Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

TRABAJO DE FIN DE CARRERA

TÍTULO DEL TFC : Estudio de políticas de asignación de retraso para Ground Delay Program

TITULACIÓN: Ingeniería Técnica Aeronáutica, especialidad Aeronavegación

AUTOR: José María Marco Carpena

DIRECTOR: Luis Delgado Muñoz

FECHA: 17 de julio de 2012

Título : Estudio de políticas de asignación de retraso para Ground Delay Program

Autor: José María Marco Carpena

Director: Luis Delgado Muñoz

Fecha: 17 de julio de 2012

Resumen

Un desequilibrio entre la demanda para aterrizar de tráfico aéreo y la capacidad que puede soportar un aeropuerto para este objetivo puede provocar la congestión del espacio aéreo con lo que supone un mayor peligro, costes económicos y retrasos.
Cuando se prevé que puede haber una situación de estas por causas meteorológicas, obras o algún evento que haga que aumente la demanda puntualmente, la manera de prevenir este problema es utilizando un Ground Delay Program o GDP asignando retrasos a los vuelos para que los hagan en tierra antes de despegar.
La política actual de asignación de retrasos RBS ha sido estudiada y han surgido nuevas propuestas de políticas de asignación como son RBPax, RBD, RBID o RBPAc. Lo que se intenta con estos nuevos modelos es reducir el retraso que se produce teniendo en cuenta diferentes aspectos.
En este trabajo se describen estas políticas nuevas y cómo funcionan. Además, tras el desarrollo de una aplicación informática se simula un GDP con datos de tráfico real variando la tipología de políticas para ver los resultados. Se llega a la conclusión que por lo que respecta a equidad entre vuelos, RBS es la mejor política posible pero no es óptima por lo que al pasajero. Para este, la mejor política aplicable es RBPax. También se obtiene que RBD es la política que, para el caso estudiado, peor resultados dio y que RBID, política propuesta por investigadores de la EETAC - UPC tiene mejores resultados que RBS por lo que respecta al retraso de los pasajeros.
Por otra parte, se puede ver reflejado el trabajo realizado antes y durante el desarrollo de la aplicación con documentación como requisitos de sistema, diagramas de casos de uso, clases o secuencia.

Title : Study of allocation delay politicies for Ground Delay Program

Author: José María Marco Carpena

Director: Luis Delgado Muñoz

Date: 17 de julio de 2012

Overview

An imbalance between demand of flights for arrival and capacity of the airport could generate a congestion of the airborne that this is translating an economics repercussion, insecurity of the air traffic and delays. When this is preview, Ground Delay Program GDP is activated and it allocated delay for flights that will be affected. They should do this delay at the original airport before departure.
The current policy delays allocation of RBS has been studied and there are new allocation policy proposals such as RBPax, RBPAc, RBD or RBID. The intention with these new models is reduce the delay that can occur taking into account different aspects.
This paper describes these new policies and how they work. In addition, following the development of a software application by the student, it is simulate a GDP with real traffic data varying the type of policies to see the results.
It concludes that with respect to equality between flights, RBS is the best policy possible but is not optimal for passenger's delay. The best policy to have less passenger's delay is RBPax. Furthermore, for the case that it was studied, RBD is the policy that had worse results and RBID, policy proposed by researchers EETAC - UPC , is better than RBS respect to delayed that passengers suffers.
Moreover, it can see reflected the work done before and during the application development whit documentation like system requirements, diagrams of use case, class or sequence.

A mis padres, José M. Marco y Puri Carpena, por apoyarme y darme ánimos durante la realización de este trabajo y toda la carrera a pesar de la distancia que nos separa; a mi tutor de TFC, Luis Delgado, por haberme dirigido durante el trabajo, hacer que me esforzase para obtener un buen resultado, darme consejos y soporte; a María, por animarme y haber colaborado con la corrección de faltas; y a todas aquellas personas cercanas que de alguna manera me han querido ayudar emocionalmente.

ÍNDICE GENERAL

INTRODUCCIÓN 1

GLOSARIO Y ABREVIATURAS 3

CAPÍTULO 1. Ground Delay Program 5

1.1. Introducción 5

1.2. Definición de GDP 6

1.3. Parámetros que definen un GDP 8

1.4. Funcionamiento de un GDP 9

1.5. Tipos de políticas 10

1.5.1. Ration By Schedule (RBS) 10

1.5.2. Ration By Passangers (RPax) 11

1.5.3. Ration By Aircraft size (RBAc) 11

1.5.4. Ration By Distance (RBD) 11

1.5.5. Ration By Inverse Distance (RBID) 12

1.5.6. Ejemplo gráfico 12

CAPÍTULO 2. Desarrollo del software 15

2.1. Introducción 15

2.2. Ciclo de vida 16

2.3. Arquitectura del programa 17

2.4. Algoritmos de cálculo 19

2.5. Posibles ampliaciones y mejoras 24

CAPÍTULO 3. Estudio de casos, simulaciones y resultados 25

3.1. Políticas individuales 29

3.1.1. RBS 29

3.1.2. RBPax 30

3.1.3. RBAc 30

3.1.4. RBD 31

3.1.5. RBID . . . 32
3.1.6. Comparativas y conclusiones . . . 32

3.2. Políticas combinadas . . . 33
3.2.1. RBS - RBPax . . . 33
3.2.2. RBPax - RBS . . . 34
3.2.3. Comparativas y conclusiones . . . 35

CAPÍTULO 4. Gestión del proyecto . . . 37

4.1. Planificación del proyecto . . . 37

4.2. Programas utilizados a lo largo del TFC . . . 39

CAPÍTULO 5. Conclusiones . . . 41

BIBLIOGRAFÍA . . . 43

APÉNDICE A. Requisitos . . . 1

A.1. Introducción . . . 1
A.1.1. Propósito . . . 1
A.1.2. Ámbito del sistema . . . 1
A.1.3. Definiciones, Acrónimos y Abreviaturas . . . 1
A.1.4. Referencias . . . 1
A.1.5. Visión general del documento . . . 2

A.2. Descripcción general . . . 3
A.2.1. Perspectiva del producto . . . 3
A.2.2. Funciones del producto . . . 3
A.2.3. Características de usuario . . . 3
A.2.4. Restricciones . . . 4
A.2.5. Suposiciones y dependencias . . . 4
A.2.6. Requisitos futuros . . . 4

A.3. Requisitos específicos . . . 4
A.3.1. Requisitos de Interfaz Externa . . . 5
A.3.2. Requisitos Funcionales . . . 11
A.3.3. Requisitos de Rendimiento . . . 16
A.3.4. Requisitos de Diseño . . . 16
A.3.5. Requisitos del Sistema . . . 16
A.3.6. Otros Requisitos . . . 19

APÉNDICE B. Diagramas UML **21**

B.1. Introducción **21**
B.1.1. Propósito 21
B.1.2. Definiciones, Acrónimos y Abreviaturas 21
B.1.3. Referencias 21
B.1.4. Visión general del documento 21

B.2. Diagramas de casos de uso **22**
B.2.1. Casos de uso de alto nivel 22
B.2.2. Casos de uso de bajo nivel 23

B.3. Diagramas de clases **24**
B.3.1. Interfaz Hombre Máquina 24
B.3.2. IHM - Domininio 25
B.3.3. Dominio 26
B.3.4. Dominio - Base de datos 27
B.3.5. Base de datos 27

B.4. Diagramas de secuencia **28**
B.4.1. Cargar Tráfico 28
B.4.2. Crear GDP 30
B.4.3. Crear política 31
B.4.4. Guardar datos 32
B.4.5. Asignar slots despues de aplicar GDP 34

APÉNDICE C. Guía de uso de *ACREL* **35**

C.1. Introducción **35**
C.1.1. Propósito 35
C.1.2. Definiciones, Acrónimos y Abreviaturas 35
C.1.3. Referencias 35
C.1.4. Visión general del documento 36

C.2. Instalación de la aplicación y archivos adjuntos **36**
C.2.1. Instalación para ejecución directa 36
C.2.2. Instalación de la aplicación para ejecucion desde editor *Java* 37
C.2.3. Archivo de tráfico 37

C.3. ¿Cómo ejecutar la aplicación? **38**

C.4. Descripción de línea de ejecución **39**

C.5. Visión rápida de las opciones del menú **41**

C.6. Descripción rápida de los procesos del cálculo principal 44
C.7. Posibles fallos durante la ejecución y sus soluciones 44

APÉNDICE D. Áctas y resumen de reuniones 49
D.1. Reunión 1 de TFC 49
D.2. Reunión 2 de TFC 51
D.3. Reunión 3 de TFC 52
D.4. Reunión 4 de TFC 53
D.5. Reunión 5 de TFC 54
D.6. Reunión 6 de TFC 56
D.7. Reunión 7 de TFC 57
D.8. Reunión 8 de TFC 58
D.9. Reunión 9 de TFC 60
D.10. Reunión 10 de TFC 62
D.11. Reunión 11 de TFC 64
D.12. Reunión 12 de TFC 65
D.13. Reunión 13 de TFC 67
D.14. Reunión 14 de TFC 69
D.15. Reunión 15 de TFC 71
D.16. Reunión 16 de TFC 73
D.17. Reunión 17 de TFC 75
D.18. Reunión 18 de TFC 77
D.19. Reunión 19 de TFC 79
D.20. Reunión 20 de TFC 81

ÍNDICE DE FIGURAS

1.1 Previsión de demanda superior a la capacidad regulada. [1] 6
1.2 Regulación del tráfico durante el GDP. [1] 7
1.3 Curvas demanda - capacidad. 8
1.4 Resumen gráfico del funcionamiento de activación de un GDP 10
1.5 Ejemplo de asignacion de slots. Caso inicial. 12
1.6 Ejemplo de asignacion de slots. Caso final tras usar RBS. 13
1.7 Ejemplo de asignacion de slots. Caso final tras usar RBAc. 13

2.1 Ciclo de vida en V 17
2.2 Esquema de programación por capas 18
2.3 Gráfica demanda vs capacidad puntos del GDP 21
2.4 Clasificación de vuelos no afectados, excluidos y controlados 21

3.1 Definicion y tiempos del GDP 26
3.2 Curvas de demanda y capacidad visión de tráfico total analizado 27
3.3 Curvas de demanda y capacidad enfocado en la región del GDP. 27
3.4 Vuelos acumulados cada hora. Todo el tráfico 28
3.5 Vuelos acumulados cada hora. Primera parte del tráfico. 28

4.1 Planificación inicial del proyecto. Diagrama de Gantt 37
4.2 Planificación final del proyecto. Diagrama de Gantt 38
4.3 Porcentajes mensuales de horas dedicadas durante el TFC 39

A.1 Ejemplo de ventana principal 8
A.2 Ejemplo ventana Gestionar GDP. 9

B.1 Casos de uso alto nivel 22
B.2 Gestionar Trafico 23
B.3 Gestionar GDP 23
B.4 Gestionar resultados 24
B.5 IHM 24
B.6 Transacciones IHM - Dominio 25
B.7 Dominio 26
B.8 Transacciones Dominio - Base de datos 27
B.9 Base de datos 27
B.10Cargar tráfico (parte 1) 28
B.11Cargar tráfico (parte2) 29
B.12Definir y crear GDP 30
B.13Crear política 31
B.14Guardar datos (parte1) 32
B.15Guardar datos (parte2) 33
B.16Asignar slot despues de aplicar GDP 34

C.1 Ventana de cmd.exe antes de ejecutar ACREL 38
C.2 Menú principal de ACREL en de cmd.exe 39
C.3 Gráfico de opción 7 del menú principal. 42

C.4 Gráfico de opción 8 del menú principal. 43

ÍNDICE DE CUADROS

3.1 Horario local del *Cluster 1* de GDP de EWR. [14] 25
3.2 Horario propuesto en UTC para la aplicación de GDP en EWR. 25
3.3 Definición de la política RBS. 29
3.4 Resultados obtenidos con RBS de retrasos del GDP de vuelos controlados. . . 29
3.5 Definición de la política RBPax. 30
3.6 Resultados obtenidos con RBPax de retrasos del GDP de vuelos controlados. . 30
3.7 Definición de la política RBAc. 30
3.8 Resultados obtenidos con RBAc de retrasos del GDP de vuelos controlados. . 31
3.9 Definición de la política RBD. 31
3.10 Resultados obtenidos con RBD de retrasos del GDP de vuelos controlados. . . 31
3.11 Definición de la política RBID. 32
3.12 Resultados obtenidos con RBID de retrasos del GDP de vuelos controlados. . . 32
3.13 Resumen de resultados globales de retrasos del GDP con vuelos controlados. . 32
3.14 Resumen de resultados de retrasos del GDP con vuelos controlados. 33
3.15 Definición de las políticas combinadas RBS - RBPax. 34
3.16 Resultados de retrasos obtenidos tras combinar RBS - RBPax del GDP de vuelos controlados. 34
3.17 Definición de las políticas combinadas RBS - RBPax. 34
3.18 Resultados de retrasos obtenidos tras combinar RBPax - RBS del GDP de vuelos controlados. 35
3.19 Resultados de retrasos GDP con combinación de politicas. 35

A.1 Ejemplo de requisito 2

INTRODUCCIÓN

El tráfico aéreo en el mundo ha ido creciendo de una forma acelerada durante los últimos 20 años. En el año 2000, la *FAA (Federal Aviation Administration)* de los Estados Unidos predijo que habría un incremento del 3 al 5% de tráfico al año durante mínimo los 15 años siguientes[2]. Sin embargo, el crecimiento de las infraestructuras para soportar este aumento de vuelos no ha sido el mismo.

Esto repercute en que la demanda por parte de las compañías aéreas, y la capacidad que pueden ofrecer las infraestructuras están muy próximas. Cuando la demanda es superior a la capacidad comienzan a haber problemas y en muchos casos esto se traduce en retrasos.

Desde 1981 [1], en los aeropuertos de Estados Unidos, cuando se prevee que se va a producir un desequilibrio de estas características se activa lo que se denomina *Ground Delay Program* o GDP. Este desajuste puede ser debido a mal tiempo, obras o porque se produce algún evento extraordinario que hace que la programación de vuelos, durante un cierto tiempo, sea mayor a la que hay normalmente.

Lo que hace el GDP es asignar retrasos a los vuelos que tienen programado llegar al aeropuerto cuando la demanda es superior a la capacidad para que los hagan en tierra. Si esto no fuera así, este retraso se produciría de igual manera en el aire con lo que supondría un mayor peligro y coste.

Actualmente, la manera de asignar los retrasos a estas aeronaves, es usando la política de asinación denominada *Ration By Schedul* que lo que hace es asignar nuevos slots de llegada según el orden en el que estaban programados los vuelos desde un inicio.

A lo largo del tiempo, muchos investigadores han ido proponiendo y haciendo estudios de nuevas políticas de asignación de retrasos donde comparaban la equidad entre aerolíneas y pasajeros.

Este trabajo se basa en el estudio de estas políticas de asignación de retrasos. Para ello, se describe cómo funciona un GDP y 5 de las políticas que exiten acutalmente.

Además, para poder realizar este estudio se ha desarrollado una aplicación informática para calcular retrasos donde se puede elegir el tipo de política que se va a aplicar a un tráfico determinadod durante un GDP de forma individual o combinada. Cómo ha sido creado y cómo funciona se explica en uno de los capítulos.

Posteriormente, utilizando el software desarrollado, se simula un GDP aplicando diferentes políticas a una base de datos de tráfico real para obtener así datos de retrasos y poder hacer unas comparativas entre políticas.

Finalmente, se puede encontrar un capítulo en el cual se describe la gestión del proyecto seguido de unas conclusiones de todo el trabajo.

GLOSARIO Y ABREVIATURAS

ACREL: Aplicación de Cálculo de Retrasos En Llegadas
ATC: Air Traffic Control
ATFM: Air Traffic Flow Management
ATM: Air traffic Management
CTA: Controlled Time of Arrival
CTD: Controlled Time of Departure
ETA: Estimated Time of Arrival
ETD: Estimated Time of Departure
FAA: Federal Aviation Administration
GDP: Ground Delay Program
IEEE: Institute of Electrical and Electronics Engineers
RBAc: Ratio By Aircraft
RBD: Ratio By Distance
RBID: Ratio By Inverse Distance
RBPax: Ratio By Passengers
RBS: Ratio By Schedule
RD: Requisitos de Diseño
RF: Requisitos Funcionales
RIE: Requisitos de Interfaz Externa
RIU: Requisitos Interfaz de Usuario
RR: Requisitos de Rendimiento
RSD: Requisitos de Sistema - Disponibilidad
RSF: Requisitos de Sistema - Fiabilidad
RSM: Requisitos de Sistema - Mantenimiento
RSP: Requisitos de Sistema - Portabilidad
RSS: Requisitos de Sistema - Seguridad
Slot: Intervalo de tiempo perteneciente a una aeronave para realizar una maniobra determinada
SRS: Specifaction Requirement Software
TFC: Trabajo Final de Carrera
TFM: Traffic Flow Management
Unix Time: Se define como la cantidad de segundos transcurridos desde la medianoche UTC del 1 de enero de 1970 [3]
UTC: Tiempo Universal Coordinado.
Vuelo controlado: Vuelo que tiene una hora de llegada programada definida durante la duración del GDP y al que se le aplica el retraso antes de despegar para que no lo realice en vuelo.
Vuelo excluido: Vuelo que tiene una hora de llegada programada definida durante la duración del GDP y que ya sea porque el vuelo estába volando cuando se definió el GDP, se trate de un vuelo internacional o la distancia del aeropuerto de origen al de destino es maryor al radio de acción del GDP, no se le aplica retraso en tierra y tiene preferencia sobre otros vuelos.
Vuelo no afectado: Vuelo que tiene una hora de llegada programada fuera de la duración del GDP.

CAPÍTULO 1. GROUND DELAY PROGRAM

1.1. Introducción

En las últimas décadas, el tráfico aéreo ha ido creciendo de una forma considerable. La velocidad de crecimiento de la demanda aérea no ha ido ligada con el aumento de las capacidades de las infraestructuras aéreas teniendo éstas un crecimiento mucho más lento. Como consecuencia, los sistemas aéreos europeos y nacional de los Estados Unidos se han ido y siguen congestionando.[4][1]

Un ejemplo del lento crecimiento de las infraestructura es que solo los aeropuertos de *Denver Internacional (DEN), Dallas Fort Worth (DFW) y George Bush Intercontinental (IAH)* son los únicos aeropuertos abiertos en los últimos 40 años en los Estados Unidos[1]. [1]

Y siguiendo con el ejemplo del sistema aeroespacial de los Estados Unidos, la *FAA (Federal Aviation Administration)* predijo en el año 2000 un incremento del 3 al 5% de tráfico al año en los próximos 15 años. [2]

Por ello, es normal que en algunos casos, la demanda de tráfico sea muy próxima a la capacidad de los sistemas y comiencen a producirse problemas de congestión provocando retrasos si ocurre cualquier imprevisto no programado.

Ejemplos de aeropuertos con problemas de congestión pueden ser el de *Newark (EWR)* o *Chicago O'Hare (ORD)*. Estos, como algunos otros aeropuertos del mundo, no pueden expandirse construyendo nuevas instalaciones para aumentar su capacidad debido a que no disponen de terreno o están restringidos por motivos medioambientales.[1]

Por tanto, debido al rápido crecimiento de tráfico aéreo respecto al de infraestructuras, los espacios aéreos deben ser gestionados de manera en la que se mantenga la seguridad y de la forma más eficiente posible. De esto se encarga *Air traffic Management (ATM)*.

El ATM se divide a su vez en *Air Traffic Control(ATC)* que se encargan de mantener una separación segura entre aeronaves y por otro lado *Traffic Flow Management (TFM)*.[2]

Cuando se produce un desequilibrio entre la demanda y capacidad de los aeropuertos, donde la demanda es mayor a la capacidad ya sea debido a una sobre-programación de vuelos, hay mal tiempo o porque hay baja visibilidad, entra en fucionamiento el TFM y una de las 'herramientas' de las que hace uso para solucionar este problema es el *Ground Delay Program (GDP)*.

Los primeros GDPs los aplicó la FAA por causas meteorológicas adversas y tras la huelga de controladores que hubo en 1981.[1]

[1] Datos del 2008

1.2. Definición de GDP

El *Ground Delay Program* es un mecanismo que se usa para disminuir el ratio de vuelos que llegan a un aeropuerto cuando la demanda del aeropuerto excede de la capacidad del mismo durante un período de tiempo[1].

Esta capacidad del aeropuerto puede restringirse o disminuir respecto a la demanda por motivos como mal tiempo, obras en el aeropuerto o eventos públicos en fechas concretas que hacen que el tráfico programado aumente puntualmente respecto a la programación diaria normal.

El retraso en tierra es el resultado de hacer que la hora de despegue programada de los vuelos que se dirigen a un aeropuerto donde se aplica el GDP sea retrasada. El motivo de este retraso en la hora de salida del vuelo es hacer que la espera no se produzca en el aire en un hipódromo de llegadas del aeropuerto de destino.

De esta manera se consigue que el retraso se realice de una forma más segura y barata en el aeropuerto de origen ya que la aeronave permanecería parada y sin consumir hasta la nueva hora de salida.[4][5]

A parte de regulizar el flujo de llegada durante el tiempo en el que la capacidad del aeropuerto está regularizada por debajo de la demanda del tráfico, también se sigue regularizando durante un tiempo después de que se haya cancelado esta regulación ya que si no, todos esos vuelos que habían sido retrasados durante la regularización, llegarían conjunta tras aumentar la capacidad produciéndose así o continuándo con una mayor demanda a pesar de que el ratio de aterrizaje/hora ha aumentado.

Esto último se muestra en las figuras 1.1 y 1.2

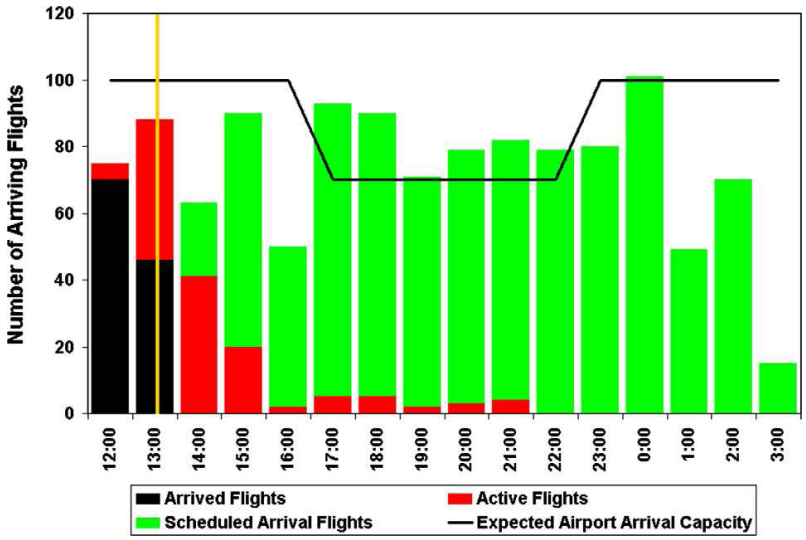

Figura 1.1: Previsión de demanda superior a la capacidad regulada. [1]

En la figura 1.1 la curva de color negro representa la capacidad del aeropuerto mientras que las barras de color verde representan la demanda de este.

Como se observa, durante la disminución de capacidad, la demanda queda por encima. Todos estos vuelos que llegan de más en este tiempo, son los vuelos que serán regulados.

Además, la linea vertical amarilla situada a las 13:00 h en este ejemplo en particular, representa la hora donde se prevee que se va a producir este desequilibrio y es cuando se da paso a la activación del GDP.

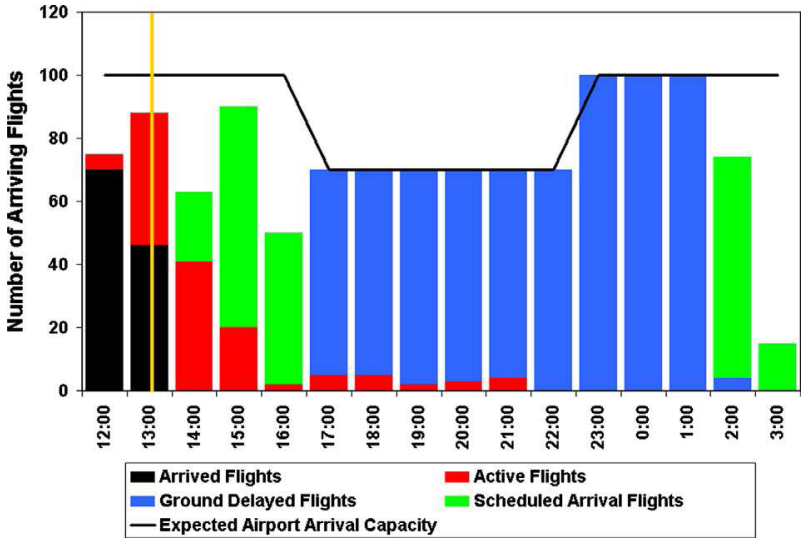

Figura 1.2: Regulación del tráfico durante el GDP. [1]

En la figura 1.2 se observa cómo quedaría el tráfico durante la regularización del GDP. Las barras azules corresponden a los vuelos controlados.

Se observa, que tras volver a la capacidad nominal, parte del tráfico se mantiene regularizado. Esto es por el motivo que se ha comentado anteriormente.

Si se grafica la demanda del aeropuerto y la capacidad de este a lo largo del tiempo, se puede obtener algo parecido a la figura 1.3 donde se muestra en color rojo la demanda, y el color azul la restriccion del GDP.

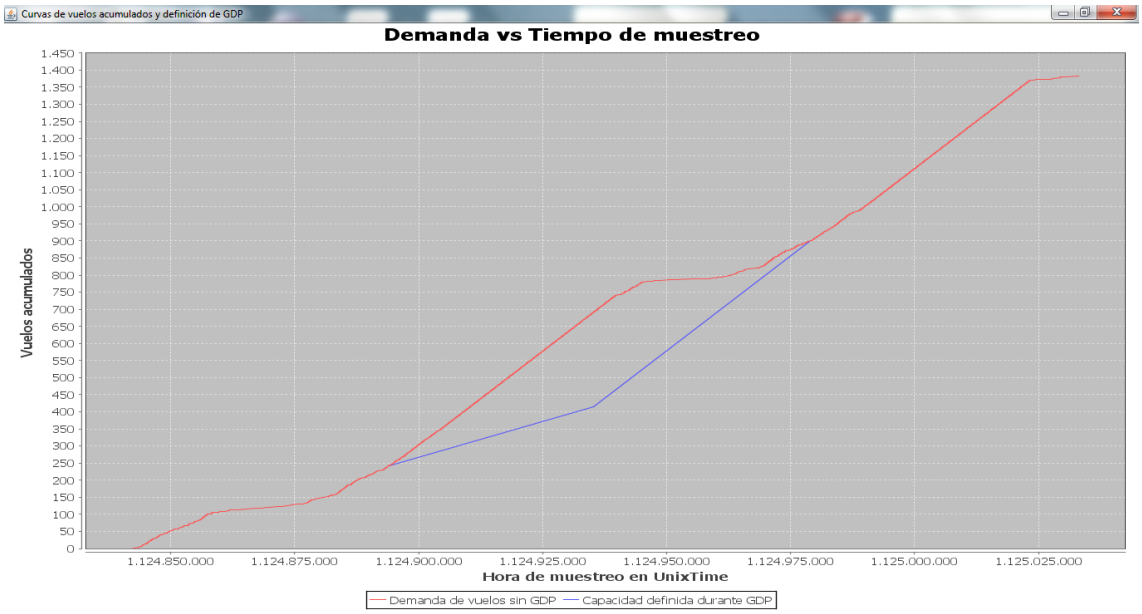

Figura 1.3: Curvas demanda - capacidad.

La primera parte de la curva azul, la de menor pendiente, representa la capacidad regulada, es decir, donde el ratio aeronaves/hora es menor.

El área que hay entre estas dos curvas representa el tiempo mínimo de retraso teórico que se puede otorgar al tráfico de manera global durante la regularización.

1.3. Parámetros que definen un GDP

Cuando se decide que es necesario apligar un GDP, se definen 3 parámetros fundamentales relacionados con él.[1]

- ***Horario de inicio y finalización del GDP***

  Estos definen cuándo empieza y cuándo acaban las restricciones, es decir, fuera de este horario todo funciona con normalidad y no se aplican retrasos en el aeropuerto de salida. Si una aeronave llega durante este horario, queda potencialmente[1] afectado por el GDP.

  La manera de determinar estos horarios son teniendo en cuenta la demanda programada y el pronóstico de las condiciones meteorológicas.

- ***alcance* o *radio de acción.***

  Esto indentifica que los vuelos de los aeropuertos de origen que estén dentro de este radio de acción definido desde el aeropuerto de destino, quedan afectados y por tanto controlados por el GDP. Fuera de este alcance los vuelos no son afectados.

[1] Si cuando se define el GDP la aeronave ya estaba volando y su hora de llegada cae dentro de la definición del GDP, este vuelo queda exlcuido del programa y aterrizará, si es posible, según llegue.

- ***Program Airport Acceptance Rate (PAAR)* o capacidad regulada.**

  Este parámentro nos indica el número de aeronaves que pueden aterrizar de forma segura en una hora mientras esté activo el GDP.[1]

1.4. Funcionamiento de un GDP

Los especialistas del ATC están continuamente monitorizando la demanda y capacidad de los aeropuertos. Cuando detectan que se va a producir un desequilibrio de estos parámetros por cualquier motivo, avisan a las aerolíneas de que se va activar el programa. Las aerolíneas, tras comprobar cómo puede afectar el GDP a sus vuelos pueden cancelarlos.

Entonces, los encargados de gestionar el GDP re-evalúan la demanada y aplican algoritmos de asignación de slots donde se les proporciona a las aerolíneas una nueva hora de llegada o *Controlled Time of Arrival (CTA)* correspondiente al slot de llegada y una nueva hora de salida o *Controlled Time of Departure (CTD)*.

De nuevo, las aerolíneas pueden cancelar sus vuelos o en algunos casos intercambiar sus slots entre vuelos de la misma compañía. Tras esto, se recalcula la asignación de slots y se activa el GDP.

No obstante, el GDP no tiene porqué terminar a la hora definida, sino cuando la capacidad del aeropuerto es superior a la demanda ya sea porque se han cancelado otros vuelos o porque las condiciones meteorológicas mejoran.

Los slot de llegada durante un GDP son intervalos necesarios para alcanzar una capacidad regulada o PAAR. Es decir, si la capacidad regulada es de 60 aeronaves por hora, quiere decir que en cada minuto (la duración de un slot) podrá aterrizar un vuelo siendo estos slots uniformemente repartidos en el tiempo.

Actualmente, la forma de asignación de slot durante un GDP se basa en ‘primero programado - primer servido’, es decir, los slots son asignados según la programación original de llegada que se tenía antes de que se activara el GDP. Este tipo de política se llama *’Ration by schedule’ (RBS)*.[1]

Todo este proceso se puede ver de forma gráfica en la figura 1.4.

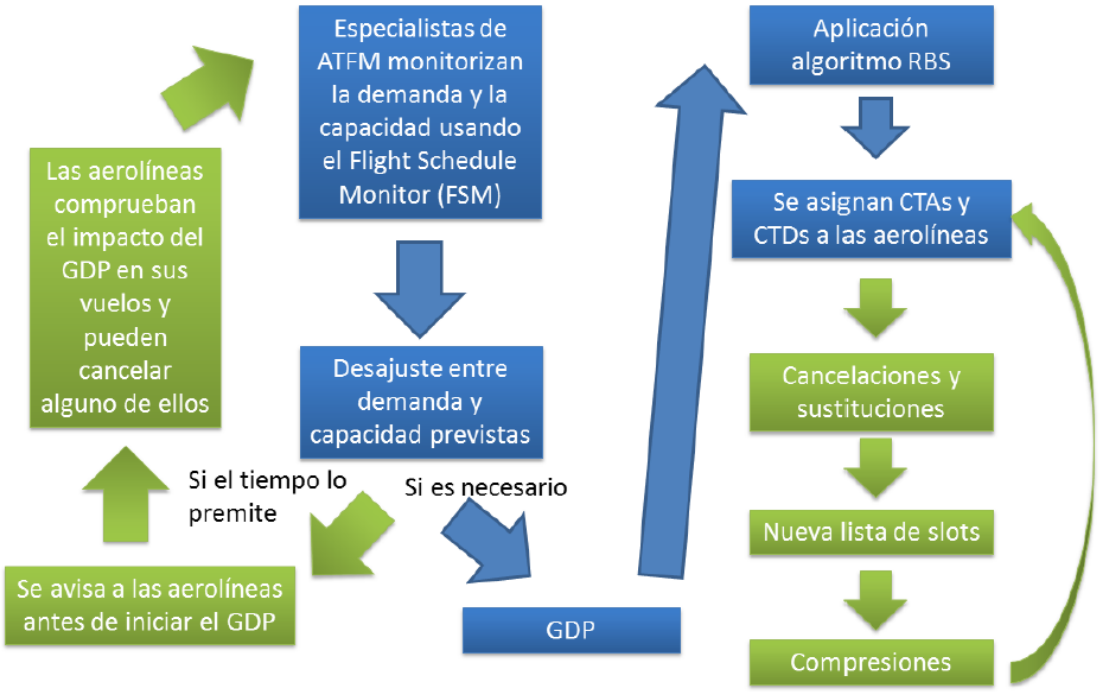

Figura 1.4: Resumen gráfico del funcionamiento de activación de un GDP

No obstante, se han hecho estudios de este tipo de asignación de slots y se han propuesto otras políticas dado que RBS puede llegar a ser 'injusta' o poco eficiente en el sentido de que da preferencia a las aerolíneas y no tiene en cuenta el retraso de los pasajeros.

1.5. Tipos de políticas

El tipo de políticas de asignación de retrasos que actualente se usa es RBS. Sin embargo, hay numerosos estudios donde se proponen otro tipo de políticas que tienen en cuenta otros parámentros como por ejemplo, el retraso global que pueden sufrir los pasajeros.

Entre estas política se encuentran Ration By Passengers, Ration By Aircraft size, Ration By Distans o Ration By Inverse Distance. A continuación se describen en qué consisten cada una de ellas.

1.5.1. Ration By Schedule (RBS)

Como ya se citó en el apartado 1.4., esta política asigna los slots a los vuelos afectados por el GDP en el orden en el que estaban programadas las llegadas. Las aeronaves que antes llegan, antes son asignadas a un slot. En el caso de que la programación de llegada de dos vuelos coincida en un mismo slot, tendrá preferencia aquel con la hora estimada de llegada *Estimated Time of Arrival (ETA)* más temprana. Si la ETA es la misma para los

dos vuelos, se asigna el slot de forma aleatoria a uno de ellos, retrasando al siguiente slot el otro vuelo.

El retraso que se optimiza con esta política es el relacionado con las compañías, es decir, con cada vuelo. En contraposición, el retraso de los pasajeros se ve perjudicado.

1.5.2. Ration By Passangers (RPax)

La asignación por pasajeros tiene en cuenta la cantidad de pasajeros que lleva el vuelo a la hora de asignar los slots. Tendrán preferencia aquellos vuelos con un número mayor de pasajeros. En el caso de que dos aeronaves coincidan en la llegada del slot y tienen el mismo número de pasajeros, se asignará la preferencia a la aeronave que tenga una ETA más temprana. Si además sucede que ambos vuelos tienen la misma hora de llegada, se asignará el slot de forma aleatoria.

Por tanto, esta política optimiza el retraso que sufren los pasajeros a cambio de perjudicar los vuelos con pocos pasajeros y por tanto, la equidad de cada vuelo ya que se modifica el orden de llegada que se tenía programado antes de que se aplicaran las restricciones.

1.5.3. Ration By Aircraft size (RBAc)

En este caso, la asignación de slots viene condicionada por el tamaño de las aeronaves. Se distinguen tre tipos *Heavy*, *Large* y *Small*. Las aeronaves de mayor tamaño tienen preferencia sobre las de menor tamaño. Es decir, se asignará primero el slot a las aeronves Heavy, después a las Large y finalmente a las Small. En el caso de que dos aeronaves coincidan en un slot y sean de la misma categoría, tendrá preferencia aquella con una ETA más temprana como pasa con RBS.

Esta política busca un objetivo similar al de la política anterior aunque no tiene en cuenta el factor de ocupación y se mira más por el tamaño de la aeronave.

1.5.4. Ration By Distance (RBD)

Esta política tiene en cuenta la distancia recorrida por la aeronave. Tendrá preferencia aquella que venga de más lejos. En el caso de que coincidan las distancias, se hará como en los casos anteriores y se mirará la ETA asignando el slot a aquella que tenía la hora programada de llegada más temprana. En caso de coincidir, se asigna de manera aleatoria.

Lo que se intenta con esta política es reducir los retrasos de los vuelos que provienen de más lejos.

1.5.5. Ration By Inverse Distance (RBID)

La asignación por distancia inversa justo es la contraria a RBD y asigna los slots dando la preferencia a aquellos vuelos que proceden de otros aeropuertos más cercanos. Como en todos los demás casos, si sucede que la distancia es la misma para dos vuelos que desean llegar en un slot, tendrá preferencia aquella con una ETA más temprana y si ambas coiciden, la asignación del slot se hace de forma aleatoria a cualquiera de los dos vuelos.

Esta política tiene el objetivo justo contario a RBD, es decir, se mejora los retrasos de los vuelos que provienen de más cerca.

1.5.6. Ejemplo gráfico

En las siguientes figuras se explica de una forma gráfica cómo funciona la asignación de slots. Cada cuadadrito representa un slot de 10 minutos. El avión A llega a las 17:03. Por

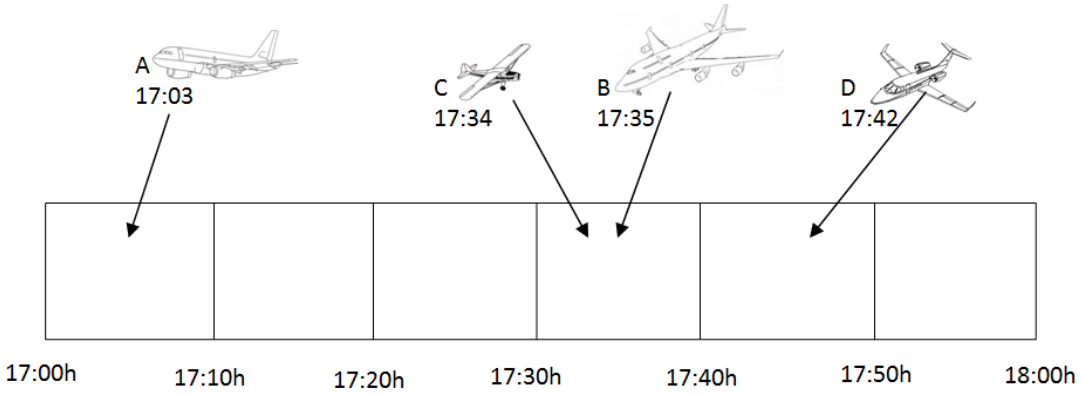

Figura 1.5: Ejemplo de asignacion de slots. Caso inicial.

tanto se le asignará el primer slot que está definido entre las 17:00 y las 17:10h. Esto es así ya que su ETA si su ETA hubiera sido las 17:15, se le hubiera asignado el siguiente slot.

Por otro lado, el avión B y el avión C llegan en el mismo slot. Aquí se producirá un conflicto y tendrán que entrar en juego las políticas de asignación de preferencia.

Por último el avión D con una ETA correspondiente a las 17:42 llega en el slot definido entre las 17:40 y las 17:50.

Si se aplicara *RBS* la aeronave C tendría preferencia sobre la aeronave B y esta última tendría que pasar al siguiente slot. Como en el siguiente slot hay otro vuelo que también 'quiere' usar ese periodo de tiempo, se vuelve a aplicar la política de asignación. Como la ETA de la aeronave B es menor que a la de la aeronave D, la aeronave D tiene que retrasarse al siguiente slot que comienza a las 17:50. La aeronave B finalmete es asignado al slot comprendido entre las 17:40 y las 17:50h. La asignación de slots finalmente quedaría utilizando RBS como se muestra en la figura 1.6.

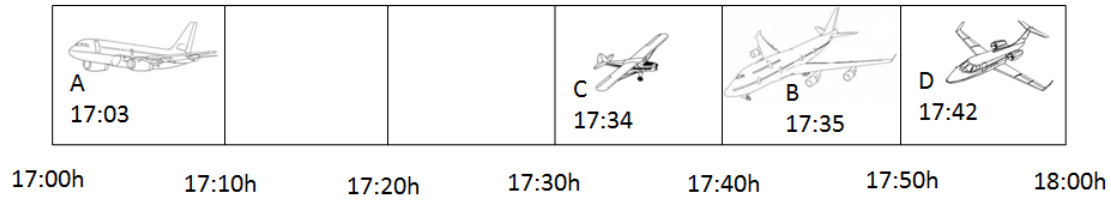

Figura 1.6: Ejemplo de asignacion de slots. Caso final tras usar RBS.

Si por el contrari en vez de usar la política de RBS se hubiera usado RBAc, la aeronave B hubiera tenido preferencia sobre la aeronave C. Esta hubiera pasado al siguiente slot y se hubiera encontrado con la aeronave D que también tiene preferencia sobre la aeronave C. El resultado si se hubiera usado RBAc sería el mostrado en la figura 1.7.

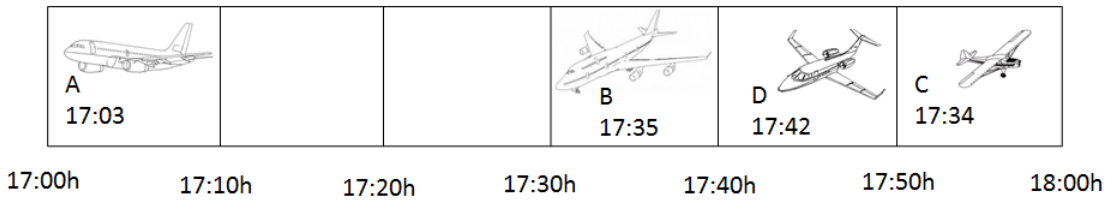

Figura 1.7: Ejemplo de asignacion de slots. Caso final tras usar RBAc.

De estos simples ejemplos se pueden observar los diferentes retrasos que se producen. Para el primer caso donde se aplica RBS, la aeronave B sufre un retraso de 5 min y la aeronave D de 8 min. Son dos aeronaves las que se ven afectadas. Por otra parte, si se utiliza la política de asignación RBAc, únicamente es el vuelo C el que sufre retraso. En concreto 16 minutos.

Con este ejemplo concreto se saca la conclusión de que el retraso global para RBS es de 13 min mientras que para RBAc es de 16 min. Por tanto, si sólo se considera el retraso de las aeronaves, para esta situación expuesta la mejor política tal vez hubiera sido RBS. Sin embargo, si se hubieran tenido en cuenta el número de pasajeros y el retraso que sufren estos (suponiendo que la aeronave B es mucho más grande que la aeronave D y esta es mayor que C y cada una tiene un número de pasajeros proporcional al tamaño de la aeronave) la política que menos retraso acumulado de los pasajeros hubiera tenido hubiera sido RBAc.

Como se puede observar, la utilización de diferentes políticas para un mismo caso puede dar lugar a varias asignaciones de retrasos.

CAPÍTULO 2. DESARROLLO DEL SOFTWARE

2.1. Introducción

Para el desarrollo de este TFC, se ha programado la aplicación denominada *ACREL* correspondiente a sus siglas en castellano a *'Aplicación de Cálculo de Retrasos en Llegadas'*.

Se trata de una herramienta hecha para poder obtener resultados de los retrasos que se producirían en tierra tras haber declarado un GDP y estudiar así algunas de las diferentes politicas de asignación de slots que se conocen.

Además, ACREL permite combinar diferentes políticas durante el GDP definido y también calcula el retraso de las aeronaves a las que no afecta el GDP por tratarse de vuelos excluidos y que aterrizan en el aeropuerto de destino mientras está activo el GDP.

Por lo que respecta al tráfico a analizar, la aplicación está preparada para poder leer archivos de tráfico con un formato tal y como se especifica en el apartado C.2.3. del apéndice C de este trabajo.

Los resultados principales que se pueden obtener son:

- Datos y estadísticas de retrasos:
    * del tráfico al que se le aplica el GDP (vuelos controlados)
    * del tráfico al que no se le aplica GDP pero llega durante éste y pueden sufrir retrasos en el aire (vuelos excluidos)
    * del tráfico que llega durante la duración del GDP pero al que no se aplica retraso en tierra pero sí se aplican las políticas de llegada mientras la capacidad del aeropuerto está reducida. Es decir, si todos los vuelos despegasen a su hora programada
    * de los pasajeros de los vuelos controlados
    * de los pasajeros de los vuelos excluidos
    * de los pasjeros si no se hubiera aplicado GDP y la capacidad del aeropuerto se viese afectada
- Un archivo de tráfico con la nueva distribución de llegadas tras aplicar el GDP y las políticas de prioridad con datos específicos de cada vuelo.
- Datos que definen las curvas de demanda y la capacidad del aeropuerto.
- Datos de los vuelos acumulados según un perído de tiempo que el usuario define.

De manera resumida, la aplicación lo que hace es crear una cola de tráfico de llegada con sus respectivos slots diferenciando si corresponden a slots de capacidad normal o

reducida dependiendo de si están fuera o dentro de la duración del GDP. La asignación de slots la hace conforme al horario de las diferentes políticas de asignación de preferencia definidas por el usuario y que pueden ser distindas y combinadas entre sí.

Después, calcula el retraso que se produce de los vuelos excluidos y controlados.

No obstante, para el desarrollo e implementación de la aplicación se ha seguido un ciclo de vida que se describe en el siguiente apartado.

2.2. Ciclo de vida

Tras la necesidad de crear un programa que combinara diferentes políticas y poder así analizar un tráfico y obtener resultados de ellas se definió qué proceso se llevaría a cabo durante toda la elaboración de esta aplicición.

Para ello, entre los diferentes ciclos de vida de creación de un sistema como pueden ser los ciclos en cascada, en espiral o en 'V' se tomó la decision de realizarlo siguiendo las pautas de este último. No obstante, cabe decir que cualquiera de los tres podría haber sido válido ya que todos están basados en los siguientes aspectos[6]:

- **Análisis de requerimientos y especificaciones.** Para ello, se redacta una serie de requisitos [7] y casos de uso que se pueden encontrar en el apéndice A y en el apartado B.2. del anexo B de este trabajo. Paralelamente se generan los test de validación del sistema.
- **Diseño.** Esto indica cómo está hecho el sistema, de qué manera y qué forma tiene. Se detalla cómo es ACREL en la sección 2.3. de este trabajo.
- **Implementación e integración.** En este punto es cuando se lleva a cabo la realización de las partes del programa diseñado y se juntan para que tenga sentido.
- **Test de unitarios.** Esto se realiza a medida que se realiza el sistema para comprobar o verificar que lo que se está haciendo, se realiza de manera correcta.
- **Test de integración.** Una vez el sistema está terminado, se realiza una serie de tests al conjunto para validar la totalidad del funcionamiento del sistema que corresponden a los requisitos y especificaciones de la primera etapa.

El resumen gráfico del ciclo de vida en V se representa en la figura 2.1 donde se puede seguir cada paso siguiendo las flechas gruesas que forman una V.

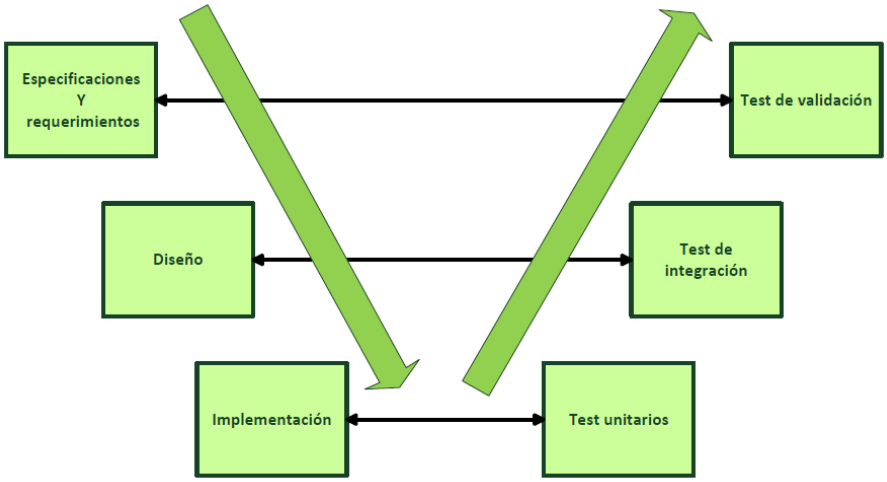

Figura 2.1: Ciclo de vida en V

Cabe decir que durante la implementación e integración, se repiten algunos de los test que se realizan tras arreglar los fallos detectados con ellos.

Por otro lado, pese a la planificación que se realiza al principio, los resultados en el caso de ACREL han variado adaptándose al tiempo disponible, a la necesidad de crear nuevos cálculos que se consideran que son necesarios y no se tuvieron en cuenta o porque al realizar los requerimientos del sistema se pensó de una forma ambiciosa. Sin embargo, los resultados mínimos y necesarios para cumplir con los objetivos de este TFC han sido cumplidos con éxito dado que ACREL se crea como herramienta para la obtención de resultados.

2.3. Arquitectura del programa

La aplicación está basada en la programación orientada a objetos siguiendo una estructura de programación por capas.

Esta estructura de programación por capas es una técnica de ingeniría de software propia de la programación orientada a objetos los cuales se organizan principalmente en 3 módulos o capas dependiendo de la funcionalidad de los objetos[8].

Estas tres capas son la de ***Interfaz Hombre Máquina (IHM)***; *capa lógica, de negocio, control o también denominada* **Dominio** y la capa de ***Base de Datos (BD)***.

La capa de IHM cumple con el objetivo de intercambiar información con el usuario usando los objetos correspondientes para poder interactuar. Es la encargada de generar los formularios o ventanas, mostrar por pantalla los resultados, recoger las peticiones del usuario y hacer que el usuario, de una forma amigable, comprenda qué se le está pidiendo o qué está realizando la aplicación.

La capa de Dominio es donde se procesa toda la información y donde se realizan la mayor

parte de todos los cálculos.

La última capa BD es la encargada de comunicarse con otras bases de datos y/o archivos para obtener o ingresar información al sistema [8].

El hecho de que ACREL o cualquier aplicación esté programada siguiendo esta división por capas hace que el mantenimiento sea más sencillo dado que cada capa representa una funcionalidad del sistema. Si por ejemplo se quiere cambiar la forma de leer un archivo de la base de datos, basta con modificar la capa de BD y adaptarla sin necesidad de modificar el resto de código que se encuentra en las otras dos capas.

Por otra parte, la comunicación entre capas se realiza a través de *transacciones* que hacen que los datos sigan un orden. La figura 2.2 representa un esquema básico de cómo estan organizadas estas capas.

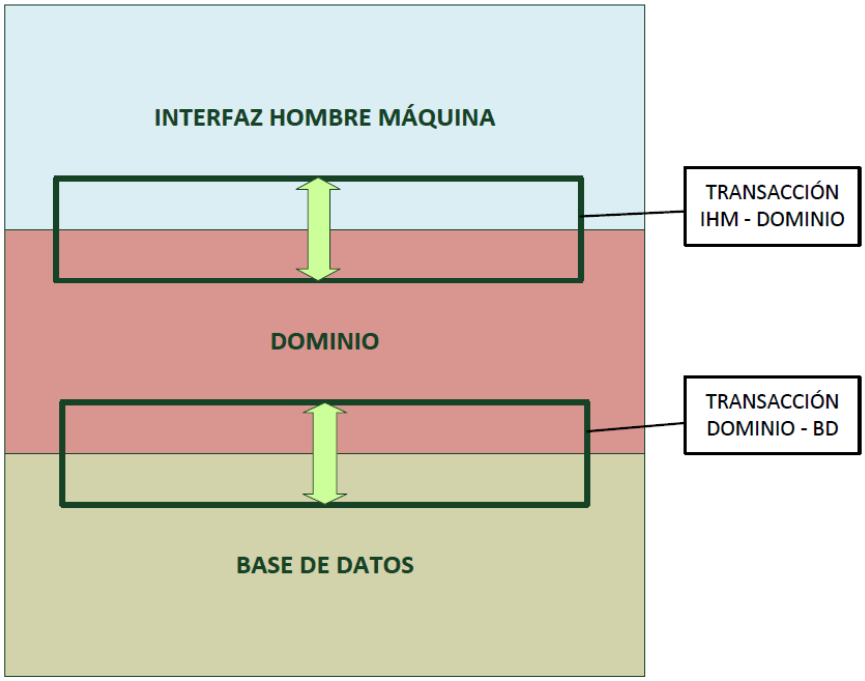

Figura 2.2: Esquema de programación por capas

Como se puede ver en la figura 2.2, la capa de IHM solo puede comunicarse con la capa de Dominio a través de la subcapa de transacciones *Transacció IHM - Dominio*.

Por otra parte, la capa BD solo puede comunicarse con la de dominio mediante la subcapa de transacciones *Transacción Dominio - BD*.

Estas transacciones tienen los dos sentidos de flujo de datos. Si fuera necesario que desde la IHM se requiera algún dato de la BD, se tendría que pasar a traves de todas las capas.

IHM <–> Transaccióno IHM - Dominio <–> Dominio <–> Transacción Dominio - BD <–> BD

Finalmente se puede ver la organización de capas con sus respectivos objetos en el apartado B.3. del apéndice B.

2.4. Algoritmos de cálculo

Este apartado explica los procesos de ejecución que se producen durante el tratamiento y cálculo de datos principales.

Para la ejecución de los cálculos principales, el programa recibe por parte del usuario una serie de parámetros separados por espacios en blanco y en una línea que hemos denominado linea de ejecución, de comandos o argumentos.

Entre esos parámetros viene especificado el archivo de tráfico que se quiere analizar, la definición el GDP y las políticas que se van a aplicar. Más información referente a esta línea de ejecución o ejemplos de ella se puede encontrar en el apartado C.4. del anexo C de este trabajo.

Una vez el usuario ha introducido esta línea de ejecución, la aplicación realiza los siguientes pasos:

1. Definición del GDP.

   Se define el GDP con los datos del usuario. En esta definición van incluidas las capacidades del aeropuerto con y sin restricciones, el horario de comienzo y finalización de las restricciones del GDP y el número de políticas que se van a aplicar.

2. Creación de tráfico.

   La aplicación carga el archivo de tráfico que va a ser analizado. (Se puede encontrar una explicación de cómo es este archivo de tráfico en el apartado C.2.3. del anexoC de este trabajo).

   Como la información del archivo de tráfico disponible no es toda la que se necesitaba para este proyecto, se creó un archivo de datos de tipos de aeronaves con información del modelo, fabricante, número de plazas y MTOW.

   Este archivo es cargado en memoria antes de comenzar a leer el archivo de tráfico. Su función es, según el tipo de aeronave que el archivo de tráfico proporciona por vuelo, dar el número de plazas y el MTOW correspondientes para que posteriormente la aplicación pueda complementar los datos del vuelo con un número de pasajeros y si se trata de una aeronave de tipo *Heavy, Large o Small*.

   La asignación del número de pasajeros es calculada de forma aleatoria gausiana alrededor de un factor de ocupación del 80 % teniendo en cuenta el número de plazas correspondientes a la aeronave.

   Para la asignación del tipo de aeronave se tiene en cuenta el peso máximo al despegue o MTOW correpondiendo a aeronaves *Small* un peso igual o inferior a 7000 kg, aeronaves *Large* entre 7000 y 136000 kg y a aeronaves con denominación *Heavy* si su MTOW es igual o mayor a 136000 kg [9][10].

Si ocurre que en esta base de datos de aeronaves no existe alguno de los modelos que aparecen en el archivo de tráfico, el número de pasajeros se asigna de forma aleatoria uniforme y para cada vuelo con un valor comprendido entre 4 y 300 pasajeros. Después, dependiendo del número de pasajeros asignado, la aplicación define la tipología del vuelo con *Small* si es un número igual o menor a 18, *Large* si la cantidad de pasajeros asignados es de entre 18 y 230, y *Heavy* para el resto. En este caso, para la asignación de tipología de aeronave no se ha tenido en cuenta el MTOW y se han tomado las consideraciones de distinción por cuenta propia para que la aplicación pueda funcionar. Este proceso también se realiza si no se dispone de base de datos de aeronaves.

3. Clasificación de demanda.

   Se crea una lista de llegada con slots con sus correspondientes vuelos ordenados según la hora de llegada sin tener en cuenta las restricciones ni la capaciad reducida del aeropuerto. Esta será la demanda del aeropuerto.

4. Cálculo hora fin de GDP.

   Con la definición del usuario y la lista de llegadas calculada en el punto anterior, el programa busca el slot donde comienza el GDP. A continuación comienza un cálculo de aeronaves que pueden llegar debido a la capacidad regulada cada slot que pasa de la lista de llegadas de demanda. Cuando se llega a la hora de fin de restricciones del GDP, es decir, cuando la capacidad regulada pasa a capacidad nominal, la aplicación continúa haciendo el cálculo de cuántas aeronaves pueden llegar y lo compara con la demanda.

   Cuando finalmente coincide el número de aeronaves que puede llegar (la capacidad) con la demanda, ese es el momento en el que finaliza el GDP.

   En la figura 2.3 se pueden observar estos puntos redondeados. El primero de ellos representa la hora de inicio del GDP y con él la restricción de capacidad con una pendiente menor a la demanda.

   El siguiente punto marcado con un círculo es donde finalizan las restricciones del GDP. Se puede observar cómo la capacidad aumenta después de este punto dado que la pendiente de la línea incrementa.

   Justo cuando la línea de capacidad coincide con la la línea de demanda, indica que finaliza el GDP puesto que ya no hace falta aplicar retrasos.

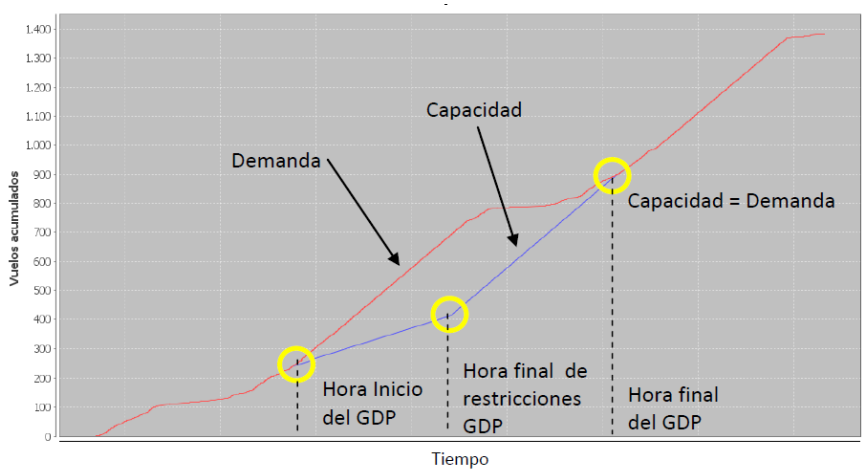

Figura 2.3: Gráfica demanda vs capacidad puntos del GDP

A la vez que la aplicación busca el punto donde coincide la demanda con la capacidad, también se va calculando el área que hay entre estas dos curvas. Este área indicará el retraso mínimo teórico global que se podría asignar a todos los vuelos afectados por el GDP.

5. Evaluación y separación de los vuelos en no afectados, excluidos y controlados.

   Tal y como se explicó en el apartado 1.3. del capítulo 1 cuando se aplica un GDP se diferencian tres tipos de vuelos. Los no afectados, los excluidos y los controlados. Podemos ver gráficamente en la figura 2.4 la separación de los no afectados del resto.

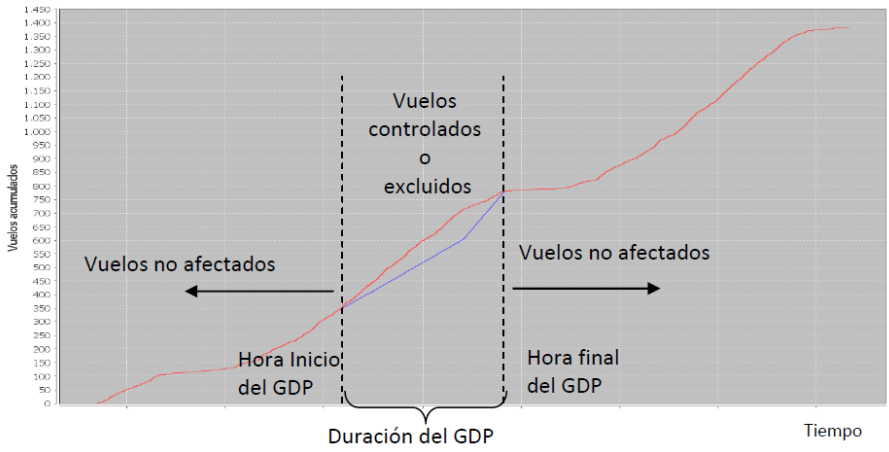

Figura 2.4: Clasificación de vuelos no afectados, excluidos y controlados

Para hacer esto, la aplicación recorre uno a uno todos los vuelos y comprueba ETA (hora de llegada). Si es menor a la hora de inicio del GDP o mayor a la hora de finalización, el vuelo es clasificado como *No afectado*.

Por otro lado, para clasificar los vuelos *Excluidos* el programa comprueba si la ETA está dentro del periodo del GDP, después mira si se trata de un vuelo nacional o

internacional, proviene de una distancia mayor al radio de acción que el usuario definió y si la hora de despegue ETD *Estimated Time of Departure* era antes de la hora de definición del GDP. Si sucede que cumple alguno de los casos anteriores, el vuelo pasa a estar como clasificado.

Finalmente, la aplicación clasifica al resto de vuelos como controlados dado que éstos llegan durante el GDP, son nacionales, provienen de una distancia menor al radio de acción del GDP y su ETD es más tarde que la hora de definición del GDP que es un parámetro que define el usuario y es anterior a la hora de inicio del GDP.

6. Definición de políticas.

   Se definen las políticas según los parámetros que el usuario introduce. (Se puede ver más información de esto en el apartado C.4. del anexoC de este trabajo)

7. Aplicación del GDP.

   Este proceso crea una nueva lista de llegadas que tiene en cuenta las restricciones del GDP, es decir, la capacidad regulada en el GDP y por tanto una disminución de aterrizajes a la hora porque los slots duran más durante el GDP. Además en este punto es donde se tienen en cuenta las políticas y se aplican si son necesarias.

   La forma de crear esta lista de llegadas es un poco costosa en recursos.

   Primero, se coge un vuelo de la lista de vuelos cargada en memoria. Después, se mira a qué hora tiene previsto llegar y se busca el slot correspondiente. Si no existe ningún slot, se van creando desde el inicio del tráfico hasta que crea el del vuelo en cuestión. Estos slots se crean teniendo en cuenta el GDP ya que, como se ha comentado antes, duran más que los de fuera del GDP.

   Si el vuelo que se quiere colocar encuentra el slot donde le tocaría ir y éste está ocupado por otro vuelo, se realiza una primera asignación de preferencia mirando si el vuelo que está colocado es controlado, excluido o no afectado. El orden de preferencia siempre será, dentro del GDP (que es donde solo pueden haber vuelos excluidos y controlados) primero excluidos y después controlados.

   Es decir, si el vuelo que se quiere colocar es controlado y el que ya está en el slot es un vuelo excluido, el vuelo controlado pasa a comprobar el siguiente slot. Si hay vuelo, se mira la preferencia y si no, se queda con el slot.

   En cambio, si el vuelo que se quiere colocar es excluido, y en el slot que le toca ir hay un controlado, el controlado “abandona” el slot y el excluido lo ocupa. Seguidamente, el vuelo controlado intenta ocupar el slot siguiente repitiéndose el proceso de preferencias y políticas.

   Si el vuelo que se quiere colocar es de la misma clasificación, se comparan entre ellos según la preferencia que asignará la política que haya en ese periodo de tiempo.

   Si resulta que no hay una política que cubra a esos vuelos (dentro del GDP) tiene preferencia la aeronave que tiene una hora de llegada más temprana. Es como si se aplicara la política RBS.

   Cuando se termina de colocar todos los vuelos, este proceso ha terminado.

8. Cálculo de retrasos.

Se recorren todos los slots desde el inicio de la lista de vuelos tras el GDP. En cada uno de ellos se mira si hay un vuelo y si es así se compara si la ETA que tenía está entre la hora de inicio y la hora de finalización de slot. Si no es así, quiere decir que ese vuelo ha sufrido un retraso.

Este retraso se calcula haciendo la diferencia entre la hora de inicio del slot donde se encuentra asignado y la ETA del vuelo inicial.

Por otro lado, la nueva hora de llegada pasa a ser la hora de inicio del slot donde está asignado y se denomina *Controlled Time of Arrival (CTA)* si se trata de un vuelo controlado.

Como se trata de un vuelo controlado, quiere decir que este retraso que se ha calculado, lo podrá realizar en tierra y su hora estimada de salida ETD del aeropuerto de origen es retrasada tanto tiempo como retraso se le haya asignado al vuelo en la hora de llegada. Esta nueva hora de salida se denomina *Controlled Time of Departure*.

Si el retraso que se calcula es de un vuelo excluido, querrá decir que éste tendrá que esperar ese tiempo antes de aterrizar si llega a la hora prevista.

9. Cálculo de estadísitcas.

   Finalmente, la aplicación junta los datos de retrasos por tipología de vuelos recorriendo los slots nuevamente de la lista.

   El retraso global del GDP (el de los vuelos controlados) es la suma de retrasos de cada uno de los vuelos controlados.

   El retraso global que se produce en el aire corresponde a la suma total de cada uno de los vuelos excluidos.

   El retraso si no se aplicara el GDP y existiera la restricción de disminución de capacidad es el resultado de la suma de las anteriores.

   Se calcula también el retraso acumulado de los pasajeros para los tres casos. Para calcular el retraso acumulado de los pasajeros de un vuelo se multiplica el número de pasajeros por el tiempo de retraso.

   Por otra parte, además de obtener los retrasos máximos que se producen de los vuelos para cada distinción, se calcula la desviación típica del retraso global. Para ello se utiliza la fórmula 2.1[11][12].

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{2.1}$$

   donde

   σ es la desviación típica
   N = numero de vuelos evaluados
   x_i = al retraso sufrido del vuelo evaluado en concreto
   $\bar{x}$ = retraso medio de los vuelos todales de la clasificación que se esta evaluando.

   Este dato de desviación típica nos indicará cómo de dispersados están los retrasos según el valor medio calculado.

Si este valor es es muy pequeño querrá decir que todos los vuelos de esa clase (ya sean excluidos, controlados o simplemente afectados por el cambio de capacidades) sufren un retraso muy próximo al retraso medio.[13]

Básicamente estos son los cálculos y los pasos que lleva a cabo la aplicación diseñada.

2.5. Posibles ampliaciones y mejoras

La aplicación está diseñada como herramienta para poder estudiar las diferentes políticas de asignación de retrasos de un GDP en este TFC. No obstante, esta aplicación podría mejorarse con una interfaz gráfica y un formulario donde se hicieran más visuales los resultados y la manera de introducir los datos.

Por otra parte, podría mejorarse también a la hora de introducir los argumentos dado que siempre que se hace hay que volverle a pasar el fichero de tráfico que se quiere analizar.

Quizás estaría bien hacer que el tráfico se cargara una sola vez al iniciar la aplicación y en caso de querer cambiar de tráfico a analizar, tener una opción que lo permita.

Como futuras ampliaciones, ACREL se ha diseñado teniendo en cuenta que el retraso de los vuelos excluidos se realiza al llegar al aeropuerto, podría hacerse durante el vuelo con una disminución de la velocidad. Es por ello que se calcula el retraso en el aire. Podría seguir por esta línea de desarrollo para cambiar los retrasos en tierra y durante el vuelo.

Por otro lado, al estar programado orientado a objetos, en caso de que se quiera añadir nuevas políticas de asignación de slots, sería muy fácil hacerlo, solo habría que crear una nueva clase en la capa de dominio teniendo en cuenta que se le pasa como parámetros los dos vuelos a comparar. Esta clase calcularía el vuelo con prioridad y devolvería el vuelo prioritario.

Además, en el objeto GDP donde se definen las políticas habría que añadir 4 líneas de código similares iguales que las que se usan para las otras políticas pero indicando que la nueva política corresponde a un número en concreto que será la referencia que le pase el usuario desde la interfaz.

CAPÍTULO 3. ESTUDIO DE CASOS, SIMULACIONES Y RESULTADOS

En este capítulo se representa el estudio de políticas con los datos obtenidos de la aplicación ACREL desarrollada con este fin y para este TFC.

Lo que se calculan son los diferentes retrasos que se producen si se usan las diferentes políticas individualmente y luego, se hace una combinatoria de de dos políticas durante un GDP cambiando el horario en la que termina la primera y comienza la siguiente para ver cómo afecta a los retrasos.

El archivo de tráfico a analizar contiene los vuelos reales que llegaron al *Aeropuerto internacional Libertad de Newark* (EWR) en Nueva York los días 24 y 25 de agosto de 2005.

Para definir nuestro GDP de referencia que usaremos para todos los casos de estudio con intención de partir de una misma base y que las comparativas solo se basen en el tipo de políticas, se tendrá en cuenta el horario de la tabla 3.1 [14].

GDP report time	GDP start time	GDP end time
10:20	12:36	00:18

Cuadro 3.1: Horario local del *Cluster 1* de GDP de EWR. [14]

Este horario es uno de los tres clusteres que representan, la hora de definición media fue a las 10:20, la hora de inicio media a las 12:36 y la hora programada de finalización del GDP a las 00:18 h [14].

No obstante, el horario que definiremos será lo más parecido a éste dado que nuestra aplicación es la encargada de calcular la hora final del GDP dado que lo que se define es la hora de finalización de restricciones (cambio de capacidad reducida a nominal). Esto es así porque la hora final del GDP respecto a la hora final de restricciones variará en función de las capacidades definidas y la demanda del aeropuerto.

El horario de la tabla está en formato local del aeropuerto. Por eso, primero que se hace para nuestro estudio es pasarlo a unixTime teniendo en cuenta el desfase horario.

La diferenca horaria entre Nueva York y Greenwich es de -5 horas en invierno y de -4 h en verano. Como nuestros datos son de agosto, pleno verano, el horario en UTC quedaría como el que se muestra en la tabla 3.2.

GDP report time	GDP start time	GDP end time
14:20	16:36	04:18

Cuadro 3.2: Horario propuesto en UTC para la aplicación de GDP en EWR.

Como ya se ha comentado antes, el horario de fin del GDP no es un parámetro que definamos directamente, sino que se trata de un parámetro que depende de la hora a la

que acaban las restricciones y cambia la capacidad.

Para calcularlo, lo que hemos hecho ha sido ir probando valores de la hora final de restricciones, que sí es un parámetro que nosotros definimos en nuestra aplicación.

Por otra parte, otro de los parámetros que se necesita son las capacidades tanto nominales como reducidas del aeropuerto.

La capacidad nominal CN que se ha seleccionado es de 50 operaciones/hora y la capaciadad regulada de 30 operaciones/hora para el aeropuerto de EWR según [2].

Se han encontrado referencias [4] con capacidades nominales de 40 op/h pero se ha visto más oportuno usar la de 50 dado que de esta forma habrá un cambio más significativo de capacidades.

También se ha fijado el último parámetro que afecta directamente al GDP. Este es el radio de acción que, para nuestro análisis, lo hemos fijado a 1090 NM tomando como referencias un estudio que analiza la vaciación del radio de acción del aeropuerto de Boston y comenta que esta distancia es la que menos retraso produce para ese caso [4].

El resultado tras aplicar todos estos parámetros al tráfico disponible es el que se muestra en la imagen 3.1 obtenida directamente de la aplicación.

```
Capacidad normal:                    50
Capacidad regulada con GDP activo:   30
Radio accion GDP:                    1090.0 NM
Hora de inicio trafico:              08/24/2005 02:14:00
Hora de final de trafico:            08/26/2005 07:12:00
Duracion del trafico:                52:58:00 h
Hora de defincion GDP:               08/24/2005 14:20:00
Hora de inicio GDP:                  08/24/2005 16:36:10
Hora de fin restricciones GDP:       08/24/2005 22:06:40
Hora de final GDP:                   08/25/2005 04:18:10
Duracion GDP:                        11:42:00 h
Duracion restricciones:              05:30:30 h
Numero de vuelos totales:            1383
Numero de vuelos no afectados:       908
Numero de vuelos excluidos:          72
Numero de vuelos controlados:        403
```

Figura 3.1: Definicion y tiempos del GDP

Aquí se puede ver un resumen de los tiempos de defínción del GDP, duraciones, del número de vuelos totales, controlados, excluidos, etc.

Uno de los datos más interesantes, es el número de vuelos controlados, que son los vuelos que realizarán en retraso en tierra.

Además, de esta definición del GDP podemos obtener de manera gráfica la representación del GDP respecto al tráfico. Esto se puede observar en la figura 3.2 y 3.3 donde esta última es una ampliación de la zona que nos interesa.

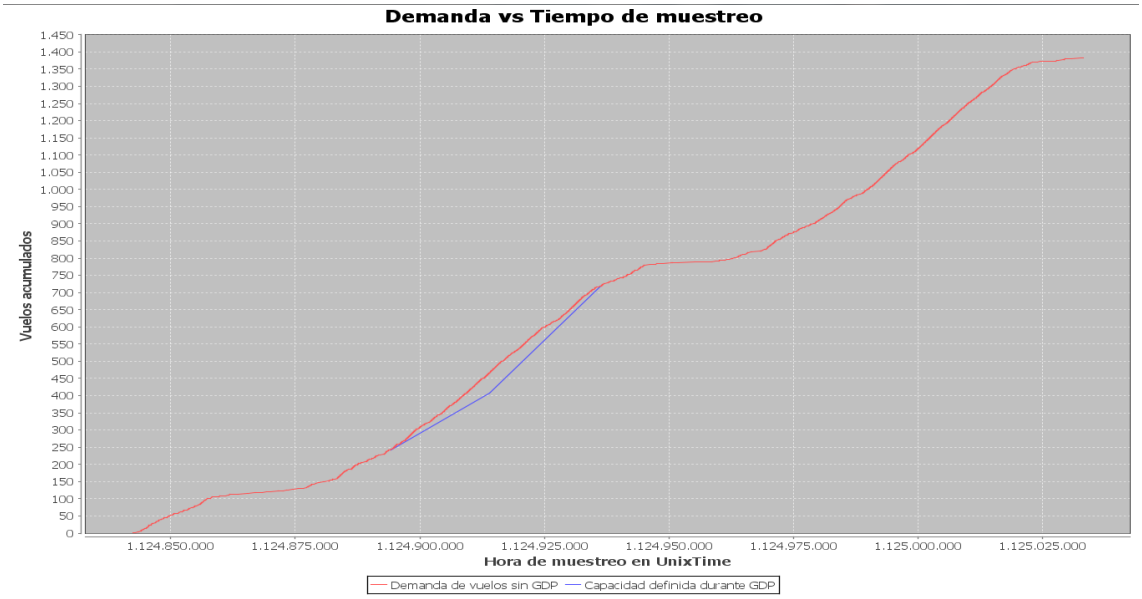

Figura 3.2: Curvas de demanda y capacidad visión de tráfico total analizado

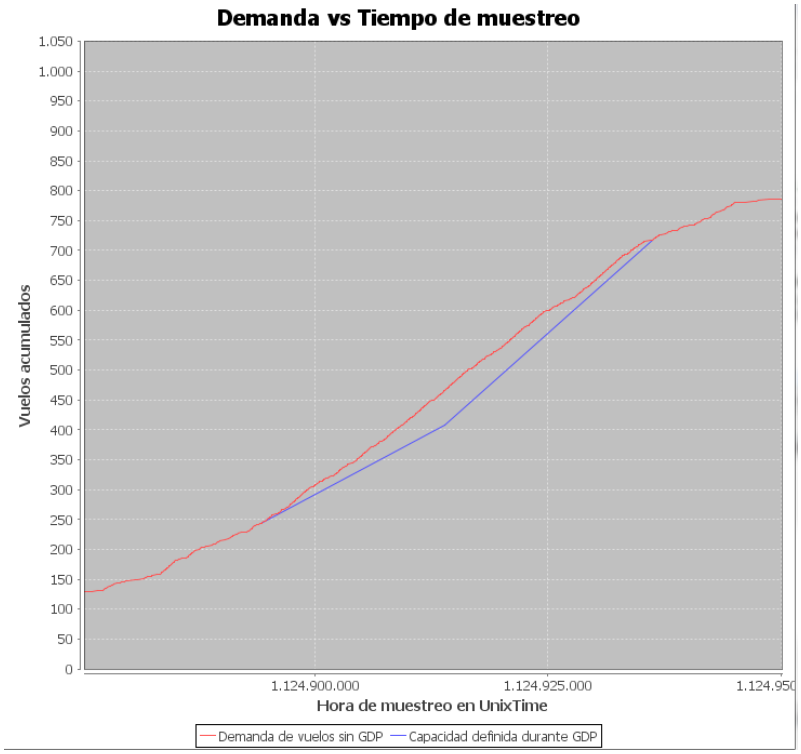

Figura 3.3: Curvas de demanda y capacidad enfocado en la región del GDP.

La gráfica 3.2 y la 3.3 muestran cómo a medida que pasa el tiempo, los vuelos van llegando.

La curva roja representa la demanda del aeropuerto, es decir, el número de vuelos que llegarían si no se aplicara restricciones.

Por otro lado, la curva azul representa las restricciones del GDP. La primera parte, con una pendite menor, es la curva que delimita la capacidad reducida, en este caso corresponde a 30 vuelos a la hora. La parte de la curva azul con mayor pendiente, posterior siempre a la de menor pendiente, está relacionada con la capacidad nominal.

Otras de las gráficas que pondemos obtener para ver el comportamiento de nuestro tráfico tras el GDP son las correspondientes a las figuras 3.4 y 3.5.

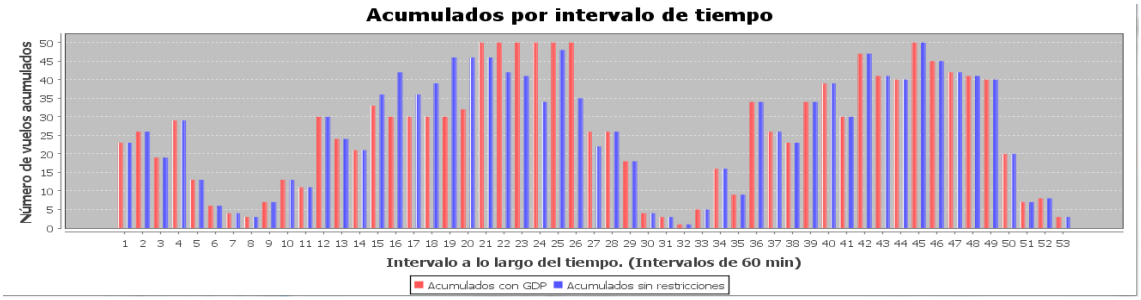

Figura 3.4: Vuelos acumulados cada hora. Todo el tráfico

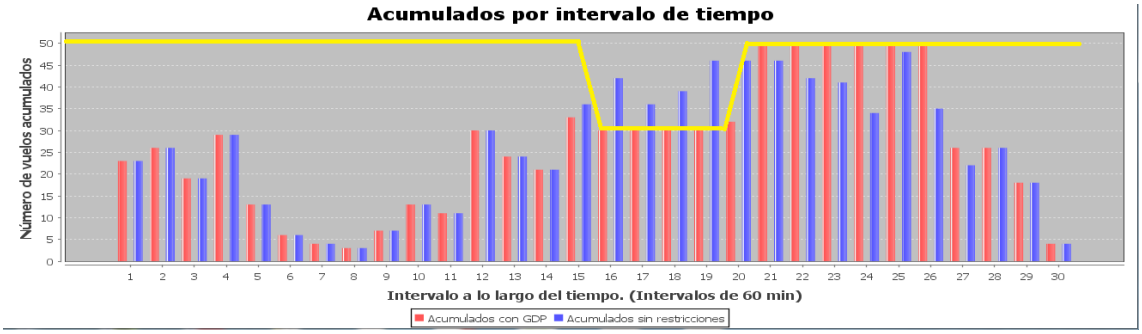

Figura 3.5: Vuelos acumulados cada hora. Primera parte del tráfico.

Ambas corresponden al GDP que hemos definido y representan la cantidad de vuelos acumulados cada hora desde el inicio del tráfico. Las barras azules representan el tráfico que llega si no se aplicara el GDP y las barras rojas si se aplicase.

Como se puede apreciar mejor en la figura 3.5, al principio todas las barras son del mismo tamaño. Esto quiere decir que el número de vuelos que se cuentan cada hora es el mismo. Esto es debido a que, como indica la línea amarilla, la capacidad del aeropuerto se encuentra por encima del número de vuelos que llegan (la demanda).

Cuando la línea amarilla desciende a 30, es porque se ha activado el GDP. Aquí es donde se aprecia que las barras rojas no superan estos 30 vuelos a la hora.

Todos estos vuelos que tenían programado llegar, representados por las barras azules, se retrasan porque no pueden son admitidos y se van retrasando.

El resultado es que, tras la finalizacion del GDP y por consiguiente el aumento de capacidad, se produzca una llegada de vuelos hasta la capacidad máxima hasta que la

demanda que estaba programada vuelve a coincidir con la demanda tras aplicar el GDP donde comienzan a igualarse las dos barras.

Una vez que ya se tiene el GDP definido, pasamos a analizar el retraso de las aeronaves utilizando las diferentes políticas.

3.1. Políticas individuales

Para esta primera parte donde obtendremos datos de las políticas aplicadas de manera individual, se considera que la duración de cada una de ellas será la duración del GDP aplicado desde la hora de inicio hasta la hora de final del GDP.

En cada sub - apartado se muestran resultados obtenidos de la aplicación ACREL. Solo se tienen en cuenta los retrasos para los vuelos controlados aunque ACREL también nos lo proporciona para vuelos excluidos y sin considerar esta distinción. Segidamente se comentarán a parte todos juntos.

3.1.1. RBS

Definición:

Tipo de politica	Inicio	Final
RBS	08/24/2005 16:36:10	08/25/2005 04:18:10

Cuadro 3.3: Definición de la política RBS.

Resultados:

	Retraso GDP de vuelos controlados
Global de vuelos	14 D 19 h 53 min 36 s
Medio por vuelo	52 min 59 s
Desv. Típica vlo.	45 min 53 s
Máx por vuelo.	20 min 00 s
Global por pasajeros.	1139 D 22 h 45 min 24 s

Cuadro 3.4: Resultados obtenidos con RBS de retrasos del GDP de vuelos controlados.

Como se puede observar en la figura 3.4, el retraso medio por aeronave es de alrededor de 53 min. Por otro lado vemos que la desviación típica es de 46 min es decir, que el retraso de la gran mayoría de vuelos está comprendido 46 min más o menos de los 53 min de media. Pero, como bien se refleja en el retraso máximo sufrido por una aeronave en concreto, pueden haber retrasos fuera de esta desviación. En este caso, se llega a un retraso de 2h 30 min superior a la media de los vuelos totales. Estas observaciones se pueden aplicar al resto de politicas expuestas a continuación pero teniendo en cuenta sus datos.

3.1.2. RBPax

Definición:

Tipo de politica	Inicio	Final
RBPax	08/24/2005 16:36:10	08/25/2005 04:18:10

Cuadro 3.5: Definición de la política RBPax.

Resultados:

	Retraso GDP de vuelos controlados
Global de vuelos	14 D 20 h 44 min 48 s
Medio por vuelo	53 min 06 s
Desv. Típica vlo.	1 h 30 min 37 s
Máx por vuelo.	10 h 20 min 12 s
Global por pasajeros.	511 D 07 h 10 min 12 s

Cuadro 3.6: Resultados obtenidos con RBPax de retrasos del GDP de vuelos controlados.

En estos datos, resalta el retraso máximo que sufre una aeronave de 10 h y 20 min cuando la media está en 53 min. Esta gran diferencia se debe a que esta política da preferencia a las aeronaves con mayor número de pasajeros. Si en la programación de los vuelos había alguna avioneta con 4 pasajeros, todas las aeronaves con más pasajeros tendrán preferencia sobre la avioneta. Como estamos analizando un aeropuerto internacional, el volumen de aeronaves comerciales con más de 100 pasajeros es mucho más elevado que el de aeronaves privadas de entre 1 y 8 tripulantes.

3.1.3. RBAc

Definición:

Tipo de politica	Inicio	Final
RBAc	08/24/2005 16:36:10	08/25/2005 04:18:10

Cuadro 3.7: Definición de la política RBAc.

Resultados:

	Retraso GDP de vuelos controlados
Global de vuelos	14 D 20 h 06 min 36 s
Medio por vuelo	53 min 01 s
Desv. Típica vlo.	1 h 01 min 26 s
Máx por vuelo.	8 h 32 min 36 s
Global por pasajeros.	848 D 16 h 32 min 24 s

Cuadro 3.8: Resultados obtenidos con RBAc de retrasos del GDP de vuelos controlados.

Esta política mejora de la anterior en la desviación típica y un poco en el retraso máximo sufrido manteniendo el mismo retraso mínimo. A cambio el retraso que sufren los pasajeros es mayor pero no es tanto como se producía con RBS.

3.1.4. RBD

Definición:

Tipo de politica	**Inicio**	**Final**
RBD	08/24/2005 16:36:10	08/25/2005 04:18:10

Cuadro 3.9: Definición de la política RBD.

Resultados:

	Retraso GDP de vuelos controlados
Global de vuelos	14 D 21 h 08 min 00 s
Medio por vuelo	53 min 10 s
Desv. Típica vlo.	1 h 42 min 03 s
Máx por vuelo.	10 h 11 min 48 s
Global por pasajeros.	1261 D 18 h 11 min 24 s

Cuadro 3.10: Resultados obtenidos con RBD de retrasos del GDP de vuelos controlados.

En este caso se observa que todos los retrasos aumentan respecto a los anteriores. Esta prioridad asigna la preferencia a los vuelos de procedencia más lejana lo que implica que los vuelos que provienen aeropuertos más cercanos sufren un mayor retraso. Posiblemente, el número de vuelos lejanos sea menor que el número de vuelos cercanos, los llamados regionales. Estos vuelos tienen mucha mayor frecuencia pero si tienen menos prioridad que uno proveniente de más lejos, acumula retrasos afectando también a todas las demás frecuencias entre esos dos aeropuertos. Por este razonamiento, es posible que surjan estos retrasos tan altos.

Por ello, se ha propuesto otra política que considera lo contrario a RBID.

3.1.5. RBID

Definición:

Tipo de politica	Inicio	Final
RBID	08/24/2005 16:36:10	08/25/2005 04:18:10

Cuadro 3.11: Definición de la política RBID.

Resultados:

	Retraso GDP de vuelos controlados
Global de vuelos	14 D 21 h 05 min 12 s
Medio por vuelo	53 min 09 s
Desv. Típica vlo.	1 h 35 min 53 s
Máx por vuelo.	8 h 2 min 48 s
Global por pasajeros.	1040 D 15 h 32 min 36 s

Cuadro 3.12: Resultados obtenidos con RBID de retrasos del GDP de vuelos controlados.

Como algunos profesores suponían, RBID mejora considerablemente el retraso de los pasajeros. Además, el retraso máximo se ha redudico en 2 horas manteniéndose el retraso medio.

3.1.6. Comparativas y conclusiones

Tras ver todos los resultados que obtuvimos con ACREL, se han seleccionado solo los referentes a los vuelos controlados, es decir, a los retrasos que se realizan en tierra.

Política	Total de vuelos	Total por pasajeros
RBS	14 D 19 h 53 min 36 s	1139 D 22 h 45 min 24 s
RBPax	14 D 20 h 44 min 48 s	511 D 07 h 10 min 12 s
RBAc	14 D 20 h 06 min 36 s	848 D 16 h 32 min 24 s
RBD	14 D 21 h 08 min 00 s	1261 D 18 h 11 min 24 s
RBID	14 D 21 h 05 min 12 s	1040 D 15 h 32 min 36 s

Cuadro 3.13: Resumen de resultados globales de retrasos del GDP con vuelos controlados.

Política	Medio por vuelo	Des. Tipica de los vuelos	Máximo por vuelo
RBS	52 min 59 s	45 min 53 s	3 h 20 min 00 s
RBPax	53 min 06 s	30 min 37 s	10 h 20 min 2 s
RBAc	53 min 01 s	1 h 01 min 26 s	8 h 32 min 36 s
RBD	53 min 10 s	1 h 42 min 03 s	10 h 11 min 48 s
RBID	53 min 09 s	1 h 35 min 53 s	8 h 2 min 48 s

Cuadro 3.14: Resumen de resultados de retrasos del GDP con vuelos controlados.

Como podemos observar en las tablas, RBS es la politica que menos retraso asigna de forma global a las aerolíneas y además, es la que ménos retraso máximo aparece. Sin embargo, perjudica a los pasajeros si lo comparamos con el resto de políticas.

También se observa que RBD es la peor política que se podría emplear dado que es la política que más retraso obtiene en todos los sentidos.

Por otra parte, el retraso que más beneficia a los pasajeros es RBPax como era de esperar. La parte negativa es que las aeronaves pequeñas pueden salir muy perjudicadas. En este caso, la aeronave que peor parada saldría utilizando esta política, tendría que posponer su vuelo 10 horas y 20 minutos. Otra venjata de esta política es que la desviación típica solo es de alrededor de 30 min. En este caso, es la mejor obtenida dado que el retraso medio por aeronave sea cual sea la política usada es muy similar.

La política RBID propuesta por los profesores de la EETAC *Luis Delgado* y *Xavier Prats* mejora la política RBD en todos los sentidos y a la politica RBS si se copara con el retraso que sufren los pasajeros.

3.2. Políticas combinadas

Tras los resultados anteriores y dado que la aplicación ACREL se ha diseñado con el objetivo también de que pudiera combinar políticas, en este apartado se hace un pequeño estudio entre dos politicas combinadas. Para ver cómo afecta en el retraso asignado.

Puesto que RBS es la política que menos retraso asigna a los vuelos, y RBPax a los pasajeros, se ha decidido combinar ambas a lo largo del GDP.

3.2.1. RBS - RBPax

En este punto se define la política RBS desde el inicio del GDP hasta la hora de fin de restriciones y RBPax el resto.

Definición:

Tipo de politica	Inicio	Final
RBS	08/24/2005 16:36:10	08/24/2005 22:06:40
RBPax	08/24/2005 22:06:40	08/25/2005 04:18:10

Cuadro 3.15: Definición de las políticas combinadas RBS - RBPax.

Como se puede comparar con la figura 3.1 la horas de cambio de política coincide con la de cambio de capacidades del aeropuerto. Esto se ha hecho así de forma intencionada. Lo mismo pasará en el siguiente caso en la tabla 3.18.

Resultados:

	Retraso GDP de vuelos controlados
Global de vuelos	14 D 20 h 17 min 00 s
Medio por vuelo	53 min 02 s
Desv. Típica vlo.	1 h 09 min 32 s
Máx por vuelo.	7 h 26 min 48 s
Global por pasajeros.	874 D 9 h 24 min 36 s

Cuadro 3.16: Resultados de retrasos obtenidos tras combinar RBS - RBPax del GDP de vuelos controlados.

Se ve reflejado que la combinatoria de las dos políticas tiene un efecto positivo. Si comparamos los resultados con las políticas individuales por separado, se obtiene que el retraso mayoritario de cada una se ve mejorado a cambio de empeorar los retrasos que la política minimizaba.

(tabla con resultados obtenidos para GDP)

3.2.2. RBPax - RBS

En este punto se define la política RBPax desde el inicio del GDP hasta la hora de fin de restriciones y RBD el resto.

Definición:

Tipo de politica	Inicio	Final
RBS	08/24/2005 16:36:10	08/24/2005 22:06:40
RBPax	08/24/2005 22:06:40	08/25/2005 04:18:10

Cuadro 3.17: Definición de las políticas combinadas RBS - RBPax.

Resultados:

	Retraso GDP de vuelos controlados
Global de vuelos	14 D 20 h 20 min 12 s
Medio por vuelo	53 min 03 s
Desv. Típica vlo.	1 h 02 min 15 s
Máx por vuelo.	6 h 35 min 48 s
Global por pasajeros.	808 D 20 h 07 min 36 s

Cuadro 3.18: Resultados de retrasos obtenidos tras combinar RBPax - RBS del GDP de vuelos controlados.

3.2.3. Comparativas y conclusiones

Como en el caso anterior, la combinatoria de políticas tiene sus pros y contras. Hace un balance entre lo mejor y lo peor de cada una complementándose. El retraso de los pasajeros se ve incrementado respecto a RBPax pero disminuido en gran medida respecto a RBS. Se observa también una mejoría de retraso por lo que respecta a las aeronaves tanto global como máximo si se compara con RBPax como única política a costa de empeorar estos dos parámetros si se compara con RBS.

Tras ver por separado la combinación de estas dos políticas, a continuación se muestran juntas para poder ver las diferencias obtenidas.

Retraso	**RBS - RBPax**	**RBPax - RBS**
Global de vuelos	14 D 20 h 17 min 00 s	14 D 20 h 20 min 12 s
Medio por vuelo	53 min 02 s	53 min 03 s
Desv. Típica vlo.	1 h 09 min 32 s	1 h 02 min 15 s
Máx por vuelo.	7 h 26 min 48 s	6 h 35 min 48 s
Global por pasajeros.	874 D 9 h 24 min 36 s	808 D 20 h 07 min 36 s

Cuadro 3.19: Resultados de retrasos GDP con combinación de politicas.

Si comparamos los dos casos como se representa en la tabla 3.19 se observa que el retrado globa de los vuelos, el retraso medio por vuelo y la desviación típica son casi idénticas. La gran diferencia se obtiene en el retraso máximo de los vuelos y el retraso global de los pasajeros.

Si se aplica RBPax antes que RBS, es decir, durante la reducción de capacidad del aeropuerto, se gana en retraso por pasajeros hasta casi 65 días. Además también se ve mejorado el retraso máximo que se produce en los vuelos en casi 1 hora.

Por tanto, se concluye con que la combinatoria RBPax - RBS es más óptima que RBS -RBPax.

Además, si se tienen miran respecto a las políticas aplicadas por separado, resultado es que se llega a un balance donde se tiene en cuenta a los pasajeros y a las aerolíneas y según como se mire, todos salen ganando.

CAPÍTULO 4. GESTIÓN DEL PROYECTO

En este capítulo se describe cómo se ha llevado la organización del TFC, la programación inicial, la final, una visión cualificativa de las horas dedicadas a lo largo del tiempo, y los programas y herramientas utilizadas para la realización del mismo.

4.1. Planificación del proyecto

Una de las primeras tareas que se realizaron fue plasmar la planificación del proyecto para seguir un orden y marcar unos hitos de entrega. Para ello, se usó un diagrama de Gantt.

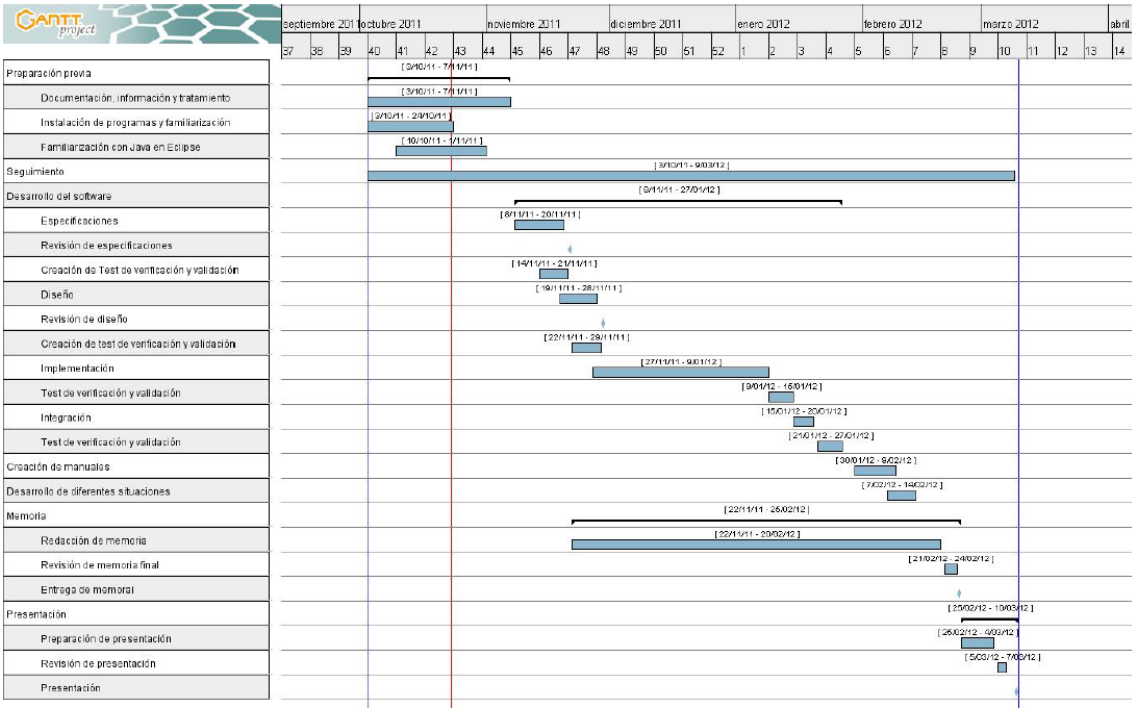

Figura 4.1: Planificación inicial del proyecto. Diagrama de Gantt

En la figura 4.1 se puede observar que se comenzó a trabajar en el proyecto el día 3 de octubre. Además, incialmente se había considerado que se presentaría al rededor de los días 8 ó 9 de marzo.

Sin embargo, debido a que el alumno asistía a clases y que además tuvo la oportuniad de realizar prácticas de empresa hasta mediados del mes de mayo, la realización del proyecto se fue posponiendo y la dedicación de tiempo no era plena.

No obstante, durante los primeros meses el tiempo empleado fue para la lectura de documentación relacionada con el GDP, instalación de programas y auto - aprendizaje de los mismos a parte de realizar algunos borradores de especificaciones y diseño del programa.

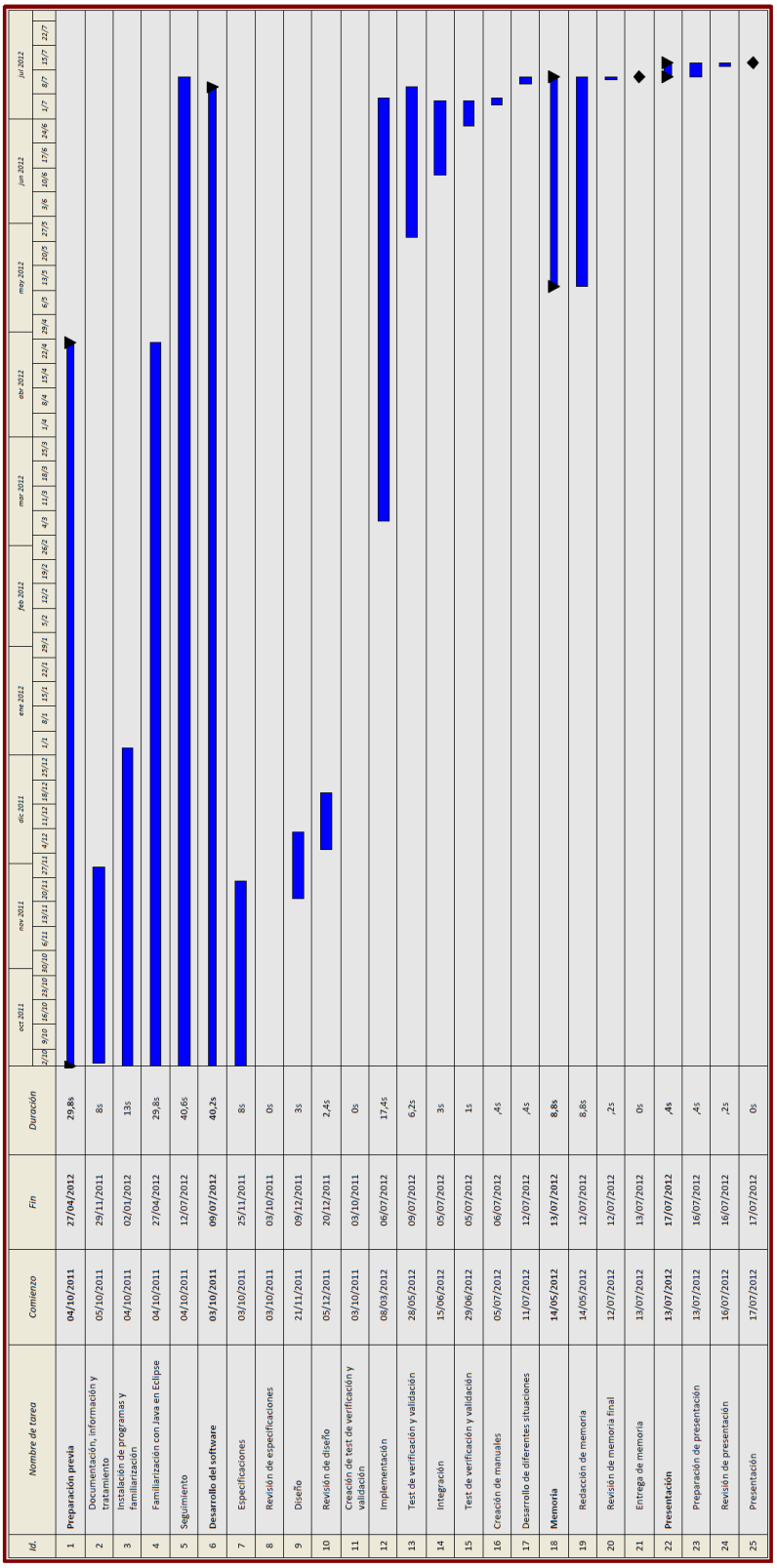

Figura 4.2: Planificación final del proyecto. Diagrama de Gantt

La planificación final quedó, de una forma orientativa, como se muestra en la figura 4.2. Se puede apreciar que a partir del 15 de mayo que es cuando el alumno finaliza las prácticas de empresa, vuelve a comenzar la actividad relacionada con el proyecto.

A parte, se puede observar que desde el principio, se ha llevado un seguimiento. Éste seguimiento consistió en la realización de reuniones periódicas entre el alumno y el director del TFC donde se comentaban los progresos, se solucionaban dudas y se asignaban nuevas tareas para hacer.

Después de cada reunión, se escribía un acta de cada una de ellas que están recogidas en el apéndice D de este TFC.

Por otro lado, también desde el inicio, se ha contabilizado el tiempo trabajado cada día. El resultado es que se han dedicado alrededor de 450 horas donde la gran parte de éstas se han realizado los tres últimos meses tal y como refleja el diagrama 4.3 obtenido de los datos gue se fueron guardando cada día de trabajo.

Cabe destacar que cada día se descontaba entre un 12 % del tiempo total empleado por considerar que era tiempo que se podía no dedicar por interrupciones u otros motivos.

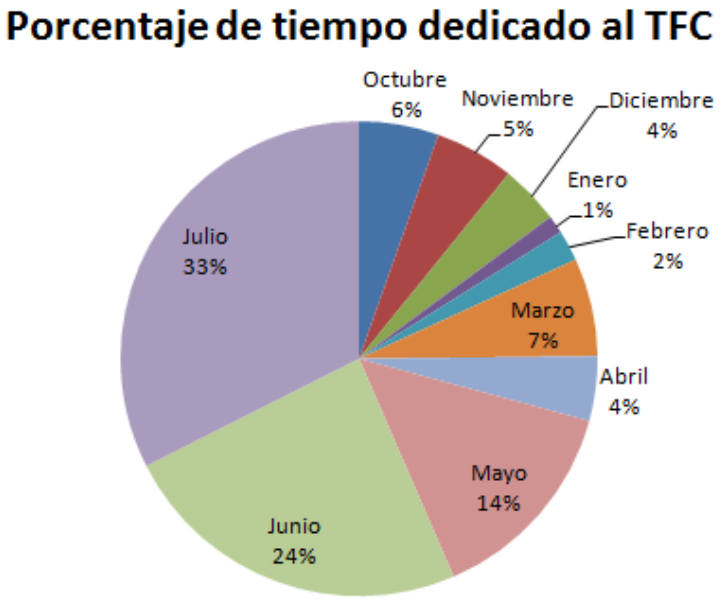

Figura 4.3: Porcentajes mensuales de horas dedicadas durante el TFC

Del gráfico 4.3, junto con el diagrama de Gantt de la planificación real llevada a cabo, se puede deducir que los últimos meses se produjo una alta carga de trabajo y se dedicaron muchas horas para que el trabajo saliera adelante y estuviera listo antes del día 20 de julio dado que era una fecha límite por motivos académicos.

4.2. Programas utilizados a lo largo del TFC

Los programas utilizados han sido varios de los cuales el alumno deconocía el funcionamiento e incluso existencia.

Internet, libros y tutoriales han servido de ayuda para el conocimiento y resolución de dudas durante la utilización de los mismos.

A continuación se enumeran dichos programas:

- **Eclipse versión Indigo:** Utilizado para realizar la programación en Java de ACREL.
- **Gantt project:** Aplicación para el diseño incial de la planificación del TFC mediante un diagrama de Gantt.
- **Mendeley:** Aplicación para gestionar toda la documentación gestionada, tenerla ordenada y poder generar la bibliografía de manera automática.
- **Microsoft Excel:** Este programa se ha usado para analizar datos de tráfico y guardar y calcular las horas de trabajo diarias.
- **Microsoft Visio 2010:** Se usó para generar los diagramas UML y la planificación final realizada del proyecto.
- **Micktex 2.29:** Implementador de código TEXpara la posible realización de esta memoria.
- **LEd:** Aplicación que usa Micktex para escribir en LATEXla memoria.
- **GSView 4.9:** Aplicación para visualizar imágenes y documentos en formato eps que son usadas en LATEX.
- **Adobe reader 9:** Visualizador de documentos pdf.

También hay que citar que para la redacción de la memoria del TFC en LATEXse ha usado la plantilla Versió: 1.2 con fecha 2006/06/14 de la EETAC creada para este propósito por el *Dr. Xavier Prats* .

CAPÍTULO 5. CONCLUSIONES

Cuando la demanda de un aeropuerto es superior a su capacidad, la manera de gestionarlo es aplicando lo que se conoce como GDP. Actualmente, la política de asignación de retrasos que se utiliza es RBS pero que según diversos estudios, no se trata de la más óptima en lo que respecta al retraso sufrido por los pasajeros. Por ello, investigadores de todo el mundo han propuesto nuevas políticas que mejoran este concepto o que dan prioridad a los vuelos dependiendo el tamaño de la aeronave, la distancia recorrida o el número de pasajeros para obtener otros resultados intentan

Para comparar algunas de las diferentes políticas, se ha creado una aplicación capaz de aplicar un GDP a un archivo de tráfico real donde se puede, además de modificar parámetros de la definición del GDP, asignar las diferentes políticas de manera individual y combinándola con otras eligiendo los tiempos de inicio y finalización.

El resultado obtenido tras varias simulaciones es que RBS es la política que menor retrasos otorga a las aeronaves y RBPax a los pasajeros. Además, se comprueba que la política RBID, propuesta por algunos de los profesores de la UPC mejora en todos los sentidos a RBD en el caso concreto estudiado.

A parte de todos los resultados teóricos obtenidos, también se puede concluir con que este TFC ha servido al alumno para aprender a realizar la gestión del diseño de un programa informático teniendo en cuenta requisitos, diseño de software y diferentes tipos de diagramas.

Además de aprender un nuevo lenguaje de programación y a cómo se gestiona y lleva a cabo un sistema, se ha aprendido un nuevo concepto de gestión del tráfico como es el GDP que hasta el inicio del TFC el alumno desconocía.

BIBLIOGRAFÍA

[1] Bengi Manley and Lance Sherry. The Impact of ground delay program rationing rules on passenger and airline equity. *2008 Integrated Communications, Navigation and Surveillance Conference*, pages 1–11, May 2008.

[2] Ken Howard, Rick Oiesen, and Michael C Wambsganss. Enhancements to the FAA Ground-Delay Program Under Collaborative Decision Making. 2001(February):57–74, 2001.

[3] Comunidad Wikipedia. http://en.wikipedia.org/wiki/Unix_time [07/12].

[4] Michael O Ball and Guglielmo Lulli. Ground Delay Programs : Optimizing over the Included Flight Set Based on Distance. pages 1–32.

[5] Joseph Rios and Kevin Ross. Delay Optimization for Airspace Capacity Management with Runtime and Equity Considerations. (August), 2007.

[6] Luis Delgado Muñoz and Jorge Ramírez Alcántara. Aviónica, Consideraciones Industriales 1. 2007.

[7] Software Engineering Standars Committee. *IEEE Recommended Practice for Software Requirements Specifications*, volume 1998. 1998.

[8] Programación Capas, Ricardo J Vargas, Del Valle, Universidad De Costa Rica, Ciencias De Computación, San José, Costa Rica, and Juan P Maltés Granados. Programación en Capas.

[9] Departamento Información Aeronáutica, Dirección de Tránsito Aéreo, and República Argentina. Designadores OACI de tipo de Aeronave. 2004.

[10] AENA. La aeronave. Tipos de aeronave. Convocatoria de controladores de la circulación aérea 2008. 2008.

[11] Luis Barrios Calmaestra. UNIDAD DIDÁCTICA: ESTADÍSTICA. DISTRIBUCIONES UNIDIMENSIONALES. [07/12].

[12] Wikipedia. Desviacion Estandar http://es.wikipedia.org/wiki/Desviación estandard - [07/12].

[13] http://www.vitutor.com/estadistica/descriptiva/a_16.html [07/12].

[14] Marc Melgosa. Estudio de Ground Delay Program y estrategias de reducción de velocidad. 2012.

Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

APÉNDICES

TÍTULO DEL TFC : Estudio de políticas de asignación de retraso para Ground Delay Program

TITULACIÓN: Ingeniería Técnica Aeronáutica, especialidad Aeronavegación

AUTOR: José María Marco Carpena

DIRECTOR: Luis Delgado Muñoz

FECHA: 17 de julio de 2012

APÉNDICE A. REQUISITOS

A.1. Introducción

Este anexo contiene los *requisitos de diseño de software* para la realización del software necesario para el TFC "*Estudio de políticas de asignación de retraso para Ground Delay Program*".

A.1.1. Propósito

El propósito de este documento es definir las especificaciones funcionales, de diseño, de rendimiento, de interfaz de usuario y de sistema para la creación de una aplicación que calculará retrasos según varias políticas de llegadas en la asignación de *slots* de forma única o combinada. Este documento está dirigido a todo aquel que desarrolle la aplicación, la utilice y/o quiera documentarse sobre ella.

A.1.2. Ámbito del sistema

ACREL (Aplicación de Cálculo de Retrasos En Llegadas) será una aplicación que tiene por objetivo tratar un archivo de vuelos simulados y obtener datos de retrasos según diferentes parámentros que el usuario defina. Con esta aplicación se pretende obtener los datos de retrasos de un fichero de tráfico aéreo según las diferentes políticas de asignación de *slots* en las llegadas como son *Ration By Schedule (RBS)*, *Ration By Distance (RBD)*, *Ration By Inverse Distance (RBID)*, *Ration By Passengers (RBPax)* y *Ration By Aircraft size (RBAc)* de manera individual y/o combinándolas entre ellas para posteriormente hacer un estudio de estos datos e intentar encontrar una manera de combinación de políticas óptima que minimice los retarsos asignados debido a un Graund Delay Program (GDP).

A.1.3. Definiciones, Acrónimos y Abreviaturas

A.1.4. Referencias

Para la realización de estas especificaciones de requisitos de software (SRS de sus siglas en inglés) se ha utilizado el documento siguiente:

Título: IEEE Recommended Practice for Software Requirements Specifications
Referencia: IEEE Std 830-1998
Fecha de puplicación: 20 de Octubre de 1998
Fecha de revisión: 09 de Diciembre de 2009

ACREL	Aplicación de Cálculo de Retrasos En Llegadas
GDP	Ground Delay Program
IEEE	Institute of Electrical and Electronics Engineers
RBAc	Ratio By Aircraft
RBD	Ratio By Distance
RBID	Ratio By Inverse Distance
RBPax	Ratio By Passengers
RBS	Ratio By Schedule
RD	Requisitos de Diseño
RF	Requisitos Funcionales
RIE	Requisitos de Interfaz Externa
RIU	Requisitos Interfaz de Usuario
RR	Requisitos de Rendimiento
RSD	Requisitos de Sistema - Disponibilidad
RSF	Requisitos de Sistema - Fiabilidad
RSM	Requisitos de Sistema - Mantenimiento
RSP	Requisitos de Sistema - Portabilidad
RSS	Requisitos de Sistema - Seguridad
SRS	Specifaction Requirement Software
TFC	Trabajo Final de Carrera

Autor: IEEE Computer Society

A.1.5. Visión general del documento

Los requisitos están organizados por temas. Además, cada uno de los requisitos está separado del resto y seguirán el siguiente modelo de tabla para cada uno de ellos.

Identificador	RE-01.00
Nombre	Requisito ejemplo
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá...

Cuadro A.1: Ejemplo de requisito

El identificador del requisito es un código que ayuda a la trazabilidad y seguimiento de los requisitos. Las letras indican el tipo de requisito que es y las cifras indica el número de requisitos. Las dos primeras cifras indica el requisito de alto nivel y las dos siguientes, después del punto, el número de requisito de bajo nivel perteneciente al de alto nivel.
El nombre del requisito indica de qué trata dicho requisito de una forma resumida.
Por otro lado, la prioridad establece la preferencia en tres niveles a la hora de cumplir dicho requisito según cómo de importante y necesario sea. Únicamente puede tener una de las tres prioridades disponibles. Un requisito con prioridad Alta/Imprescindible será aquél que debe cumplirse siempre para que la aplicación sea satisfactoria y disponga de lo

mínimo para funcionar y obtener los resultados esenciales y requeridos. Media/Deseado indica que ese requisito no es imprescindible pero que mejoraría la aplicación. La prioridad Baja/Opcional simboliza que se trata de una mejora de la aplicación pero que en caso de hacerla no tiene porqué ser exactamente como el requisito especifica.
Finalmente, el campo de descripción explica de una manera más detallada cómo es el requisito.

A.2. Descripcción general

Esta sección de SRS describe los factores generales que afectan a la aplicación y sus requisitos. Proporciona un *"background"* para los requisitos pero no los define.

A.2.1. Perspectiva del producto

El sistema realizado no estará diseñado para formar parte de otro más complejo. Inicialmente será una herramienta necesaria para el tratamiento de datos de tráfico aéreo que formará parte del TFC ***Estudio de políticas de asignación de retraso para Ground Delay Program*** en la realización de este. Por otra parte, la aplicación podría ampliarse, modificarse y usar en posteriores estudios en los que sea necesaria una aplicación con estas características.

A.2.2. Funciones del producto

La ACREL calculará el retraso que sufren vuelos a los que le afecta un GDP. Estos vuelos se obtendrán de un fichero de datos que será la entrada al programa.
Se podrá definir los parámetros del GDP como son la capacidad del aeropuerto de destino, las políticas que se van a aplicar al tráfico para la llegada así como definir los tiempos del GDP y las políticas de llegada. El resultado será un nuevo fichero de datos donde aparecerán los vuelos afectados por el GDP, el retraso que ha sufrido el vuelo y otros datos de interés.
Con la variación de parámetros de un mismo fichero de tráfico aéreo se esperan obtener diferentes valores de retrasos para un posterior estudio que forma parte del TFC "*Estudio de políticas de asignación de retraso para Ground Delay Program*"para el que estará diseñada ACREL.

A.2.3. Características de usuario

El programa estará dirigido principalmente al estudiante y director del TFC para la elaboración del mismo. No obstante, una vez evaluado el TFC dicho programa quedará disponible a toda aquella persona que quiera utilizarlo con fines educativos.

Dicha aplicación tendrá un nivel educacional, de experiencia y estudio técnico para personas que estén relacionadas con el tema del GDP, asignación de slots en llegadas y cálculo de retrasos.

A.2.4. Restricciones

Puesto que se tratará de un software de prueba para el tratamiento y obtención de datos, la aplicación estará restringida a realizar un procesado en el tiempo de ejecución, es decir, no se podrá procesar a la vez dos ficheros de datos.
Para su utilización, será necesario un ordenador de usuario básico. No será necesario altavoces, micrófono, escáner, webcam u otros dispositivos de entrada que no sean ratón y teclado.
Los requisitos de fiabilidad no serán restrictivos dado que se tratará de una aplicación para un estudio con fines educativos.
La realización de la aplicación será crítica en el sentido de que es necesaria para el desarrollo del TFC *Estudio de políticas de asignación de retraso para Ground Delay Program*. Una mala programación, funcionamiento o fallos inesperados no supondrá una catástrofe. No obstante, se buscará que el programa sea robusto, entendible, amigable y sobretodo útil para el propósito de su desarrollo.
Como consideraciones de seguridad, tendremos en cuenta que no se usará para aplicar y tomar decisiones según los datos obtenidos de la ACREL en la gestión actual de las operaciones de tráfico aéreo. Además, mientras el TFC no sea presentado, dicha aplicación no debe ser distribuída y solo debe tener acceso el director y autor de dicho TFC.

A.2.5. Suposiciones y dependencias

No se especifican.

A.2.6. Requisitos futuros

No se tienen previstos requisitos futuros.

A.3. Requisitos específicos

Aqui empieza la parte de requisitos en sí. Cada requisito estará organizado según corresponda a los siguientes apartados.

A.3.1. Requisitos de Interfaz Externa

Identificador	RIE-01.00
Nombre	Organización del fichero de tráfico
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	El archivo de tráfico deberá disponer de líneas de datos correspondientes a un vuelo.

Identificador	RIE-01.01
Nombre	Líneas fichero de tráfico
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	El fichero de tráfico deberá tener las líneas correspondientes a los datos de un vuelo seguidas de las otras con salto de línea sin dejar líneas en blanco.

Identificador	RIE-01.02
Nombre	Separación de datos de líneas del fichero de tráfico
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	Los diferentes datos correspondientes al vuelo de cada una de las líneas del fichero de tráfico deberán estar separadas por espacio.

Identificador	RIE-02.00
Nombre	Formato tiempos del fichero de tráfico
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	Los datos de fechas que aparezcan en el archivo de tráfico deberán aparecer en formato *unixTime*

Identificador	RIE-03.00
Nombre	Extensión del archivo de resultados
Prioridad	☐Alta/Imprescindible ☐Media/Deseado ✓Baja/Opcional
Descripción	Los archivos obtenidos de la aplicación deberán tener una extensión .txt

Identificador	RIE-04.00
Nombre	Líneas de archivo de resultados
Prioridad	☐Alta/Imprescindible ☐Media/Deseado ✓Baja/Opcional
Descripción	Los resultados del análisis deberán guardarse línea por línea correspondiendo cada una de ellas a un vuelo.

Identificador	RIE-04.01
Nombre	Separación entre argumentos de resultados
Prioridad	☐Alta/Imprescindible ☐Media/Deseado ✓Baja/Opcional
Descripción	La separación entre argumentos de cada una de las líneas correspondientes a un vuelo deberán estar espaciados por *tabulador*

A.3.1.1. Requisitos de Interfaz de Usuario

Identificador	RIU-01.00
Nombre	Ejecución por línea de comandos
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá poder ejecutarse por linea de comandos desde una consola como puede ser *cmd.exe de Windows*.

Identificador	RIU-01.01
Nombre	Contenido línea de comandos
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La línea de comandos contendrá: el nombre de la aplicación para ejecutar, la ruta hasta el fichero de tráfico que deseamos cargar, el nombre fichero de tráfico que deseamos cargar, el momento de inicio del tráfico a cargar, el momento de finalización de tráfico a cargar, la capacidad definida del aeropuerto sin y con GDP, el radio de acción del GDP, el momento cuando se define el GDP, el momento cuando comienza el GDP, el momento cuando finalizan las restricciones del GDP, el número *n* de políticas de llegadas que se va a aplicar, elección del tipo de política 1 que se va a escoger, el momento en el que la política 1 comenzará, el momento final donde la politica 1 terminará, (...) elección del tipo de política *n* que se va a escoger, el momento en el que la política *n* comenzará, el momento final donde la politica *n* terminará.

Identificador	RIU-01.02
Nombre	Distinción elementos en línea de comandos
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	Cada elemento definido en el *RIU-01.01* estará separado del siguiente mediante un espacio.

Identificador	RIU-01.03
Nombre	Formato tiempos en línea de comandos
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	El momento de inicio del tráfico a cargar, el momento de finalición de tráfico a cargar, el momento cuándo se define el GDP, el momento cuándo comienza el GDP, el momento cuándo finaliza el GDP, el momento en el que la política *n* comenzará y el momento final donde la politica *n* terminará deben estar en formato *Unix time*

Identificador	RIU-01.04
Nombre	Formato capacidades en línea de comandos
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La capacidad del aeropuerto sin GDP y la capacidad del aeropuerto con GDP deben ser números naturales (*1, 2, 3, ...*).

Identificador	RIU-01.05
Nombre	Formato radio de acción en línea de comandos
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	El formato para el radio de acción al que afecta el GDP deben ser números positivos (*1, 2.5, 3, 900.45 ...*).

Identificador	RIU-01.06
Nombre	Formato número de políticas en línea de comandos
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La cantidad de políticas definidas debe ser un número natural (*1, 2, 3, ...*).

Identificador	RIU-01.07
Nombre	Formato elección política en línea de comandos
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	La elección de la política de llegada debe ser un número natural comprendido entre el (*1 y 5*) ambos inclusive donde 1 será *Ration By Schedule (RBS)*, 2 *Ration By Passengers (RBPax)*, 3 *Ration By Distance (RBD)*, 4 *Ration By Inverse Distance (RBID)* y 5 *Ration By Aircraft (RBAc)*.

Identificador	RIU-02.00
Nombre	Ejecución por ventana
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá poder ejecutarse a través de una interfáz gráfica con ventanas

Identificador	RIU-02.01
Nombre	Ventana principal de la aplicación
Prioridad	☐Alta/Imprescindible ☐Media/Deseado ✓Baja/Opcional
Descripción	La ventana principal de la aplicación debe parecerse a la de la figuraA.1.

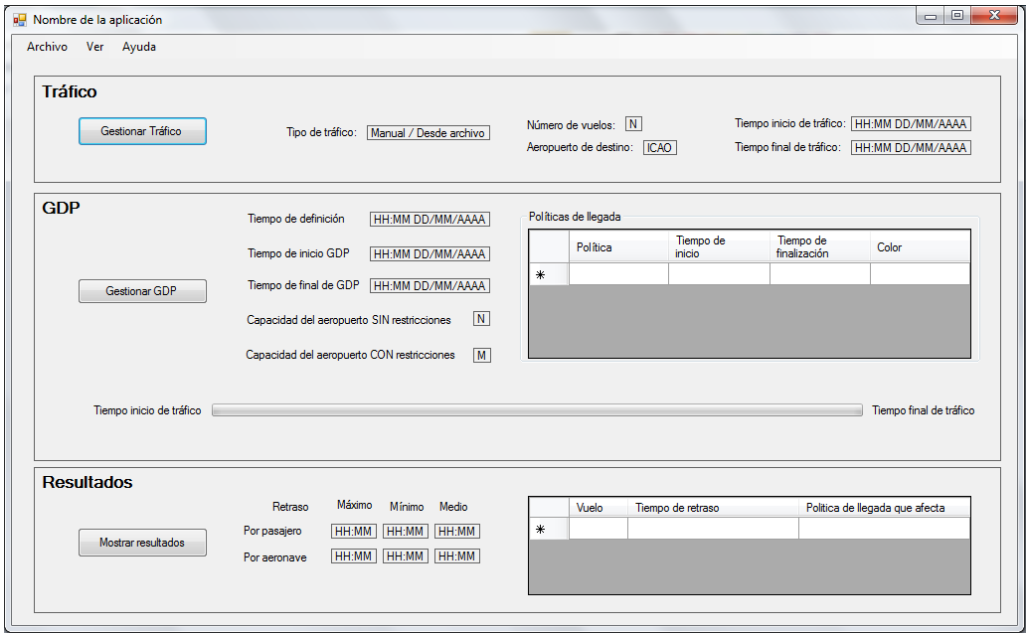

Figura A.1: Ejemplo de ventana principal

Identificador	RIU-02.02
Nombre	Ventana principal de la aplicación
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	La ventana principal de la aplicación debe contener tres apartados diferenciados que serán: 1) Tráfico; 2) GDP y 3)Resultados

Identificador	RIU-02.03
Nombre	Ventana principal de la aplicación - datos
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	La ventana principal de la aplicación solo debe mostrar datos no pudiendo modificarlos directamente.

Identificador	RIU-02.04
Nombre	Carga de tráfico
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	La ventana principal de la aplicación debe contener un botón que abra una ventana secundaria para cargar un tráfico desde un archivo específico.

Identificador	RIU-02.05
Nombre	Gestionar GDP
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	La ventana principal de la aplicación debe contener un botón que abra una ventana secundaria que permita gestionar el GDP.

Identificador	RIU-02.06
Nombre	Ventana secundaria gestión de GDP
Prioridad	☐Alta/Imprescindible ☐Media/Deseado ✓Baja/Opcional
Descripción	La ventana secundaria de gestión del GDP debe parecerse a la de la figura A.2.

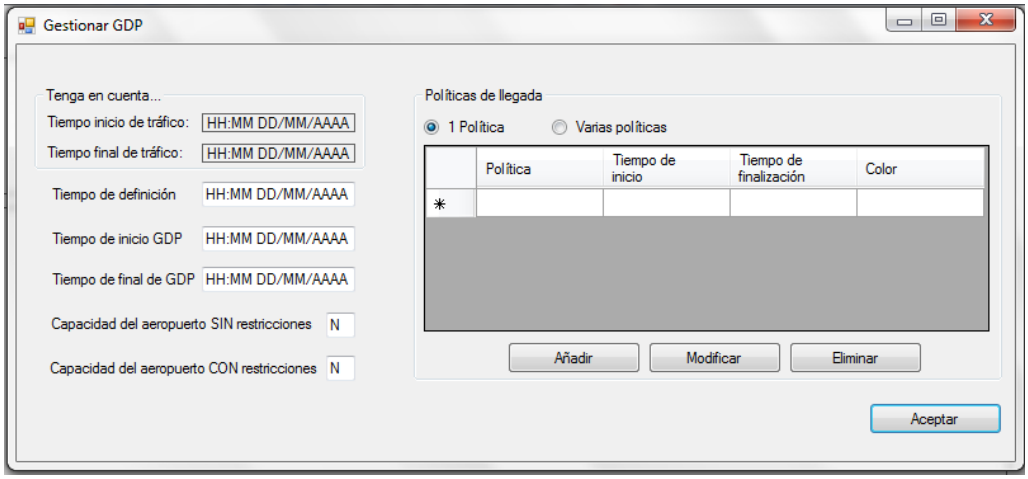

Figura A.2: Ejemplo ventana Gestionar GDP.

Identificador	RIU-02.07
Nombre	Ventana secundaria gestión de GDP - Datos
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	La ventana secundaria de gestión del GDP debe poder modificar los momentos de definición del GDP y la capacidad del aeropuerto de destino con y sin restricciones.

Identificador	RIU-02.08
Nombre	Ventana secundaria gestión de GDP - Políticas
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	Desde la ventana secundaria de gestión del GDP se debe poder añadir, modificar y borrar nuevas políticas de asignación de slots para el GDP.

Identificador	RIU-02.09
Nombre	Ventana secundaria gestión de GDP - Visión de políticas
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	La ventana secundaria de gestión del GDP debe mostrar un listado ordenado en orden cronológico de las políticas de asignación de slots indicando al menos el tipo de política y el momento de comienzo y término de cada una.

Identificador	RIU-02.10
Nombre	Activación botones gestionar GDP y calcular
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	El botón de gestionar GDP y el de calcular retrasos de la ventana principal no se activarán hasta que se haya cargado un tráfico concreto.

Identificador	RIU-02.11
Nombre	GDP por defecto
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	Una vez cargado el tráfico, el GDP debe tener unos parámetros asignados por defecto.

Identificador	RIU-03.00
Nombre	Mostrar resultados
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	Los resultados de tiempos de retraso máximo, mínimo y medio por pasajero y por vuelo deben poder mostrarse.

Identificador	RIU-03.01
Nombre	Mostrar lista final de llegada de vuelos
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	La aplicación debe poder mostrar una lista de vuelos ordenados cronológicamente a su llegada indicando, si es así, el retraso que ha sufrido.

Identificador	RIU-03.02
Nombre	Detalle del vuelo
Prioridad	☐Alta/Imprescindible ☐Media/Deseado ✓Baja/Opcional
Descripción	Se deben poder visualizar los datos de un vuelo específico como número de vuelo, número de pasajeros, cost index, hora de salida programada, hora de llegada programada, hora de llegada real, tiempo de retraso y política GDP que le afectó de la lista de vuelos ordenados cronológicamente.

Identificador	RIU-04.00
Nombre	Fondo de la aplicación
Prioridad	☐Alta/Imprescindible ☐Media/Deseado ✓Baja/Opcional
Descripción	El fondo de las ventanas de la aplicación será de color gris.

Identificador	RIU-05.00
Nombre	Tipo de letra
Prioridad	☐Alta/Imprescindible ☐Media/Deseado ✓Baja/Opcional
Descripción	Las letras de las ventanas de la aplicación serán de tipo Arial, tamaño 12 y color negro.

A.3.2. Requisitos Funcionales

En esta sección se especifican lo que el sistema debe hacer.

Identificador	RF-01.00
Nombre	Gestión de tráfico
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá gestionar datos de tráfico

Identificador	RF-01.01
Nombre	Generar tráfico desde archivo
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá generar datos de tráfico a partir de un archivo del simulador FACET

Identificador	RF-01.02
Nombre	Generar tráfico manual
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá permitir generar datos de tráfico de forma manual por el usuario.

Identificador	RF-01.03
Nombre	Generar tráfico según aeropuerto de destino
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá permitir al usuario elegir el tráfico que se generará según el aeropuerto de destino.

Identificador	RF-01.04
Nombre	Generar tráfico según características de aviones
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá permitir al usuario elegir el tráfico que se generará según las características de los aviones como número de asientos y range máximo.

Identificador	RF-01.05
Nombre	Generar tráfico según intervalos de tiempo
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá permitir elegir al usuario la hora a la que comienza el tráfico y la hora a la que acaba.

Identificador	RF-01.06
Nombre	Guardar tráfico
Prioridad	☐Alta/Imprescindible ☐Media/Deseado ✓Baja/Opcional
Descripción	La aplicación deberá permitir al usuario guardar un archivo de tráfico generado por la aplicación.

Identificador	RF-01.07
Nombre	Cargar tráfico
Prioridad	☐Alta/Imprescindible ☐Media/Deseado ✓Baja/Opcional
Descripción	La aplicación deberá permitir al usuario elegir un archivo de tráfico generado anteriormente por la aplicación.

Identificador	RF-02.00
Nombre	Aplicar GDP a tráfico
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá aplicar un GDP a un tráfico determinado.

Identificador	RF-03.00
Nombre	Gestionar GDP
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá permitir al usuario definir un GDP.

Identificador	RF-03.01
Nombre	Gestionar horario definición GDP
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá permitir al usuario definir cuándo se define el GDP.

Identificador	RF-03.02
Nombre	Gestionar horario comienzo GDP
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá permitir al usuario definir cuándo comienza el GDP.

Identificador	RF-03.03
Nombre	Gestionar horario finalización de restriccioens del GDP
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá permitir al usuario definir cuándo terminan las restricciones del GDP.

Identificador	RF-03.04
Nombre	Gestionar política de llegadas
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá permitir al usuario definir la política de asignación de slots de llegadas al aeropuerto.

Identificador	RF-03.05
Nombre	RBS
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá permitir al usuario definir la política de llegadas según RBS (*Ratio By Schedule*).

Identificador	RF-03.06
Nombre	RBD
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá permitir al usuario definir la política de llegadas según RBD (*Ratio By Distance*).

Identificador	RF-03.07
Nombre	RBID
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá permitir al usuario definir la política de llegadas según RBID (*Ratio By Inverse Distance*).

Identificador	RF-03.08
Nombre	RBPax
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá permitir al usuario definir la política de llegadas según RBPax (*Ratio By number of Passangers*).

Identificador	RF-03.09
Nombre	RBAc
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá permitir al usuario definir la política de llegadas según RBAc (*Ratio By model of Aircraft*).

Identificador	RF-03.10
Nombre	Definir capacidad del aeropuerto con GDP
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá permitir al usuario definir la capacidad declarada del aeropuerto de destino para el GDP activo.

Identificador	RF-03.11
Nombre	Definir capacidad del aeropuerto sin GDP
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá permitir al usuario definir la capacidad declarada del aeropuerto de destino para el GDP NO activo.

Identificador	RF-04.00
Nombre	Análisis y gestión de resultados
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá calcular y mostrar los retrasos del tráfico.

Identificador	RF-04.01
Nombre	Retraso de cada vuelo
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá calcular el retraso para cada vuelo.

Identificador	RF-04.02
Nombre	Retraso medio
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá calcular el retraso medio del tráfico.

Identificador	RF-04.03
Nombre	Retraso máximo
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá calcular el retraso máximo del tráfico.

Identificador	RF-04.04
Nombre	Retraso mínimo
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá calcular el retraso mínimo del tráfico.

Identificador	RF-04.05
Nombre	Retraso medio Pax
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá calcular el retraso medio de los pasajeros.

Identificador	RF-04.06
Nombre	Retraso máximo Pax
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá calcular el retraso máximo de los pasajeros.

Identificador	RF-04.07
Nombre	Retraso mínimo Pax
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá calcular el retraso mínimo de los pasajeros.

Identificador	RF-05.00
Nombre	Guardar estadísticas
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá poder elegir guardar los datos calculados de retrasos.

A.3.3. Requisitos de Rendimiento

Identificador	RR-01.00
Nombre	Tiempo de cálculo
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación debe mostrar los resultados requeridos desde el momento de la solicitud del cálculo en un tiempo máximo de 45 segundos.

A.3.4. Requisitos de Diseño

Identificador	RD-01.00
Nombre	Lenguaje de programación
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá estar programada en JAVA.

Identificador	RD-02.00
Nombre	Código de programación
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá estar programada siguiendo una *programación orientada a objetos*.

Identificador	RD-03.00
Nombre	Modo de programación
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá estar programada por capas de*Interfaz Hombre Máquina, Dominio y Base de Datos*.

A.3.5. Requisitos del Sistema

A.3.5.1. Fiabilidad

Identificador	RSF-01.00
Nombre	Fallo al iniciar la aplicación
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	La aplicación debe iniciarse correctamente 99 de cada 100 veces.

Identificador	RSF-01.01
Nombre	Aviso de fallo
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	La aplicación debe avisar en caso de que se hayan producido fallos al iniciarse.

Identificador	RSF-02.00
Nombre	Fallo al cargar fichero de tráfico
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	La aplicación debe cargar el fichero de tráfico correctamente 99 de cada 100 veces.

Identificador	RSF-02.01
Nombre	Aviso de fallo al cargar tráfico
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	La aplicación debe avisar en caso de que se hayan producido fallos al cargar el fichero de tráfico.

Identificador	RSF-02.02
Nombre	Aviso de error de fichero al cargar tráfico
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	La aplicación debe avisar en caso de que no pueda cargar el fichero de tráfico por error en los datos de este.

Identificador	RSF-03.00
Nombre	Fallo al calcular
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	La aplicación debe calcular los retrasos correctamente 999 de cada 1000 veces que se ejecute.

Identificador	RSF-03.01
Nombre	Aviso de fallo al calcular
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	La aplicación debe avisar en caso de que se hayan producido problemas al hacer el cálculo de retrasos.

A.3.5.2. Disponibilidad

Identificador	RSD-01.00
Nombre	Acceso público
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	La aplicación debe poder ser utilizada por todo el público que disponga o tenga acceso a ella (No habrá acceso por contraseña).

A.3.5.3. Seguridad

No se especifican requisitos de este tipo.

A.3.5.4. Mantenimiento

Identificador	RSM-01.00
Nombre	Código documentado
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	El código de la aplicación debe estar documentado y comentado.

Identificador	RSM-02.00
Nombre	Código ampliable
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	El código de la aplicación debe poder ampliarse si se requiere.

A.3.5.5. Portabilidada

Identificador	RSP-01.00
Nombre	Ejecución en Windows
Prioridad	✓Alta/Imprescindible ☐Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá poder ser ejecutada en Windows XP o versiones posteriores.

Identificador	RSP-02.00
Nombre	Ejecución en Linux
Prioridad	☐Alta/Imprescindible ✓Media/Deseado ☐Baja/Opcional
Descripción	La aplicación deberá poder ser ejecutada en Linux.

A.3.6. Otros Requisitos

No se especifican.

APÉNDICE B. DIAGRAMAS UML

B.1. Introducción

Este apéndice contiene los diagramas de *casos de uso*, *clases* y *secuencias* de la aplicación *ACREL* creada como herramienta para el TFC "*Estudio de políticas de asignación de retraso para Ground Delay Program*".

B.1.1. Propósito

El objetivo de este documento es dar una visión general, siguiendo un patrón UML *Lenguaje Modelado Unificado*, de lo que se puede hacer con la aplicación y una perspectiva esquemática y notación gráfica de la arquitectura de software con la representación de clases que forman el programa y algunas secuencias de ejecución. Este documento está dirigido a toda aquella persona que deseé documentarse sobre la aplicación con objetivos de usarla, ampliarla o simplemente ver cómo está diseñada.

B.1.2. Definiciones, Acrónimos y Abreviaturas

ACREL	Aplicación de Cálculo de Retrasos En Llegadas
GDP	Ground Delay Program
IEEE	Institute of Electrical and Electronics Engineers
IHM	Interfáz Hombre Máquina
TFC	Trabajo Final de Carrera

B.1.3. Referencias

Para la realización de este documento se ha seguido como referencia:

Título: IEEE Recommended Practice for Software Requirements Specifications
Referencia: IEEE Std 830-1998
Fecha de puplicación: 20 de Octubre de 1998
Fecha de revisión: 09 de Diciembre de 2009
Autor: IEEE Computer Society

B.1.4. Visión general del documento

Los diagramas están separados en tres apartados:

- Diagramas de casos de uso
- Diagramas de clases
- Diagramas de secuencias

A su vez, los diagramas de casos de uso parten de una visión general de alto nivel y continuan con diagramas que representan los requisitos funcionales de la aplicación.

Por otro lado, los diagramas de clases están separados por capas y sus transacciones tal y como está constituida la aplicación.

Finalmente se representan algunas iteraciones temporales que se puede llevar a cabo en la aplicación en forma de diagramas de secuencias.

B.2. Diagramas de casos de uso

B.2.1. Casos de uso de alto nivel

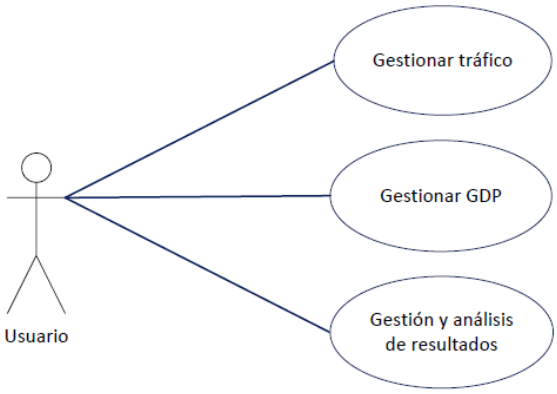

Figura B.1: Casos de uso alto nivel

B.2.2. Casos de uso de bajo nivel

B.2.2.1. Gestionar Tráfico

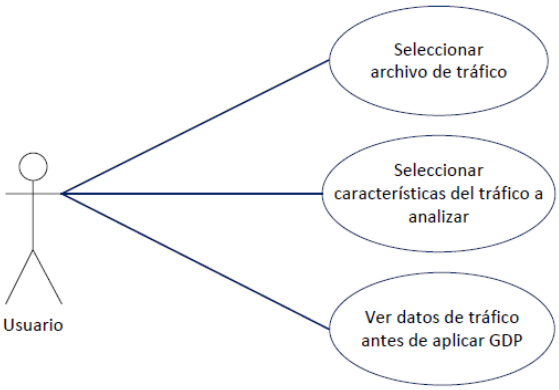

Figura B.2: Gestionar Trafico

B.2.2.2. Gestionar GDP

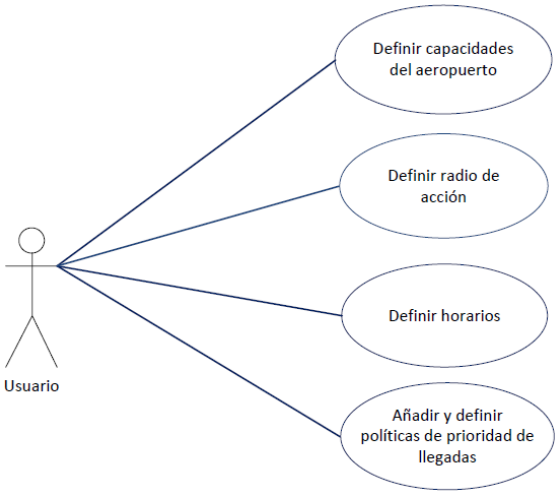

Figura B.3: Gestionar GDP

B.2.2.3. Gestión y análisis de resultados

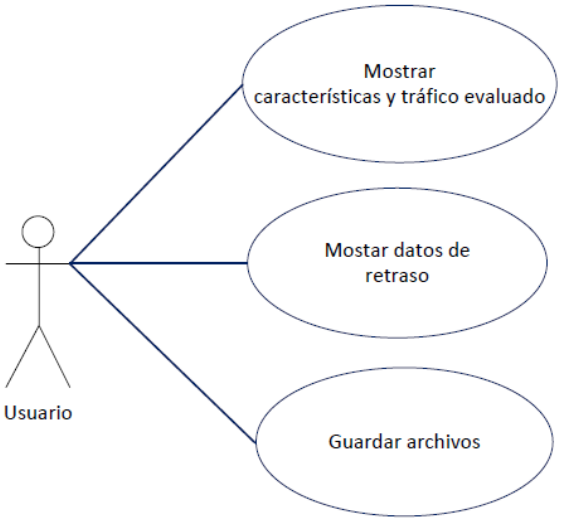

Figura B.4: Gestionar resultados

B.3. Diagramas de clases

B.3.1. Interfaz Hombre Máquina

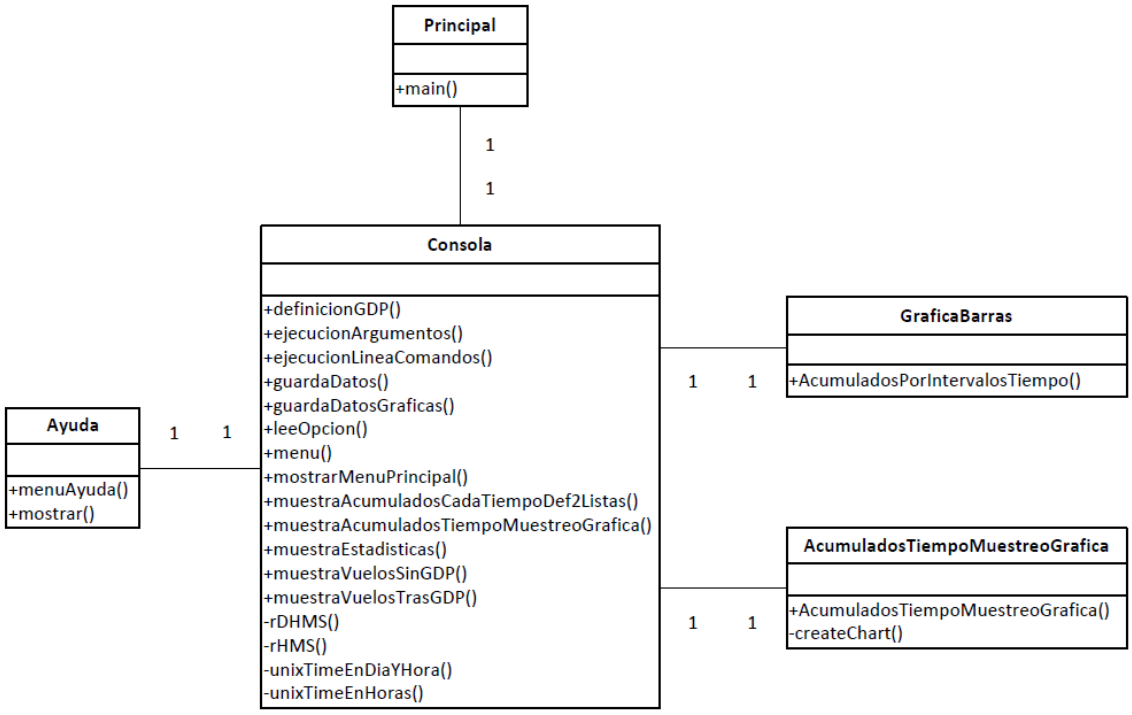

Figura B.5: IHM

B.3.2. IHM - Domininio

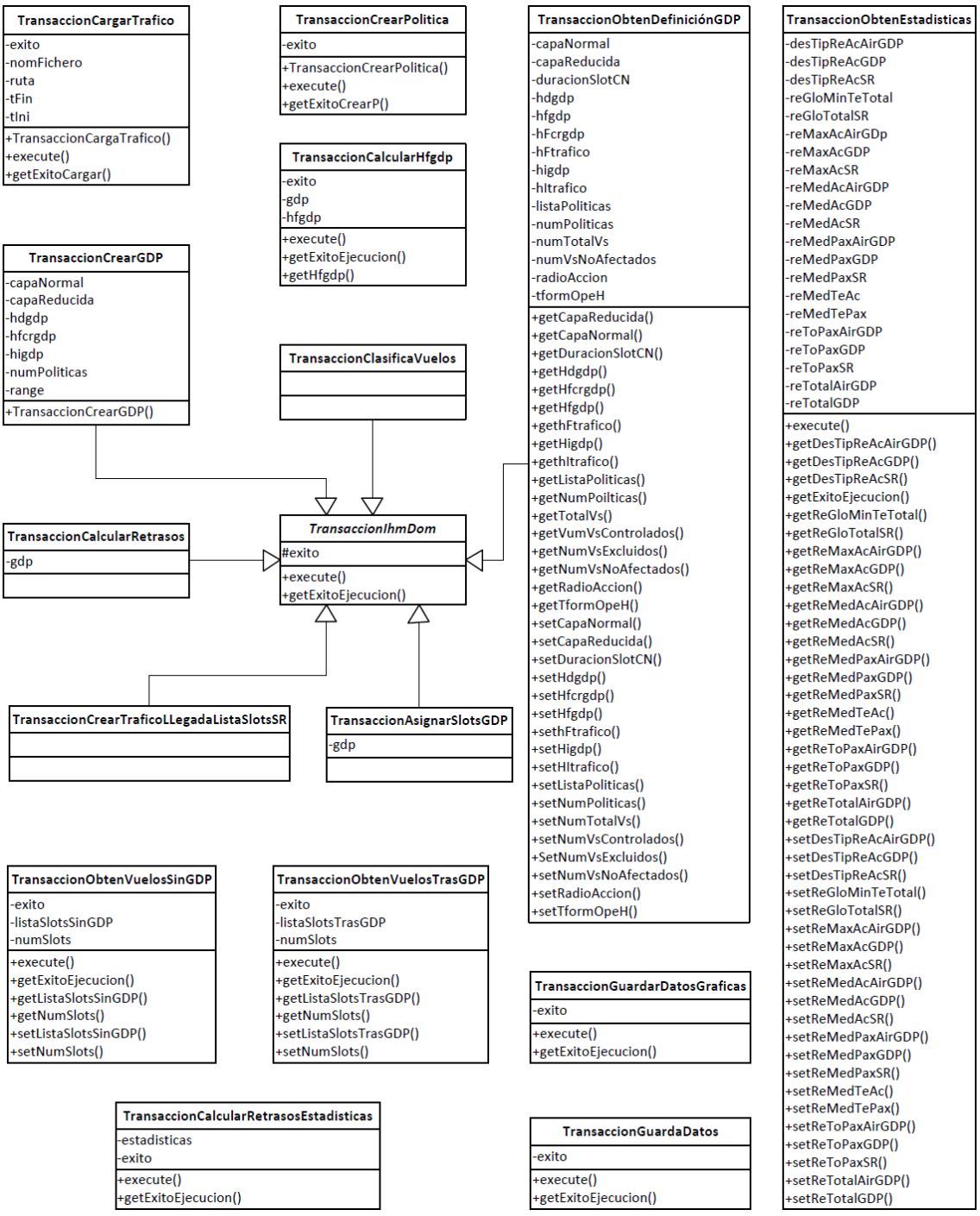

Figura B.6: Transacciones IHM - Dominio

B.3.3. Dominio

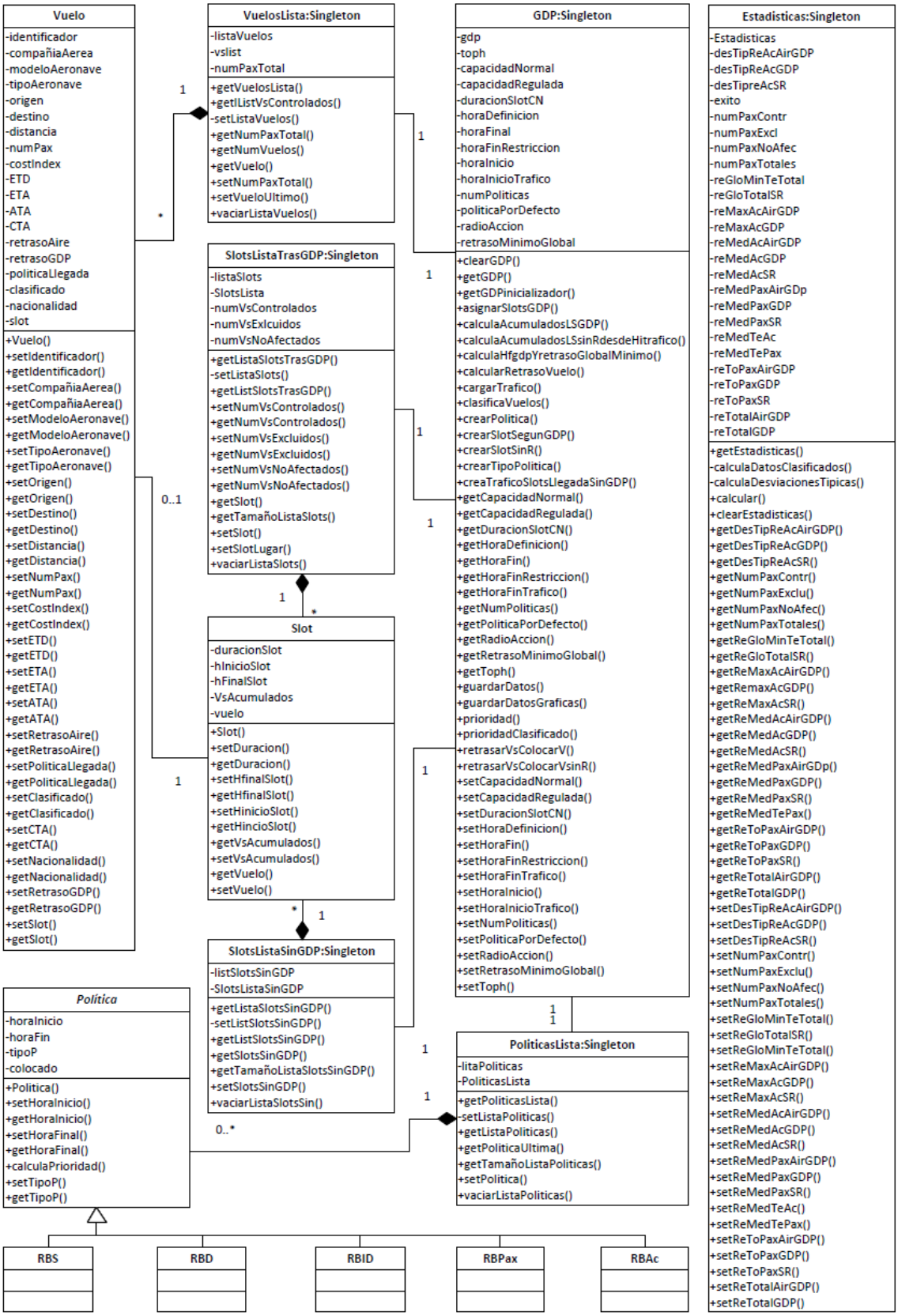

Figura B.7: Dominio

B.3.4. Dominio - Base de datos

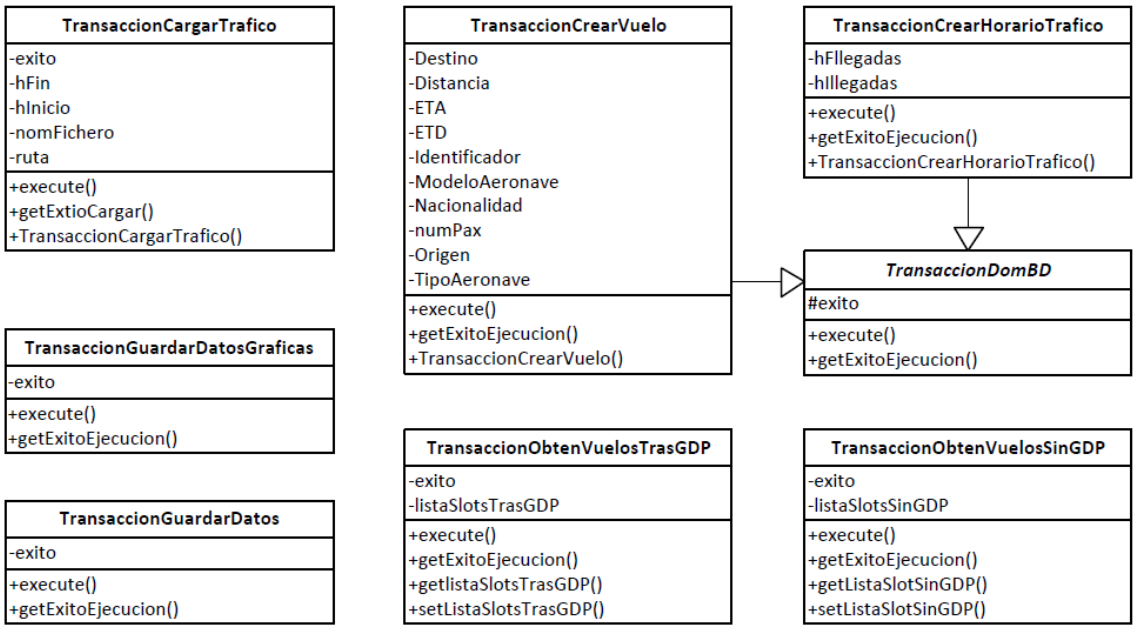

Figura B.8: Transacciones Dominio - Base de datos

B.3.5. Base de datos

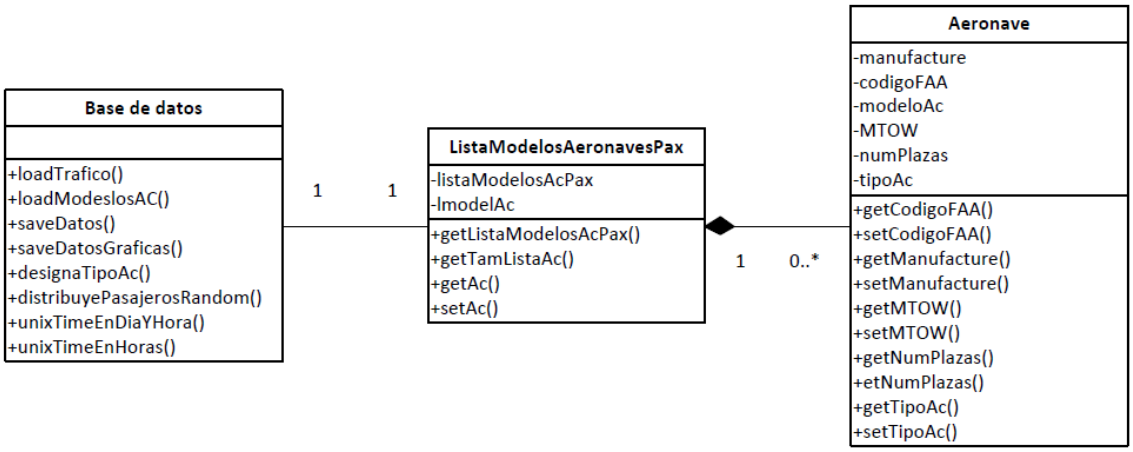

Figura B.9: Base de datos

B.4. Diagramas de secuencia

B.4.1. Cargar Tráfico

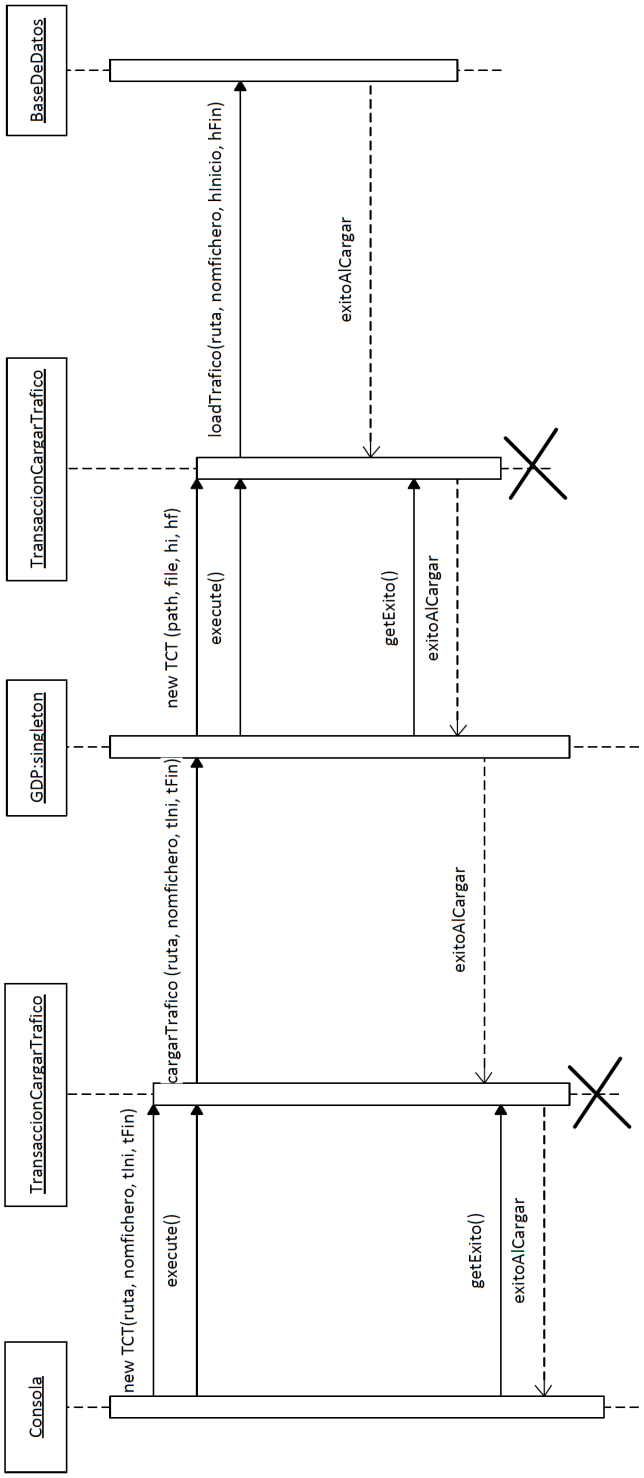

Figura B.10: Cargar tráfico (parte 1)

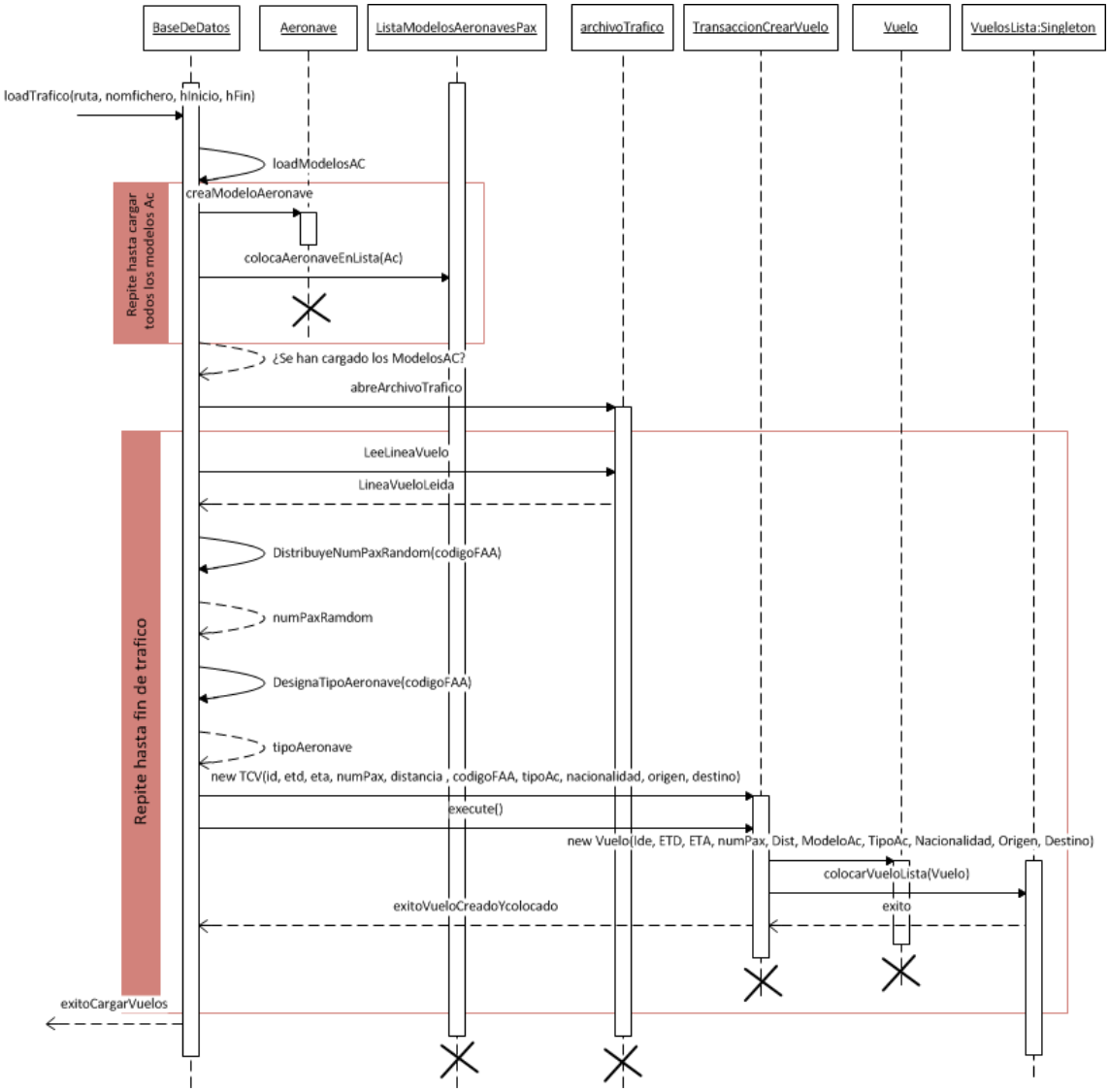

Figura B.11: Cargar tráfico (parte2)

B.4.2. Crear GDP

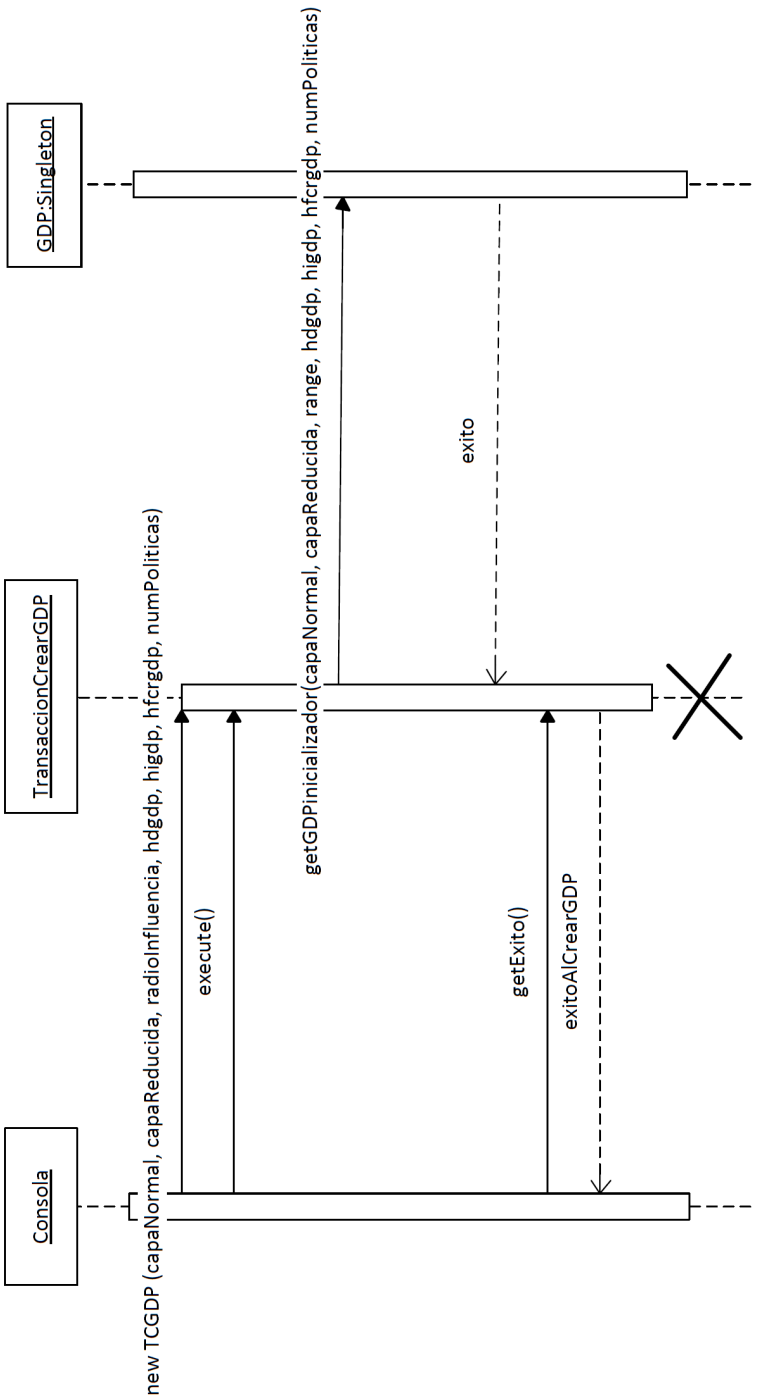

Figura B.12: Definir y crear GDP

B.4.3. Crear política

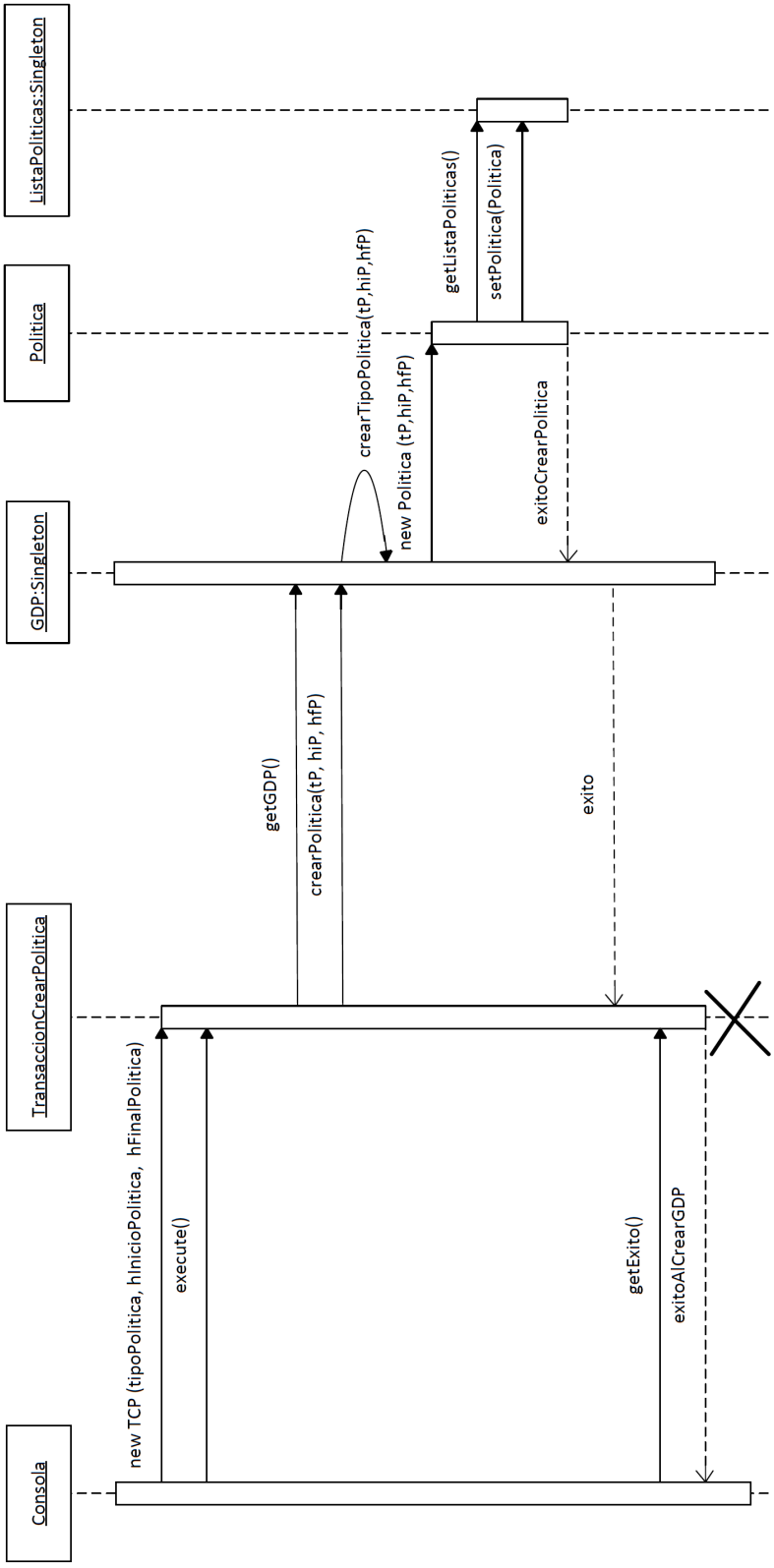

Figura B.13: Crear política

B.4.4. Guardar datos

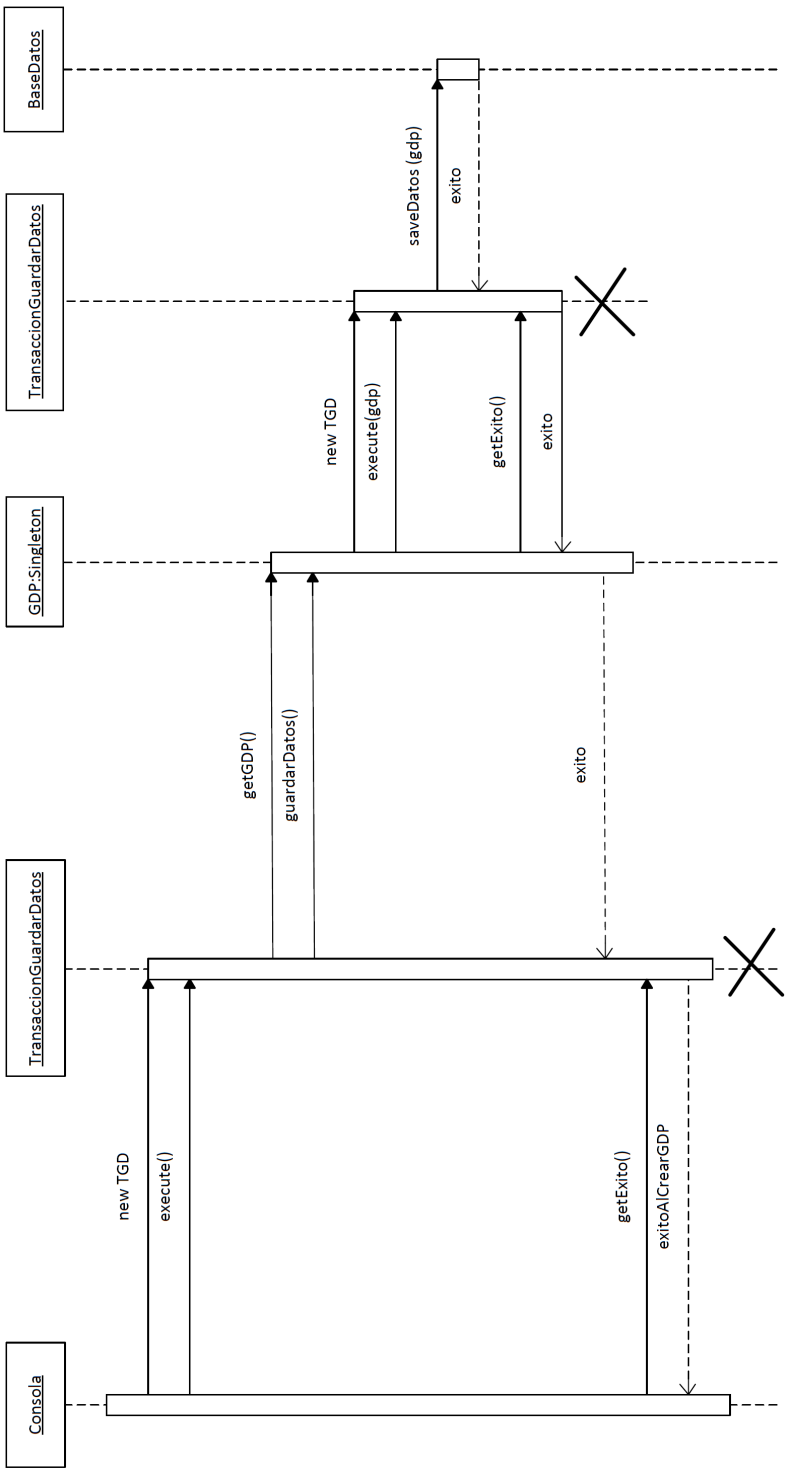

Figura B.14: Guardar datos (parte1)

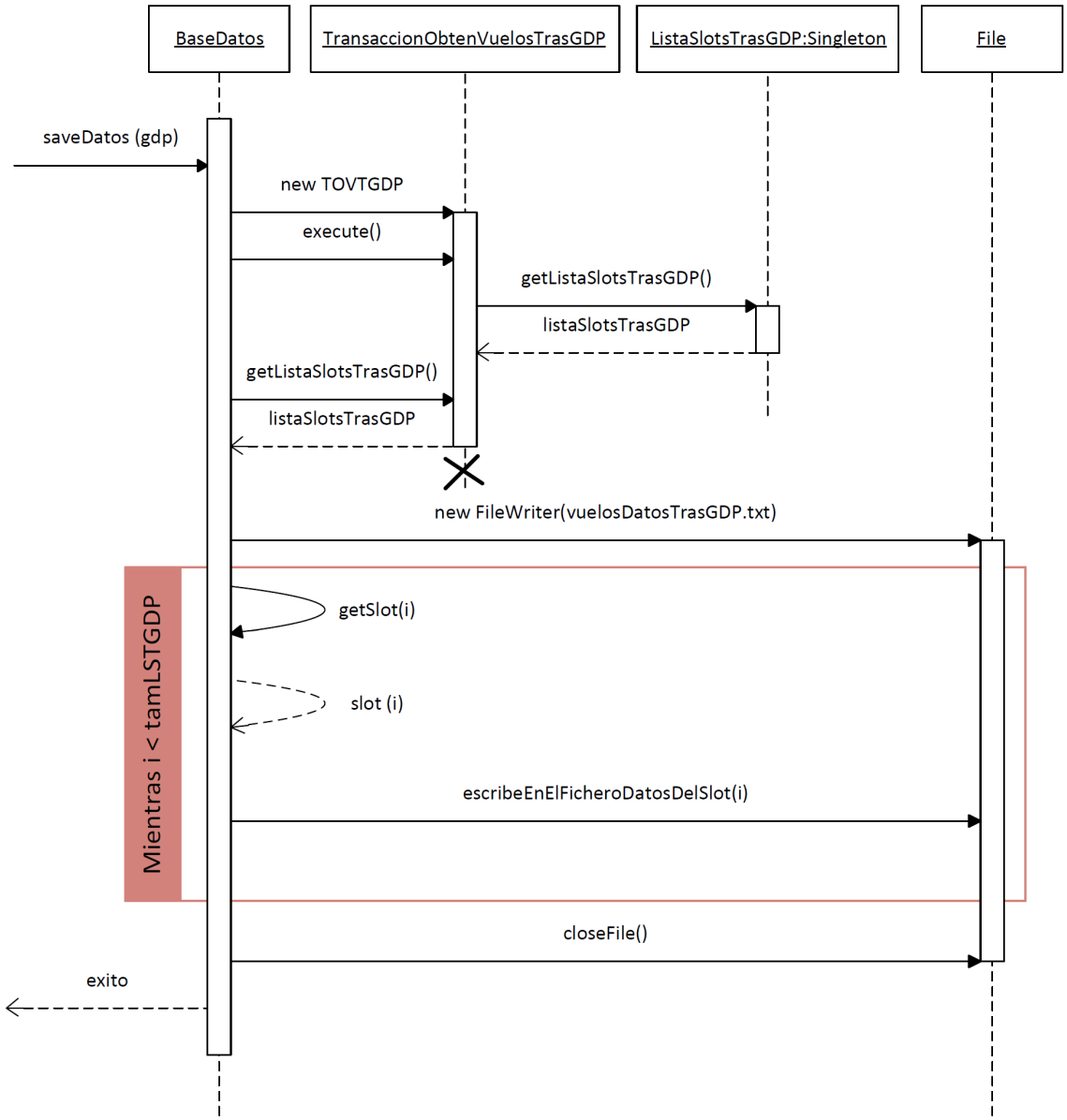

Figura B.15: Guardar datos (parte2)

B.4.5. Asignar slots despues de aplicar GDP

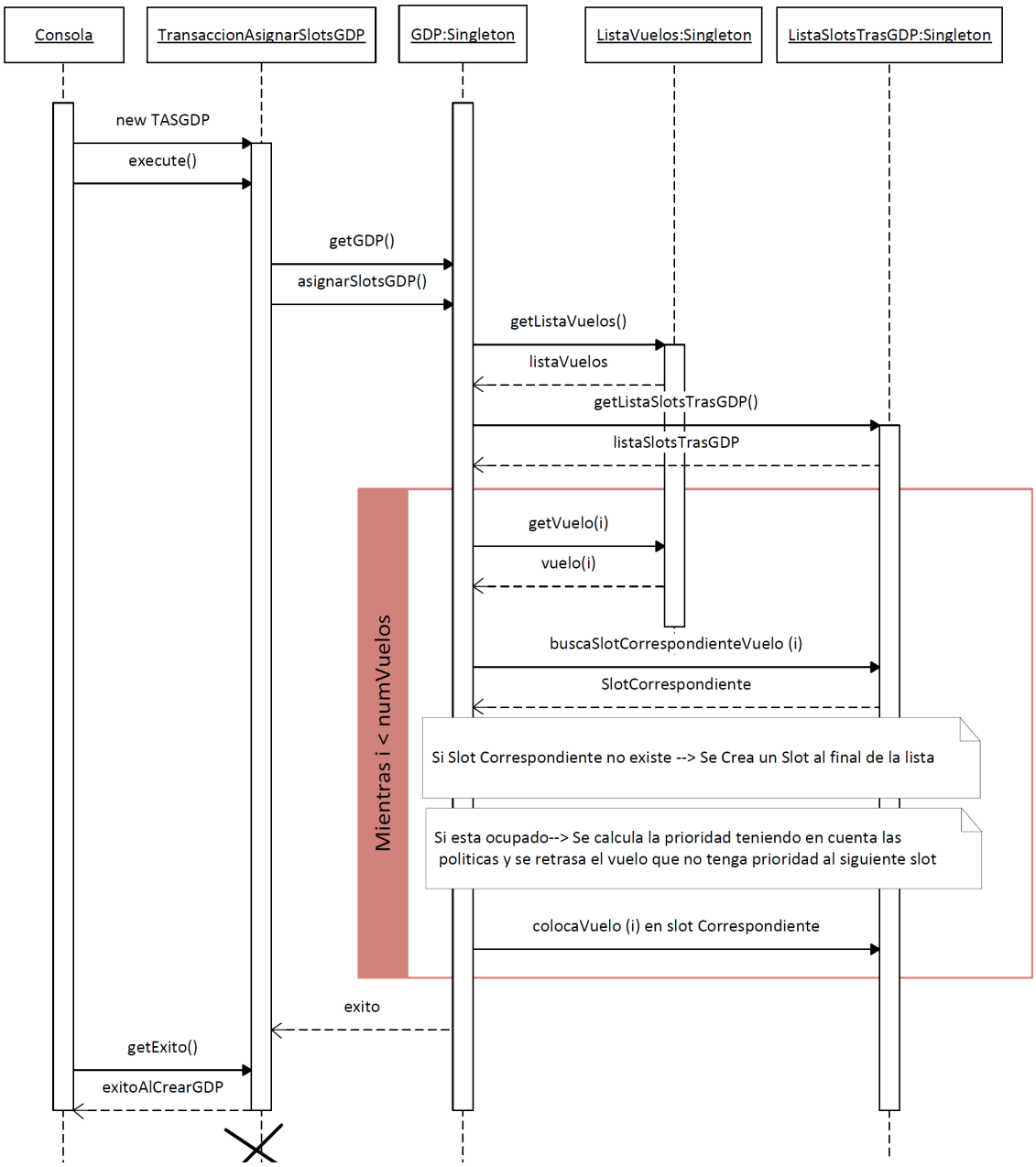

Figura B.16: Asignar slot despues de aplicar GDP

APÉNDICE C. GUÍA DE USO DE *ACREL*

C.1. Introducción

Este apéndice contiene una guía rápida para la utilización y ayuda de la aplicación *ACREL* creada como herramienta para el TFC ***'Estudio de políticas de asignación de retraso para Ground Delay Program'***.

C.1.1. Propósito

El propósito fundamenatal de este apéndice es proporcionar una guía rápida para la ejecución de la aplicación anteriormente citada. Además se puede encontrar una breve descripción de cada opción que posee el menú principal del programa así como una pequeña observación de cada proceso que se llevan a cabo durante el cálculo principal. Finalmente se enumeran algunos de los posibles errores con sus respectivas soluciones.

Este documento está dirigido a toda aquella persona que vaya a utilizar la aplicación *ACREL* por primera vez, necesite ayuda durante su ejecución, quiera documentarse para tener una visión general de cuál es su funcionamiento y/o tenga intenciones de ampliarla.

C.1.2. Definiciones, Acrónimos y Abreviaturas

ACREL	Aplicación de Cálculo de Retrasos En Llegadas
BD	Base de datos
GDP	Ground Delay Program
IEEE	Institute of Electrical and Electronics Engineers
IHM	Interfáz Hombre Máquina
TFC	Trabajo Final de Carrera

C.1.3. Referencias

Para la realización de este documento se ha seguido como referencia:

Título: IEEE Recommended Practice for Software Requirements Specifications
Referencia: IEEE Std 830-1998
Fecha de puplicación: 20 de Octubre de 1998
Fecha de revisión: 09 de Diciembre de 2009
Autor: IEEE Computer Society

C.1.4. Visión general del documento

Esta guía de ejecución y ayuda de *ACREL* está separada en los siguientes apartados:

- Instalación de la aplicación y archivos adjuntos
- ¿Cómo ejecutar la aplicación?
- Descripción de línea de ejecución
- Visión rápida de las opciones del menú
- Descripción rápida de los procesos del cálculo principal
- Posibles fallos durante la ejecución y sus soluciones

C.2. Instalación de la aplicación y archivos adjuntos

La aplicación puede ser ejecutada de dos formas diferentes, a través de un editor *Java* como es *Eclipse* donde se puede ver y modificar todo el código de las clases que forman la aplicación o desde una consola como puede ser *cmd.exe* de *Windows* donde directamente se ejecuta. Cada uno de ellos usarán unos archivos diferentes.

C.2.1. Instalación para ejecución directa

Para la instalación de la aplicación de ejecución directa es necesario descomprimir y copiar la carpeta *ACREL* incluida en el CD de la memoria del TFC en algún directorio de nuestro ordenador. Esta carpeta debe contener los siguientes archivos:

- ***Acrel*** Es el archivo principal con extensión *.jar* y contiene todo el código compilado
- ***Acrel_lib*** Es una carpeta que contiene librerías que usa ACREL
- ***baseDatosAeronaves*** Es un archivo de texto con extensión *.txt* que utiliza ACREL para proporcionar información complementaria y datos reales de aeronaves a la hora de cargar el archivo de tráfico.
- ***trafico*** Es el archivo de tráfico por defecto perteneciente a los días 24 y 25 de agosto de 2005 del *Aeropuerto Internacional Libertad de Newark* (KEWR) de Nueva York. Tiene una extensión *.dat*.

C.2.2. Instalación de la aplicación para ejecucion desde editor *Java*

En este caso es necesario descomprimir y copiar la carpeta *ACREL_codigo* incluida en el CD de la memoria del TFC en algún directorio de nuestro ordenador. Esta carpeta contiene dos carpetas dos carpetas llamadas *src* y *bin* y un archivo de texto baseDatosAeronaves con extensión *.txt* a parte de otros posibles archivos. Dentro de las carpetas *src* y *bin* se encontrarán otras carpetas que contienen las capas y el código de las clases por las que está formada la aplicación.

Nota: dentro de la carpeta *ACREL_codigo* se puede añadir el archivo ***trafico*** ubicado en el CD del TFC que se nombra en el apartado C.2.1. para disponer de un archivo de datos de tráfico por defecto.

C.2.3. Archivo de tráfico

Si se quiere analizar otro tráfico, el archivo que lo contenga debe estar en un formato que *Java* pueda leer. Durante la programación de *ACREL* se comprobó que es capaz de leer archivos .txt y .dat

Además, ese archivo debe tener una línea por tráfico y donde cada argumento esté separo por espacios. Las líneas de tráfico deberán ir seguidas unas debajo de las otras sin dejar saltos de línea vacíos. Un ejemplo de línea[1] de vuelo es la siguiente:

NWA102 A320 1124842440000 1124849940000 18.0
MSP.DLL1.DLL.J34.BAE.J34.CRL.J584.SLT.FQM1.EWR/0208 892.528

A continuación se enumeran los argumento según el orden en la línea correspondiente al vuelo.

- **Identificador:** Corresponde al indentificador del vuelo.
- **Código FAA de la aeronave:** Indica el tipo de aeronave de la que se trata.
- **Hora de aparición** Expresada en milisegundos del tiempo *UnixTime* indica el momento en el que aparece el vuelo. Debería corresponder a la hora de despegue del vuelo.
- **Hora de aterrizaje:** Expresada en milisegundos del tiempo *UnixTime* indica la hora programada de llegada al aeropuerto de destino.
- **Duración de ascenso:** Corresponde al tiempo que tarda el vuelo hasta ascender hasta la altitud de crucero. Está expresada en minutos. Este parámetro podría ser -1 si a la hora de obtener los datos del tráfico el vuelo ya estaba volando.
- **Ruta:** Este parámetro está constituido de los puntos por los que pasa el vuelo. Está separado por puntos y el primer argumento del parámetro corresponde al aeropuerto de origen. El último al aeropuerto de destino.

[1] En el ejemplo está puesta en dos líneas porque no es posible hacerlo en una debido al formato del documento

- **Distancia de la ruta:** Este número indica en millas náuticas la distancia recorrida del vuelo. Puede también ser -1 en el caso de que no se disponga de la distancia o se trate de un vuelo internacional.

  Nota: Si se trata de un vuelo internaciona, la duración de ascenso también es -1.

C.3. ¿Cómo ejecutar la aplicación?

En este apartado se explica cómo hacerlo para la ejecución directa dado que se presupone que la persona que la ejecute a través de un editor de *Java* como puede ser *Eclipse* tiene los conocimientos suficientes para poder realizarlo habíendose leído este documento y esta sección dado que la forma de proceder, una vez dentro del programa, no es diferente.

ACREL está diseñada para que pueda comenzar a procesar el tráfico justo cuando se ejecuta la aplicación. Sin embargo puede hacerse una vez esté en funcionamiento.

A continuación se describen los pasos a seguir para ejecutarla en Windows. Asegúrese de haber realizado lo indicado en el apartado C.2.1.

- Abrir *cmd.exe*[1]
- Abrir la carpeta ACREL que anteriormente hemos instalado.[2]

Una vez aquí tendremos una ventana similar a la de la figura C.1

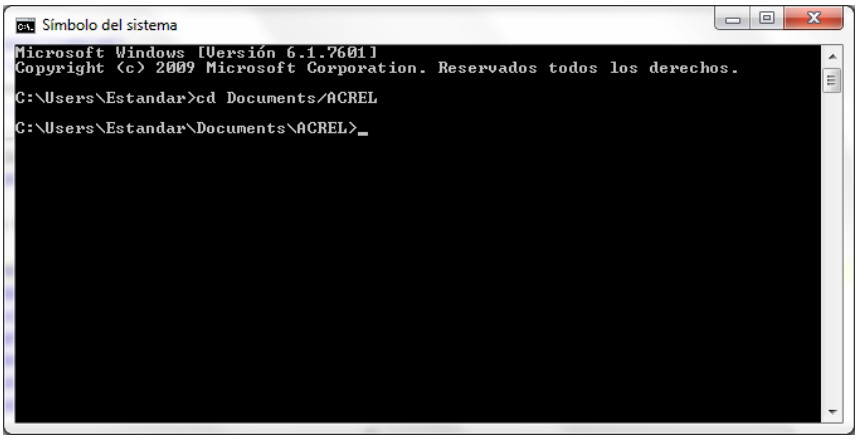

Figura C.1: Ventana de cmd.exe antes de ejecutar ACREL

El último paso a seguir tiene dos posibilidades:

[1] Si no sabe dónde buscarlo puede utilizar el buscador de programas y archivos de *Windows*.

[2] Para obtener ayuda de cmd.exe puede escribir en el mismo programa *help*. El comando a utilizar para cambiar de carpeta es *cd*. Ejemplo: cd /Documents/ACREL.

- Ejecutar la aplicacion y mostrar el menú principal sin procersar tráfico ó
- Ejecutar la aplicacion y a la vez procesar el tráfico

Para la primera opción se debe escribir en la pantalla de *cmd.exe:*

- java -jar Acrel.jar

y para la segunda opción:

- java -jar Acrel.jar *la línea de comandos*

una vez escrito lo deseado y tras pulsar el botón de *Intro*, aparecerá el menú principal[1] de la aplicación mostrado en la fingura C.2

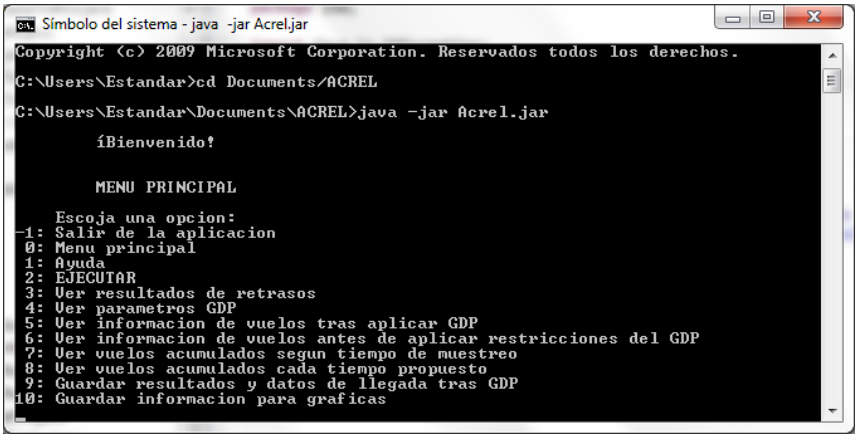

Figura C.2: Menú principal de ACREL en de cmd.exe

En este punto, la aplicación está en marcha pudiendo seleccionar cualquiera de las opciones del menú.

C.4. Descripción de línea de ejecución

La línea de ejecución es un conjunto de parámetros que se le pasa a la aplicación para que ésta pueda realizar los cálculos. Estos parámetros están separados por espacio. Todos los tiempos deben ser expresados en segundos usando *unixTime*.

Esta línea puede ser introducida antes de iniciar la aplicación como se explicó en el apartado C.2.1. de este documento o se puede pasar tras seleccionar la opción 2 del menú una vez la aplicación está activa.

[1] En el caso de haber introducido la linea de comandos, comenzarán los procesos de cálculo a mostrarse a medida que se van ejecutando antes de mostrar el menú principal. Puede tardar varios segundos.

A continuación se enumeran todos los parámetros y se comentan algunos de ellos en orden de aparición en la línea.

- **Ruta del archivo de tráfico:** Aquí se escribe la ruta completa donde está guardado el archivo de tráfico a analizar. Puede ser -1 si se quiere analizar algún archivo de la ruta por defecto[1].
- **Archivo de tráfico:** Este es el nombre del archivo de tráfico donde también se le debe poner la extensión que tiene. Puede ser -1 si se quiere analizar el archivo tráfico por defecto.
- **Hora inicio del tráfico a analizar:** Con este argumento se indica desde qué momento se quiere tener en cuenta el tráfico para analizar. Todos los vuelos anteriores a ese momento no serán tenidos en cuenta. Puede ser -1 si se desea que la hora de inicio del tráfico a analizar coincida con el del primer vuelo disponible.
- **Hora fin del tráfico a analizar:** Este argumento es el contrario al anterior. Indica el momento hasta el cual se tendrán en cuenta los vuelos a analizar. Si el argumento es -1 no se descartará ningún vuelo desde la hora de inicio de tráfico[2].
- **Capacidad normal:** Este argumento indica la el número de aeronaves/hora que pueden aterrizar en el aeropuerto que define el GDP en situaciones nominales. Es decir, la máxima capacidad de aterrizaje.
- **Capacidad regulada:** Indca el número de aeronaves/hora que pueden aterrizar en el aeropuerto que define el GDP en situaciones donde la capacidad se ve afectada y reducida. Es un número menor a la capacidad nominal.
- **Radio de influencia:** Es la distancia en millas náuticas la cual todo vuelo que se dirija al aeropuerto donde está definido el GDP y provenga de una distancia mayor a esta, queda excluido del programa GDP y tiene prioridad para aterrizar lo más próximo a hora de llegada.
- **Hora de definición del GDP:** Es el momento en el que se define el GDP
- **Hora de inicio del GDP:** Es el momento en que la capacidad del aeropuerto pasa a ser reducida y se cominzan a aplicar las políticas de asignación de prioridad de aterrizaje.
- **Hora de fin de restricciones del GDP:** Es la hora en la cual las restricciones del GDP se cancelan y se pasa de capacidad reducida a la capacidad nominal del aeropuerto.
- **Número de políticas:** Indica el la cantidad de políticas de asignación de aterrizaje que habrá durante el GDP. Puede ser 0 y se aplicará como política por defecto *RBS*. Este argumento condiciona a los siguientes. Si es 0, no se deben definir políticas. El otro caso se denerán definir tantas políticas como número de políticas se haya definido.

[1] Esta ruta coincidirá con la ruta que tiene la carpeta ACREL donde esté alojada.

[2] Siempre y cuando haya tráfico disponible

- **Tipo de política:** Este argumento indica qué política es la que se va a definir. Tiene que ser un número natural. Puede ser:

    1. *RBS - Ration By Schedule*
    2. *RBPax - Ration By Passengers*
    3. *RBD - Ration By Distance*
    4. *RBID - Ration By Inverse Distance*
    5. *RBAc - Ration By Aircraft size*

- **Hora inicio política:** Define cuándo comienza la política del tipo de política anterior seleccionado anteriormente. Puede ser -1 para que empiece al inicio del GDP o si hay alguna política definida anteriormente, en la hora de fin de esta.

- **Hora de fin de política:** Define cuándo termina la política seleccionada justo antes a la hora de inicio de política. Puede ser -1 indicando que termine justo cuando termina el GDP[1].

Ejemplos de líneas de ejecución:

Línea de ejecución con 1 política que dura todo el GDP

C:\Users\Estandar\esktop\ DemandDistanceEWR2425082005.dat 1124882400 1124927000 40 30 4000 1124884000 1124894160 1124935280 1 1 -1 -1

Línea de ejecución con tráfico por defecto desde que comienza hasta que acaba. Indicación de 2 políticas donde la última comienza cuando termina la primera y acaba cuando termina el GDP definido.

-1 -1 -1 -1 40 30 4000 1124884000 1124894160 1124935280 2 3 -1 1124925280 2 -1 -1

Línea de ejecución con 0 políticas

C:\ACREL\ trafic.dat -1 -1 40 30 2500 1124874200 1124883360 1124925480 0

C.5. Visión rápida de las opciones del menú

Este apartado porporciona una visión genérica de lo que cada opción del menú hace.

- **-1: Salir de la aplicación.** Esta opción termina con todos los procesos y finaliza ACREL.

- **0: Menú principal.** Muesta el menú principal de la aplicación.

- **1: Ayuda.** Muestra un sub-menú con diferentes opciones entre las cuales se encuentran algunos apartados de este documento. También contiene un .Acerca de..."donde muestra la versión, el propósito y otros datos de interés.

[1] La hora final del GDP es calculada por la aplicación y no corresponde con la hora de fin de restricciones que introduce el usuario.

- **2: EJECUTAR.** Tras la selección de esta opción se pide que se introduzca una línea de ejecución con los argumentos detallados en la anterior sección C.4. de este documento. Tras su introducción y el pulsado de la tecla *Intro* la aplicación carga el tráfico, define el GDP y sus políticas y calcula el retraso y asigna el nuevo modelo de llegadas de de los vuelos. Los procesos pueden tardar varios segundos dependiendo de la cantidad de vuelos a analizar.
- **3: Ver resultados de retrasos.** Muestra los datos de retrasos calculados.
- **4: Ver parámetros GDP.** Muestra la definición del GDP y otros calculados por la aplicación como pueden ser la hora de finalización del GDP, su duración, la cantidad de vuelos controlados, excluidos y no afectados por el GDP entre otros.
- **5: Ver información de vuelos tras aplicar GDP.** Esta opción muestra un listado de la disposición de los vuelos en sus respectivos slots y características de estos como pueden ser el retraso final, la hora inicial de llegada, el modelo de aeronave, la distancia recorrida o número de pasajeros entre otros.
- **6: Ver información de vuelos antes de aplicar restricciones del GDP.** Muestra información similar a la opción anterior pero antes de aplicar el GDP.
- **7: Ver vuelos acumulados según tiempo de muestreo.** Calcula y muestra por pantalla el tiempo de muestreo, el slot correspondiente y los vuelos acumulados hasta ese momento tras aplicar el GDP. Al final de esta lista se muestra también tres líneas separadas con los mismos datos pero haciendo referencia al slot de inicio del GDP, al de final de restricciones y el del final del GDP. Por otra parte se activa un gráfico que muestra la demanda y la capacidad definida. Este gráfico es de dimension variable y es similar al de la figura C.3

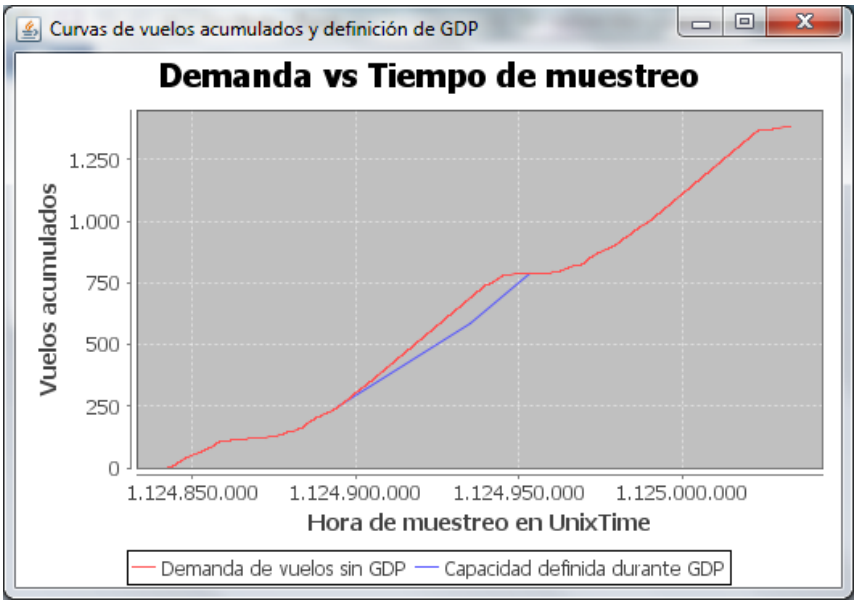

Figura C.3: Gráfico de opción 7 del menú principal.

- **8: Ver vuelos acumulados cada tiempo propuesto** Al ejecutar esta opción, lo primero que te solicita es que indiques cada cuánto tiempo se quiere contar los

vuelos desde el inicio hasta el final del tráfico. Es decir, si se indica 60, se mostrarán cuántos vuelos llegan cada 60 minutos.

La aplicación muestra por pantalla cuatro conlumnas con, siguiendo el orden, la hora de inicio, hora de fin de tiempo de conteo, vuelos acumulados tras el GDP y vuelos acumulados sin restricciones entre los dos tiempos anteriores. La forma de leerlo es: *Desde las X h hasta las Y h hay n vuelos si se aplica el GDP y m vuelos si no se aplicase*. Además, al igual que en la opción anterior, se abre una ventana similar al de la figura C.4 que muestra de forma gráfica estos resultados. Esta ventana es de dimensión variable pudiendo hacer el gráfico más grande.

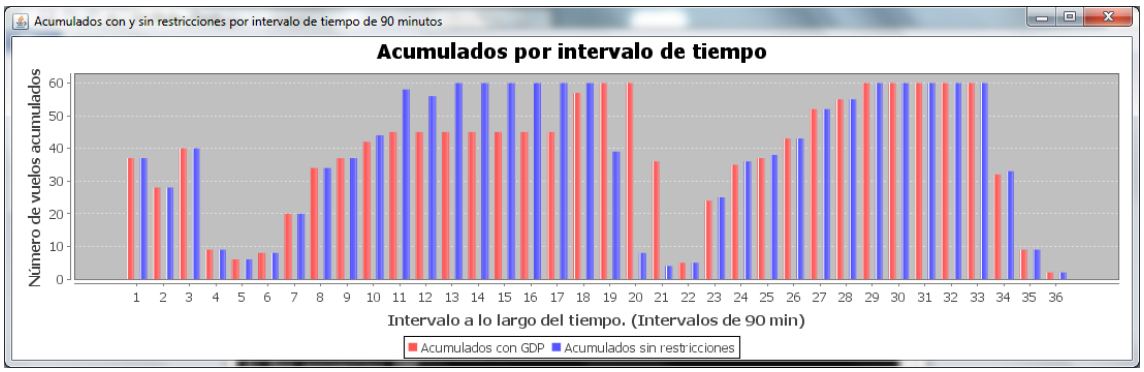

Figura C.4: Gráfico de opción 8 del menú principal.

- **9: Guardar resultados y datos de llegada tras GDP.** Esta opción guarda en un fichero txt, en la carpeta donde está instala la aplicación, el resultado del tráfico tras aplicar el GDP definido en la línea de ejecución.

  El nombre del archivo generado es *"vuelosDatosTrasGDP.txt"* y contiene, en el mismo orden que es citado, el número de slot, los vuelos acumulados hasta ese slot, hora de inicio y hora final del slot, duración de este, el identificador del vuelo que hay, la hora de aparición del vuelo, la hora estimada de llegada, la hora controlada de llegada, la hora real de llegada, el retraso en tierra, retrasos en el aire, modelo de aeronave, tipo de aeronave, número de pasajeros a bordo, distancia recorrida, si es nacional o internacional y si se trata de un vuelo no afectado, excluido o controlado según el GDP.

- **10: Guardar informacion para graficas.** Esta última opción permite generar dos ficheros con los datos que se muestran en las opciones 7 y 8. Está pensado para poder genera otro tipo de gráficas.

  Los dos archivos que se generan se guardan en la carpeta donde esta ubicada la aplicación y se guardan en formato *.txt*

  Los archivos son*"acumuladosTiempoMuestreo.txt"* y *"acumuladosCadaHora2Listas.txt"*

C.6. Descripción rápida de los procesos del cálculo principal

Durante la ejecución de la línea de comandos, la aplicación realiza principalmente 8 procesos en cadena, es decir, en el caso de que falle uno, el resto no se ejecuta.

- **Proceso 1 de 8:** La aplicación define y carga el GDP definido por el usuario en la línea de ejecución. Tras hacer esto, carga el tráfico de la base de datos asignándole un número de pasajeros por vuelos según el modelo de aeronave del que tenga asignado ese vuelo. Esto lo hace porque el archivo de tráfico no contiene el número de pasajeros.

  Finalmente también carga el horario de inicio y final de tráfico.

- **Proceso 2 de 8:** La aplicación crea el tráfico de llegada si no hubiera GDP ni restricciones. Crea una lista de slots de llegadas y asigna a cada vuelo un slot.

- **Proceso 3 de 8:** Se calcula la hora final del GDP y el retraso mínimo global que puede asignarse a los vuelos.

- **Proceso 4 de 8:** Se evalúa el tráfico según el horario del GDP y se definen si son controlados, excluidos o no afectados por el GDP.

- **Proceso 5 de 8:** Se crean las políticas de asignación de prioridad de llegada definidas por el usuario.

- **Proceso 6 de 8:** Se aplica las restricciones del GDP y las políticas de asignación al tráfico. Se crea una lista de slots de llegadas con vuelos asignados a esta.

- **Proceso 7 de 8:** Se calculan los retrasos de los vuelos tras haber sido evaluados y asignado a los nuevos slots de llegada.

- **Proceso 8 de 8:** Se calculan las estadísticas y datos de los retrasos de los vuelos tras haber evaluados por el GDP y las políticas de asignación de llegada.

  Los procesos 1, 2 y 6 suelen ser los que más tiempo necesitan y pueden tardar varios segundos en ejecutarse. Si tras 45 segundos la aplicación no muestra un mensaje de éxito en la ejecución, es posible que se haya producido un fallo interno.

  Esto último es orientativo y teniendo en cuenta de que la aplicación se ha creado con el objetivo de analizar tráfico de 2 días y una cantidad aproximada de 1400 vuelos. No se ha intentado aplicar a más tráfico y se desconoce el comportamiento de la aplicación en tal caso.

C.7. Posibles fallos durante la ejecución y sus soluciones

El propósito de esta sección es enumerar los posibles fallos que puede mostrar la aplicación ACREL durante su ejecución y cómo proceder para solventarlos.

Los errores más comunes suelen estar relacionados con los argumentos de la línea de ejecución descrita en el apartado C.4.. Recuerde que todas las horas de definición deben estar en segundos del formato *unixTime*.

Además debe tener en cuenta que caca argumento debe estar separado del siguiente por un espacio en blanco.

- *ERROR al introducir linea de comandos.* Este error se producce cuando el número de argumentos de la línea de ejecución es menor de los que se necesitan. Recuerde, se necesitan al menos 10 argumento, es decir, hasta indicar el número de políticas de asignación de retraso que habrán durante el GDP.
- *ERROR. Formato de definicion del GDP incorrecto.* Esto sucede cuando en la línea de ejecución se introducen letras donde tienen que ir números o viceversa.
- *ERROR al definir el horario del GDP.* Posiblemente se haya introducido las definiciones de los horarios de forma inadecuada. Es probable que la hora de inicio del GDP sea menor a la de definición o que la hora de final de restricciones sea menor que la hora de inicio. Asegúrese de que la hora de definición es menor que la hora de inicio del GDP y ésta menor a la hora de fin de restricciones.
- *El tiempo de las restricciones del GDP es demasiado corto. Aumente la capacidad reducida o aumente el tiempo de restricciones. GDP no definido.*

A continuación se numeran los errores que pueden producirse durante la ejecución del cálculo principal. Para ello se dividen por procesos.

- ***Proceso 1 de 8***
  - *ERROR (-1). El sistema no encuentra el archivo de modelos de aeronaves*

    Esto sucede cuando el archivo de base de datos ***'baseDatosAeronaves.txt'*** del que se hace referencia en C.2.1. no existe o no se encuentra en la carpeta correspondiente. Asegúrese de que exista en el directorio correcto.
  - *ERROR(-2). Fallo en la lectura de datos de modelos de aeronaves.*

    Este error puede darse si el archivo ***'baseDatosAeronaves.txt'*** ha sido modificado o está dañado. Borre el archivo del directorio de origen y vuelva a colocar una copia en buen estado de este fichero.
  - *ERROR (-3). El sistema no encuentra el archivo de vuelos a analizar.*

    Si en la línea de ejecución ha escrito la ruta y el nombre del fichero, es posible que se haya equibocado. Recuerde poner al final de la ruta 'ý separarla del archivo de tráfico por espacio. En el nombre del fichero debe especificarse la extensión de este.

    Si por el contrario en la línea de ejecución ha seleccionado la ruta y fichero por defecto con '-1' en ambos casos, es posible que el archivo de tráfico ***'trafico.dat'*** no se encuentre en la carpeta done está instalada la aplicación. Revísela y si es necesario, reistale el archivo.

- *ERROR (-4). Fallo en la lectura del archivo de vuelos a analizar.*

  Al igual que en el error '-2', es posible que el archivo de tráfico esté dañado o haya sido modificado. La solución más sencilla es reinstalar el archivo.

- *ERROR (-5). Fallo al crear el horario del tráfico de llegadas.*

  Este error no ha sido mostrado durante la implementación del sistema. En caso de que ocurriese, vuelva a ejecutar la aplicaición cambiando los parámetros y si persiste el error, puede ponerse en contacto con el desarrollador de la aplicación y autor de este TFC.

- *ERROR (-6). 2 Fallos: Fallo al crear el horario de tráfico y fallo en la lectura del fichero de BD de modelos de aeronaves.*

  Este error junta los errores -2 y -5 anteriormente nombrados.

- *ERROR (-7). 2 Fallos: Fallo al crear el horario de tráfico y el sistema no encuentra el fichero de BD de modelos de aeronaves.*

  Este error junta los errores -1 y -5 anteriormente nombrados.

- *ERROR (-8). Fallo. No se ha podido crear algún vuelo en memoria.*

  Este error no ha sido mostrado durante el desarrollo de la aplicación. En caso de que suceda vuelva a ejecutar la aplicación cambiando algún parámetro.

- *ERROR (-9). Fallo. El horario de analisis propuesto no incluye trafico.*

  Se muestra este error cuando en la línea de ejecución se ha escrito un horario que no incluye ningún vuelo representado en el archivo de tráfico que se está analizando.

  Redefina el horario de análisis de la línea de ejecución ó defínalo por defecto con '-1'.

- *ERROR (-10). Fallo. El horario de inicio de analisis debe ser menor al horario final de analisis.*

  Se ha introducido incorrectamente el horario en la línea de ejecución. Revíselo y asegúrese que la hora de inicio es menor que la hora de fin de análisis.

- *Fallo no crítico (1). Se ha cargado el tráfico PERO hay alguna aeronave que no está en la base de datos y se ha puesto 'Numero de pasajeros' y 'Tipo de aeronave' por defecto de manera aleatoria.*

  Este error se produce cuando en ***'baseDatosAeronaves.txt'*** alguna aeronave del archivo de tráfico. Puede buscar de qué modelo de aeronave se trata e introducirlo manualmente en la base de datos de aeronaves con sus respectivas características.

- ***Proceso 2 de 8. ERROR al crear el trafico de llegada sin GDP aplicado.***

  Intente a volver a ejecutar la aplicación con una nueva deficinión de línea de comandos.

- ***Proceso 3 de 8***

  - *ERROR. No se ha podido calcular la Hora Final del GDP. Hora de inicio GDP no encontrada en la lista de slots.*

    Intente a volver a ejecutar la aplicación con una nueva deficinión de línea de comandos.

  - *ERROR. No se ha podido calcular la Hora Final del GDP. Trafico insuficiente.*

    Este mensaje se muestra cuando la aplicación ha analizado todos los slots y no ha conseguido encontrar la hora de final de GDP.

    Intente a volver a ejecutar la aplicación con una nueva deficinión de línea de comandos u otro tráfico.

- ***Proceso 4 de 8. ERROR. El trafico no ha sido evaluado por fallo al clasificarlo.***

  Se ha producido un error interno y no se ha podido clasificar el tráfico en controlados, excluidos o no afectados. Por consiguiente, se finaliza el proceso.

  Intente a volver a ejecutar la aplicación con una nueva deficinión de línea de comandos u otro tráfico.

- ***Proceso 5 de 8***

  - *ERROR al definir las politicas. No ha escrito todo lo necesario para definirlas.*

    Para definir una política son necesarios al menos tres parámetros: el tipo de política, la hora de inicio y la hora de fin. Intente a volver a ejecutar la aplicación con una nueva deficinión de línea de comandos.

  - *Otros errores referentes a las horas de definición de politicas.*

    Intente a volver a ejecutar la aplicación con una nueva deficinión de línea de comandos.

- ***Proceso 6 de 8. ERROR. El trafico no ha podido ser asignado a slots de llegada tras aplicar el GDP***

  Error no mostrado durante el desarrollo de la aplicación. Intente a volver a ejecutar la aplicación con una nueva deficinión de línea de comandos.

- ***Proceso 7 de 8. ERROR. No se han podido asignar los retrasos de los vuelos.***

  Error no mostrado durante el desarrollo de la aplicación. Intente a volver a ejecutar la aplicación con una nueva deficinión de línea de comandos.

- ***Proceso 8 de 8. ERROR. No se ha podidido calcular los datos y estadisticas de retrasos.***

  Error no mostrado durante el desarrollo de la aplicación. Intente a volver a ejecutar la aplicación con una nueva deficinión de línea de comandos.

Todos los errores anteriores son algunos de los fallos que pueden aparecer durante la ejecución de la aplicación. No obstante pueden aparecer otros diferentes. En este caso, la recomendación primera es intentar ejecutar una nueva línea de comandos variando las definiciones de horarios.

Para más información a cerca de la aplicación, puede ponerse en contacto con el desarrollador de la aplicación y autor de este TFC.

APÉNDICE D. ÁCTAS Y RESUMEN DE REUNIONES

D.1. Reunión 1 de TFC

Día: Lunes 03 de octubre de 2011
Asistentes

ID	Nombre	E-mail	Organización
LD	Luis Delgado Muñoz	Luis.delgado@upc.edu	Director
JM	José M Marco Carpena	mc.josemarco@gmail.com	Alumno

Objetivos de la reunión:

1. Asistir a la reunión
2. Hablar sobre la propuesta del TFC
3. Establecer el modo de trabajo
4. Establecer un día y lugar para las próximas reuniones

Resumen de puntos:

LD: Propone como TFC el estudio de la combinatoria del GDP (Ground Daley Program), RBS (Ration By Schedule) y RBID (Ration By Inverse Distance) con un simulador de tráfico aéreo donde habría que crear un código de los diferentes programas de asignación de slots que interactuase con dicho simulador.
JM: Acepta la propuesta.
LD: Propone asignar un día semanal para reunirnos. Cada lunes a las 17h00.
Además para las próximas dos semanas debe estar mirado por parte del alumnos unos documentos referentes al GDP, haber instalado y ver el funcionamiento del programa Mendeley, tener en mente y buscar un programa para hacer la planificación con un diagrama de Gantt teniendo en cuenta los ciclos de desarrollo de un proyecto. Para esto último el alumno debe mirarse los apuntes de la asignatura aviónica.
Por otro lado, el alumno debe instalar el programa Eclipse, familiarizarse con él y hacer algunas pruebas programando en Java.
Además de todo esto, el profesor pide que el alumno lleve un control sobre las horas dedicadas y que después de cada reunión se haga un acta como es esta primera.

Resumen de puntos:

- Instalación Mendeley

- Instalación de Eclipse y pruebas con Java
- Leer documentos sobre GDP
- Documento de horas dedicadas
- Documentación de ciclos de desarrollo
- Planificación diagrama de Gantt

Próxima reunión: lunes 10 de octubre de 2011 a las 17h00 en el despacho 120 del edificio C3.

D.2. Reunión 2 de TFC

Día: Martes 11 de octubre de 2011
Asistentes

ID	Nombre	E-mail	Organización
LD	Luis Delgado Muñoz	Luis.delgado@upc.edu	Director
JM	José M Marco Carpena	mc.josemarco@gmail.com	Alumno

Objetivos de la reunión:

1. Asistir a la reunión
2. Revisión del inicio del TFC
3. Resolución de dudas
4. Asignación de nuevos puntos para realizar antes de la siguiente reunión

Resumen de puntos:

- Definir el diagrama de Gantt previsto para el proyecto*
- *Terminar de definir diagrama de Gantt.
- Tener en cuenta los test en el diagrama para la integración de código.
- Buscar la clasificación de requisitos
- Si es posible, mirar cómo hacer la trazabilidad del proyecto con matrices
- *Descargar plug- in Omondo para UML. Editor eclipse UML.
- Preparar plantillas de documentos como requisitos y test.
- *Programación JAVA. Realizar un pequeño programa estilo IC (Introducción a los computadores)

*Puntos con prioridad.

Próxima reunión: lunes 24 de octubre de 2011 a las 17h00 en el despacho 120 del edificio C3.

D.3. Reunión 3 de TFC

Día: Martes 18 de octubre de 2011
Asistentes

ID	Nombre	E-mail	Organización
LD	Luis Delgado Muñoz	Luis.delgado@upc.edu	Director
JM	José M Marco Carpena	mc.josemarco@gmail.com	Alumno

Objetivos de la reunión:

1. Asistir a la reunión
2. Revisión de puntos de la reunión pasada
3. Resolución de dudas
4. Asignación de nuevos puntos para realizar antes de la siguiente reunión

JM enseña lo realizado hasta el momento.
LD propone lo que hay que mejorar, nuevos puntos a tener en cuenta y hace un pequeño ejemplo en Eclipse.

Resumen de puntos:

- Definir el diagrama de Gantt real para el seguimiento del proyecto
- Realizar pruebas en Eclipse*
- Ver la API de Java
- Documentarse sobre UML (lenguaje de modelado) Diagramas de clase, secuencias y casos de uso*
- Instalar LED Latex editor (no necesario en estos momentos)
- Buscar información de CASA de la CFMU como analogía a GDP

*Puntos con prioridad.

Próxima reunión: lunes 17 de octubre de 2011 a las 17h00 en el despacho 120 del edificio C3.

D.4. Reunión 4 de TFC

Día: Lunes 24 de octubre de 2011
Asistentes

ID	Nombre	E-mail	Organización
LD	Luis Delgado Muñoz	Luis.delgado@upc.edu	Director
JM	José M Marco Carpena	mc.josemarco@gmail.com	Alumno

Objetivos de la reunión:

1. Asistir a la reunión
2. Revisión de puntos de la reunión pasada
3. Resolución de dudas
4. Asignación de nuevos puntos para realizar antes de la siguiente reunión

Resumen de puntos:

- Realizar los puntos de la reunión pasada por no estar completos.
- Descargar otra versión más completa de UML para Eclipse.
- Realizar platilla de todos los documentos Anexos. Si es posible en Latex
- Tener clara la clasificación de requisitos
- Diagrama de casos de uso
- Actores
- Casos del sistema
- Comenzar con requisitos del TFC
- Continuar con conceptos de programación Java

*Puntos con prioridad.

Próxima reunión: lunes 31 de octubre de 2011 a las 17h00 en el despacho 120 del edificio C3.

D.5. Reunión 5 de TFC

Día: Jueves 03 de noviembre de 2011
Asistentes

ID	Nombre	E-mail	Organización
LD	Luis Delgado Muñoz	Luis.delgado@upc.edu	Director
JM	José M Marco Carpena	mc.josemarco@gmail.com	Alumno

Objetivos de la reunión:

1. Asistir a la reunión
2. Revisión de puntos de la reunión pasada
3. Resolución de dudas
4. Asignación de nuevos puntos para realizar antes de la siguiente reunión

Resumen de puntos:

JM: Muestra los estándares encontrados para la realización del TFC.
LD: Indica que se tomen solo los que el alumno vea conveniente teniendo en cuenta la documentación que genera. Indica que deben ser los justos y necesarios para poder centrarnos en el proyecto en sí.
JM: Comenta que tenía problemas con la utilización de LedLaTex dado que esto solo es un editor y no tenía código. Se muestra la solución hecha instando MiKTex.
LD: Verifica la solución, explica algunos comando y la estructuración para trabajar en LaTex y proporciona a JM un documento de apoyo escrito en LaTex.
JM: Enseña UML en Eclipse para diagrama de casos de usuario.
LD: Da ejemplos de algunos requisitos a tener en cuenta e indica que JM debe empezar con los diagramas de caso para de ahí obtener los requisitos.
LD: Pide a JM que se ponga las pilas y comience cuanto antes con los requisitos.

Resumen de puntos:

- Comenzar con diagramas de casos de uso
    * De Alto nivel
    * De Bajo nivel
- Realizar platilla de Anexos
- Comenzar con requisitos tras hacer los diagramas
- Continuar con familiarización en Java

*Puntos con prioridad.

Próxima reunión: A concretar vía e-mail cuando estén listos los diagramas,requisitos y plantillas.

D.6. Reunión 6 de TFC

Día: Viernes 02 de diciembre de 2011
Asistentes

ID	Nombre	E-mail	Organización
LD	Luis Delgado Muñoz	Luis.delgado@upc.edu	Director
JM	José M Marco Carpena	mc.josemarco@gmail.com	Alumno

Objetivos de la reunión:

1. Asistir a la reunión
2. Revisión de puntos de la reunión pasada
3. Resolución de dudas
4. Asignación de nuevos puntos para realizar antes de la siguiente reunión

Resumen de puntos:

JM: Muestra el diagrama de casos de uso planteado y la plantilla de requisitos hecha en LaTeX.
LD: Pide reestructurar el diagrama de casos de uso. Guía cómo podrían ser. Enfatiza que es necesario tenerlos cuanto antes al igual que los requisitos. Soluciona algunas dudas respecto a LaTeX.
Explica qué se tendría que hacer una vez hecho los requisitos. Comenzar con los diagramas de clase e ir pensando los parámetros necesarios para el programa. También se comenta la interfaz que podría tener el programa al hablar de los requisitos.
JM y LD: Fijan conceptos y qué debería hacer el programa.
JM: Queda en avisar a LD vía e-mail el viernes 9 para concretar la futura reunión.
Resumen de puntos:

- Reestructurar los diagramas de casos de uso*
- Hacer requisitos*
- Generar test para requisitos una vez acabados estos.
- Comenzar a plasmar una lista de parámetros necesarios para el programa.
- Intentar comenzar con los diagramas de clases

*Puntos con prioridad.

Próxima reunión: A concretar vía e-mail.

D.7. Reunión 7 de TFC

Día: Miércoles 14 de diciembre de 2011
Asistentes

ID	Nombre	E-mail	Organización
LD	Luis Delgado Muñoz	Luis.delgado@upc.edu	Director
JM	José M Marco Carpena	mc.josemarco@gmail.com	Alumno

Objetivos de la reunión:

1. Asistir a la reunión
2. Revisión de puntos de la reunión pasada
3. Resolución de dudas
4. Asignación de nuevos puntos para realizar antes de la siguiente reunión

Resumen de puntos:

Se revisa la nueva versión de diagramas de Casos de Uso.
LD: Indica unos fallos de presentación a corregir y da el visto bueno.
Se revisa la primera versión de los Requisitos Funcionales.
LD: Indica a JM putos que habría que mejorar.
LD y JM: Debaten algunos puntos de Requisitos Funcionales.
LD: Da ejemplos de cómo podría JM concretar requisitos usando los que ya tiene hechos.
LD y JM comentan otros requisitos posibles de qué debería poder hacer la aplicación.
LD: Indica a JM que comencemos con una aplicación por consola donde lea líneas de comandos y hace un ejemplo.
JM: Muestra los datos del fichero que cree él que hace falta para la aplicación.
LD: Aconseja a JM que use tiempos Unixtime para la aplicación.
Resumen de puntos:

- Retocar los diagramas de casos de uso*
- Terminar requisitos más concretos*
- Buscar información sobre problema con las tablas en LaTeX*
- Generar test para requisitos una vez acabados estos.
- Buscar información de Programación por capas.
- Definir dominio. Intentar comenzar con los diagramas de clases

*Puntos con prioridad.

Próxima reunión: A concretar vía e-mail.

D.8. Reunión 8 de TFC

Día: Jueves 08 de Marzo de 2012
Asistentes

ID	Nombre	E-mail	Organización
LD	Luis Delgado Muñoz	Luis.delgado@upc.edu	Director
JM	José M Marco Carpena	mc.josemarco@gmail.com	Alumno

Objetivos de la reunión:

1. Asistir a la reunión
2. Revisión de puntos de la reunión pasada
3. Resolución de dudas
4. Asignación de nuevos puntos para realizar antes de la siguiente reunión

Resumen de puntos:

En un encuentro de 5 min el viernes 27 de enero de 2012 LD le dijo a JM que buscara información y se mirara el patrón ”Singleton”. Además, que procurara comenzar con los diagramas de clases.
Hoy se revisa lo hecho hasta el momento.
JM muestra la última versión de requisitos con imágenes de la posible interfaz. Además muestra el comienzo del diagrama de clases.
LD comenta que debemos empezar a programar cuanto antes y que tiene mayor importancia que el programa funcione por línea de comandos, es decir, que la interfaz gráfica es secundaria. Por otra parte explica y soluciona dudas a JM referentes al diagrama de clases poniéndole ejemplos.
LD propone a JM tener una primera versión de los diagramas de clases, el comienzo de los diagramas de secuencia y el inicio del programa de una forma simplificada.
JM queda en enviar a LD vía e-mail la versión actual de los requisitos. Finalmente LD asigna el TFC a JM.

Resumen de puntos:

- Se asigna el TFC con el nombre .Estudio de políticas de asignación de retraso para Ground Delay Program”
- Primera versión de diagrama de clases para la próxima reunión.*
- Comienzo de la primera versión del diagrama de secuencias.
- Comienzo del programa de forma básica para RBS.

*Puntos con prioridad.

Próxima reunión: jueves 15 de marzo de 2012 a las 15h30 en el despacho 120 del edificio.

D.9. Reunión 9 de TFC

Día: Jueves 15 de Marzo de 2012
Asistentes

ID	Nombre	E-mail	Organización
LD	Luis Delgado Muñoz	Luis.delgado@upc.edu	Director
JM	José M Marco Carpena	mc.josemarco@gmail.com	Alumno

Objetivos de la reunión:

1. Asistir a la reunión
2. Revisión de puntos de la reunión pasada
3. Resolución de dudas
4. Asignación de nuevos puntos para realizar antes de la siguiente reunión

Resumen de puntos:

Se ha revisado la primera versión de diagramas de clases. LD ha indicado unos cambios respecto a este diagrama de clases.
JM ha enseñado el inicio de las clases en Java. LD ha explicado algunas dudas y ha dado indicaciones para la corrección de la organización del código (los packages). Además, comenta que además de la capa de dominio, que es la que se ha estado haciendo hasta ahora, hay que incluir las capas de IHM (Interfaz Hombre - Máquina) y la de base de datos.
LD ha mostrado capítulos importantes a tener en cuenta del libro que JM ha llevado de programación en Java (Capítulo 3 y 16 de El Lenguaje de Programación JAVA. 3**?** Edición).
LD indica a JM que busque información de los Iteradores (pág. 410) y vea cómo se programan.
LD indica a JM que busque información del modelo de Transacción y repase el patrón Singleton procurando añadirlo al código (Indica que posiblemente se use en la clase GDP).
LD indica que si es posible, comience a hacer un pequeño programa con RBS para ver el funcionamiento.

Resumen de puntos:

- Corregir y refinar el diagrama de clases.*
- Corregir paquetes de clases en código del programa.*
- Crear las clases de los otros dominios.
- Terminar la primera versión del diagrama de secuencias.*

- Documentarse de programación JAVA (Cap. 3 ,16 e Iteradores pág. 410)
- Incorporar patrón Singleton a las clases necesarias.
- Intentar programa sencillo para RBS.

*Puntos con prioridad.

Próxima reunión: jueves 22 de marzo de 2012 a las 15h30 en el despacho 120 del edificio C3.

D.10. Reunión 10 de TFC

Día: Miércoles 28 de Marzo de 2012
Asistentes

ID	Nombre	E-mail	Organización
LD	Luis Delgado Muñoz	Luis.delgado@upc.edu	Director
JM	José M Marco Carpena	mc.josemarco@gmail.com	Alumno

Objetivos de la reunión:

1. Asistir a la reunión
2. Revisión de puntos de la reunión pasada
3. Resolución de dudas
4. Asignación de nuevos puntos para realizar antes de la siguiente reunión

Resumen de puntos:

JM muestra a LD las mejoras sobre los diagramas de clases. LD sugiere que se cambie la nomenclatura de poner y obtener por set y get de los diagramas y del código Java.
LD comenta que faltan dos capas intermedias entre IHM - Dominio y Dominio - Base de datos.
JM también muestra la primera versión de diagrama de secuencias donde LD explica y da un nuevo enfoque a lo que JM está haciendo dado que no está incluyendo la función en las transiciones y deben aparecer. Explica que se trata de escribir el código y no una visión tan general como lo que JM ha hecho hasta ahora.
En el código Java, LD resuelve unas dudas referentes al patrón Singleton el cual se ha empezado a introducir pero no está en todas las clases que deben estar.
LD pide a JM que para la próxima reunión debe estar hecho el diagrama de secuencias de cargar datos y lo más importante, que la aplicación lea desde un fichero una serie de datos y las muestre en pantalla según las lea. Una vez hecho esto, que lo cargue en el dominio y a continuación se muestre por pantalla los datos de los vuelos.

Resumen de puntos:

- Cambiar nomenclatura de los diagramas de clases y código.*
- Añadir diagramas de clases entre IHM - Dominio y Dominio - Base de datos.
- Replantear los diagramas de secuencias.
- Terminar de introducir en código el patrón Singleton donde corresponda.*
- Hacer que la aplicación lea un fichero y lo muestre por pantalla.*

- Hacer que la aplicación cargue un fichero y lo muestre por pantalla.

*Puntos con prioridad.

Próxima reunión: A concretar por JM tras realizar pruebas de lectura de un fichero de datos.

D.11. Reunión 11 de TFC

Día: Miércoles 18 de Abril de 2012
Asistentes

ID	Nombre	E-mail	Organización
LD	Luis Delgado Muñoz	Luis.delgado@upc.edu	Director
JM	José M Marco Carpena	mc.josemarco@gmail.com	Alumno

Objetivos de la reunión:

1. Asistir a la reunión
2. Revisión de puntos de la reunión pasada
3. Resolución de dudas
4. Asignación de nuevos puntos para realizar antes de la siguiente reunión

Resumen de puntos:

En esta reunión se han tratado varias dudas que JM tenía.
Principalmente LD ha aclarado y explicado las transiciones entre capas. Ha corregido a JM indicando que IHM tendrá contenido el ”main”de la aplicación.
Para ver qué quería decir ejecutar el programa por línea de comandos, LD ha mostrado un par de ejemplos y ha intentado ejecutar lo que JM había hecho desde el cmd.exe
Resumen de puntos:

- Terminar de refinar una versión inicial de la aplicación que se ejecute por línea de comandos y muestre en pantalla un fichero. Luego que lo cargue.
- Intentar avanzar lo máximo posible con el código de la aplicación.

*Puntos con prioridad.

Próxima reunión: A concretar vía e-mail.

D.12. Reunión 12 de TFC

Día: Miércoles 02 de Mayo de 2012
Asistentes

ID	Nombre	E-mail	Organización
LD	Luis Delgado Muñoz	Luis.delgado@upc.edu	Director
JM	José M Marco Carpena	mc.josemarco@gmail.com	Alumno

Objetivos de la reunión:

1. Asistir a la reunión
2. Revisión de puntos de la reunión pasada
3. Resolución de dudas
4. Asignación de nuevos puntos para realizar antes de la siguiente reunión

Resumen de puntos:

JM muestra lo que ha hecho hasta el momento con la aplicación y un diagrama de secuencia. Enseña a LD que la aplicación ya lee por teclado, lee el archivo txt de la base de datos, la muestra por pantalla y se crean objetos de tipo vuelo sabiendo que no los guarda en ningún lugar.
LD indica que existe una clase vector para hacer lo que JM estaba intentando hacer para guardar los vuelos de una manera más sencilla. Le dice que busque información de la clase en la librería de Java STD.
Además, explica el porqué a JM no se le ejecuta la aplicación a través del cmd.exe de Windows y hace la prueba.
Por otra parte, LD indica a JM que está creando los vuelos con más código del necesario y le dice que debe cambiarlo.
Por otra parte, JM pregunta a LD el tema de las transiciones a través de las capas y muestra cómo lo ha planteado. LD hace unas objeciones explicando que cada transición debe corresponderse con un método y con una única petición que desde la IHM se hará.
Indica que debe haber una clase abstracta de transición donde el método principal sea execute donde tenga un boolean que indique que la transición ha ido bien (true) o mal (false).
Finalmente JM muestra a LD el diagrama de secuencia de la transición çargarz dice que tiene que corregir un par de puntos pero que está bastante bien.
Resumen de puntos:

- Modificar código para que ejecute la aplicación en 3 casos. Directamente con los argumentos, por consola o por ventana.
- Arreglar en el código de la aplicación las transacciones.*

- Buscar información sobre el ”Split”de un ”String”para aplicarlo a la hora de leer el archivo de la base de datos.
- Crear objetos vuelo y guardarlos en el vector (usando clase ”vector”) para posteriormente mostrarlo por pantalla.*
- Intentar hacer lo mismo que el punto anterior pero con objetos de tipo slot.
- Corregir y continuar con los diagramas de secuencia.*

*Puntos con prioridad.

Próxima reunión: A concretar vía e-mail.

D.13. Reunión 13 de TFC

Día: Viernes 11 de Mayo de 2012
Asistentes

ID	Nombre	E-mail	Organización
LD	Luis Delgado Muñoz	Luis.delgado@upc.edu	Director
JM	José M Marco Carpena	mc.josemarco@gmail.com	Alumno

Objetivos de la reunión:

1. Asistir a la reunión
2. Revisión de puntos de la reunión pasada
3. Resolución de dudas
4. Asignación de nuevos puntos para realizar antes de la siguiente reunión

Resumen de puntos:

JM muestra lo que ha hecho hasta el momento. Le enseña a LD desde çmd.exe"que se ejecuta directamente si se le pasan parámetros, si se le pasa un número incorrecto muestra error y si no se le pasa nada la aplicación muestra un menú de ejecución.
JM explica a LD que ha intentado ejecutar la aplicación desde una carpeta externa a "bin"del código y que no le sale. LD lo intenta y tampoco le sale.
Finalmente LD dice que esto no tiene mayor importancia y que podemos hacerlo tal y como lo estábamos haciendo.
JM comenta a LD que se quiere pasar a la superior y por tanto que la lectura del TFC debe ser antes del 20 de julio de este año. LD dice que es factible incluso hacerlo antes siempre y cuando JM le dedique tiempo. JM comenta también que el martes 15 de mayo termina las prácticas y que solo se dedicará al TFC.
Por otro lado, JM muestra el código de la aplicación y pregunta a LD si está bien cómo ha hecho las transacciones entre capas. LD dice que está correcto pero que si JM quiere, puede cambiar los returns"de las funciones si han ido bien por métodos que le pregunte a la función si ha ido bien.
Respecto a la creación de los objeto vuelos, JM comenta con LD que estuvo buscando información de cómo guardar los objetos en una lista y que creía que "LinkedList.^{er}a el más apropiado dado que la clase "Vector.^{es} un poco anticuada y que tardaría más en eliminar el primer objeto de la lista. LD lo ve bien.
Finalmente LD indica a JM un par de tareas que ya se habían comentado en otras reuniones.

Resumen de puntos:

- Cambiar los returns"de las transacciones por funciones que indiquen si ha ido bien el proceso o no.

- Terminar todas las transacciones.
- Crear los objetos vuelo y guardarlos en el ”LinkedList”.
- Mostrar los objetos vuelo por pantalla para ver si han cargado bien.

*Puntos con prioridad.

Próxima reunión: A concretar vía e-mail tras hacer que funcione lo propuesto o si surge alguna duda.

D.14. Reunión 14 de TFC

Día: Jueves 17 de Mayo de 2012
Asistentes

ID	Nombre	E-mail	Organización
LD	Luis Delgado Muñoz	Luis.delgado@upc.edu	Director
JM	José M Marco Carpena	mc.josemarco@gmail.com	Alumno

Objetivos de la reunión:

1. Asistir a la reunión
2. Revisión de puntos de la reunión pasada
3. Resolución de dudas
4. Asignación de nuevos puntos para realizar antes de la siguiente reunión

Resumen de puntos:

Esta reunión fue de 15 minutos dado que no se disponía de más tiempo. Se realizó este día dado que LD se tenía que marchar al día siguiente de congreso y no sería posible hacer una reunión hasta junio.
JM mostró a LD que el programa ya cargaba los vuelos desde la base de datos y también los mostraba una vez cargados.
Por otra parte LD resolvió unas cuantas dudas que JM tenía. Una de ellas era si era necesario crear una clase lista de tráfico dado que si se cargan dos veces seguidas, se añaden los vuelos aunque estén repetidos.
La posible solución que se dijo era que cada vez que cargara un tráfico desde la base de datos, que borrara lo que la aplicación tenga en memoria o si los vuelos no están cargados, que se añadan a la lista de tráfico. Además se habló que posiblemente habría que implementar otra función de GDP para cargar el tráfico.
Finalmente quedaron en que durante las dos semanas siguientes JM iría avanzando por su cuenta procurando seguir con el código e ir avanzando en la memoria en la parte de anexos correspondientes a la aplicación. Finalmente LD y JM quedaron en que estarían en contacto vía e-mail.
Resumen de puntos:

- Mejorar algunas funciones como la de leer por línea de comandos para que compruebe si tiene los argumentos necesarios.
- Añadir la nueva función de ”dameGDP”para poder cargar el tráfico en la lista mejorando la forma actual.
- Intentar realizar un Ratio By Schedul”.

- Seguir redactando anexos de la aplicación. bien.

*Puntos con prioridad.

Próxima reunión: A concretar vía e-mail.

D.15. Reunión 15 de TFC

Día: Martes 05 de Junio de 2012
Asistentes

ID	Nombre	E-mail	Organización
LD	Luis Delgado Muñoz	Luis.delgado@upc.edu	Director
JM	José M Marco Carpena	mc.josemarco@gmail.com	Alumno

Objetivos de la reunión:

1. Asistir a la reunión
2. Revisión de puntos de la reunión pasada
3. Resolución de dudas
4. Asignación de nuevos puntos para realizar antes de la siguiente reunión

Resumen de puntos:

JM muestra a LD los progresos conseguidos. La aplicación es capaz de aplicar las 5 políticas correctamente al tráfico de manera individual o combinada entre ellas. Se listan los vuelos ordenados en sus respectivos slots y con el retraso asignado.
LD soluciona un par de dudas respecto a los slots del GDP y si JM había hecho las consideraciones correctas cuando dos vuelos tenían la misma prioridad.
JM propone añadir dos nuevas políticas RBTF (Ration By Time of Flight) y RBITF (Ration By Inverse Time of Flight) que consideraría los tiempos de vuelo.
Aunque son similares a las políticas que tienen en cuenta las distancias cree que podría ser interesante dado que sería una forma de incluir la velocidad de la aeronave y la distancia que recorre. LD comenta que las comentaremos más adelante dado que supondría una ampliación y tenemos mucho trabajo que hacer todavía.
Por otra parte, LD explica qué ampliaciones debe hacer JM al algoritmo para que podamos comenzar a comparar políticas y vayamos concluyendo la aplicación.
Explica que se deben tener en cuenta los vuelos que ya están volando antes de la definición del GDP, los internacionales, los que están fuera de un alcance determinado, los que llegan al aeropuerto de destino antes o después del GDP y los vuelos que se ven afectados. Además LD indica a JM que se ha de tener en cuenta un tiempo anterior al de finalización del GDP donde se aplique las políticas pero se aumente la capacidad del aeropuerto.
Por último JM pregunta por qué incluir en la memoria y LD propone hablarlo la siguiente reunión. JM procurará llevar un guión de memoria. LD dice a JM que puede ir redactando en el caso de que no pueda avanzar con la aplicación pero que el mayor tiempo se debe dedicar a esta por el momento.
Resumen de puntos:

- Ampliar algoritmo de asignación de slots teniendo en cuenta vuelos controlados y excluidos.*
- Procurar hacer la aplicación más robusta detectando y corrigiendo fallos.
- Hacer guión de memoria y redactar explicación del algoritmo.

*Puntos con prioridad.

Próxima reunión: A definir vía e-mail procurando que sea esta misma semana

D.16. Reunión 16 de TFC

Día: Lunes 11 de Junio de 2012
Asistentes

ID	Nombre	E-mail	Organización
LD	Luis Delgado Muñoz	Luis.delgado@upc.edu	Director
JM	José M Marco Carpena	mc.josemarco@gmail.com	Alumno

Objetivos de la reunión:

1. Asistir a la reunión
2. Revisión de puntos de la reunión pasada
3. Resolución de dudas
4. Asignación de nuevos puntos para realizar antes de la siguiente reunión

Resumen de puntos:

JM explica a LD que se ha modificado el código de la aplicación introduciendo las nuevas listas de vuelos excluidos y vuelos controlados. Además explica el funcionamiento de este.
LD comenta que JM lo podía haber hecho de una manera más sencilla aunque, dice también que lo que JM ha hecho está bien y aporta de esta manera robustez a la aplicación.
LD intentará proporcionarle un fichero de datos reales a JM lo antes posible dado que ya se pueden extraer datos, con la finalidad de comprobar que la aplicación está funcionando correctamente. LD indica que JM puede ir testeando la aplicación, mientras este genera el fichero final, con el tráfico que JM se ha inventado.
Por otra parte, se solucionan unas cuantas dudas que JM había tenido durante la semana para comprobar que las consideraciones que había hecho eran correctas. Una de las consideraciones que JM había tenido en cuenta era aplicar las políticas al tráfico excluido. LD apunta que esto no era necesario para lo que quería obtener con el TFC. Sin embargo, tras una explicación de JM de porqué lo había hecho así, LD dice que es buena idea y que si se dispone de tiempo, podría considerarse para hacer el estudio requerido. Básicamente es calcular el retraso que puedan tener los vuelos excluidos para que se puedan tener en cuenta en futuros estudios. Estos estudios podrían ser la reducción de velocidad en vuelo para evitar entrar en los hipódromos de espera de un aeropuerto de destino ya que el retraso no se puede realizar en tierra del aeropuerto de origen.
LD indica a JM cuál es el siguiente paso con el algoritmo. Se trata de buscar el punto donde la demanda y la capacidad tras un GDP coinciden. Además, también indica que sería conveniente calcular el retraso óptimo global.
Se habla también de que la aplicación no dispondrá de interfaz gráfica y será todo por línea de comandos por dos motivos principales que son, primero, la aplicación será una herramienta para hacer obtener datos y hacer el estudio para el TFC y este no trata de realizar un programa para uso público, y segundo, la realización de esta interfaz llevaría

bastante tiempo extra y es más importante invertir este tiempo en terminar el TFC e incluir las nuevas ideas para extraer resultados.
Por último JM ha presentado un borrador de índice de la memoria del TFC y ha explicado los puntos por encima. LD dice que de momento le parece bien y que vaya redactando.
Resumen de puntos:

- Buscar punto de fin de GDP donde coinciden la capacidad nominal sin GDP y el final de la capacidad nominal del GDP.*
- Calcular área (retraso mínimo óptimo global) de las curvas de llegadas con GDP y sin GDP.
- Escribir correo electrónico a Luis con el tipo de argumentos y orden de estos del fichero de la base de datos de vuelos.*

*Puntos con prioridad.

Próxima reunión: A definir vía e-mail.

D.17. Reunión 17 de TFC

Día: Miércoles 20 de Junio de 2012
Asistentes

ID	Nombre	E-mail	Organización
LD	Luis Delgado Muñoz	Luis.delgado@upc.edu	Director
JM	José M Marco Carpena	mc.josemarco@gmail.com	Alumno

Objetivos de la reunión:

1. Asistir a la reunión
2. Revisión de puntos de la reunión pasada
3. Resolución de dudas
4. Asignación de nuevos puntos para realizar antes de la siguiente reunión

Resumen de puntos:

JM muestra a LD lo que tiene del programa y explica qué hace y cómo lo hace. Muestra además unas primeras gráficas mostrando que no consigue encontrar el punto de HFGDP (hora final del GDP).
LD dice a JM que para calcular el HFGDP no hace falta tener en cuenta los slots. Que como lo está haciendo JM está bien pero es más complicado. Además, comenta que primero hay que calcular el HFGDP antes de hacer las listas de vuelos excluidos y afectados. A parte, una vez tengamos esto se pueden crear los slots vacíos porque se dispondrá de la duración total del GDP.
Sin embargo, la manera de crear los slots según se vaya asignando el tráfico es correcta y puede ofrecer ventajas ya que tras aplicar el GDP se pueden crear slots más tarde de la HFGDP por culpa de las restricciones.
Además, JM pregunta a Luis si hacen falta dos transacciones para mostrar datos, es decir, una para llamar y otra para que nos devuelvan. LD comenta que con tan solo una es suficiente dado que puede llevar hasta la IHM una lista y allí gestionarla mediante un bucle llamando a los métodos de la misma.
Después LD muestra a JM una aplicación por internet (5pm con la que trabaja el grupo Ícarus) para la gestión y seguimiento del TFC. Comenta que esto se tenía que haber usado desde un principio pero no se había dado cuenta. Sin embargo, se procederá a usarlo hasta la finalización del TFC.
Finalmente se habla de una orientación de fechas de plazo donde la semana del 16 al 20 de julio de 2012 podría estar el día de presentación dado que el 20 de julio sería la fecha límite que nos habíamos propuesto por motivos académicos.
Resumen de puntos:

- Replantear la búsqueda del punto de fin de GDP HFGDP.*

- Calcular área.

*Puntos con prioridad.

Próxima reunión: A definir vía e-mail.

D.18. Reunión 18 de TFC

Día: Martes 26 de Junio de 2012
Asistentes

ID	Nombre	E-mail	Organización
LD	Luis Delgado Muñoz	Luis.delgado@upc.edu	Director
JM	José M Marco Carpena	mc.josemarco@gmail.com	Alumno

Objetivos de la reunión:

1. Asistir a la reunión
2. Revisión de puntos de la reunión pasada
3. Resolución de dudas
4. Asignación de nuevos puntos para realizar antes de la siguiente reunión

Resumen de puntos:

JM enseña a LD un par de gráficas donde se mostraban, en la primera, la demanda y en la segunda demanda y regulación calculada. En esta última se comenta que se está haciendo algo mal dado que ambas curvas no se cruzan en ningún punto (el que corresponde a HFGDP). JM se percata del error y explica a LD que para graficar los datos se ha tenido en cuenta los vuelos acumulados según los slots tras aplicar el GDP pero que para calcular la HFGDP no se tiene en cuenta. LD explica de qué manera se puede dibujar esta curva dado que es independiente de los slots y del tráfico. Se trata de una curva fija que solo depende de las capacidades y el horario del GDP definidas.
Por otra parte se comenta que a lo largo de esta semana la aplicación debería estar terminada para poder comenzar a obtener datos y estadísticas tras analizarlos la semana que viene. Paralelamente se puede ir escribiendo la memoria.
JM pregunta a LD, viendo las gráficas de demanda, qué horario podemos aplicar en el GDP para hacer las estadísticas. LD da a JM unos valores de capacidades que usa el aeropuerto de los datos disponibles NWK (Aeropuerto internacional Libertad de Newark, Nueva York) y unos horarios reales utilizados. Además queda en enviar a JM nuevos datos del aeropuerto de SFO (San Francisco) puesto que para los análisis del GDP podrían ser más interesantes debido al gran cambio real que hay entre capacidades nominales y reducidas.
Además, LD comenta a JM que podemos comenzar a montar el tribunal para tener a los miembros avisados. LD propone a Xevi Prats como segundo miembro. Si acepta, se procederá a la asignación del jurado y por consiguiente del tercer miembro que se hace a sorteo.
Finalmente se asignan y dan de alta tareas en la aplicación 5pm porque LD estará durante dos semanas ocupado.
Resumen de puntos:

- Refinar la generación de datos para que se muestre la capacidad teórica y poder visualizar el punto correspondiente a HFGDP.*
- Buscar número de pasajeros y MTOW (peso máximo al despegue) de la lista de aeronaves para poder cargarla desde la base de datos.*
- Enviar gráficas a LD cuando se muestre el punto HFGDP.*
- LD enviar a JM datos del aeropuerto de SFO.
- Terminar aplicación.

*Puntos con prioridad.

Próxima reunión: A definir vía e-mail.

D.19. Reunión 19 de TFC

Día: Miércoles 04 de Julio de 2012
Asistentes

ID	Nombre	E-mail	Organización
LD	Luis Delgado Muñoz	Luis.delgado@upc.edu	Director
JM	José M Marco Carpena	mc.josemarco@gmail.com	Alumno

Objetivos de la reunión:

1. Asistir a la reunión
2. Revisión de puntos de la reunión pasada
3. Resolución de dudas
4. Asignación de nuevos puntos para realizar antes de la siguiente reunión

Resumen de puntos:

JM muestra a LD la aplicación prácticamente acabada. Durante la ejecución de esta, se detectan que hay datos que no se ajustan a valores coherentes. JM también comenta que los valores de retraso mínimo teóricos están saliendo mayores que los calculados y entre los dos asistentes llegan a la conclusión de que puede ser que sea así por diseño del sistema aunque si es posible debería revisarse.
Además, durante la sesión se realizan pequeñas pruebas y algunos ajustes de código. LD se da cuenta de que JM estaba realizando una comparación de Strings de una forma errónea y explica cómo debe hacerlo.
Siguiendo con la aplicación, JM enseña una parte de código a LD donde se intenta mostrar gráficas pero que JM no consigue hacerlo. LD comenta a JM que no sabe cómo hacerlo pero que si no se consigue no pasa nada. JM comenta a LD que estaría bien que se mostraran esas gráficas que está intentando para quien ejecute el programa vea algo gráfico y no solo una lista de números. No obstante, JM se pone como fecha límite esa misma noche para mostrar revisar el código de las gráficas. Si no consigue mostrarlas, descartará esta opción dado que los datos se obtienen en formato para poder procesarlos con Excel que es como se han hecho hasta el momento.
Los datos disponibles para analizar corresponden a dos días de tráfico del aeropuerto EWR. LD proporciona a JM las capacidades y horas reales de GDPs para proceder con la obtención de datos. Pero, se comenta que estaría mejor analizar los datos de SFO por la diferencias de capacidades con y sin restricciones. LD comenta que si le es posible enviará a JM los datos de este aeropuerto para poder analizarlos.
Por otro lado, durante esta reunión se habla de las fechas de entrega de la memoria y revisión de exposición antes de esta dado que ya se ha asignado el jurado completo y la fecha final de defensa del trabajo.
Resumen de puntos:

- Arreglar los datos que se muestran de retrasos.*
- Revisar el cálculo de retrasos por pasajeros.*
- Añadir en la opción de mostrar datos del GDP la duración de este.
- Revisar el código en busca de comparaciones de Strings y arreglarlo si procede.*
- LD enviar datos de tráfico del aeropuerto de SFO si es posible.
- Intentar mostrar gráficas en la aplicación.

*Puntos con prioridad.

Próxima reunión: A definir vía e-mail.

D.20. Reunión 20 de TFC

Día:Jueves 12 de Julio de 2012
Asistentes

ID	Nombre	E-mail	Organización
LD	Luis Delgado Muñoz	Luis.delgado@upc.edu	Director
JM	José M Marco Carpena	mc.josemarco@gmail.com	Alumno

Objetivos de la reunión:

1. Asistir a la reunión
2. Revisión de la memoria
3. Resolución de dudas
4. Asignación de nuevos puntos para realizar antes de la siguiente reunión

Resumen de puntos:

JM muestra lo que tiene de memoria hasta el momento. LD la mira por encima y comenta unos cuantos cambios a JM como hacer una pequeña introducción a las políticas. Además, comenta que JM podría añadir un esquema de cómo funciona un GDP hecho por él y usado en sus clases.
Por otra parte, JM explica los últimos apartados a redactar que tiene planificados.
Finalmente, JM queda en enviar la memoria como primer borrador a LD por la noche para que éste la revise y le marque las posibles correcciones que JM hará al día siguiente antes de imprimirla y entregársela al jurado.
Resumen de puntos:

- Terminar el capítulo de estudio de casos.
- Hacer la introducción
- Redactar el resumen
- Corregir las posibles faltas
- Preparar el contenido del CD que se entregará junto a la memoria.

Este es la última acta de reuniones realizadas a pesar de que están programadas otras dos antes de la presentación final. El motivo es que la entrega de la memoria final, donde van incluidos estos documentos, se realiza antes de que tengan lugar estas futuras reuniones.