

# EXPLICACIÓ I JUSTIFICACIÓ DE LES ESTRUCTURES DE DADES UTILITZADES

## Aeroports:

```
map<int, Airport> m_int_airports;  
map<string, Airport*> m_str_airports_IATA;  
map<string, Airport*> m_str_airports_ICAO;
```

Utilitzo mapes ja que necessitaré poder fer moltes cerques d'Aeroports per codi (ja sigui OpenFlightsID, IATA o ICAO) de manera ordenada i eficient.

Faig servir un mapa principal amb clau OpenFlights ID i de valor un objecte Aeroport.

A més a més utilitzo dos mapes secundaris que només guarden de valor un punter a un objecte Aeroport de l'estructura principal. Tots dos fan servir un string com a clau.

El primer mapa secundari fa servir codis IATA per la clau.

El segon mapa secundari fa servir codis ICAO per la clau.

## Aerolínies:

```
map<int, Airline> m_airlines;  
map<string, Airline*> m_str_airlines_IATA;
```

Utilitzo mapes ja que necessitaré poder fer moltes cerques d'Aerolínies per codi (ja sigui OpenFlightsID, IATA) de manera ordenada i eficient.

Faig servir un mapa principal amb clau OpenFlights ID i de valor un objecte Aerolínia.

A més a més utilitzo un mapa secundari que només guarda de valor un punter a un objecte Aerolínia de l'estructura principal. Fa servir un string com a clau, que serà un codi IATA.

## Avions:

```
map<string, Plane> m_planes;  
set<string> s_repetits;
```

Utilitzo un mapa ja que necessitaré poder fer moltes cerques d'Avions per codi IATA, de manera ordenada i eficient.

A més a més utilitzo un set com a estructura auxiliar per guardar-hi avions repetits. Aquesta estructura serveix només durant la lectura del fitxer d'avions i funciona com a complement de l'estructura principal, de manera que ajuda a filtrar tots aquells avions el codi IATA dels quals aparegui més d'un cop al fitxer. El set és esborrat de memòria un cop acabada la lectura del fitxer.

## Rutes:

```
list<Route> l_routes;
```

Utilitzo una llista ja que només necessitaré accedir-hi per fer recorreguts sencers i una llista serà molt ràpida a l'hora de guardar les dades llegides de fitxer ja que no necessito mantenir-les ordenades.

## Vols:

```
multimap<string, Flight> mm_flights_DATE; // DATE IS KEY WITH FORMAT [YYYYMMDD]
multimap<string, Flight*> mm_flights_AL; // AIRLINE'S IATA IS KEY
multimap<string, Flight*> mm_flights_AP; // AIRPORT'S IATA IS KEY
```

Utilitzo multimapes ja que necessitaré poder fer moltes cerques de Vols per codi (ja sigui per data, codi IATA d'Aerolínia o codi IATA d'Aeroport) de manera ordenada i eficient, i al mateix temps m'ha de permetre tenir diferents Vols guardats que tinguin la mateixa clau.

Faig servir un mapa principal amb clau DATA i de valor un objecte Flight.

A més a més utilitzo dos multimapes secundaris que només guarden de valor un punter a un objecte Flight de l'estructura principal.

El primer multimapa secundari fa servir codis IATA de l'Aerolinia del Vol per la clau.

El segon multimapa secundari fa servir codis IATA dels Aeroports del Vol per la clau.