

Rodrigo Rodrigues - up202108749

Luís Relvas - up202108661

João Sequeira - 202108823

Grupo : G45

# Trabalho prático 2: Transportes Aéreos



# ÍNDICE:

- DIAGRAMA DE CLASSES/SECÇÕES: FICHEIROS ENVOLVIDOS
- LEITURA DE DADOS/CLASS GESTOR
- GRAFO “AUX”
- GRAFO “AUX”-ILUSTRAÇÃO
- EXEMPLOS DE INTERATIVIDADE COM O UTILIZADOR
- DESTAQUE DE FUNCIONALIDADES
- PRINCIPAIS DIFICULDADES



# DIAGRAMA DE CLASSES/SECÇÕES: FICHEIROS ENVOLVIDOS

Dados fornecidos:

- **Ficheiro aeroportos.csv (aeroportos contidos no ficheiro)**

```
Code,Name,City,Country,Latitude,Longitude
CDG,Charles De Gaulle,Paris,France,49.012779,2.550000
ORY,Orly,Paris,France,48.725278,2.359444
LGA,La Guardia,New York,United States,40.777245,-73.872608
JFK,John F Kennedy Intl,New York,United States,40.639751,-73.778925
LHR,Heathrow,London,United Kingdom,51.477500,-0.461389
```

- **Ficheiro Airlines.csv (companhias aéreas contidas no ficheiro)**

```
Code,Name,City,Country,Latitude,Longitude
CDG,Charles De Gaulle,Paris,France,49.012779,2.550000
ORY,Orly,Paris,France,48.725278,2.359444
LGA,La Guardia,New York,United States,40.777245,-73.872608
JFK,John F Kennedy Intl,New York,United States,40.639751,-73.778925
LHR,Heathrow,London,United Kingdom,51.477500,-0.461389
```

- **Ficheiro flights.csv (voos contidos no ficheiro)**

```
Source,Target,Airline
CDG,JFK,AAL
JFK,CDG,AAL
ORY,LHR,IBE
ORY,LHR,BAW
OPO,LIS,TAP
```

Tarefas a implementar:

- implementar um sistema capaz de providenciar ajuda para quem quer usar a rede de voos das companhias aéreas de todo o mundo
- Dar informações sobre os aeroportos
- Pesquisa em largura (BFS)
- Tabelas de Hash



# LEITURA DE DADOS/ CLASS GESTOR



`gestor_airports`  
Esta função serve para ler o ficheiro  
“airports.csv” e guarda a informação em 6  
`unordered_maps` distintos

`gestor_flights`  
Esta função serve para ler o ficheiro “flights.csv” e  
guarda a informação criando um grafo

`gestor_airlines`  
Esta função serve para ler o ficheiro “Airlines.csv”  
e guarda a informação num `unordered_set`

# GRAFO “AUX”

Criou-se um **grafo** onde os “nodes” representam **aeroportos** e os “edges” um **set de “Airlines”**

```
void Gestor::gestor_flights(Graph &aux) {  
  
    ifstream flightsf( s: "../data/flights.csv");  
    string line;  
    getline( & flightsf, & line);  
    while (getline( & flightsf, & line)) {  
        istringstream iss( str: line);  
        string airport_from_code, airport_to_code, airline_code;  
  
        getline( & iss, & airport_from_code, delim: ',');  
        getline( & iss, & airport_to_code, delim: ',');  
        getline( & iss, & airline_code);  
  
        Airport airport_from = airports.at( k: airports_codes.at( k: airport_from_code));  
        Airport airport_to = airports.at( k: airports_codes.at( k: airport_to_code));  
  
        int dist = (int) distancia( lat1: airport_from.getLatitude(), lon1: airport_from.getLongitude(),  
                                   lat2: airport_to.getLatitude(), lon2: airport_to.getLongitude());  
  
        aux.addEdge( src: airports_codes.at( k: airport_from_code), dest: airports_codes.at( k: airport_to_code),
```

```
class Graph {  
public:  
    struct Edge {  
        int dest; // Destination node  
        int weight; // An integer weight  
        set<string> airlines;  
    };  
  
    struct Node {  
        list<Edge> adj; // The list of outgoing edges (to adjacent nodes)  
        bool visited; // As the node been visited on a search?  
        int distance;  
        bool in_stack;  
        int num;  
        int low;  
    };  
  
    int n; // Graph size (vertices are numbered from 1 to n)  
    vector<Node> nodes;
```

# GRAFO “AUX”- ILUSTRAÇÃO





# EXEMPLOS DE INTERATIVIDADE COM O UTILIZADOR

1

```
*****
**----- INFO -----**
** Escolha uma opcao:          **
**-----**
**                               **
** 1 - De um Aeroporto         **
** 2 - De uma Cidade           **
** 3 - De um Pais              **
** 4 - De um raio de X Km de uma coordenada **
**                               **
** 0 - Sair                    **
**                               **
*****

3
3
Pais:Portugal
Portugal
```

2

```
*****
**----- INFO -----**
** Escolha uma opcao:          **
**-----**
**                               **
** 1 - De um Aeroporto         **
** 2 - De uma Cidade           **
** 3 - De um Pais              **
** 4 - De um raio de X Km de uma coordenada **
**                               **
** 0 - Sair                    **
**                               **
*****

1
1
Aeroporto:CDG
CDG
```

3

```
*****
**----- PESQUISA -----**
** Escolha uma opcao:          **
**-----**
**                               **
** 1 - Rota com menos escalas  **
** 2 - Quantos aeroportos sao possiveis alcançar? **
**                               **
** 0 - Sair                    **
**                               **
*****

1
1
```

```
*****
**----- PESQUISA -----**
** Origem do Voo:              (Menos Escalas) **
**-----**
**                               **
** 1 - Um Aeroporto            **
** 2 - Uma Cidade              **
** 3 - Um Pais                 **
** 4 - 0 ponto mais proximo de umas coordenadas **
** 5 - Todos os pontos num raio de X Km          **
** 0 - Sair                    **
**                               **
*****

2
2
Cidade:Paris
```

```
2942 - NEW ZEALAND
2943 - NEW ZEALAND
2944 - NEW ZEALAND
2945 - NEW ZEALAND
2946 - NEW ZEALAND
2947 - NEW ZEALAND
2948 - NEW ZEALAND
2949 - NEW ZEALAND
2950 - NEW ZEALAND
2951 - NEW ZEALAND
2952 - NEW ZEALAND
2953 - NEW ZEALAND
2954 - NEW ZEALAND
2955 - NEW ZEALAND
2956 - BAHRAIN

=====
Opcao:2956
```

# DESTAQUE DE FUNCIONALIDADES

O que mais nos orgulhou no projeto foi o facto de sermos capazes de responder a vários pedidos do utilizador bastante úteis para nós.

Um exemplo é quando o utilizador precisa de saber quantos e quais aeroportos existem numa certa latitude/longitude com um raio de x km de distância.

```
=====
|----- INFO -----|
|
| Latitude: 41.2481
| Longitude: -8.68139
| Raio: 500
|
|-----|
|
| Numero de aeroportos: 15
|
| Aeroporto 1: Vigo - VGO
| Aeroporto 2: Porto - OPO
| Aeroporto 3: Santander - SDR
| Aeroporto 4: Valladolid - VLL
| Aeroporto 5: A Coruna - LCG
| Aeroporto 6: Lisboa - LIS
| Aeroporto 7: Faro - FAO
| Aeroporto 8: Asturias - OVD
| Aeroporto 9: Sevilla - SVQ
|
```

```
=====
|----- INFO -----|
|
| Latitude: 47.485
| Longitude: 9.56077
| Raio: 200
|
|-----|
|
| Numero de aeroportos: 15
|
| Aeroporto 1: Marktoberdorf BF - OAL
| Aeroporto 2: St Gallen Altenrhein - ACH
| Aeroporto 3: Bolzano - BZO
| Aeroporto 4: Stuttgart - STR
| Aeroporto 5: Franz Josef Strauss - MUC
| Aeroporto 6: Bale Mulhouse - MLH
| Aeroporto 7: Zurich - ZRH
| Aeroporto 8: Augsburg - AGB
| Aeroporto 9: Entzheim - SXB
| Aeroporto 10: Innsbruck - INN
| Aeroporto 11: Allgau - FMM
|
```

No inicio o grupo teve dificuldade em achar os pontos de articulação. Mas com um grande esforço e dedicação conseguimos chegar a uma solução certa.

```
=====
|----- ESTATISTICAS -----|
|
| Pontos de Articulacao
|
|-----|
|
| Existem 309 pontos de articulacao
|
| Aeroportos com pontos de articulacao:
| [Orly] -> ORY, France
| [Port Moresby Jacksons Intl] -> POM, Papua New Guinea
| [Narsarsuaq] -> UAK, Greenland
| [Reykjavik] -> RKV, Iceland
| [Kugluktuk] -> YCO, Canada
| [Iqaluit] -> YFB, Canada
| [Gaspe] -> YGP, Canada
| [Natashquan] -> YNA, Canada
| [Quebec Jean Lesage Intl] -> YQB, Canada
| [Resolute Bay] -> YRB, Canada
| [Pierre Elliott Trudeau Intl] -> YUL, Canada
| [Norman Wells] -> YVQ, Canada
|
```

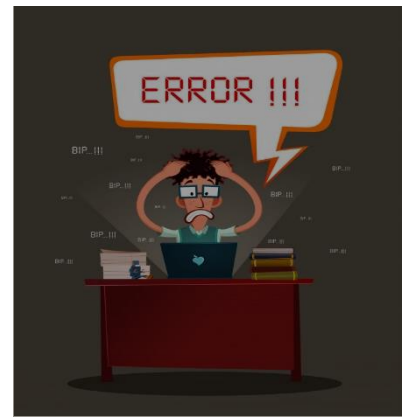
```
[Venetie Airport] -> VEE, United States
[Toowoomba] -> TWB, Australia
[Craig Seaplane Base] -> CGA, United States
[Quinhagak Airport] -> KWN, United States
[Tulaksak Airport] -> TLT, United States
[Postville Airport] -> YSO, Canada
[Dangriga Airport] -> DGA, Belize
[Placencia Airport] -> PLJ, Belize
[Jijiga Airport] -> JIJ, Ethiopia
[Friday Harbor Seaplane Base] -> FBS, United States
[Arctic Bay Airport] -> YAB, Canada
[Redacao Airport] -> RDC, Brazil
[Sigiriya Airport] -> GIU, Sri Lanka

|-----|
|
| Pressione ENTER para continuar...
```



# PRINCIPAIS DIFICULDADES

- Fazer o passo “Pontos de articulação”
- Concluir por completo o doxygen
- Escolher a melhor maneira de criar o grafo com os dados fornecidos
- Desenvolver alguns tópicos avançados com base no conhecimento que temos



## DIVISÃO DO PROJETO POR TODOS OS INTEGRANTES

Ambos os alunos do grupo participaram de uma maneira dividamente semelhante na realização do projeto. Todos os membros ajudaram na construção de todas as classes usadas no projeto. Enquanto, depois de finalizado a parte do código no projeto, o Luís Relvas ficou responsável por testar o código e resolver todos os problemas que apareciam no Terminal. Já o João Sequeira fez a parte do projeto onde incluía trabalhar com o Doxygen. O Rodrigo Rodrigues ficou encarregue da apresentação do trabalho no formato de PDF.

