la información sea definida en la base de datos. Es por eso que las zonas, arcos, aeronaves y funciones de distribución necesarias no están especificados dentro del propio programa. De esta forma el software permite calcular los tiempos de taxiout estimados y número de paradas estimadas no sólo para el aeropuerto de Barcelona en configuración oeste, sino de cualquier otra configuración de pista o cualquier aeropuerto definido en la base de datos. En el propio programa se han especificado tres configuraciones de zona diferentes para realizar el cálculo del número de paradas según los aviones en zona para poder mostrar la flexibilidad de este.

- Cuatro zonas (dos por cada terminal, definido en el mapa).
- Dos zonas (cada zona son los terminales).
- Número de zonas es igual a número de arcos (Cada arco se considera una zona en si misma).

### PROGRAMACIÓN CONCURRENTE

Un hilo o Thread es la secuencia de código más pequeña que puede ser gestionada por el planificador del sistema operativo. La implementación de estos procesos ligeros es dependiente al sistema operativo, y por tanto difiere de uno a otro. Para el mejor rendimiento del algoritmo principal se optó por hacerlo multihilo, permitiendo ejecutar diferentes instancias de la aplicación de forma concurrente con la finalidad de optimizar el rendimiento de este y realizar el total de iteraciones en menor tiempo (total de iteraciones = Iteraciones \* número de threads). De esta forma se aprovechan las capacidades Hyperthreading o multinúcleo de las máquinas actuales.

### USO DE LA LIBRERÍA PARALLEL COLT

La librería Parallel Colt es una implementación multihilo de la librería Colt que es capaz de aprovechar las capacidades de las máquinas con más de un procesador. Colt es una librería de código libre escrita en Java y desarrollada por el CERN. Se ha utilizado esta librería para obtener valores de las diferentes funciones de distribución necesarias para realizar la simulación orientada a eventos discretos.

### INTERFAZ DE USUARIO

La interfaz de usuario ha sido desarrollada con el objetivo de mostrar únicamente información relevante al usuario. La interfaz está dividida en dos secciones, una superior y otra inferior que a su vez se dividen en objetos y pestañas.

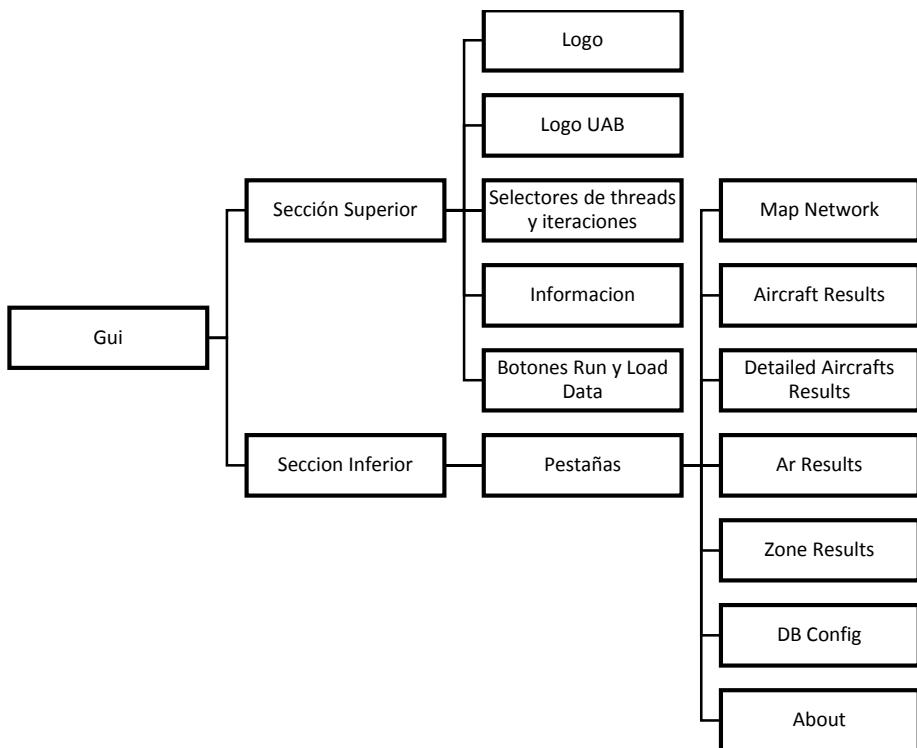


ILUSTRACIÓN 5. COMPOSICIÓN DE LA INTERFAZ DE USUARIO

## SECCIÓN SUPERIOR

Esta sección superior está compuesta por cinco componentes.

- Logos del programa y Universitat Autònoma de Barcelona.
- Selección de threads e iteraciones: Permite al usuario definir el número total de iteraciones a realizar.
- Consola: Muestra información referente al estado general del programa (cuando empieza el algoritmo, cuando acaba y posibles errores como los debidos a una carga incorrecta de los datos).
- Botones referentes a la carga de datos y a la ejecución del algoritmo.



ILUSTRACIÓN 6. SECCIÓN SUPERIOR DE LA INTERFAZ DE USUARIO

## SECCIÓN INFERIOR



ILUSTRACIÓN 7 . SECCIÓN INFERIOR DE LA INTERFAZ DE USUARIO

La sección inferior está compuesta por las pestañas que muestran los resultados, configuraciones, etc.

### PESTAÑAS

#### MAP NETWORK

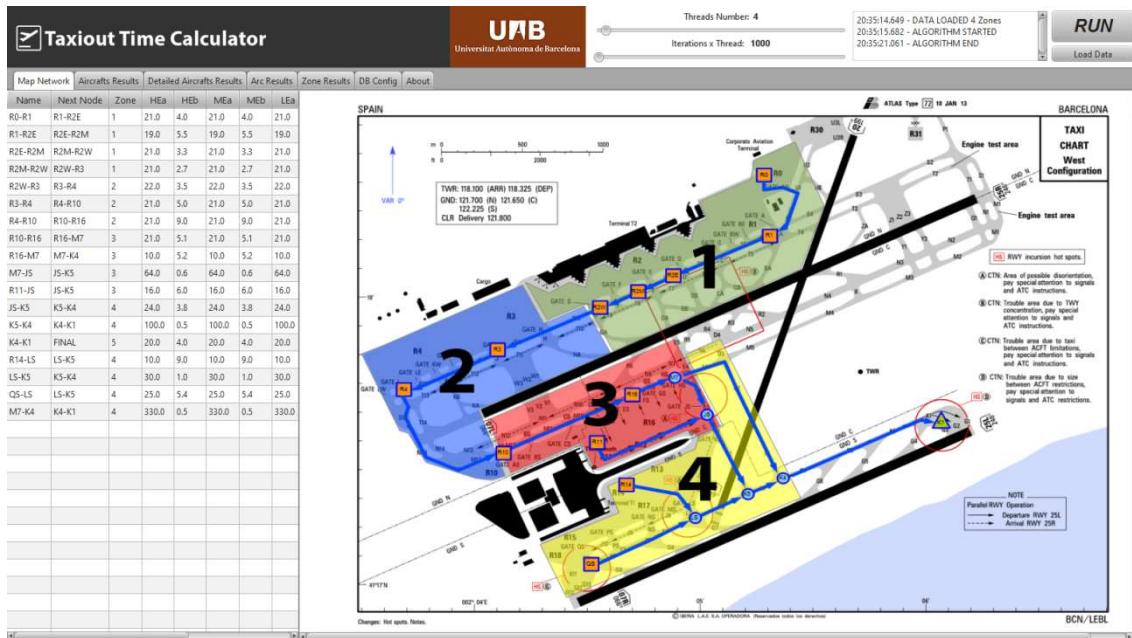


ILUSTRACIÓN 8. PESTAÑA MAP NETWORK

La pestaña “Map Network” muestra la información de los arcos cargados en una tabla situada a la izquierda y el mapa de arcos y zonas a la derecha.

## AIRCRAFT RESULTS

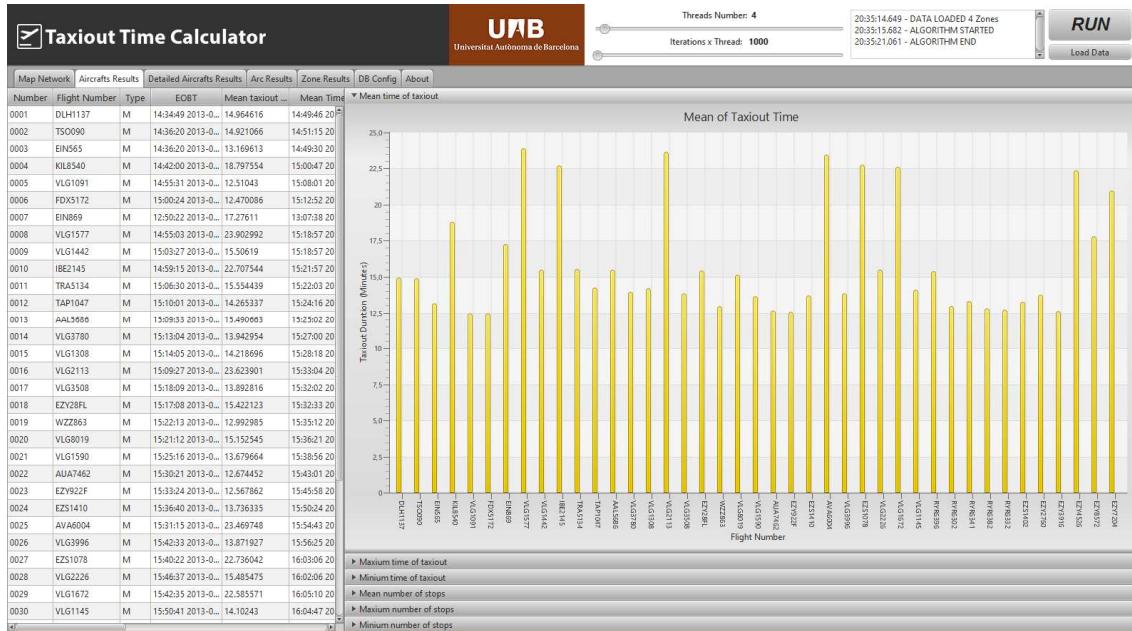


ILUSTRACIÓN 9. PESTAÑA "AIRCRAFT RESULTS"

La pestaña "Aircraft Results" muestra a la izquierda de la pantalla una tabla con la información de la aeronave y los outputs del algoritmo y a la izquierda un acordeón donde se pueden visualizar los siguientes gráficos.

- Mean time of taxiout.
- Maximum time of taxiout.
- Minium time of taxiout.
- Mean number of stops.
- Maximum number of stops.
- Minium number of stops.

## DETAILED AIRCRAFT RESULTS

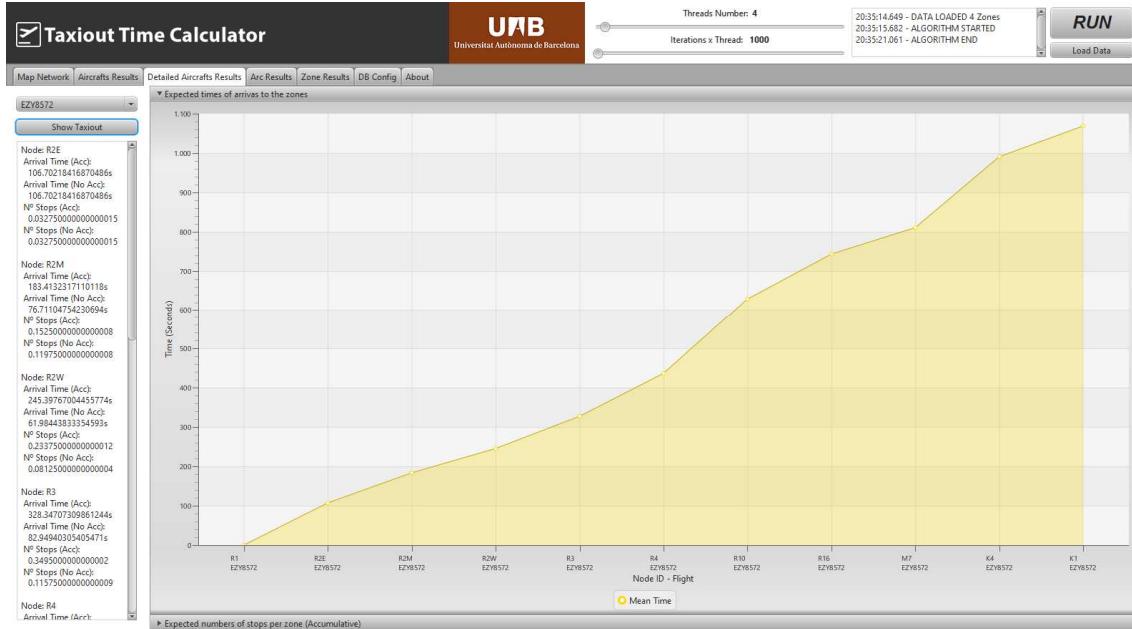


ILUSTRACIÓN 10. PESTAÑA "DETAILED AIRCRAFTS RESULTS"

Esta pestaña muestra a la izquierda de la pantalla el selector de vuelo, un botón para mostrar la información del vuelo seleccionado. En la izquierda tenemos un acordeón con los siguientes gráficos.

- Expected times of arrivals to the zones.
- Expected numbers of stops per zone.

## ARC RESULTS

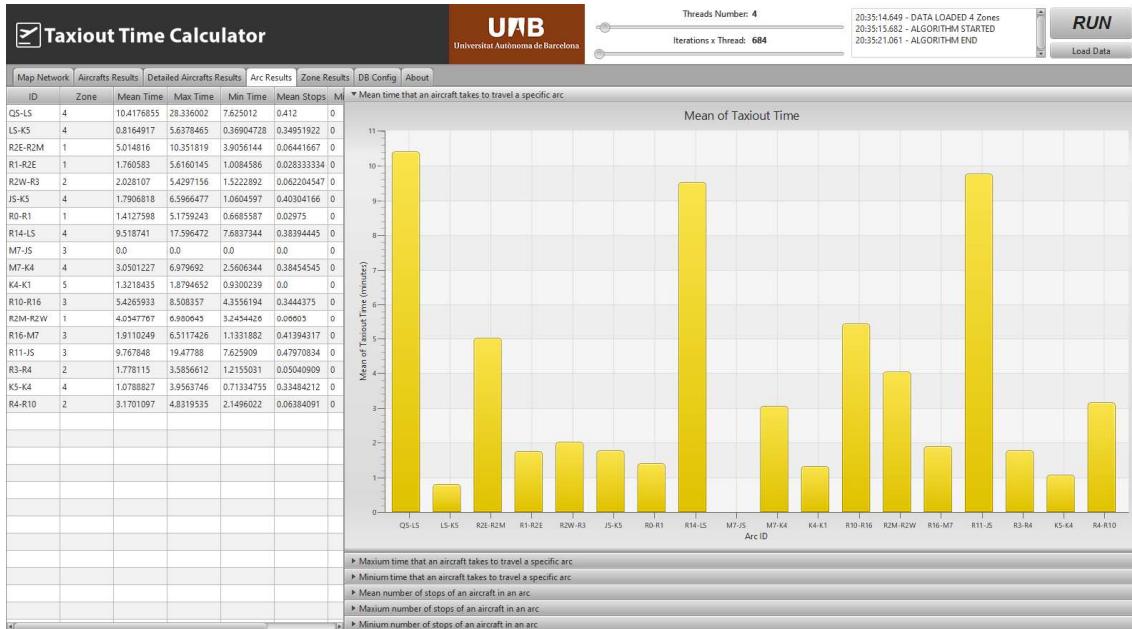


ILUSTRACIÓN 11. PESTAÑA "ARC RESULTS"

En la pestaña “Arc Results” muestra a la izquierda una tabla con los outputs del programa referente a los arcos y a la derecha un acordeón con los siguientes gráficos.

- Mean time that an aircraft takes to travel a specific arc.
- Maximum time that an aircraft takes to travel a specific arc.
- Minimum time that an aircraft takes to travel a specific arc.
- Mean number of stops of an aircraft in an arc.
- Maximum number of stops of an aircraft in an arc.
- Minimum number of stops of an aircraft in an arc.

## ZONE RESULTS

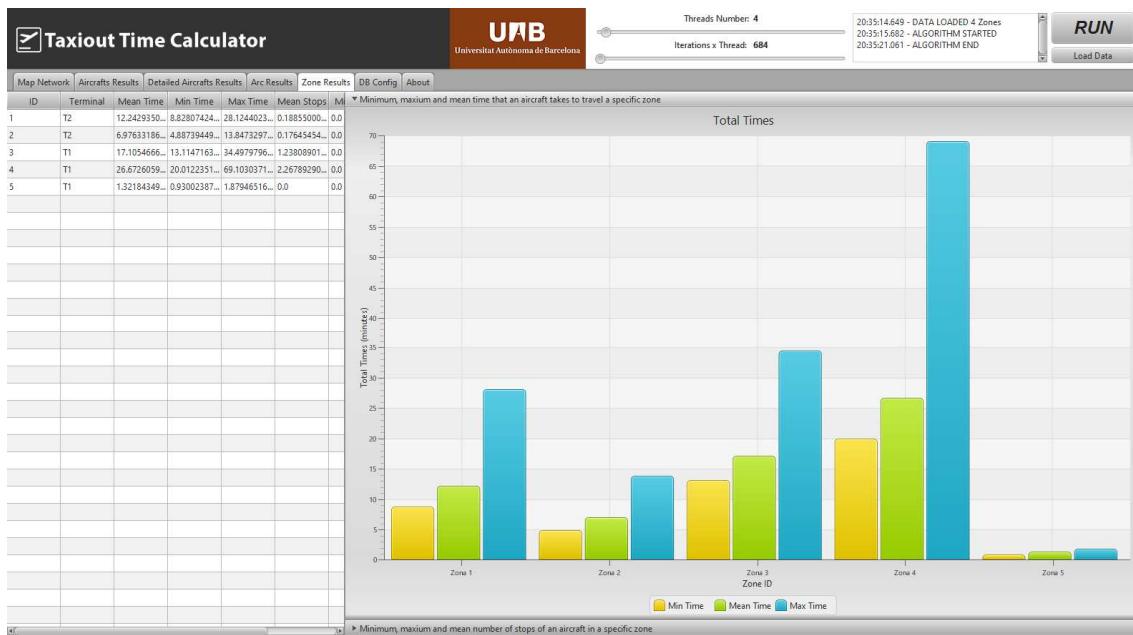


ILUSTRACIÓN 12. PESTAÑA "ZONE RESULTS"

La pestaña “Arc Results” muestra en la parte izquierda una tabla con los outputs calculados de las zonas y en la izquierda un acordeón con los siguientes gráficos.

- Minimum, maximum and mean time that an aircraft takes to travel a specific zone.
- Minimum, maximum and mean number of stops of an aircraft in a specific zone.

## DB CONFIG

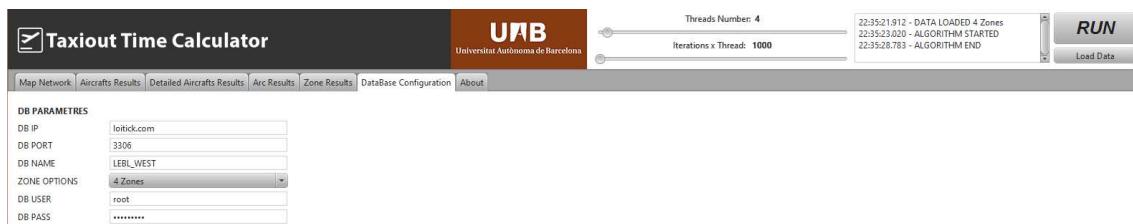


ILUSTRACIÓN 13. PESTAÑA "DATABASE CONFIGURATION"

En la parte derecha muestra los campos para la configuración de la base de datos así como el selector de zona.

## ABOUT

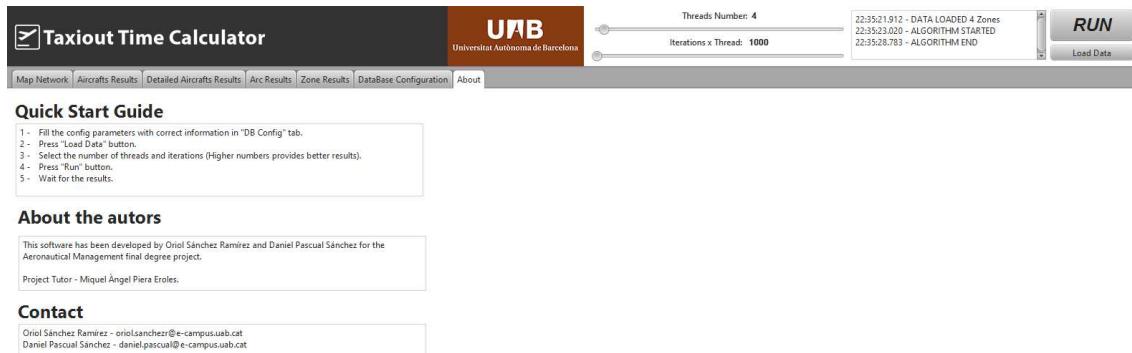


ILUSTRACIÓN 14. PESTAÑA "ABOUT"

La pestaña “About” muestra a la izquierda una guía de uso rápido e información acerca de los autores.

## CONCLUSIONES

### CONCLUSIONES DEL MODELO

Las siguientes conclusiones se derivan del estudio realizado previo al desarrollo del programa.

- Una buena predicción de los tiempos de taxi requiere información muy precisa de la pista, la ruta de taxi y el parking de salida de la aeronave.
- Desarrollar un mejor modelo de predicción de taxi out es posible y más fiable que el uso de datos estáticos que no tengan en cuenta el estado del aeropuerto en tiempo real.
- Una mejor predicción de los tiempos de taxi influirá positivamente sobre la capacidad del aeropuerto.
- El modelo desarrollado en este proyecto se puede ampliar teniendo en cuenta factores como la climatología, limitaciones de servicios en tierra y alteraciones en los nodos que componen la red.
- El modelo desarrollado puede extender a configuraciones específicas para el Aeropuerto de Barcelona así como a otros aeropuertos cambiando la definición de los arcos, nodos y zonas que componen la red.

### CONCLUSIONES DEL DESARROLLO DE LA APLICACIÓN

- El uso del api de JavaFx ha permitido un rápido desarrollo de la interfaz gráfica.
- El consumo de memoria por parte de la aplicación es excesivo si se almacenan las simulaciones realizadas por los threads, en una primera versión se almacenaba esta información pero finalmente se hizo uso de cálculos parciales.
- El uso de threads incrementa notablemente la velocidad de la aplicación.
- Los posibles usuarios de la aplicación desarrollada pueden ser:
  - CDM: Obtendría una mejor gestión de los recursos del aeropuerto.
  - Modelos AMAN-DMAN: Cualquier modelo que necesite realizar una simulación de tiempos de taxi puede hacer uso del modelo propuesto para este propósito.
  - Aerolíneas: Una mejor predicción de los tiempos de taxi puede ayudar a ahorrar costes derivados al consumo de combustible en la maniobra de taxi.
- 

## RESULTADOS OBTENIDOS

Resultados obtenidos a partir de 41 aeronaves (Listado de vuelo extraído de la base de datos incluida en el anexo).

Parámetros: 1000 iteraciones, 4 threads (4000 iteraciones totales), 4 zonas.

Duración del cálculo: 5,872 segundos.

Máquina: Intel(R) Core(TM)2 Quad CPU Q6600 @2.4GHz – 4GB (RAM) – Widows 8

FLIGHT NUMBER	OFF BLOCK TIME (UNIX TIME)	MEAN TIME (UNIX TIME)	T.DEVIATION (SECONDS)	MEAN STOPS	STOPS DEVIATION
DLH1137	1368448489	1368449396	185	1,0000	1,7343
TSO090	1368448580	1368449483	173	0,9040	1,6211
EIN565	1368448580	1368449380	195	0,8330	1,5940
KIL8540	1368448920	1368450053	245	1,0160	1,9417
VLG1091	1368449731	1368450491	187	1,0295	1,8888
FDX5172	1368450024	1368450774	175	0,8015	1,6644
EIN869	1368442222	1368443265	235	0,8495	1,7572
VLG1577	1368449703	1368451141	268	1,9182	2,7420
VLG1442	1368450207	1368451139	189	1,5325	2,1753
IBE2145	1368449955	1368451313	240	1,6245	2,3719
TRA5134	1368450390	1368451320	189	1,5182	2,2475
TAP1047	1368450601	1368451452	207	1,4717	2,2153
AAL5686	1368450573	1368451500	193	1,4032	2,1446
VLG3780	1368450784	1368451617	218	1,4102	2,1794
VLG1308	1368450845	1368451695	210	1,4537	2,2052
VLG2113	1368450567	1368451984	239	1,3227	2,0894
VLG3508	1368451089	1368451924	228	1,3722	2,1762
EZY28FL	1368451028	1368451952	190	1,2822	1,9990
WZZ863	1368451333	1368452112	212	1,3022	2,1217
VLG8019	1368451272	1368452184	176	1,0035	1,7024
VLG1590	1368451516	1368452341	222	1,1932	2,0477
AUA7462	1368451821	1368452583	194	1,0197	1,8718
EZY922F	1368452004	1368452759	186	0,8640	1,7458
EZS1410	1368452200	1368453024	177	0,9753	1,7424
AVA6004	1368451875	1368453281	239	1,2612	2,1362
VLG3996	1368452553	1368453387	188	1,1532	1,9777
EZS1078	1368452422	1368453787	239	1,8775	2,6507
VLG2226	1368452797	1368453725	183	1,5310	2,1929
VLG1672	1368452555	1368453906	228	1,5340	2,2423
VLG1145	1368453041	1368453892	200	1,5545	2,1880
RYR6396	1368452980	1368453905	179	1,4435	2,1189
RYR6302	1368453224	1368454000	197	1,3302	2,0904
RYR6341	1368453285	1368454086	152	0,8928	1,4939
RYR6382	1368453345	1368454114	196	1,1410	1,9550
RYR6332	1368453582	1368454339	180	0,9720	1,7961
EZS1402	1368453530	1368454323	145	0,7565	1,3653
EZY2760	1368453910	1368454731	172	1,0127	1,7608
EZY3916	1368454083	1368454832	161	0,8448	1,6227

<b>EZY4526</b>	1368453623	1368454960	221	1,2522	2,1783
<b>EZY8572</b>	1368453901	1368454961	236	1,3847	2,3178
<b>EZY7204</b>	1368454550	1368455807	197	0,7140	1,5001

**TABLA 13. RESULTADOS OBTENIDOS PARA LAS AERONAVES**

Los resultados obtenidos y mostrados en la tabla 13 concuerdan con las observaciones realizadas en el periodo de obtención de datos, no obstante observamos que la desviación media del número de paradas es demasiado elevada para la media obtenida. Este hecho está principalmente motivado a que hace falta un estudio con datos fiables sobre el comportamiento que describe la variable número de paradas en referencia al número de aeronaves en zona. En el modelo se propone que la probabilidad de parada se incremente uniformemente según el número de aeronaves en zona. El incremento está definido en el código del programa (actualmente 0.1) así como un número de paradas máximas a realizar por arco. Aparentemente los datos se aproximan a realidad, tanto en tiempo como en número de paradas (datos obtenidos de IATA<sup>3</sup>). No obstante no es posible la validación del modelo sin datos exactos de fuentes fidedignas y no basadas en observaciones.

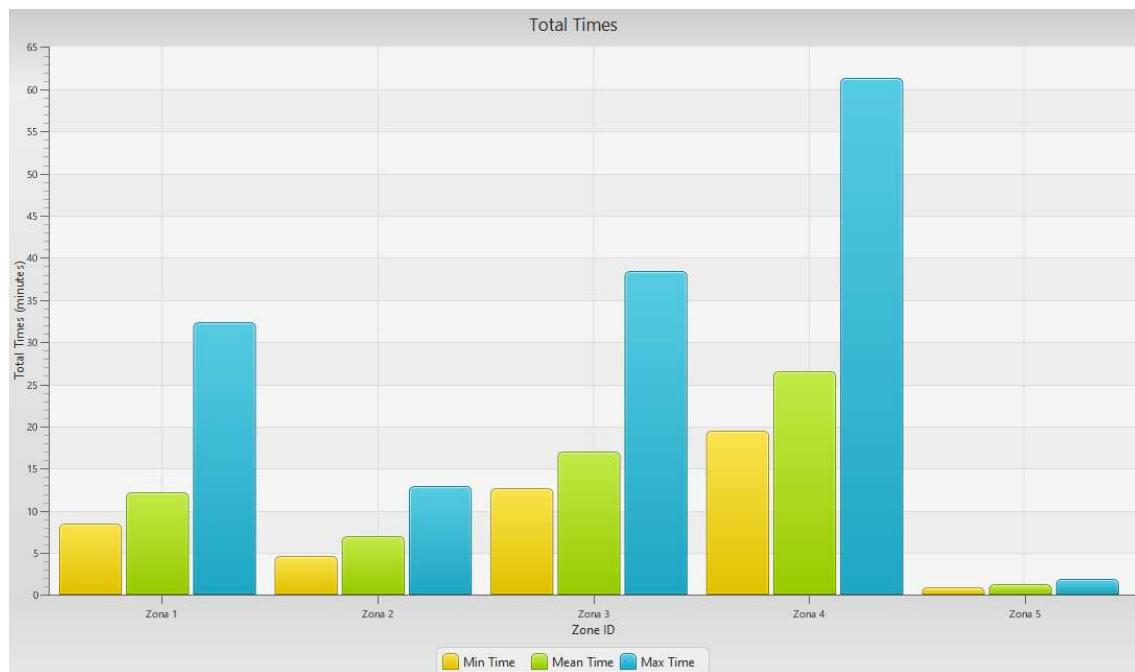
ARC ID	MEAN TIME PER AIRCRAFT (SECONDS)	MEAN STOPS PER AIRCRAFT
<b>QS-LS</b>	624,9890	0,4139
<b>LS-K5</b>	49,6750	0,357442308
<b>R2E-R2M</b>	301,1276	0,07225
<b>R1-R2E</b>	105,2134	0,026
<b>R2W-R3</b>	121,8322	0,067318182
<b>JS-K5</b>	106,5441	0,404583333
<b>R0-R1</b>	84,9651	0,0265
<b>R14-LS</b>	571,4811	0,391638889
<b>M7-K4</b>	183,2053	0,391340909
<b>K4-K1</b>	78,9084	0
<b>R10-R16</b>	326,1145	0,3483625
<b>R2M-R2W</b>	243,6217	0,069475
<b>R16-M7</b>	115,3000	0,421477273
<b>R11-JS</b>	586,4657	0,485375
<b>R3-R4</b>	106,0618	0,047977273
<b>K5-K4</b>	65,3772	0,345684211
<b>R4-R10</b>	189,4537	0,061090909

**TABLA 14. RESULTADOS OBTENIDOS PARA LOS ARCOS**

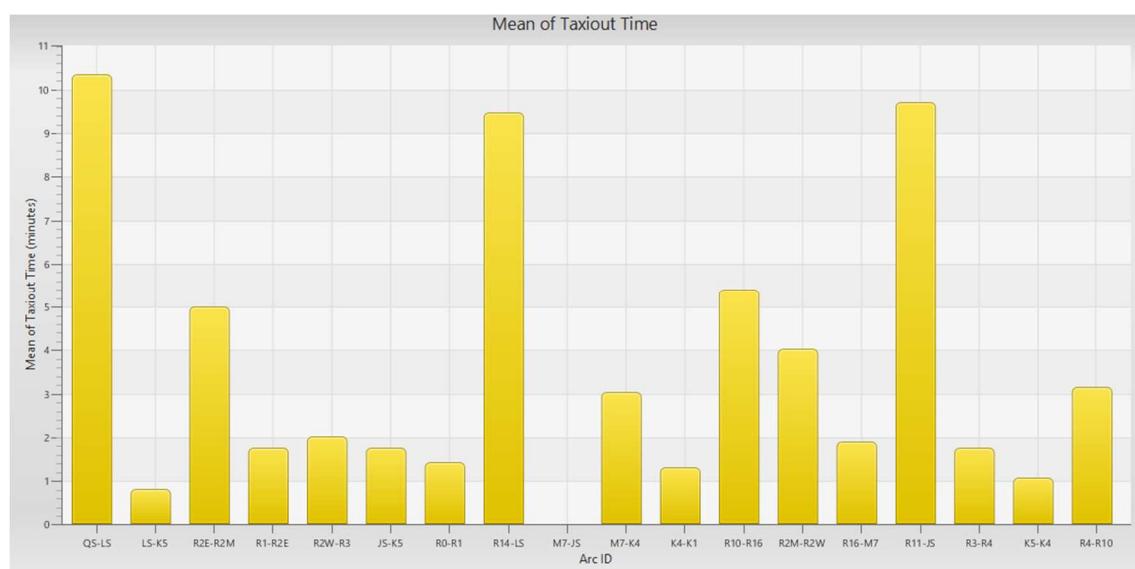
El arco RS-QS aparece como el arco en el cual una aeronave tarda más en recorrer de media para los inputs introducidos. Esto se debe a que es el nodo fuente con mayor recorrido y situado en la zona donde según la simulación es la más congestionada (ver Ilustración 15). Debido a que en los tiempos de taxi están incluidos los tiempos de pushback los arcos con nodos fuente de origen en zonas más congestionadas normalmente serán los que tengan tiempos medios de recorrido por aeronave más elevados. La tabla arcos nos corrobora esta información.

<sup>3</sup> IATA Winter Season 2011-2012.

Finalmente el grafico el tiempo medio, máximo y mínimo que una aeronave tarda en recorrer una zona.



**ILUSTRACIÓN 15. TIEMPOS DE RECORRIDO MEDIO POR AERONAVE Y ZONA**



**ILUSTRACIÓN 16. TIEMPOS MEDIO DE AERONAVE POR ARCO**

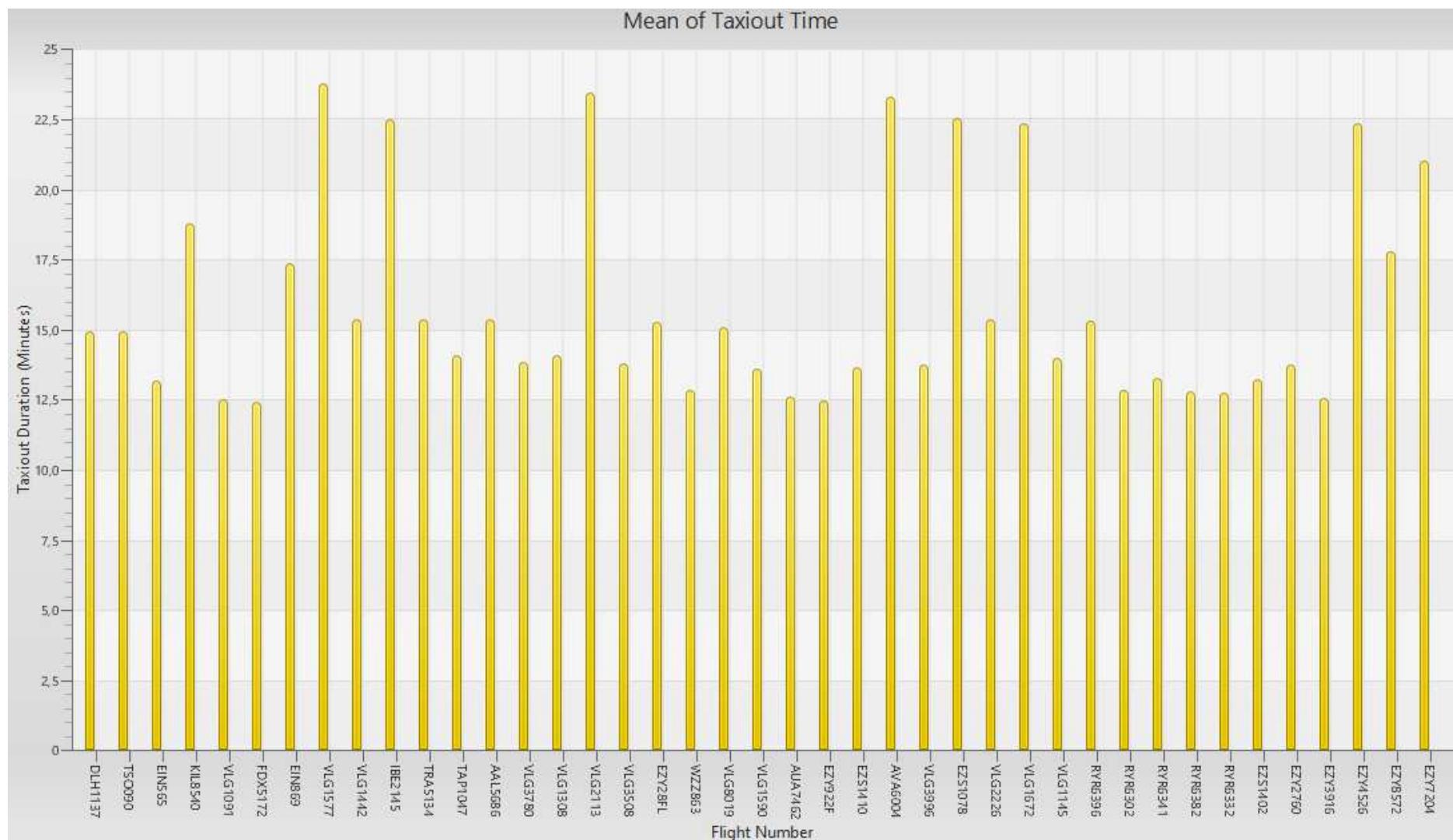


ILUSTRACIÓN 17. TIEMPOS DE TAXI MEDIOS

## BIBLIOGRAFÍA

Thuan V. Truong. The Distribution Function of Airport Taxi-Out Times and Selected Applications. *Journal of the Transportation Research Forum*, Vol. 50, No. 2 (Summer 2011), pp. 33-44.

Wayne W. Cooper, Jr., Dr. Ellen A. Cherniavsky, James S. DeArmon, J. Glenn Foster, Dr. Michael J. Mills, Dr. Satish C. Mohleji and Frank Z. Zhu. *Determination of Minimum Push-Back Time Predictability Needed for Near-Term Departure Scheduling using DEPARTS*. Center for Advanced Aviation System Development.

Rajesh Ganesan, Lance Sherry, Center for Air Transportation Systems Research, George. A *STOCHASTIC DYNAMIC PROGRAMMING APPROACH TO TAXI-OUT PREDICTION USING REINFORCEMENT LEARNING*. Mason University, Fairfax, VA, USA.

Harshad Khadikar. *Analysis and Modeling of Airport Surface Operations*. Submitted to the Department of Aeronautics and Astronautics in partial fulfillment of the requirements for the degree of Master of Science in Aeronautics and Astronautics at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY. 2011.

Amal Srivastava. Improving Departure Taxi Time Predictions Using ASDE-X Surveillance Data. Center for Advanced Aviation System Development. 2010.

Jeffrey Legge, Benjamin Levy. *DEPARTURE TAXI TIME PREDICTIONS USING ASDE-X SURVEILLANCE DATA*. Sensis Corporation. 2011.

Husni Idris, John-Paul Clarke, Rani Bhuva and Laura Kang. *Queuing Model for Taxi-Out Time Estimation*. Massachusetts Institute of Technology. 2001.

Sander Niemeijer (KLM) and Gerardo Valé (LVNL). Functional Design of Dynamic Taxi-time Prediction Sub-project of Timeline at Amsterdam Schiphol Airport. 2007.

Laura Kang, John-Paul Clarke. *Taxi Time Prediction for CDM*. Massachusetts Institute of Technology. International Center for Air Transportation.

Arjun Chauhan. *Modeling and Predicting Taxi Times at Airports*. University of South Florida.