



# Universidad Rey Juan Carlos

Escuela Técnica Superior Ingeniería  
Informática

XXXXXX XXXX

**TFG**

GIS - Carlos Vázquez Sánchez

Móstoles - 4 de marzo de 2017

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Descripción del problema</b>	<b>4</b>
<b>3. Versiones anteriores del problema y nuevos objetivos</b>	<b>5</b>
3.1. Versión anterior . . . . .	5
3.2. Nuevos objetivos . . . . .	5
<b>4. Tecnologías utilizadas</b>	<b>7</b>
4.1. C++ . . . . .	7
4.2. MySQL . . . . .	7
4.3. Scripting . . . . .	7
4.4. HTML y JavaScript . . . . .	7
4.5. Git . . . . .	7
<b>5. Heurístico</b>	<b>8</b>
<b>6. Resultados experimentales</b>	<b>9</b>
<b>7. Conclusiones</b>	<b>10</b>
<b>8. Bibliografía</b>	<b>11</b>

## 1. Introducción

## **2. Descripción del problema**

### 3. Versiones anteriores del problema y nuevos objetivos

#### 3.1. Versión anterior

Este Trabajo Final de Grado es la continuación del trabajo que llevaron a cabo Diego Ruiz Aguado y Gonzalo Quevedo García en 2012 en sus Proyectos Finales de Carrera, los cuales se apoyaron a su vez en la Tesis Doctoral de Alba Agustín Martín.

A continuación se da una breve descripción del trabajo de Diego Ruiz Aguado y Gonzalo Quevedo García:

1. Se mejoró la BBDD que contenía toda la información del problema, pasando de un modelo no relacional y con redundancias a uno relacional y bien estructurado.
2. Para obtener los datos que necesitaba el problema, se realizó un programa en JAVA que se conectaba a la BBDD y creaba varios ficheros .txt en la que se volcaba toda la información necesaria para el posterior modelado del problema.
3. A continuación, se leían estos ficheros .txt y se creaban las estructuras de datos necesarias (árbol de rutas, vuelos, waypoints, etc).
4. Posteriormente una subrutina en C se encargaba de definir un problema de CIPLEX con la función objetivo y las restricciones necesarias.
5. Finalmente se obtenía la mejor solución del problema.

#### 3.2. Nuevos objetivos

La versión anterior del problema adolecía de un importante inconveniente: no podía salir de los máximos locales, ya que el heurístico que utilizaba para lanzar los vuelos era un algoritmo voraz. Por tanto los objetivos marcados para este TFG han sido los siguientes (ordenados en decreciente prioridad):

1. **Mejorar heurístico:** utilizar heurísticos más elaborados que mejoren la solución del problema.
2. **Desacoplar el programa de CIPLEX:** con la implementación de los nuevos heurísticos no es necesaria la librería de optimización. Se pasará de un sistema clásico de optimización (función objetivo y restricciones) a una estructura de objetos que permitan un manejo óptimo de las estructuras de datos durante el heurístico.
3. **Mejorar el sistema de lectura de datos:** el sistema actual crea ficheros .txt que pueden superar las 100.000 líneas. Hay que mejorar este sistema.

4. **Representación gráfica:** aunque no es estrictamente necesaria, si se dispusiera de tiempo suficiente se añadiría una representación gráfica de la solución del problema.

## 4. Tecnologías utilizadas

Las tecnologías utilizadas han sido las siguientes:

### 4.1. C++

Aunque se comenzó utilizando C en las primeras versiones del TFG, finalmente se optó por utilizar C++. Esto fue debido a que el cambio a C++ permitió mantener el trabajo ya realizado en C, y posee estructuras de datos ya implementadas como mapas o vectores que reducen en gran medida el tiempo de programación, además de permitir la orientación a objetos.

### 4.2. MySQL

Al igual que en la versión anterior del proyecto, la BBDD que usamos será la relacional que realizó Diego Ruiz Aguado en el 2012.

### 4.3. Scripting

Se crearon pequeños scripts que se encargan de importar la BBDD, exportar las tablas en el formato necesario y compilar el proyecto

### 4.4. HTML y JavaScript

Para la representación gráfica del problema se ha utilizado JS para leer y parsear la información almacenada en un fichero y HTML para su visualización (para la creación del grafo se ha utilizado la librería [vis.js](#)).

### 4.5. Git

Se ha utilizado Git como sistema de versiones, y todo el código se puede descargar y consultar en [Github](#)

## 5. Heurístico

El algoritmo que se ha creado para el problema está inspirado en el metaheurístico constructivo *GRASP* (*Greedy Randomized Adaptive Search Procedure*) se resume en el siguiente diagrama:

->IMAGEN DEL ALGORITMO EN UML<-

Los pasos son:

1. **Se intentan colocar vuelos con su mejor solución:** se lanzan todos los vuelos de manera aleatoria sin permitir retrasos o desvíos, solo la solución inicial.
2. **Intercambio de vuelos:** se intenta sustituir uno de los vuelos exitosos del paso anterior por 2 o más vuelos aleatorios a los que no se les halló solución. Para ello:
3. **Se intentan colocar vuelos permitiendo retrasos:** se lanzan los vuelos que aun no tienen solución, permitiendo retrasos en sus rutas, pero no desvíos.
4. **Se intentan colocar vuelos permitiendo retrasos y desvíos:** se lanzan los vuelos que aun no tienen solución, permitiendo retrasos y desvíos en sus rutas.
5. **Se intentan colocar vuelos permitiendo retrasos:** se lanzan los vuelos que aun no tienen solución, permitiendo retrasos en sus rutas, pero no desvíos. En este punto si un vuelo tiene alguna solución factible, se le asignará.
6. **Se buscan los waypoints sin usar y se les asigna una ruta:** se localizan los waypoints por los que no pasa ningún vuelo a lo largo de todo el problema. Si algún vuelo tiene alguna solución que utilice alguno de estos waypoints, y es factible, se la asigna
7. **Retrasar vuelos con solución para colocar 2 o más cancelados:** se intenta retrasar alguno de los vuelos con solución factible para poder encontrar de forma aleatoria uno o más vuelos que estaban cancelados.

Este esquema se repite el número de iteraciones indicadas. para que no sea un proceso totalmente aleatorio, se han añadido 2 técnicas utilizadas en los algoritmos metaheurísticos: La *búsqueda tabú* y el *arranque multistart*



## **6. Resultados experimentales**

A continuación se detallan los resultados obtenidos con distintos parámetros del problema

## 7. Conclusiones

## 8. Bibliografia