



Escola Politècnica Superior
de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL DE FI DE CARRERA

TÍTOL DEL TFC : Algoritmo de cálculo de rutas óptimas para aeronaves en el plano horizontal

TITULACIÓ: Enginyeria Tècnica Aeronàutica, especialitat Aeronavegació

AUTOR: Ismael González Rodríguez

DIRECTOR: Luis Delgado

DATA: 24 de enero de 2011

Título : Algoritmo de cálculo de rutas óptimas para aeronaves en el plano horizontal

Autor: Ismael González Rodríguez

Director: Luis Delgado

Data: 24 de enero de 2011

Resum

En este trabajo se ha desarrollado una aplicación que permite conocer la ruta más corta entre dos puntos del espacio aéreo europeo, si es que la hay. Además de la ruta, la aplicación también calcula la distancia que ha recorrido el aeronave y el tiempo de vuelo, para ello antes hay que introducir diferentes parámetros como el mach y flight level de crucero, que la misma aplicación ya te los pide. A la hora de escoger los puntos de origen y destino, hay dos opciones, volar entre aeropuertos o volar entre waypoints y navaid, hay que elegir la segunda opción, porque el programa no permite calcular la ruta entre dos aeropuertos porque pese a que están cargados, no están conectados con los demás puntos del grafo ya que la base de datos de la que se ha extraído la información no contenía estas conexiones.

La información necesaria para crear el grafo se ha extraído de la base de datos de SkyView, que es un programa que contiene toda la información sobre los waypoints, navaid, aerovías, sectores aéreos, etc. del espacio aéreo europeo. Además este programa se ha utilizado también para hacer las comprobaciones de los resultados a la hora de calcular diferentes rutas con la aplicación y así poder asegurar que funciona correctamente.

La aplicación se ha programado con Eclipse en el sistema operativo Linux, por lo que solo funciona en este sistema operativo. El lenguaje de programación que se ha utilizado es c++ y sql a la hora de leer las bases de datos extraídas del SkyView, y por último el diseño gráfico se ha realizado con qtDesigner.

A la hora de programar la aplicación no se ha hecho de manera libre, sino que se han desarrollado una serie de documentos, que se pueden encontrar en los apéndices, que han facilitado su creación. Primeramente se elaboró un documento con los casos de uso del software, dónde se especifican los actores del programa y cuáles serán las diferentes funcionalidades que tendrá. A partir de este documento se elaboró un documento de especificaciones de requisitos del software, donde están detallados los diferentes requisitos funcionales y no funcionales de la aplicación. Para la fase de diseño se elaboró un documento de arquitectura del software donde se puede encontrar los diagramas de clase y diagramas de secuencia de la aplicación. Por último se elaboraron unos test de validación, para una vez finalizada la aplicación comprobar que cumplía con los requisitos iniciales.

La memoria ha sido realizada en \LaTeX .

Title : Algorithm for calculating optimal routes for aircraft in the horizontal plane

Author: Ismael González Rodríguez

Director: Luis Delgado

Date: January 24, 2011

Overview

In this project has been developed an application which allows to know the shortest route between two points of the European airspace, if it exists. The application also computes the distance which has travel the airplane and the time of the flight, for this we must enter different parameters before as mach and flight level of cruising. At the time for choosing the departure and arrival points, there are two options, fly between airports or fly between waypoints or nav aids, we must choose the second option, because the program doesn't allows to calculate the route between two airports because although they were loaded, they aren't connected with other points of the graph as the database from which information has been extracted did not contain these connections.

The information necessary for create the graph has been extracted of a SkyView database. SkyView is a program which contains all information of waypoints, nav aids, airways, boundaries, etc. of the European airspace. Also this program has been used for make checks of the results at the time of calculate the different routes with the application and so to ensure that works correctly.

The application has been programmed with Eclipse in the Linux operating system, so it only works in this operating system. The programming language which has been used is c++ and sql for read the database, and finally the graphic design has been realized with qtDesigner.

At the time of program the application hasn't been developed without control, but has been developed a series of documents, which can find in the appendices, which has been facilitate its creation. Firstly it was developed a document with use software cases, where specifies the program stakeholders and which will be the different features. From this document has been developed a document of requirements software specification, where are detailed the different functional and nonfunctional requirements of the application. For the design phase has been developed a document of software architecture where was found class diagrams and sequence diagrams of the application. Finally, have been developed validation tests, for once the application check that it met the initial requirements.

The memory has been written in \LaTeX .

ÍNDICE GENERAL

INTRODUCCIÓN	1
CAPÍTULO 1. Glosario	3
CAPÍTULO 2. Rutas aéreas	5
2.1. Introducción	5
2.2. Fases de un vuelo	7
2.2.1. Rodaje	7
2.2.2. Despegue y ascenso	7
2.2.3. En ruta	9
2.2.4. Llegada y aproximación	9
2.2.5. Aterrizaje	10
2.3. Clasificación de las rutas ATS	11
2.3.1. Rutas convencionales	11
2.3.2. Rutas RNAV	12
2.4. Zonas de estatuto particular	14
2.5. Rutas condicionales	15
2.6. Meteorología	16
CAPÍTULO 3. Desarrollo del software	17
3.1. Introducción a la aplicación	17
3.2. Ciclo de vida de la aplicación	19
3.3. Arquitectura del programa	21
3.4. Algoritmo de cálculo	26
3.5. Ejemplos y comprobaciones	29
3.6. Posibles ampliaciones y mejoras	34
CAPÍTULO 4. Gestión del proyecto	39

CAPÍTULO 5. Conclusiones 41

BIBLIOGRAFÍA 43

ÍNDICE DE FIGURAS

2.1 Rutas ATS del espacio superior de España	6
2.2 Ejemplo SID Mallorca	8
2.3 Segmentos de despegue [1]	9
2.4 Ejemplo STAR Mallorca	10
3.1 Pantalla principal de la aplicación	18
3.2 Selección del origen y el destino	18
3.3 Características de vuelo	19
3.4 Restricciones de vuelo	19
3.5 Resultado del calculo de la ruta más corta	19
3.6 Ciclo en V de la aplicación	20
3.7 Estructura del programa	21
3.8 Esquema general de la estructura del programa	22
3.9 Capa de presentación	23
3.10 Transacción presentación - dominio	23
3.11 Capa de dominio	24
3.12 Transacción dominio - datos	25
3.13 Capa de datos	25
3.14 Ejemplo de grafo	27
3.15 Distancia a lo largo de un círculo máximo	28
3.16 Ruta entre los puntos WIK y BPK	29
3.17 Ruta entre los puntos SUL y ABERU	30
3.18 Ruta entre los puntos KU y ING	31
3.19 Ruta entre los puntos WAL y ALL	31
3.20 Ruta entre los puntos TFN y TLD	32
3.21 Ruta entre los puntos DINRO y ADOKI	32
3.22 Ruta entre los puntos DINRO y ARD	33
3.23 Ruta entre los puntos BLASO y MEBUT	34
3.24 Procedimientos de salida y llegada del aeropuerto de Girona	36
3.25 Procedimientos de salida del aeropuerto de Girona	37
3.26 Procedimientos de llegada del aeropuerto de Girona	37
4.1 Planificación inicial del proyecto	39
4.2 Diagrama de Gantt del proyecto	40

INTRODUCCIÓN

En este trabajo hay dos partes claramente diferenciadas, la primera es una introducción sobre las rutas aéreas y las fases de un vuelo. En ella se puede encontrar como se establece una ruta aérea, su clasificación en rutas convencionales o rutas RNAV, las diferentes zonas de estatuto particular, como por ejemplo zonas prohibidas o zonas restringidas, las rutas condicionales, que son y cuando se pueden utilizar. Además también se puede encontrar las fases de un vuelo IFR explicadas detalladamente y una pequeña explicación sobre la afectación de la meteorología en un vuelo. Toda esta información se puede encontrar en el capítulo 2

La segunda parte está referida al desarrollo de la aplicación, en la que se puede encontrar cómo funciona la aplicación y sus diferentes funcionalidades para que así el usuario pueda utilizarla sin tener ninguna duda. Además hay diferentes ejemplos de cálculos de ruta y su comprobación con el programa SkyView. La aplicación está preparada para que pueda ser mejorada y ampliada, por ello existe un apartado donde se recomiendan posibles mejoras y ampliaciones. También está explicada la arquitectura del programa, para que así el usuario conozca que la aplicación está programada en tres capas y cuáles son las diferentes clases que las componen. Por último existe una explicación en detalle del algoritmo utilizado y el porqué de su elección, ya que existen diferentes algoritmos que puedan calcular la ruta más corta entre dos puntos. Para acabar con esta segunda parte, se ha confeccionado un apartado para explicar el ciclo de vida de la aplicación, es decir el desarrollo de cada una de las fases para finalmente llegar a tener la aplicación en funcionamiento. Esta información se puede encontrar en el capítulo 3

Además de estas dos grandes partes, existe un pequeño apartado de gestión del proyecto, en el que se puede encontrar la planificación inicial del proyecto comparada con el desarrollo que se ha llevado a cabo en realidad, que se puede encontrar en el capítulo 4

Por último están los apéndices, que son diferentes documentos técnicos con los cuales se ha podido desarrollar la aplicación. Hay cuatro documentos, el de casos de uso del software, especificación del software, arquitectura del software y los test de validación.

CAPÍTULO 1. GLOSARIO

ADF: Automatic Direction Finder

AIP: Publicación de Información Aeronáutica

AMC: Airspace Management Cell

ATC: Air Traffic Control

ATIS: Sistema de Transmisión de Información de Aeropuerto

ATS: Air Traffic Services

AUP: Airspace Use Plan

CDR: Conditional Rute

CDR1: Conditional Rute type 1

CDR2: Conditional Rute type 2

CDR3: Conditional Rute type 3

GRAM: Conditional Routes Availability Message

DME: Distance measuring equipment

FAF: Final Approach Fix

FPL: Filed Flight Plan

GPS: Global Positioning System

IAF: Initial Approach Fix

IFR: Instrumental Flight Rules

ILS: Instrument Landing System

INS: Inertial Navigation System

ISA: International Standard Atmosphere

LLC: Localizer

LORAN: Long Range Navigation

NDB: Non Directional Beacon

OACI: Organización de Aviación Civil Internacional

RNAV: Area Navigation

RNP: Required Navigation Performance

RPL: Repetitive flight plan

SID: Standard Instrument Departures

STAR: Standard Instrument Arrivals

TSA: Temporary segregated areas

VFR: Visual Flight Rules

VOLMET: Sistema de Transmisión de Información Meteorológica a aeronaves en vuelo

VOR: Very High Frequency Omnidirectional Range

WGS-84: World Geodetic System 84

CAPÍTULO 2. RUTAS AÉREAS

2.1. Introducción

Las aeronaves a la hora de realizar un vuelo, se pueden basar en dos reglas, en reglas de vuelo visual (VFR) o en reglas de vuelo instrumental (IFR). En las primeras, como dice su nombre, la única ayuda para el vuelo es la vista y las segundas están basadas en radioayudas. Las fases de un vuelo IFR están explicadas en detalle en el apartado 2.2.. Las radioayudas son equipos electrónicos que se encuentran en tierra, estos equipos emiten señales que son recibidas por las aeronaves a través del equipo de a bordo. Hay diferentes tipos de radioayudas, y se pueden clasificar según el tipo de información que proporcionan: [2]

- Dirección a un punto fijo: en este tipo de radioayudas se indica mediante una aguja la dirección en la que tendría que volar el piloto para llegar a un punto de referencia dado. Esta información es proporcionada por el sistema ADF/NDB.
- Azimutales: como indica su nombre, esta radioayuda indica el azimut, que es el ángulo horizontal formado entre un eje de referencia (por ejemplo el vector radioayuda-norte magnético), y el vector radioayuda-aeronave. El VOR y el ILS/LLC son ejemplos de radioayudas de este tipo. En el apartado 2.3.1. está explicado más detalladamente el funcionamiento de un VOR.
- Genitales: en este caso la radioayuda proporciona el ángulo vertical entre el eje de referencia radioayuda-horizonte y el vector radioayuda-aeronave. Esta información la proporciona el sistema ILS/GS.
- De distancia: este tipo de radioayudas proporcionan la distancia entre la radioayuda y la aeronave. La radioayuda que realiza esta función es el DME.

A partir de estas radioayudas, conocidas también como nav aids, las rutas aéreas son establecidas. Además de las nav aids, existen los waypoints, que son un conjunto de coordenadas que identifican un punto en concreto, estos puntos se utilizan también para establecer rutas del tipo RNAV que está explicado en más detalle en el apartado 2.3.2.. La determinación de las rutas ATS está de acuerdo con lo establecido por OACI en el anexo 11 y el documento 8168 (edición 4 de 1993). [3]

La anchura de las rutas ATS está de acuerdo a lo especificado en el Anexo 11 de OACI para las rutas basadas en un VOR, que depende de la distancia a la que estén separadas estas radioayudas. En las rutas basadas en un NDB la anchura de cada tramo es 5 NM a cada lado del eje en las primeras 42 NM, ensanchando hacia cada lado con un ángulo de $6,9^\circ$ hasta alcanzar el máximo valor en el punto medio de su cobertura. Por último, la anchura de las rutas RNAV también está especificado en el Anexo 11. [3]

En las rutas ATS se especifican sus límites verticales. El límite inferior asegura como mínimo, en condiciones de atmósfera estándar internacional (ISA), una separación vertical

de 300 m (1000 ft), 600 m (2000 ft) en terreno montañoso, por encima del obstáculo más alto o del límite superior de las zonas restringidas. La altitud/nivel de vuelo mínimo utilizable por un vuelo IFR durante el vuelo es 500 ft o 1500 ft por encima del límite inferior de la ruta, dependiendo de si la ruta requiere nivel par o impar, a menos que en la casilla de observaciones del plan de vuelo se especifique otra altitud/nivel de vuelo. En la figura 2.1 se puede observar un ejemplo de rutas ATS del espacio aéreo superior de España. [3]

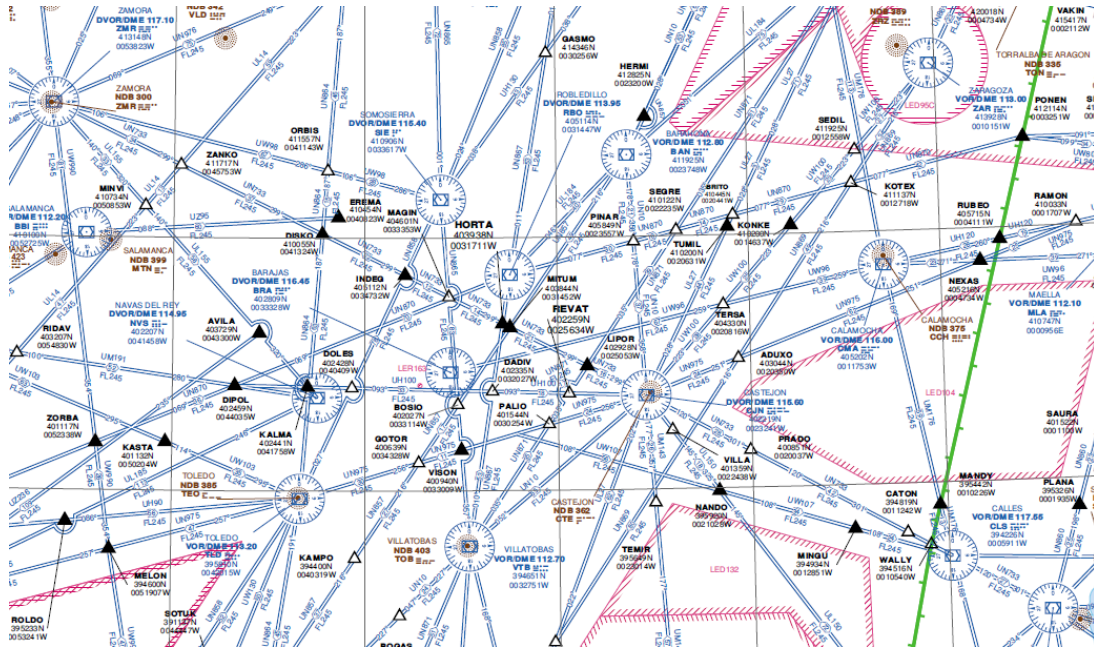


Figura 2.1: Rutas ATS del espacio superior de España

Para el establecimiento de una red de rutas ATS, en la mayoría de los casos, se suele seguir los siguientes pasos: [4]

- Los explotadores, es decir las compañías aéreas, determinan sus necesidades presentes y previstas respecto a las rutas entre los aeródromos que utilizan.
- Las demandas, alguna vez bastante contradictorias, de los distintos explotadores se unifican en una configuración racionalmente coherente de sus necesidades en cuanto a las rutas.
- Esas necesidades se evalúan luego con las otras demandas relacionadas con el espacio aéreo atravesado por esas rutas (sectores militares, evitación de volar por encima de instalaciones en tierra relacionadas con la seguridad nacional, etc.) y se estudian otras posibilidades para poder alinear exactamente las distintas rutas.
- Esas otras posibilidades se ponen en conocimiento y negocian con los explotadores interesados hasta llegar racionalmente a una fórmula de compromiso.
- En casos comparativamente raros, cuando se ve que las ofertas que puedan hacerse a los explotadores no son aceptables, hay que convenir en que las necesidades originales sean objeto de consideración ulterior por todas las partes interesadas, hasta el momento en que circunstancias más favorables permiten llegar a una alineación que sea bastante aproximada a la solicitada por los explotadores.

2.2. Fases de un vuelo

En este apartado se describirán las diferentes fases de un vuelo IFR para que pueda entenderse como se desarrolla, es decir, los diferentes pasos que hay que seguir en cada una de las fases.

2.2.1. Rodaje

En esta primera fase el piloto recibe información sobre la meteorología. Lo primero que tiene que hacer es pedir autorización a la torre de control del aeropuerto para iniciar el rodaje, que consiste en cuando la aeronave se mueve sobre la superficie del aeródromo con fuerza propia antes de despegar. Esta fase de vuelo incluye las siguientes sub-fases: [5] [6] [7]

- Rodaje hacia la pista: Empieza cuando la aeronave comienza a moverse con su propia potencia alejándose de la puerta, rampa, plataforma, o área de estacionamiento y termina al llegar a la pista de aterrizaje. Este proceso también se conoce como taxi. [6]
- Rodaje a la posición de despegue: Desde que entra a la pista hasta que llega a la posición de despegue. [6]

2.2.2. Despegue y ascenso

El despegue comienza desde la aplicación de potencia de despegue hasta una altitud de 35 ft por encima de la elevación de la pista o hasta la recogida del tren de aterrizaje, lo que ocurra primero. [7]

El controlador local asigna un punto de ruta, es decir, el último punto de la SID (Standard Instrument Departure). Una SID es una serie de puntos de salida y procedimientos normalizados definidos para algunos aeropuertos, que sirven para volar desde cada una de las pistas de despegue hasta cualquiera de los puntos de salida. De esta manera es más fácil gestionar el tráfico de salida del aeropuerto, permitiendo enlazar el despegue con la navegación en ruta. La figura 2.2 es un ejemplo de una SID del aeropuerto de Mallorca extraída del AIP de España. [7]

Los segmentos en que se suele dividir el despegue son: [7]

- 1^{er} segmento: Comienza cuando se separan las ruedas de la pista. En este primer segmento el tren de aterrizaje está fuera o recogido, los flaps en posición de despegue y el empuje o potencia de despegue (con un motor inoperativo).
- 2^o segmento: Empieza en el momento en que el tren está completamente recogido. El cambio que existe entre el 1^{er} y 2^o segmento corresponde a una configuración diferente del avión. Este segmento puede acabar:

los flaps y se ha alcanzado la velocidad del segmento final, y hasta a una altura de 1500 ft como mínimo. El empuje o potencia es el máximo continuo. En el caso que el 3er segmento se realice en una altura superior a 1500 ft, el segmento final no existe, y la senda de despegue termina al final del 3er segmento.

En la figura 2.3 se puede observar claramente los diferentes segmentos del despegue.

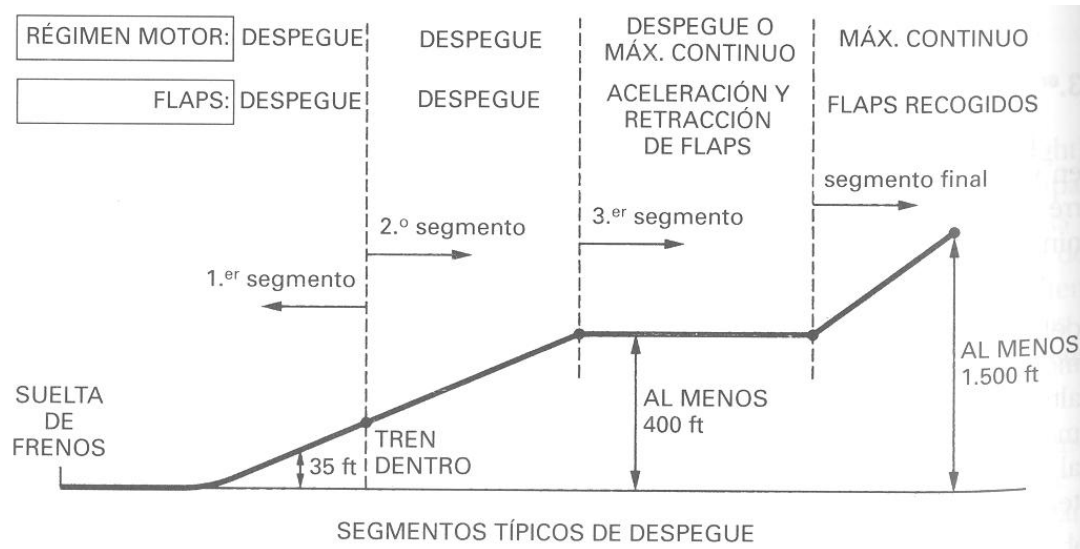


Figura 2.3: Segmentos de despegue [1]

2.2.3. En ruta

La fase de ruta va desde que se termina el ascenso inicial pasando por la altitud de crucero hasta llegar al primer punto de la STAR (Standard Terminal Arrival Route). Esta fase de vuelo incluye las siguientes sub-fases: [6]

- Ascenso a crucero: Desde la conclusión del ascenso inicial hasta llegar a la altitud inicial de crucero asignada.
- Crucero: Cualquier segmento de vuelo horizontal después de alcanzar la altitud crucero inicial hasta comenzar el descenso al punto de destino.
- Cambio del nivel de crucero: Cualquier ascenso o descenso durante el crucero después del ascenso inicial a crucero, pero antes del descenso al punto de destino.

2.2.4. Llegada y aproximación

La llegada consiste en el descenso desde crucero hasta la posición inicial de la aproximación (IAF) o de entrada al circuito VFR. Este procedimiento se denomina STAR (Standard Terminal Arrival Route) y al igual que la SID, consiste en una serie de puntos de llegada y procedimientos normalizados definidos para algunos aeropuertos. La figura 2.4 es un ejemplo de una STAR del aeropuerto de Mallorca extraída del AIP de España. [6] [7]

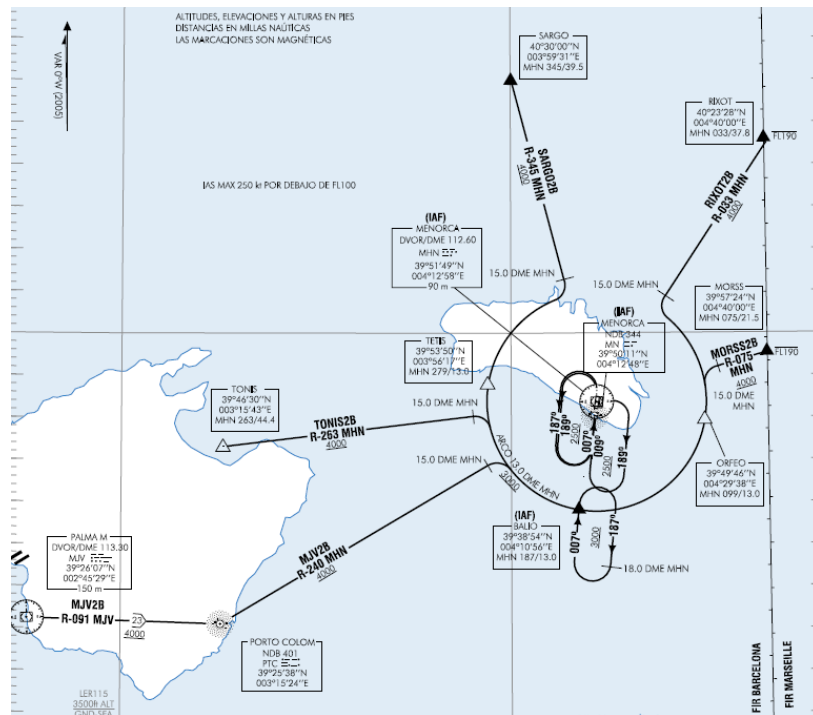


Figura 2.4: Ejemplo STAR Mallorca

Al acabar la llegada hay que empezar con la aproximación, que va desde el punto de posición inicial de la aproximación (IAF), hasta el comienzo del enderezamiento para aterrizar. Esta fase de vuelo incluye las siguientes sub fases: [6] [7]

- Aproximación inicial: De la IAF a la posición final de la aproximación (FAF).
- Aproximación final: Desde la FAF hasta el comienzo del enderezamiento para aterrizar.

2.2.5. Aterrizaje

Esta fase va desde el comienzo del enderezamiento para aterrizar hasta que la aeronave sale de la pista de aterrizaje o se detiene en la pista. Esta fase de vuelo incluye las siguientes sub-fases: [6]

- Enderezamiento para aterrizar: Transición de la posición nariz-baja a nariz-alta inmediatamente antes de aterrizar, hasta el momento de tocar el suelo.
- Recorrido de aterrizaje: Después del momento de tocar el suelo hasta que la aeronave salga de la pista de aterrizaje o se detenga, lo que ocurra primero.

Por último el control de tierra controla el avión hasta su estacionamiento en la plataforma. Las normas para el rodaje posterior al aterrizaje son las mismas que las que se aplican en el rodaje previo al despegue. [6]

2.3. Clasificación de las rutas ATS

Las rutas aéreas se pueden clasificar básicamente en dos tipos: Rutas convencionales y rutas RNAV.

2.3.1. Rutas convencionales

Este método de navegación se basa en zonas del espacio aéreo que están definidas por puntos que, o bien están balizados por radioayudas, o bien se pueden identificar visualmente. [8]

Los sistemas de navegación que se utilizan para realizar esta navegación son el VOR, el NDB y el DME. El VOR (Very High Frequency Omnidirectional Range) es tal vez el más difundido. Consta de dos equipamientos, un transmisor fijo en tierra y el receptor montado en el avión enlazados por ondas de radio de muy alta frecuencia comprendidas entre 108 y 118 MHz. [9]

La estación de tierra es un transmisor omnidireccional, esto quiere decir que transmite en todas las direcciones, captándose si esta dentro del alcance, sus señales independientemente uno se encuentra en el sur, norte, este u oeste.

La información transmitida es tal que tras decodificarla por el sistema de recepción montado en el avión, pueda determinarse en qué posición se encuentra el avión respecto de la estación de tierra.

La idea en principio es muy simple, por ejemplo para viajar de una ciudad a otra en automóvil solo basta con saber que ruta las une, luego se sigue esa ruta para llegar al destino, con los VOR ocurre lo mismo, con conocer cuales radiales de cuales VOR nos conducen al destino, con solo seguirlos llegaremos al punto deseado. De igual forma se pueden determinar posiciones e intercepciones.

Pero todo no es tan fácil, ya que esta forma de operación no es muy adecuada porque como dice en [10]:

- Es inflexible: Depende de la posición geográfica de las ayudas de navegación, que son fijas.
- Es costoso: Las aeronaves consumen más combustible para ir del origen al destino.
- Introduce retrasos: Las aeronaves invierten mayor cantidad de tiempo en su viaje.
- No crece con facilidad: La creación de nuevas rutas aéreas está supeditada a la construcción e instalación de nuevas estaciones de radioayudas en tierra.
- Sobrecarga los servicios ATS: La mayor parte del tráfico aéreo pasa por ciertos nodos importantes de la red de rutas, donde confluyen muchas aeronaves simultáneamente.

- Desaprovecha la evolución tecnológica: La tecnología de los sistemas de navegación aérea ha avanzado vertiginosamente en los últimos años, pero estos avances no se habían tomado en cuenta para mejorar las operaciones.

Con el fin de acabar con estos inconvenientes y responder a la mayor demanda de capacidad y eficacia en la gestión del espacio aéreo, numerosas autoridades aeronáuticas en todo el mundo empezaron a alcanzar acuerdos y a desarrollar estrategias para mejorar la situación.

2.3.2. Rutas RNAV

Las siglas RNAV corresponden a *Area Navigation* o Navegación de Área. Se trata de un método de navegación que permite que una aeronave se desplace en cualquier trayectoria deseada, sin la necesidad de pasar sobre puntos predefinidos por la existencia de radioayudas en tierra. [10] [11]

Este tipo de operación se hará dentro del área de cobertura de las radioayudas utilizadas como puntos de referencia o dentro de los límites de la capacidad de los sistemas de navegación autónomos (como el sistema inercial), o una combinación de ambos. [10]

Por tanto, RNAV emplea los datos que obtiene de las distintas radioayudas o del equipo de a bordo para hacer cálculos de posición, permitiendo conocer la ubicación de la aeronave mientras se cumplan ciertos requisitos y siempre con un área de incertidumbre que vendrá dada por los propios sistemas empleados. OACI, la Organización de Aviación Civil Internacional, establece los márgenes de error permitidos en base a la tipología de RNAV. [11]

Los equipos de navegación a bordo determinarán automáticamente la posición de la aeronave según la información recibida y la controlarán para que siga la ruta pre-establecida. Los sistemas que pueden utilizarse (de manera individual o combinada) son: [10] [11]

- VOR/DME
- DME/DME
- INS
- LORAN-C
- GPS

RNAV tiene como objetivo la optimización del uso de la red ATS (tanto en ruta como en las áreas terminales), para proporcionar una mayor capacidad del espacio aéreo, junto una mayor eficiencia en las operaciones. Esto se logra porque se reducen las separaciones laterales entre las trayectorias de las aeronaves y se utilizan nuevas rutas que no están atadas al sobrevuelo de determinadas radioayudas. [10]

Entre los numerosos beneficios de esta forma de operación, tal y como se dice en [10] se pueden citar, entre otros:

- Rutas flexibles, modificables rápidamente.
- Rutas más directas.
- Procedimientos SID y STAR optimizados.
- Carga de trabajo menor, tanto para los pilotos como para los controladores.
- Menor impacto ambiental gracias al ahorro de combustible (rutas más cortas) y reducción del ruido (procedimientos más óptimos).

Sin embargo, para que se pueda implementar adecuadamente el concepto RNAV y que éste sustituya a la radionavegación convencional, es necesario que se cumplan primero una serie de premisas básicas: [10]

- Los sistemas de navegación de a bordo de las aeronaves deben tener la capacidad de operar bajo este esquema.
- Las tripulaciones de vuelo deben estar adecuadamente entrenadas para pilotar según las técnicas RNAV.
- La infraestructura de navegación debe proporcionar una cobertura adecuada.
- Las coordenadas de navegación deben cumplir los requerimientos de exactitud e integridad establecidos por la OACI. Esto implica que todos los datos de navegación disponibles deben tener como referencia el sistema geodético WGS-84.
- Se han diseñado (y publicado) procedimientos y rutas compatibles con los sistemas RNAV a bordo de las aeronaves.

Dado que el cumplimiento de las condiciones anteriores requiere de un proceso de evolución que tomará un cierto tiempo, la implantación de RNAV se ha dividido en diferentes fases. La primera de éstas se ha denominado B-RNAV y su exactitud (RNP-5) es aproximadamente igual a la que se obtiene utilizando las técnicas de navegación convencionales. [10]

Este tipo de RNAV básica también es llamada RNAV 2D, pues en este caso el sistema de navegación debe ser capaz de cumplir los requerimientos de exactitud en un plano horizontal. Se tiene previsto que posteriormente se incorporen sistemas RNAV 3D (incorporando exactitud en el plano vertical) y RNAV 4D (agregando parámetros de exactitud en la variable tiempo). [10]

Cuando hablamos del concepto RNP (Required Navigation Performance), se refiere a la exactitud de las prestaciones de navegación requeridas para poder operar dentro de un espacio aéreo determinado. Para determinar estos requisitos en las prestaciones primero se define un objetivo de seguridad que se desea alcanzar. Este objetivo de seguridad puede interpretarse como una región de confinamiento dentro de la cual la aeronave debe ser capaz de mantenerse durante su vuelo. Formalmente, cada tipo RNP define un error total del sistema de navegación que se permite en el eje lateral, longitudinal y vertical, este solo en algunos casos. Ahora bien, no es el error cometido el único parámetro que ha de tomarse en cuenta al evaluar un sistema de navegación dado, ya que los parámetros que definen el concepto RNP son: [12]

- Exactitud
- Integridad
- Continuidad
- Disponibilidad

2.4. Zonas de estatuto particular

Todo espacio aéreo en el cual pudiera existir un peligro potencial para la operación de aeronaves y/o restringirse temporal o permanentemente la operación de aeronaves civiles, como se dice en el documento de Zonas prohibidas, restringidas y peligrosas del AIP de España [13], se clasifica en:

- Zona prohibida (P): Espacio aéreo de dimensiones definidas sobre el territorio o las aguas jurisdiccionales de España, dentro del cual está prohibido el vuelo de aeronaves, excepto las españolas de estado debidamente autorizadas por el Ministerio de Defensa.

Ninguna aeronave volará en una zona prohibida (P) sin haber obtenido el preceptivo permiso del estado español. (ej. Alrededor de una central nuclear)

- Zona restringida (R): Espacio aéreo de dimensiones definidas sobre el territorio o las aguas jurisdiccionales de España, dentro del cual está restringido el vuelo de las aeronaves de acuerdo con determinadas condiciones especificadas, excepto las aeronaves españolas de estado cuando así lo exija el cumplimiento de su cometido o por razones de emergencia.

Ninguna aeronave volará en una zona restringida (R) sin ajustarse a las restricciones establecidas, salvo que haya obtenido permiso de la autoridad ATS competente. (ej. Paracaidismo, artillería)

- Zona peligrosa (D): Espacio aéreo de dimensiones definidas en el cual pueden desplegarse en determinados momentos actividades peligrosas para el vuelo de las aeronaves.

Antes de iniciar un vuelo que afecte a una zona peligrosa (D), el piloto obtendrá toda la información necesaria sobre las actividades de la misma y horario en que tendrán lugar. (ej. Ejercicios acrobáticos)

- Zonas D y R manejables: Zonas militares que manteniendo su concepto D o R actual pueden gestionarse por la AMC dentro de los periodos publicados en el AIP.
- Áreas temporalmente segregadas (TSA): Espacio aéreo de dimensiones definidas cuya activación requiere una reserva de espacio aéreo para uso exclusivo de usuarios específicos durante un período de tiempo determinado.

2.5. Rutas condicionales

Las rutas condicionales (CDR) son rutas o tramos de rutas ATS, de carácter no permanente, que sólo se pueden planificar y utilizar bajo ciertas condiciones específicas dentro de los periodos publicados. [14]

Como se dice en el documento del Uso flexible del espacio aéreo del AIP de España [14], se dividen en tres categorías en función de su posible utilización en los planes de vuelo:

- CDR1: Se establecen en la fase estratégica. Están disponibles la mayor parte del tiempo, por lo que se pueden planificar permanentemente en los planes de vuelo (RPL y FPL).

A diario se distribuye el AUP y el CRAM con las rutas CDR1 que se cierran. Los RPL afectados por rutas cerradas temporalmente deberán cancelarse, y se presentará un nuevo FPL que incluya en la casilla 15, la ruta alternativa publicada que corresponda a cada CDR1 no disponible. Cuando una CDR1 se deba cerrar al tráfico con poco tiempo de preaviso, el ATC dará instrucciones a los vuelos para utilizar rutas alternativas en la fase táctica.

- CDR2: Se gestionan en la fase pretáctica. No se pueden planificar permanentemente. Las CDR2 sólo se podrán incluir en los FPL de acuerdo con las condiciones que se publiquen diariamente, el día antes de la operación, en el Mensaje de Disponibilidad de Rutas Condicionales (CRAM).
- CDR3: Se utilizan en la fase táctica. No se pueden planificar en plan de vuelo. Solo se pueden utilizar bajo autorización ATC, previa coordinación civil-militar.

Las CDR atraviesan Zonas Temporalmente Segregadas (TSA) o Zonas Manejables Peligrosas y Restringidas. Los periodos en que dichas rutas o tramos se clasifican como CDR2 o CDR3 deben ser coincidentes con los periodos de actividad de las zonas que atraviesan.

Un mismo tramo de ruta ATS puede ser condicional 1, 2 o 3 en periodos horarios diferentes. En España, fuera de los horarios y los límites verticales publicados como CDR, la ruta ATS es de utilización normal.

Las distintas fases en las que se establecen las diferentes rutas, provienen de la existencia de tres niveles de gestión del espacio aéreo:

- Nivel 1 - Estratégico: en el que se establece una planificación a largo plazo de la política nacional de gestión del espacio aéreo y sus estructuras, mediante un proceso conjunto civil/militar.
- Nivel 2 - Pretáctico: en el que se realiza la gestión día a día, en el día anterior a las operaciones, y la asignación temporal de espacio aéreo a través de la Célula de Gestión de Espacio Aéreo (AMC) que recoge y analiza todas las peticiones de espacio aéreo y negocia y decide diariamente su asignación.

- Nivel 3 - Táctico: en el que se gestiona el uso del espacio aéreo en tiempo real.

2.6. Meteorología

A pesar de las condiciones y características de una aeronave, potencia, sistemas de navegación, etc., la seguridad en vuelo se encuentra todavía sujeta a las condiciones atmosféricas existentes, tanto en la ruta de vuelo como en los aeródromos de salida, aterrizaje. Por lo tanto, para poder planificar un vuelo se precisa de dicha información, que es proporcionada por las oficinas de información aeronáutica. Asimismo, en determinados casos, es posible obtener la información meteorológica necesaria utilizando: el sistema automático ATIS (aeródromos de salida, llegada y alternativa), VOLMET (informes meteorológicos en vuelo), teléfono o fax, etc. [15]

Con anterioridad al inicio de un vuelo es preciso efectuar una evaluación de la situación meteorológica existente en el aeródromo de salida, la existente y prevista a lo largo de su ruta de vuelo, así como las condiciones previstas del aeródromo de destino y de los posibles aeródromos alternativos. A partir de esta evaluación, se podrá planificar la ruta. [15]

Uno de los fenómenos meteorológicos más importantes es el viento. El viento afectará a las aeronaves dependiendo de su dirección y de su velocidad. Si existe viento de cola, el avión alcanzará antes el destino, si el viento es de cara, lo que ocasionará es que el avión se retrase y tarde más en llegar a su destino, y por último existe la posibilidad del viento lateral, que lo que ocasiona son desvíos laterales del avión. Contra mayor sea la velocidad del viento mayores serán estos efectos. Por lo tanto, habrá que diferenciar entre la distancia terrestre, distancia física entre dos puntos, y la distancia aire, distancia teniendo en cuenta el efecto del viento.

CAPÍTULO 3. DESARROLLO DEL SOFTWARE

3.1. Introducción a la aplicación

El programa que se ha diseñado e implementado es una aplicación que permite conocer la ruta más corta entre dos puntos del espacio aéreo europeo, si es que la hay. Estos puntos son waypoints y nav aids. Además al acabar de calcular la ruta, el programa informará dirá la distancia que se ha recorrido, la ruta que ha seguido y el tiempo que se ha tardado, eligiendo previamente el mach y flight level de crucero, el equipo de navegación de la aeronave, el día y la hora de salida del vuelo.

La aplicación permite calcular la ruta entre dos puntos del espacio aéreo europeo. Este cálculo llevará a obtener los siguientes resultados:

- **Calculo de rutas:** La aplicación calculará la ruta más corta entre los dos puntos del espacio aéreo europeo seleccionados y la mostrará en pantalla.
- **Calculo de distancia:** La aplicación calculará la distancia que ha recorrido la aeronave y la mostrará en la pantalla.
- **Calculo de tiempo:** La aplicación calculará el tiempo que tarda la aeronave en recorrer la ruta, eligiendo previamente el mach de crucero, y lo mostrará en pantalla.

Para poder realizar el programa, en primer lugar o se necesita una base de datos de donde se pueda extraer la información necesaria de todo el espacio europeo para poder cargarla en nuestro programa. La base de datos que se ha utilizado se ha obtenido de SkyView. Se trataba de una base de datos en formato acces por lo que se ha tenido que utilizar el SQLite para convertir la base de datos en un formato de base de datos para linux para así poder leerla usando comandos SQL. Además del eclipse para el entorno de desarrollo, para la parte gráfica se ha utilizado el qt designer.

El programa tiene principalmente tres funcionalidades, crear un vuelo, añadir restricciones al vuelo y calcular la ruta. En la figura 3.1 se puede observar el aspecto de la pantalla principal de la aplicación.

Primeramente hay que elegir un waypoint o nav aid de salida y otro de destino. También existe la opción de elegir un aeropuerto de salida y otro de llegada, pero el programa no permite calcular la ruta entre dos aeropuertos porque, pese a que están cargados, no están conectados con los demás puntos del grafo. En la figura 3.2 se puede observar cómo seleccionar el punto de origen y de destino.

Una vez elegidos la salida y el destino, el siguiente paso es crear el vuelo y para ello hay que clicar el botón "Características de vuelo". En esta ventana podremos introducir diferentes parámetros que afectaran a nuestra ruta. Estos parámetros son el equipo de navegación, el mach de crucero, el flight level, el día de la semana que se realizará el vuelo y la hora del día en que se volará. De todos estos parámetros los únicos que afectarán serán el mach y el flight level a la hora de calcular el tiempo en vuelo de nuestra ruta,

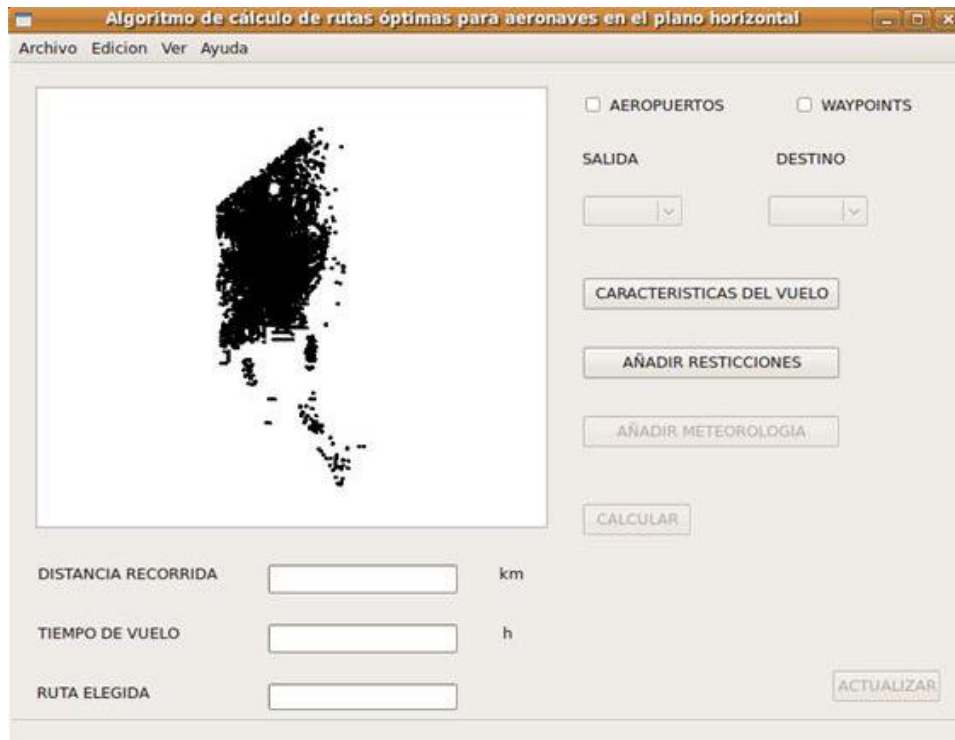


Figura 3.1: Pantalla principal de la aplicación



Figura 3.2: Selección del origen y el destino

ya que los otros tres parametros se permiten elegir para una posible ampliación de la aplicación. En la figura 3.3 se puede observar la ventana de características de vuelo.

Una vez elegidas las características del vuelo ya podemos calcular la ruta. Antes de calcularla, se pueden añadir restricciones si se quiere, aunque en esta versión de la aplicación las restricciones no afectan ya que no están programadas las funciones para que lo realice, no obstante está preparado para poder añadir esta nueva funcionalidad. En la figura 3.4 se puede observar la ventana de añadir restricciones.

Por último, calculamos nuestra ruta, dónde nos podemos encontrar con dos situaciones: que haya una ruta entre los dos puntos elegidos, entonces el programa nos mostrará el identificador de los puntos por los cuales ha tenido que pasar para llegar de uno a otro, la distancia que ha recorrido en kilometros y el tiempo estimado de vuelo en horas o que no haya ruta, entonces el programa nos dirá que no hay ruta entre estos dos puntos. En la figura 3.5 se puede observar el resultado que nos da la aplicación al encontrar la ruta más corta entre dos puntos.



The screenshot shows a dialog box titled "Características de vuelo". It contains several input fields: "EQUIPO DE NAVEGACIÓN" with a dropdown menu set to "Equipo convencional"; "MACH DE CRUCERO" with a spinner box set to "0.72"; "FLIGHT LEVEL DE CRUCERO" with a spinner box set to "280"; "DIA DE LA SEMANA" with a dropdown menu set to "Viernes"; and "HORA DE SALIDA" with a spinner box set to "11:17". At the bottom, there are two buttons: "Aceptar" and "Cancelar".

Figura 3.3: Características de vuelo



The screenshot shows a dialog box titled "Restricciones de vuelo". It has two sections. The first section, "Rutas condicionales", has three radio buttons: "No usarlas nunca" (checked), "Usarlas cuando se pueda", and "Usarlas siempre". The second section, "Anulos de giro máximo", has two radio buttons: "No tenerlos en cuenta" (checked) and "El ángulo mínimo entre aerovías deberá ser:" followed by a spinner box set to "1". At the bottom, there are two buttons: "Aceptar" and "Cancelar".

Figura 3.4: Restricciones de vuelo



The screenshot shows a summary of flight calculation results. It consists of three rows, each with a label, a text input field, and a unit. The first row is "DISTANCIA RECORRIDA" with the value "2921.75" and unit "km". The second row is "TIEMPO DE VUELO" with the value "3.6886" and unit "h". The third row is "RUTA ELEGIDA" with the value "EMITOS-LEPOLOP-LEALT".

Figura 3.5: Resultado del calculo de la ruta más corta

3.2. Ciclo de vida de la aplicación

El desarrollo de la aplicación se ha llevado a cabo a partir de un ciclo en V, que esta compuesto por diferentes fases: planificación, especificación, diseño, implementación, integración y calificación. En la figura 3.6 se puede observar el ciclo que se ha seguido para desarrollar la aplicación.

Antes de empezar con la primera fase de especificación, se ha realizado un diagrama de Gantt dónde se especifica cuanto se tarda en hacer cada tarea y para cuándo estará lista.

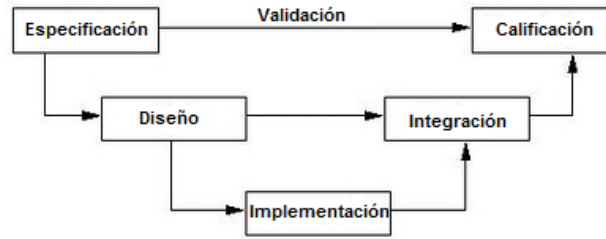


Figura 3.6: Ciclo en V de la aplicación

La planificación no ha sido como se pensó en un principio, pero esto está explicado en detalle en el capítulo 4.

En la primera fase, la fase de especificación, se trataba de especificar cómo sería la aplicación y que sería capaz de hacer. Para realizar esta tarea se desarrollaron dos documentos: Casos de uso del software y Especificación de requisitos del software. El primer documento lo podemos encontrar en el apéndice *Casos de uso del software* y explica cuáles son los actores del programa, es decir, habrá un usuario que utilizará la aplicación, un administrador y una base de datos de dónde extraer la información necesaria. Además en este documento también podemos encontrar cuáles serán las diferentes funcionalidades del programa, como por ejemplo crear un vuelo y calcular una ruta. El documento de Especificación de requisitos de software lo podemos encontrar en el apéndice *Especificación de requisitos del software* y en él están detallados los diferentes requisitos funcionales y no funcionales de la aplicación. Además en esta fase se ha realizado un test de validación de los casos de uso y de los requisitos, que al llegar a la última fase habrá que utilizarlos para verificar que cumple con las especificaciones.

En la tercera fase, el diseño, se trataba de diseñar la arquitectura del programa, es decir como estaría estructurada la aplicación. Para ello se elaboró un documento de Arquitectura del software que lo podemos encontrar en el apéndice *Arquitectura del software* y en el que podemos encontrar como está estructurada la aplicación, las clases que formaran cada una de las capas, los atributos y métodos que habrá en cada una de las clases, los diagramas de clase para así saber cómo están relacionadas y algunos diagramas de secuencia para poder interpretar que hará la aplicación cuando el usuario ejecute una determinada orden.

En la fase intermedia, la implementación, la tarea ha sido realizar el código correspondiente para cada clase de cada una de las capas. Esta fase no se ha llevado a cabo libremente, sino que ha tenido que seguir todos los pasos que se han hecho en fases anteriores para así poder obtener el producto deseado.

En la penúltima fase, la integración, lo que se ha hecho es unir todas las partes del programa para que así se pudiera poner en marcha la aplicación.

Por último, esta la fase de calificación, se trataba de pasar los test de validación que se pueden encontrar en el apéndice *Test de validación del software* y que se crearon en la fase de especificación y comprobar que cumple con cada uno de los requisitos y realiza todo lo que debía realizar, si no es así, hay que volver a la fase de implementación y a la de integración para arreglarlo, y al acabar se vuelve a pasar el test para comprobar que

los errores ya no existen, sino es así, habría que volver de nuevo hacia atrás.

3.3. Arquitectura del programa

La estructura del programa está separada en tres capas, esto recibe el nombre de programación por capas. Este estilo de programación tiene como objetivo principal la separación de la capa de presentación, capa de negocio o dominio y capa de datos. En la figura 3.7 se puede observar las diferentes capas. [16]



Figura 3.7: Estructura del programa

La ventaja principal de este tipo de arquitectura es que el desarrollo del programa se puede llevar a cabo en varios niveles, y en el caso que se quiera hacer un cambio solo habrá que modificar el nivel requerido. Además este estilo de programación también nos asegura avanzar en la programación del proyecto de una forma ordenada, lo cual nos beneficia por la reducción de tiempo, debido a que se podrá avanzar de forma más segura en el desarrollo ya que la aplicación general está dividida en varios módulos y capas que pueden ser tratados de manera independiente y hasta en forma paralela. [16] [17]

- **Capa de presentación:** Es la capa que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario. La interfaz debe ser amigable y fácil de utilizar, ya que el usuario final es el que se va encargar de utilizar el sistema. Esta capa se comunica únicamente con la capa de negocio. La capa de presentación de la aplicación se puede observar en la figura 3.9.
- **Capa de negocio:** Es la capa donde se encuentra toda la lógica del programa, así como las estructuras de datos y objetos encargados para la manipulación de los datos existentes, y el procesamiento de la información ingresada o solicitada por el usuario en la capa de presentación. Esta capa es el corazón de la aplicación ya que se comunica con todas las demás capas para llevar a cabo la aplicación, se comunica con la capa de presentación para recibir las solicitudes del usuario y presentar los resultados, y con la capa de datos para solicitar a la base de datos que almacene o recupere datos. La capa de presentación de la aplicación se puede observar en la figura 3.11.

- Capa de datos: Es la capa donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. Esta capa se comunica únicamente con la capa de negocio. La capa de presentación de la aplicación se puede observar en la figura 3.13.

La comunicación entre las diferentes capas se realizará utilizando un modelo de transacciones, de manera que un conjunto reducido de clases hará de interfaz entre las diferentes capas. En la figura 3.8 tenemos un esquema general de como está estructurada la aplicación:

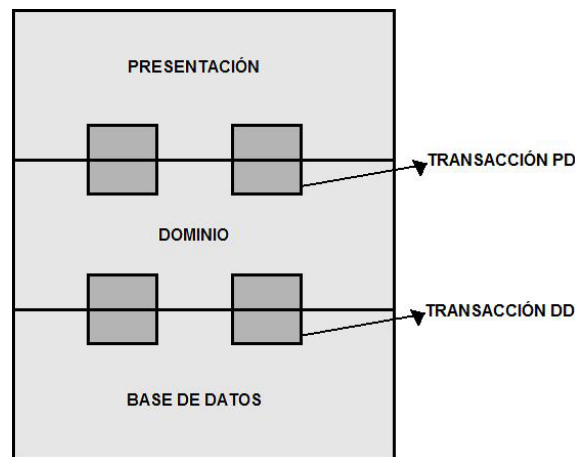


Figura 3.8: Esquema general de la estructura del programa

Tal y como se comentaba antes, la capa de presentación solo se puede comunicar con la capa de dominio y lo hará a través de las transacciones PD (presentación - dominio) que se pueden observar en la figura 3.10 y la capa de datos únicamente también con la capa de dominio y a través de las transacciones DD (dominio- datos) que se puede observar en la figura 3.12.

Además de este esquema general, antes de diseñar la aplicación se realizaron los diagramas de clase, donde se especifican las clases que va haber en cada capa, y los atributos y métodos que tendrán estas clases. En estos diagramas también se puede apreciar cómo están relacionadas las clases entre sí. Para más detalle sobre la arquitectura del software consultar el Apéndice Arquitectura de software.

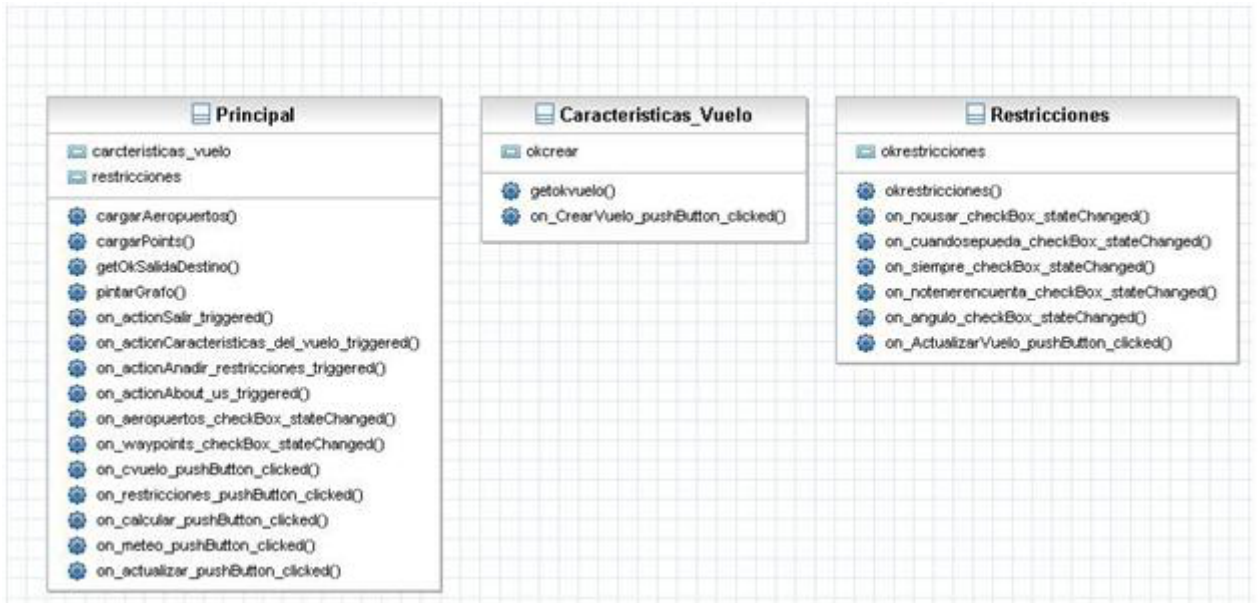


Figura 3.9: Capa de presentación

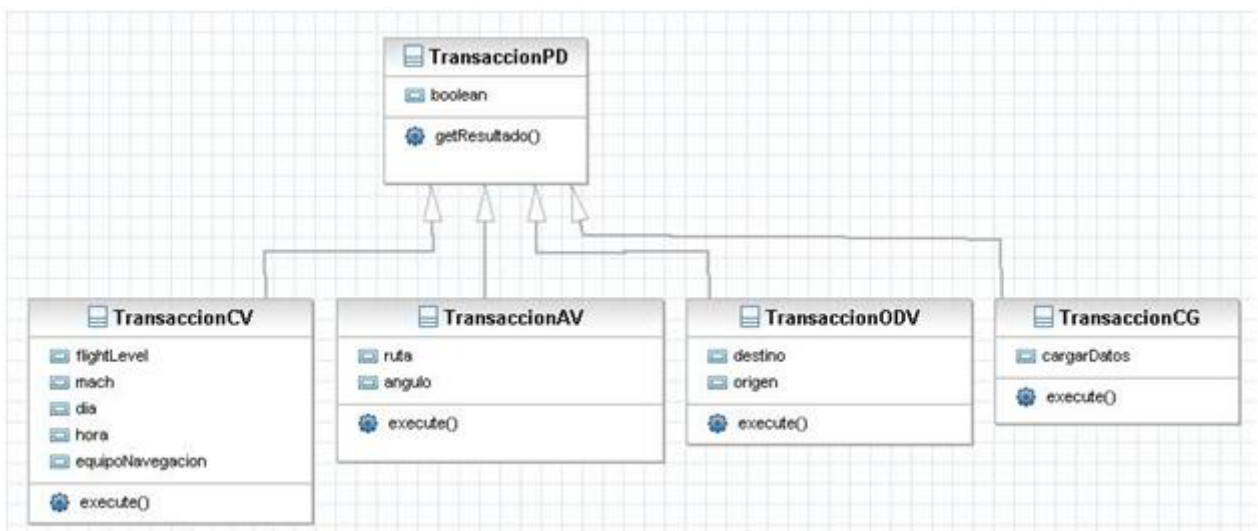


Figura 3.10: Transacción presentación - dominio

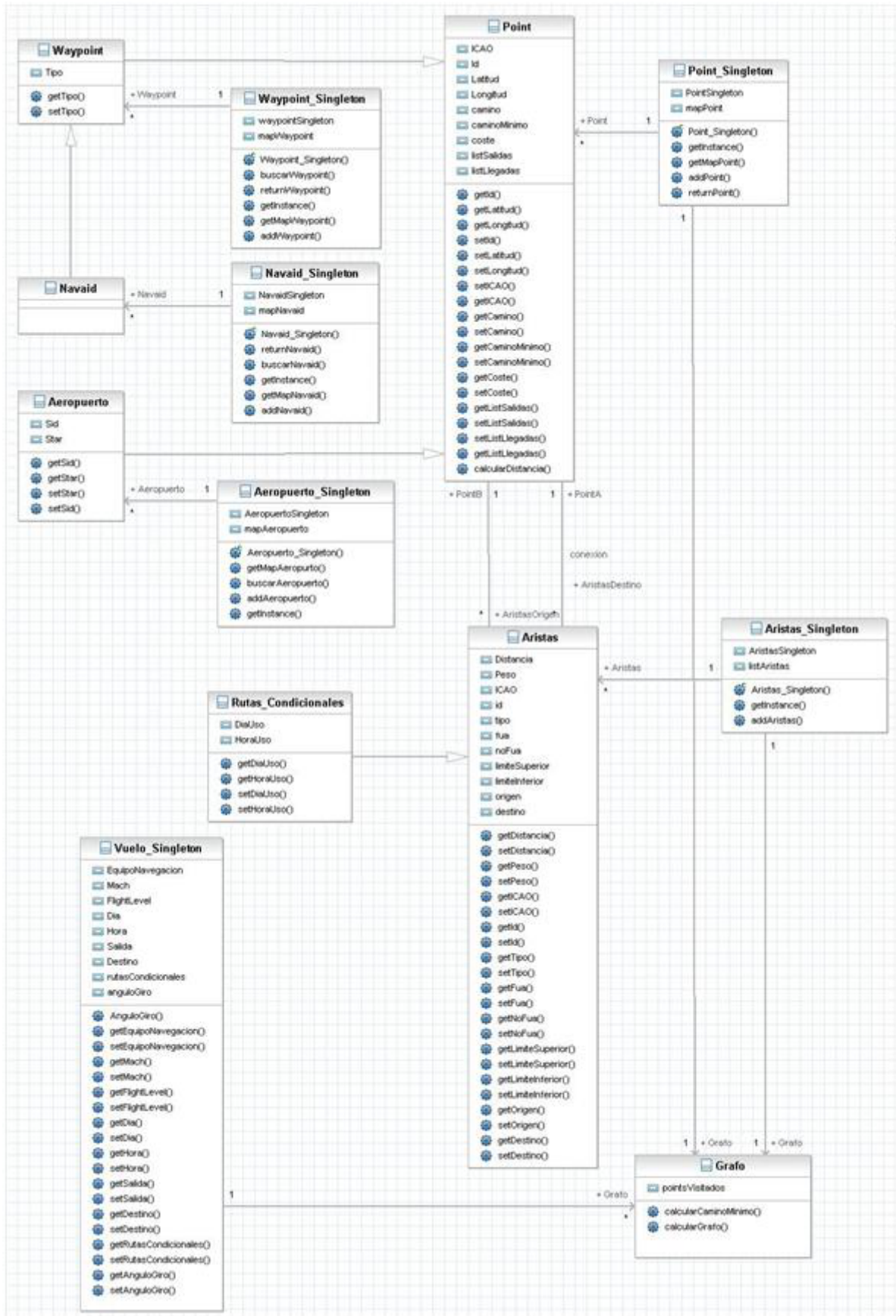


Figura 3.11: Capa de dominio



Figura 3.12: Transaccion dominio - datos



Figura 3.13: Capa de datos

3.4. Algoritmo de cálculo

El algoritmo que se ha utilizado para realizar la búsqueda del camino más corto entre dos puntos del espacio aéreo europeo es el A* (A estrella). Se trata de un algoritmo de búsqueda para grafos que fue presentado por primera vez en 1968 por Peter E. Hart, Nils J. Nilsson y Bertram Raphael. Este algoritmo utiliza una función de evaluación para determinar el orden en que la búsqueda visita nodos en el grafo. Esta función es el resultado de la suma de otras dos funciones, $g(n)$ que indica el coste del camino seguido hasta un cierto nodo y $h'(n)$ o heurístico que es una estimación admisible de la distancia hasta el destino. La función resulta entonces: [18] [19]

$$f(n) = g(n) + h'(n) \quad (3.1)$$

Como es lógico para intentar encontrar el camino más corto, el primer nodo a visitar es el nodo con menor valor de $g(n) + h'(n)$ de los nodos expandidos por el origen, mientras los demás nodos se van almacenando en una cola de prioridad. Esta prioridad es determinada por $f(n)$. El proceso continua hasta que el nodo de destino tiene el valor $f(n)$ menor, es decir, es el primero en la cola de prioridad o hasta que se han recorrido todos los nodos, lo que significaría que no hay solución. [19] [20]

Una de las características que me ha llevado a elegir este algoritmo es que ningún otro algoritmo óptimo garantiza expandir menos nodos que A*, por lo que ocupará menos memoria y será más rápido. [18]

Para que A* sea óptimo, $h'(n)$ debe ser una heurística admisible, es decir, que nunca sobrestime el coste de alcanzar el nodo de destino. Las heurísticas admisibles son por naturaleza optimistas, porque piensan que el coste de resolver el problema es menor o igual al que es en realidad. Ya que $g(n)$ es el coste exacto para alcanzar n , tenemos como consecuencia inmediata que $g(n)$ nunca sobrestima el coste verdadero de la solución a través de n . [18] [21]

La heurística que he utilizado es la distancia en línea recta que hay hasta nuestro punto de destino. Esta heurística es admisible porque el camino más corto entre dos puntos cualesquiera es una línea recta, entonces la línea recta no puede ser una sobrestimación.

De no cumplirse dicha condición, el algoritmo pasa a denominarse simplemente A, y a pesar de seguir siendo completo, no se asegura que el resultado obtenido sea el camino de coste mínimo. [21]

Seguidamente en la figura 3.14 se muestra un ejemplo de un problema resuelto con el algoritmo A*.

El objetivo es llegar del punto A al punto G por el camino más corto. Para ello hay que empezar a analizar el nodo A sabiendo que la distancia entre dos puntos es:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3.2)$$

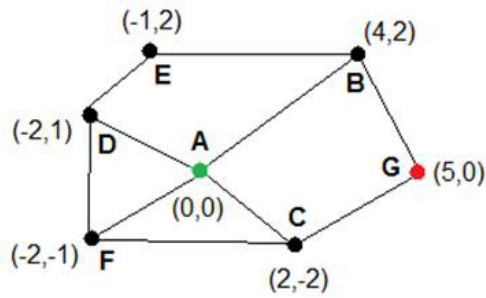


Figura 3.14: Ejemplo de grafo

Y que la heurística que utilizaré es la distancia en línea recta que hay hasta el punto G.

A esta comunicado con los nodos B, C, F y D.

$$d_{AB} = \sqrt{(4-0)^2 + (2-0)^2} = 4,47 \quad (3.3)$$

$$d_{AC} = \sqrt{(2-0)^2 + (-2-0)^2} = 2,83 \quad (3.4)$$

$$d_{AD} = \sqrt{(-2-0)^2 + (1-0)^2} = 2,24 \quad (3.5)$$

$$d_{AF} = \sqrt{(-2-0)^2 + (-1-0)^2} = 2,24 \quad (3.6)$$

Y también necesitaremos:

$$d_{BG} = \sqrt{(5-4)^2 + (0-2)^2} = 2,24 \quad (3.7)$$

$$d_{CG} = \sqrt{(5-2)^2 + (0-(-2))^2} = 3,60 \quad (3.8)$$

$$d_{DG} = \sqrt{(5-(-2))^2 + (0-1)^2} = 7,07 \quad (3.9)$$

$$d_{FG} = \sqrt{(5-(-2))^2 + (0-(-1))^2} = 7,07 \quad (3.10)$$

Ahora hay que calcular la función de evaluación para cada uno de los casos:

$$f(B) = g(B) + h'(B) = 4,47 + 2,24 = 6,71 \quad (3.11)$$

$$f(C) = g(C) + h'(C) = 2,83 + 3,60 = 6,43 \quad (3.12)$$

$$f(D) = g(D) + h'(D) = 2,24 + 7,07 = 9,31 \quad (3.13)$$

$$f(F) = g(F) + h'(F) = 2,24 + 7,07 = 9,31 \quad (3.14)$$

Ahora tenemos que el siguiente nodo a visitar es el nodo C, y si no hubiésemos utilizado $h'(n)$ el siguiente nodo sería el D o el F y no estaríamos alejando del destino.

Tenemos que volver a empezar con los pasos de antes pero con el nodo C.

$$d_{BG} = \sqrt{(5-2)^2 + (0-(-2))^2} = 3,60 \quad (3.15)$$

$$d_{GG} = \sqrt{(5-5)^2 + (0-0)^2} = 0 \quad (3.16)$$

$$f(G) = g(G) + h'(G) = 6,43 + 3,60 + 0 = 10,03 \quad (3.17)$$

Seguimos recorriendo nodos, ahora es el turno de B, ya que ahora es el más bajo que hay.

$$d_{BG} = \sqrt{(5-4)^2 + (0-2)^2} = 2,24 \quad (3.18)$$

$$d_{GG} = \sqrt{(5-5)^2 + (0-0)^2} = 0 \quad (3.19)$$

$$f(G) = g(G) + h'(G) = 6,71 + 2,24 + 0 = 8,95 \quad (3.20)$$

Ahora el nodo más bajo es G y G es el destino, eso quiere decir que ya hemos encontrado el camino más corto. El resultado es:

A - B - G

Con un coste de 8.95.

Con este ejemplo queda demostrado la gran ventaja de este algoritmo, y es que si no hubiésemos utilizado $h'(n)$ habría que haber recorrido más nodos, y esto en un grafo de únicamente 7 puntos.

Cabe destacar que en la aplicación la distancia entre dos puntos no se calcula del mismo modo, ya que en la aplicación cada punto está definido por una latitud y una longitud. Para calcular la distancia más corta entre dos puntos cualesquiera de la superficie terrestre se ha calculado la distancia a lo largo del círculo máximo que los une. La figura 3.15 muestra un círculo máximo que contiene dos puntos de cualesquiera.

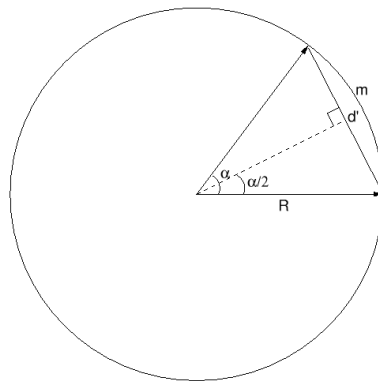


Figura 3.15: Distancia a lo largo de un círculo máximo

La distancia que nosotros necesitamos es la correspondiente al arco m que aparece en la figura 3.15. Esta distancia es:

$$m = 2R \arcsin \sqrt{\frac{(\cos \gamma_2 \cos \beta_2 - \cos \gamma_1 \cos \beta_1)^2 + (\cos \gamma_2 \sin \beta_2 - \cos \gamma_1 \sin \beta_1)^2 + (\sin \gamma_2 - \sin \beta_2)^2}{2}} \quad (3.21)$$

Dónde γ_1 es la latitud del punto 1, β_1 es la longitud del punto 1, γ_2 es la latitud del punto 2, β_2 es la longitud del punto 2 y R es el radio de la Tierra.

3.5. Ejemplos y comprobaciones

Para comprobar que la aplicación que se ha diseñado funciona correctamente y calcula la ruta más corta en todos los casos y no la calcula cuando no existe conexión entre los puntos se han llevado a cabo diferentes ejemplo.

Ejemplo 1: Vuelo entre dos puntos del Reino Unido (WIK - BPK)

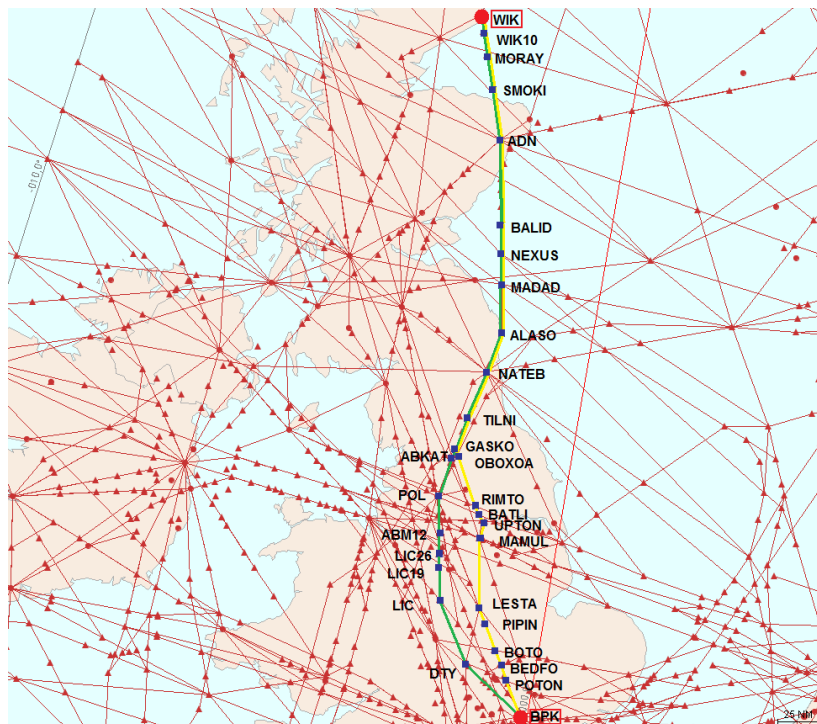


Figura 3.16: Ruta entre los puntos WIK y BPK

En la figura 3.16 podemos observar un camino verde que sirve para ir de WIK a BPK de la forma más corta, y otro amarillo que sirve para ir de BPK a WIK de la forma más corta también. La diferencia de rutas se debe a que no todas las aerovías son de doble sentido. En el primer caso se recorren 813.54 km y en el segundo 792.89 km, por lo que en este caso el cambio no es muy significativo. De igual manera el cambio en el tiempo de vuelo tampoco es significativo, ya que la primera ruta se tarda en recorrer 0.95 h y la segunda

0.93 h, todo esto, teniendo en cuenta que el vuelo se ha realizado a un mach de crucero de 0.80 y un flight level de 350.

Ejemplo 2: Vuelo entre dos puntos de Alemania (SUL - ABERU)



Figura 3.17: Ruta entre los puntos SUL y ABERU

En la figura 3.17 podemos observar una ruta de ida y vuelta casi completamente igual. Esto se debe a que las aerovías entre estos dos puntos son de doble sentido. El camino verde sirve para ir de SUL a ABERU y recorre una distancia de 333.58 km en un tiempo de 30 minutos, teniendo en cuenta que el vuelo se ha realizado a un mach de crucero de 0.80 y un flight level de 350. Y el camino amarillo sirve para ir de ABERU a SUL, que como se puede observar es mínimamente más corto, exactamente 11.63 km, lo que resulta insignificante en el tiempo de vuelo, ya que en este caso se tardan 29 minutos.

Ejemplo 3: Vuelo entre Groenlandia y Islandia (KU - ING)

En la figura 3.18 lo que se puede observar es que no existe ruta ninguna para poder viajar entre estos dos puntos, por lo tanto, la aplicación al introducir estos dos puntos nos ha dicho que no hay ruta.

Ejemplo 4: Vuelo entre el Reino Unido y Dinamarca (WAL - AAL)

En la figura 3.19 pasa completamente igual como en el ejemplo 2, es una ruta de ida y vuelta casi igual, la única diferencia que existe es debido a que la aerovía de LIBSO a UPTON es de un único sentido por lo que para ir de UPTON a LIBSO la aeronave tiene que hacer un pequeño desvío que le obliga a hacer 8 km más en la ruta de ida.

Ejemplo 5: Vuelo entre Tenerife y Toledo (TFN - TLD)

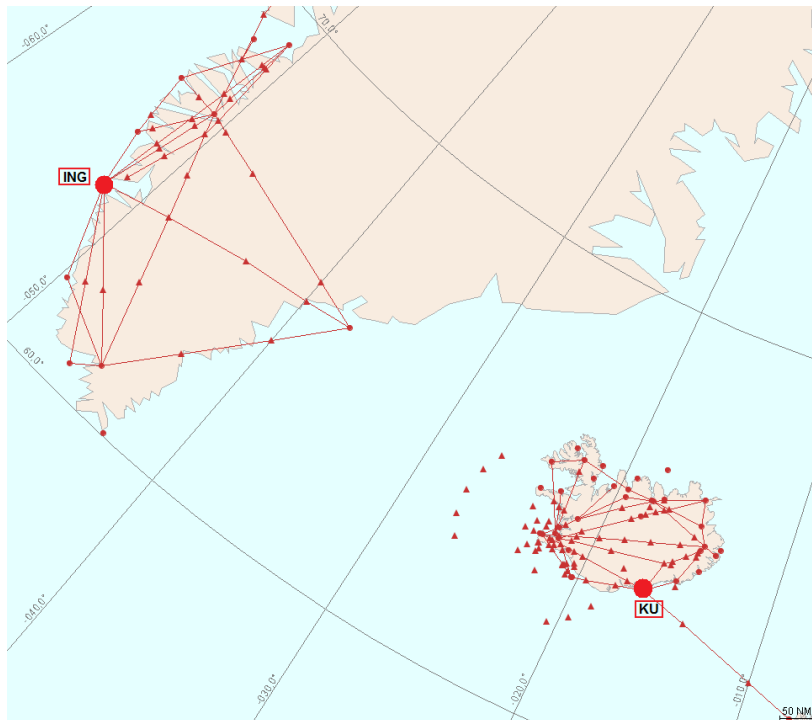


Figura 3.18: Ruta entre los puntos KU y ING

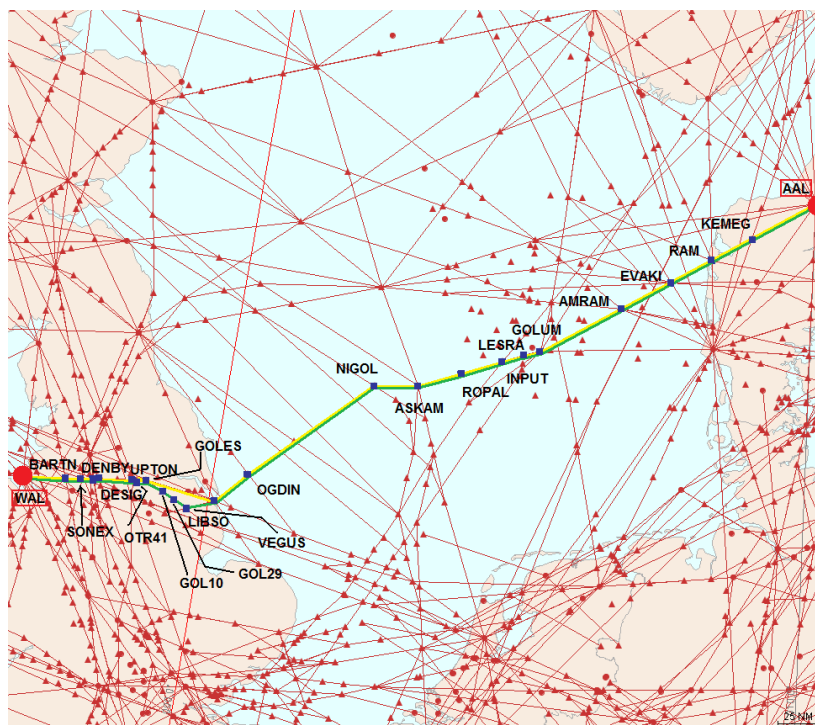


Figura 3.19: Ruta entre los puntos WAL y ALL

En la figura 3.20 vuelve a pasar lo mismo que en los ejemplos 2 y 4, se trata de una ruta casi completamente igual debido a que la mayoría de las aerovías son de doble sentido. En este caso en la ruta de Toledo a Tenerife, la amarilla, el aeronave esta recorriendo 8 km más que le suponen un incremento en el tiempo de vuelo de 1 minuto.

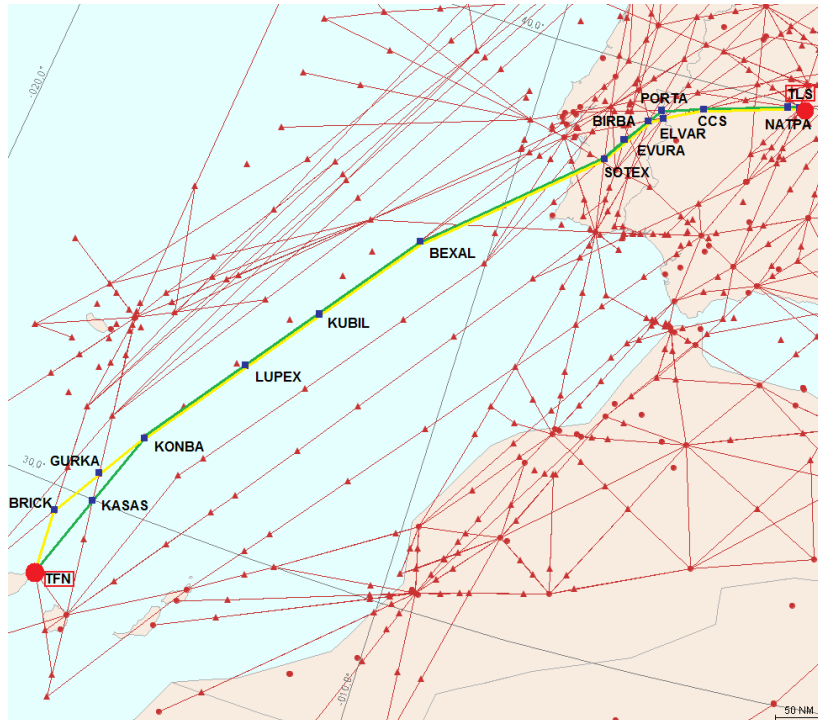


Figura 3.20: Ruta entre los puntos TFN y TLD

Ejemplo 6: Vuelo entre Bulgaria y Polonia (DINRO - ADOKI)



Figura 3.21: Ruta entre los puntos DINRO y ADOKI

En la figura 3.21 se puede observar uno de los ejemplos más interesantes, ya que las rutas solo se cruzan en un único punto, LUXAR. La ruta verde sirve para ir de DINRO

a ADOKI y se recorre una distancia de 1085.65 km y se tarda 1.62 h, en cambio por el camino amarillo que es la vuelta, se recorren 954.34 km y se tarda 1.43 h, teniendo en cuenta el valor del mach y el flight level de crucero que es igual que en los demás ejemplos.

Ejemplo 7: Vuelo entre Bulgaria y Rumanía (DINRO - ARD)



Figura 3.22: Ruta entre los puntos DINRO y ARD

En la figura 3.22 volvemos a tener un ejemplo en el que las rutas son muy diferentes, en este caso las rutas solo coinciden por tres aerovías. Con esto podemos deducir que alrededor del punto DINRO la mayoría de aerovías solo son de un sentido, ya que en el ejemplo anterior nos ha pasado lo mismo con este punto. En este caso la diferencia de distancia recorrida no es muy grande ya que las rutas son muy parecidas, en el caso del vuelo de ida se recorren 13 km más lo que le supone un minuto y medio más de vuelo.

Ejemplo 8: Vuelo entre dos puntos de las Islas Baleares (BLASO - MEBUT)

En la figura 3.23 es en la que se puede observar mayor diferencia entre las rutas. En la ruta verde, de BLASO a MEBUT se recorren 253.56 km y en la ruta amarilla de MEBUT a BLASO se recorren 427.16 km. La segunda ruta es un 68% más larga que la primera. Esto se debe que para salir de BLASO solo se puede salir por VERSO y por eso hay que dar esa vuelta. Hablando en tiempo, la primera ruta se recorre en 12 minutos menos que la segunda.

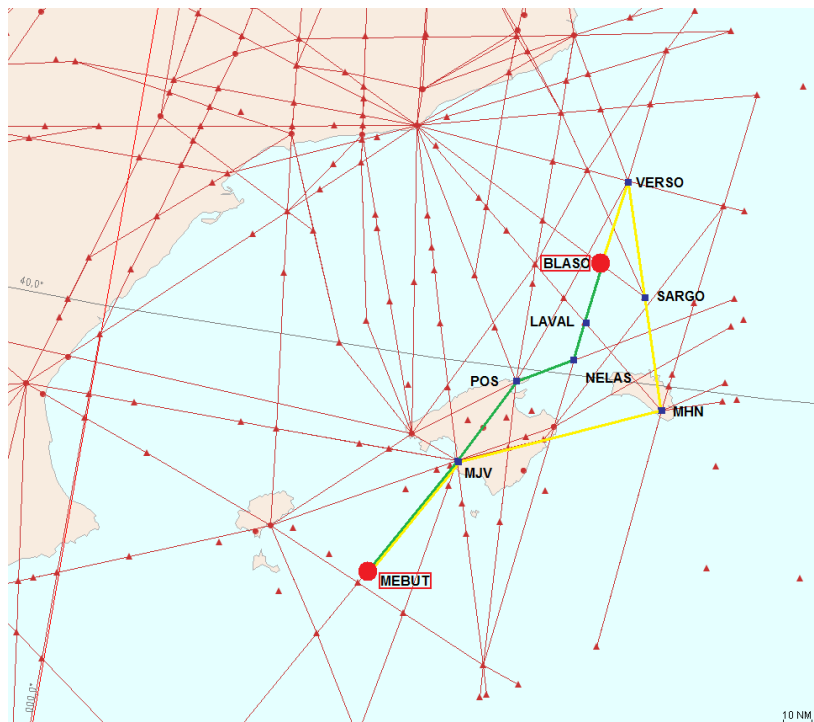


Figura 3.23: Ruta entre los puntos BLASO y MEBUT

3.6. Posibles ampliaciones y mejoras

La aplicación está preparada para poder añadirle diferentes mejoras y nuevas funcionalidades.

Las posibles mejoras son:

- **Equipo de navegación:** Por ahora la aplicación te permite elegir un equipo de navegación, RNAV o equipo convencional, pero no tiene ningún efecto a la hora de calcular la ruta. Lo que debería hacer es cuando el equipo sea RNAV basar la ruta únicamente en waypoints y cuando sea equipo convencional basarla en VOR, NDB y DME. Lo único que hay que hacer para que esto funcione es añadir una función en el archivo Grafo.h y Grafo.cpp que verifique al visitar un nuevo nodo que se esté cumpliendo la condición del equipo de navegación.
- **Flight level:** Por ahora la aplicación te permite elegir un flight level entre 250 y 460, y únicamente tiene efecto cuando calcula el tiempo de la ruta, ya que se necesita la temperatura para poder calcular la velocidad del sonido y para calcularla necesitamos la altitud. Pero el flight level puede tener más utilidades, y es que a la hora de calcular la ruta puede ser que algún punto del espacio aéreo europeo no esté útil para la altitud de crucero que hemos elegido, diferenciando entre rutas del espacio aéreo superior y rutas del espacio aéreo inferior. Lo único que hay que hacer para que esto funcione es añadir una función en el archivo Grafo.h y Grafo.cpp que verifique al visitar un nuevo nodo que esta útil para la altitud que hemos elegido.
- **Día y hora del vuelo:** Por ahora la aplicación te permite elegir el día y la hora a la

que se efectuara el vuelo, pero no tiene ningún efecto a la hora de calcular la ruta. Lo que debería hacer es si las rutas condicionales se pueden usar, porque en caso contrario no tendrían ningún efecto estos dos parámetros, restringir las rutas que estén dentro del día y hora de vuelo. Para ello lo único que hay que hacer es añadir una función en el archivo Grafo.h y Grafo.cpp que verifique al visitar un nuevo nodo que la ruta esta útil el día y a la hora que hemos elegido. Además de estas dos funciones hay que encontrar un archivo que contenga esta información para poder cargarla en el programa, porque en la base de datos actual esta información no está reflejada, ya que únicamente aparece si es ruta condicional o no, y en el caso de que sea su categoría.

- Restricciones de vuelo: Por ahora la aplicación permite elegir las restricciones de vuelo, tanto las referentes a las rutas condicionales como a las del ángulo de giro, pero ninguna de las dos tiene efecto a la hora de calcular la ruta. Para que estas restricciones funcionen, lo único que hay que hacer es añadir una función en Grafo.h y Grafo.cpp que verifique al visitar un nuevo nodo que cumple con las condiciones que se han establecido en las restricciones y si no se cumplen dicho nodo no se pueda utilizar.
- Uso de aeropuertos: Una mejora importante para el programa sería que se pudiera calcular una ruta entre dos aeropuertos. Por ahora el programa carga los aeropuertos europeos pero no deja efectuar un vuelo entre ellos ya que no tienen cargadas las sids y las stars, por lo tanto los aeropuertos no están conectados con el grafo formado por los waypoints, las nav aids y las aerovías. Lo que hay que hacer es buscar un archivo que contenga esta información para poder cargarla en la aplicación, porque la base de datos actual no contiene dicha información. Es decir, hay que unir los aeropuertos a waypoints o nav aids que estén conectados al grafo, simplemente con esto el programa ya calcularía la ruta entre un aeropuerto y cualquier otro punto del grafo, ya que el grafo está compuesto por points y aeropuerto hereda de la clase points.

En la figura 3.24 aparece un ejemplo de como se tendría que conectar un aeropuerto con el grafo. En verde más clarito aparecen las salidas y el verde más oscuro se refiere a las llegadas. Estos procedimientos han sido obtenidos a partir de la AIP. Por lo tanto lo único que habría que hacer es conectar estos puntos con el aeropuerto de Girona y ya estaría disponible. En la figura 3.25 se puede observar la carta de salidas de Girona y ver como coinciden las tres salidas con las de la imagen 3.24, y lo mismo con la carta de llegadas que se puede observar en la figura 3.26.

Las nuevas funcionalidades son:

- Añadir meteorología: Esta función también estaba pensada para que estuviese disponible, pero por cuestión de tiempo no se ha podido realizar. Es muy interesante ya que nos permitiría comprobar que no siempre la ruta más corta sería la que recorriese menos distancia, ya que añadiendo viento de cara la velocidad del avión se vería reducida y por lo tanto habría otra ruta en la cual no hubiese este viento y fuera más rápida. Para añadir este factor la idea es crear diferentes bases de datos, donde se indicaría por cada nivel de vuelo la dirección y velocidad del viento, y entonces cuando apretase el botón de añadir meteorología te dejara elegir

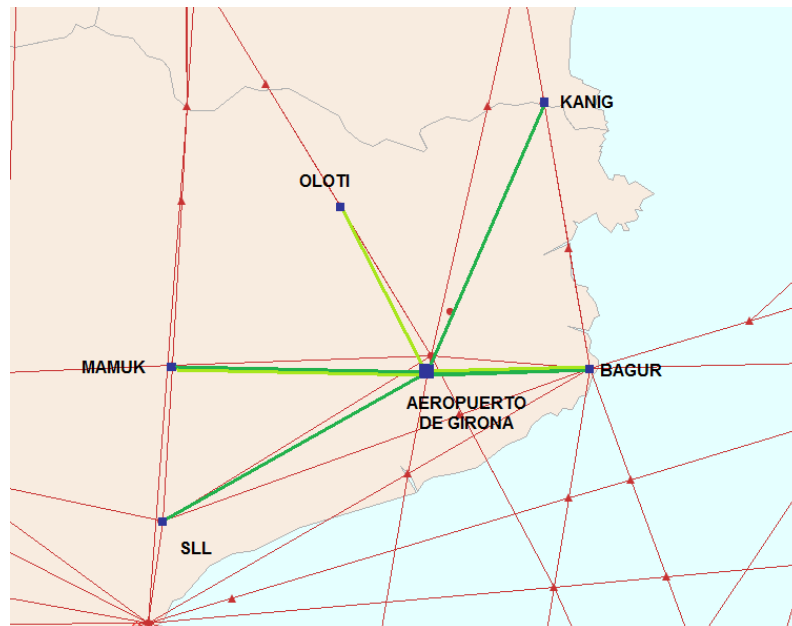


Figura 3.24: Procedimientos de salida y llegada del aeropuerto de Girona

entre uno de estos archivos y a la hora de calcular la ruta se tendría que tener en cuenta, de manera que las aristas se ponderarían teniendo en cuenta el efecto del viento.

- **Actualización:** Esta función también estaba pensada para que estuviese disponible, pero al igual que la anterior, no ha habido tiempo para poder realizarla. La idea de esta función es que al apretar este botón te pida una contraseña que solo el administrador conocerá y lo que te permitirá es cambiar la base de datos actual por una nueva, y así poder actualizar los datos de la aplicación.
- **Cambiar contraseña:** Esta función solo la podrá utilizar únicamente el administrador y lo que te permitirá es como indica su nombre, cambiar la contraseña para acceder a cambiar la base de datos actual.
- **Visualización:** Una nueva funcionalidad y muy importante sería poder añadirle zoom al grafo, ya que de esta manera al cargar un grafo tan extenso como el del espacio aéreo europeo se pueda apreciar como la aplicación calcula una ruta y cuál es su resultado. A parte del zoom, también sería importante tener una funcionalidad que centrará el grafo en la parte en la cual aparece la ruta.

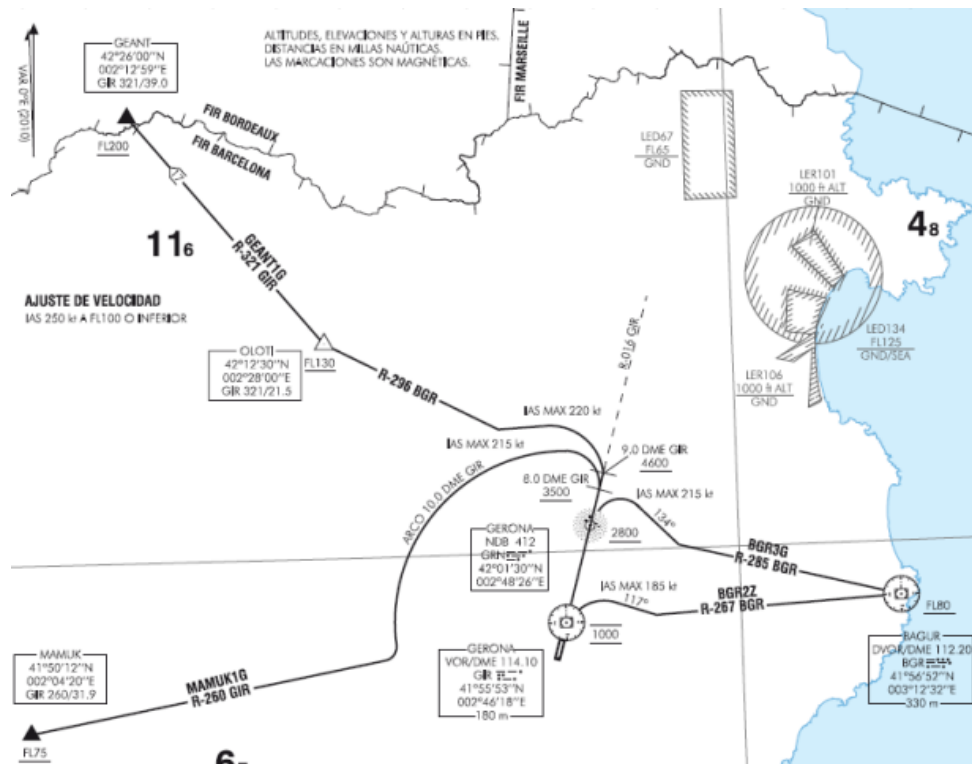


Figura 3.25: Procedimientos de salida del aeropuerto de Girona



Figura 3.26: Procedimientos de llegada del aeropuerto de Girona

CAPÍTULO 4. GESTIÓN DEL PROYECTO

En este apartado se va a realizar una comprobación de la planificación inicial del proyecto con el desarrollo que se ha llevado a cabo en realidad. En la figura 4.1 se puede observar la planificación inicial del proyecto y en la figura 4.2 el diagrama de Gantt.

		Nombre de tarea	Duration	Start	Finish
1		☐ Documentación	18 days	Mon 2/22/10	Thu 3/11/10
2	📄	Búsqueda de información	8 days	Mon 2/22/10	Mon 3/1/10
3		Creación Índice	1 day	Tue 3/2/10	Tue 3/2/10
4	📄	Tratamiento de información	9 days	Wed 3/3/10	Thu 3/11/10
5		☐ Especificación	15 days	Fri 3/12/10	Fri 3/26/10
6	📄	Funcionalidad del programa	2 days	Fri 3/12/10	Sat 3/13/10
7	📄	Especificaciones técnicas	7 days	Sun 3/14/10	Sat 3/20/10
8		Creación test de validación	6 days	Sun 3/21/10	Fri 3/26/10
9		☐ Diseño	23 days	Sat 3/27/10	Sun 4/18/10
10	📄	Arquitectura del sistema	7 days	Sat 3/27/10	Fri 4/2/10
11	📄	Diseño algoritmo de Dijkstra	10 days	Sat 4/3/10	Mon 4/12/10
12		Creación test de integración	6 days	Tue 4/13/10	Sun 4/18/10
13		☐ Programación	47 days	Mon 4/19/10	Fri 6/4/10
14		Programación algoritmo de Dijkstra	35 days	Mon 4/19/10	Sun 5/23/10
15		Programación funcionalidades extra	12 days	Mon 5/24/10	Fri 6/4/10
16		Programación HMI	25 days	Tue 5/11/10	Fri 6/4/10
17		Creación test de verificación	6 days	Mon 5/24/10	Sat 5/29/10
18		☐ Validación y Verificación	50 days	Mon 4/19/10	Mon 6/7/10
19	📄	Test de verificación	50 days	Mon 4/19/10	Mon 6/7/10
20	📄	Test de validación	50 days	Mon 4/19/10	Mon 6/7/10
21		Conclusiones	5 days	Tue 6/8/10	Sat 6/12/10
22	📄	Redacción Memoria	109 days	Wed 3/3/10	Sat 6/19/10
23		Memoria	0 days	Sat 6/19/10	Sat 6/19/10
24		Preparación presentación	10 days	Sun 6/20/10	Tue 6/29/10
25		Presentación	0 days	Tue 6/29/10	Tue 6/29/10

Figura 4.1: Planificación inicial del proyecto

Como se puede observar, primeramente el proyecto estaba planificado para realizarlo en 4 meses y medio, y finalmente se han convertido en 11 meses, esta gran diferencia es debida únicamente a que a partir del mes de Abril empecé a trabajar en el aeropuerto de Barcelona, lo que me suponía quedarme sin tiempo, por la mañana trabajaba y por las tardes estudiaba. A partir de septiembre que ya no tengo clases le he podido dedicar más tiempo al proyecto para así poder finalizarlo. A parte de la diferencia del tiempo, también existe otra gran diferencia y es la proporción de tiempo que me ha llevado cada tarea, ya que la especificación y la programación me han llevado más tiempo del previsto.

Para acabar con este apartado, se va a realizar un estudio del coste del proyecto si este hubiese sido encargado por una empresa.

En este proyecto se han invertido alrededor de unas 500 horas y teniendo en cuenta que el precio de la hora es de 40 Euros, este proyecto hubiera tenido un coste de 20.000 Euros.

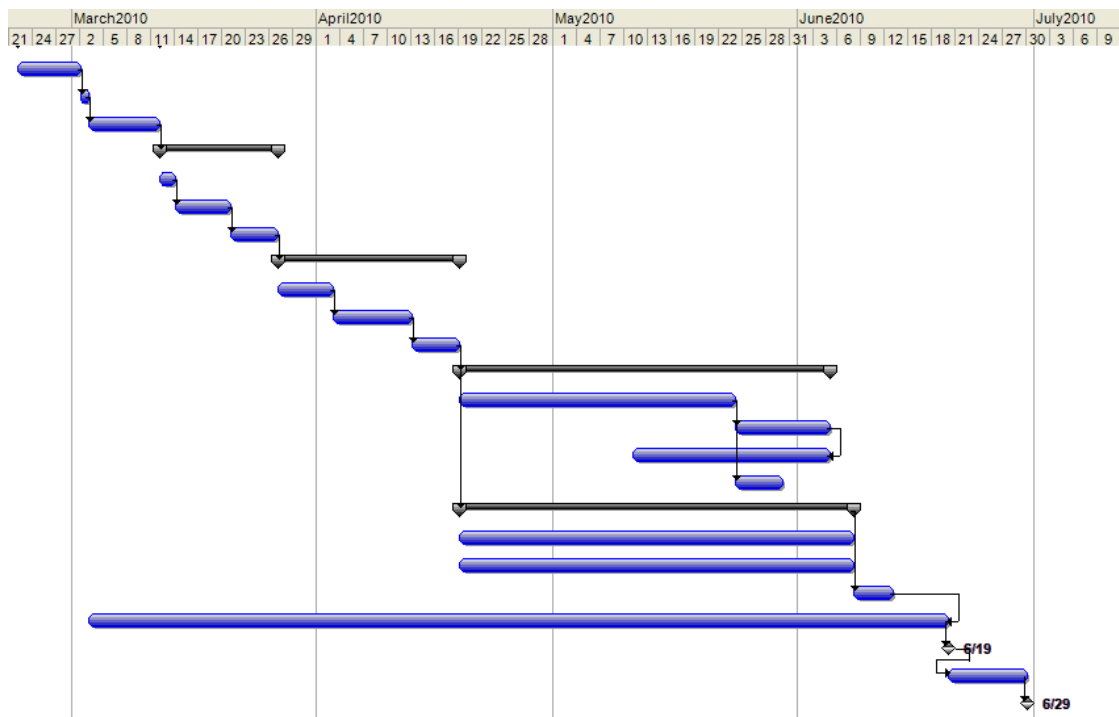


Figura 4.2: Diagrama de Gantt del proyecto

CAPÍTULO 5. CONCLUSIONES

El objetivo inicial de este trabajo era crear una aplicación que calculara la ruta más corta entre dos puntos del espacio aéreo europeo y así ha sido, se ha desarrollado correctamente dicha aplicación, que además de la ruta calcula otros parámetros.

Para crear dicha aplicación se ha seguido un ciclo de vida de ingeniería de software, lo cual ha permitido llevar el desarrollo de la aplicación de una manera ordenada.

La aplicación tiene un diseño modular y está estructurada en diferentes capas, lo cual permite una mayor comprensión del código para cualquier persona que quiera modificarlo, ya que la aplicación también está preparada para añadir nuevas funcionalidades que no se han podido añadir en este proyecto.

A nivel personal este proyecto me ha hecho aprender cómo funciona realmente el ciclo de vida de un proyecto y además también he ampliado mi conocimiento sobre algunos aspectos de programación.

BIBLIOGRAFÍA

- [1] Anibal Isidoro Carmona. *Aerodinámica y actuaciones del avión*. Paraninfo, 2007.
- [2] <http://nacc.upc.es/navegacion-aerea/x52.html>, Mayo 2010.
- [3] AENA. Rutas ats. Technical report, AIP España, 2007.
- [4] OACI. *Manual de Planificación de Servicios de tránsito aéreo*, 1995.
- [5] http://www.aena.es/csee/Flash/html/controlAereo02_04.jsp, Abril 2010.
- [6] Organización de Aviación Civil Internacional. Definiciones de la fase de vulo y notas de uso. Technical report, Equipo de Seguridad de la Aviación Comercial, 2002.
- [7] Luis Delgado. Apuntes transporte aéreo. Technical report, EPSC, 2007.
- [8] Pablo Hernández. Rutas aéreas. Technical report, Miscelanea Navegación, 2008.
- [9] Navegación vor. Technical report, -, 2010.
- [10] <http://nacc.upc.es/rnav/c31.html>, Mayo 2010.
- [11] http://www.seguridadaerea.es/AESA/LANG_CASTELLANO/NAVEGACION/PROGRAMAS/NAVEGACION_AREA/, Junio 2010.
- [12] http://www.seguridadaerea.es/AESA/LANG_CASTELLANO/NAVEGACION/PROGRAMAS/NAVEGACION_AREA/conceptos.htm, Junio 2010.
- [13] AENA. Zonas prohibidas, estringidas y peligrosas; Áreas temporalmente segregadas. Technical report, AIP España, 2002.
- [14] AENA. Uso flexible del espacio aéreo. Technical report, AIP España, 2002.
- [15] Joaquín C. Adsuar. *Meteorología*. Paraninfo, 2002.
- [16] http://es.wikipedia.org/wiki/Programaci%C3%B3n_por_capas, Enero 2011.
- [17] Juan P. Maltés Granados Ricardo J. Vargas del Valle. Ingeniería del software. Technical report, Universidad de Costa Rica, 2005.
- [18] http://es.wikipedia.org/wiki/Algoritmo_de_b%C3%BAsqueda_A*, Enero 2011.
- [19] Vicente J. Ferrer Dalmau. El problema del viajante del comercio. inteligencia artificial. Technical report, -, 2008.
- [20] José L. Ruiz Reina José Antonio Alonso Jiménez, Francisco Jesús Martín Mateo. Técnicas heurísticas de búsqueda. Technical report, Universidad de Sevilla, 2010.
- [21] <http://poiritem.wordpress.com/2010/01/14/6-5-2-busqueda-heuristica-algoritmo-a/>, Enero 2011.



Escola Politècnica Superior
de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

APÈNDIXS

TÍTOL DEL TFC : Algoritmo de càlculo de rutes òptimas para aeronaves en el plano horizontal

TITULACIÓ: Enginyeria Tècnica Aeronàutica, especialitat Aeronavegació

AUTOR: Ismael González Rodríguez

DIRECTOR: Luis Delgado

DATA: 24 de enero de 2011