

# Gestão de Informação de uma Companhia Aérea

## Algoritmos e Estruturas de Dados

### Alunos - G82

João Teixeira - up202005437

Lucas Sousa - up202004682

Rui Soares - up202103631

### Orientadores

Pedro Manuel Pinto Ribeiro

Ana Paula Cunha da Rocha

Sofia Cardoso Martins



# Temas Abordados

- Descrição do **Problema**;
- Descrição da **Solução**;
- **Classes** Utilizadas e suas **Relações**;
- Estrutura de **Ficheiros**;
- **Funcionalidades**;
- **Funcionalidade** em **Destaque**;
- **Dificuldades** Encontradas e **Contributos**;

# Descrição do Problema

- Uma **Companhia Aérea** precisa de um sistema de gestão de informação relativa a **aviões**, **voos**, **passageiros**, **bagagens** e **aeroportos**.
- **Aviões** caracterizam-se por matrícula, capacidade, lista de voos que realiza e serviços (Manutenção/Limpeza) a que estão sujeitos.
- **Voos** caracterizam-se por data de partida, duração de voo, origem e destino.
- Os **Passageiros** são caracterizados pelos bilhetes.
- No **Check-in**, as bagagens dos passageiros são transportadas por um carrinho de transporte, com dadas carruagens e dadas pilhas de uma certa capacidade.
- Os **Aeroportos** caracterizam-se pelo seu nome, local e pelos transportes disponíveis.

# Descrição da Solução

O primeiro passo foi dividir o problema nas classes que achávamos necessárias, as quais estão no diapositivo seguinte. Implementamos as devidas estruturas de dados para cada uma das classes, por exemplo, a BST para os transportes, a stack no carrinho de bagagem e as listas em vários pontos do código.

Criamos uma classe (BaseDados) que guarda todos os objetos e implementa as várias funções de load e save, para obter uma melhor organização do código. Esta classe contém os métodos relacionados com a interação do menu.

Após isso, preparamos a solução para conseguir ler os seus dados a partir de ficheiros e criamos um menu que permitia aceder às operações CRUD. Também implementamos listagens totais e parciais, com diferentes opções de ordenação.

Criamos também a documentação com recurso a Doxygen.

# Descrição da Solução

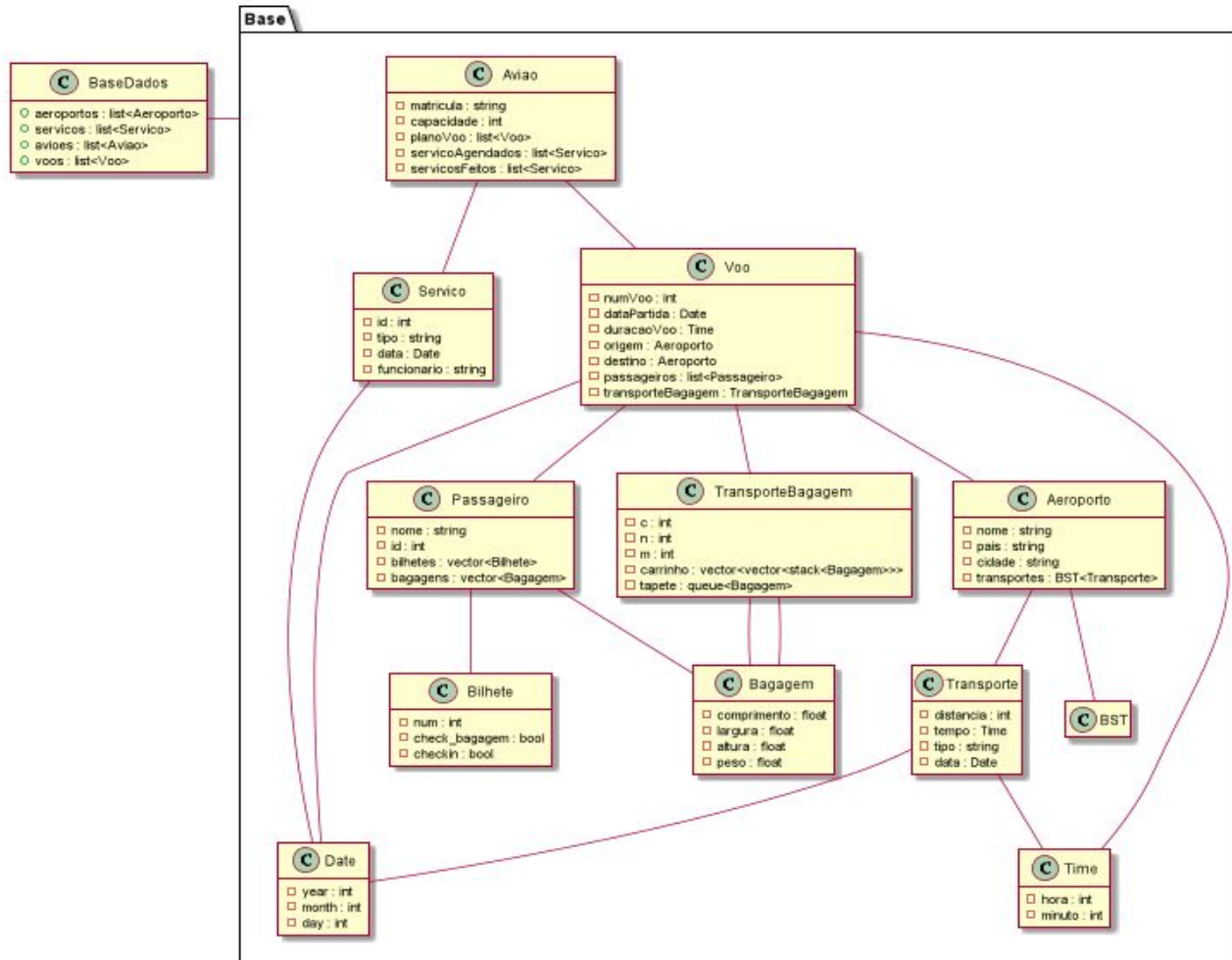
Foi implementado operador< e operador== em muitas das classes, de forma a utilizar o `std::sort()`, (Quicksort + Heapsort) + Insertionsort. Foram implementadas outras funções booleanas para poder fornecer escolhas de ordenação, por exemplo, ordenar por data, alfabeticamente, ou localização.

Usamos pesquisa linear para pesquisar em Linked Lists e pesquisa linear com iteração em ordem na pesquisa em Binary Search Trees.

Finalmente, asseguramos que todos os ficheiros necessários são lidos no início do programa automaticamente e são guardados no final do programa, também automaticamente, desde que o programa seja fechado pelo menu.

# Classes e Relações

- BaseDados
- Date
- Time
- Aviao
- Serviço
- Voo
- Aeroporto
- Passageiro
- Bagagem
- Bilhete
- TransporteBagagem
- BST



# Estrutura de Ficheiros - Aeroporto & Transportes

Um ficheiro “aeroporto\_all\_save.txt”  
que lista todos os aeroportos pelo nome,  
cidade e país. Estes só podem ter uma palavra,  
por isso ‘\_’ podem ser necessários.

Um ficheiro  
“aeroporto\_{nome}\_{cidade}\_{pais}\_save.txt”,  
por objeto da classe Aeroporto, que lista os  
transportes nesse aeroporto, pela distância,  
tipo, dia, mês, ano, hora e minuto.

```
Francisco_sa_carneiro Porto Portugal  
Humberto_delgado Lisboa Portugal  
Internacional_de_faro Faro Portugal  
London_Heathrow London UK
```

aeroporto\_all\_save.txt

```
London_Heathrow London UK  
9 metro 15 12 2021 11 30  
10 autocarro 15 12 2021 15 30
```

aeroporto\_London\_Heathrow\_London\_UK\_save.txt

# Estrutura de Ficheiros - Voo

Um ficheiro “voo\_all\_save.txt” onde estão presentes os números de todos os voos existentes.

Um ficheiro “voo\_{número do voo}\_save.txt” com a estrutura:

Número\_do\_voo Data Duração\_do\_voo Número\_de\_passageiros

Nome Cidade País (do aeroporto de origem)

Nome Cidade País (do aeroporto de destino)

C N M (características do transporte de bagagem)

ID\_Passageiro Nome\_Passageiro Número\_de\_Bilhetes Número\_de\_Bagagens

Número\_do\_Bilhete Check\_Bagagem Checkin\_Automatico

...

Comprimento Largura Altura Peso (da bagagem)

...

```
1 10 5 2021 2 2
Francisco_Sa_Carneiro Porto Portugal
London_Heathrow London UK
2 4 6
1 Joao 2 2
1 1 1
2 1 1
10 10 10 20
5 10 5 8
2 Lucas 1 1
3 1 1
5 5 5 15
```



# Estrutura de Ficheiros - Avião & Serviços

Um ficheiro “avioes\_all\_save.txt” e outro “servicos\_all\_save.txt” que lista todos os Aviões e Serviços, respetivamente, com a seguinte estrutura:

avioes\_all\_save.txt

(Número de Aviões)

(Matrícula1) (Capacidade) (Nº de Voos) (Nº de Ser. Agendados) (Nº de Ser. Terminados)  
(id Voo1)

...

(ID Serviço Agendado 1)

...

(ID Serviço Terminado 1)

...

servicos\_all\_save.txt

(Nº de Serviços)

(ID1) (Funcionário1) (Tipo1) (Data1)

...

# Lista de Funcionalidades

	CRUD	Listagem Total	Listagem Parcial
Aeroporto	✓ completa	✓ completa	✓ completa
Avião	✓ completa	✓ completa	✓ completa
Voo	✓ completa	✓ completa	✗ parcial
Transporte	✗ parcial	✓ completa	✗ parcial
Serviço	✗ parcial	✓ completa	✓ completa

# Funcionalidade em Destaque

## Load e Save Automático

No início do programa, todos os dados são carregados automaticamente e colocados na devida estrutura de dados.

No final do programa, todos os dados são guardados automaticamente nos devidos ficheiros, desde que o programa seja fechado pelo menu.

```
    bd.saveAll();  
    return 0;  
}
```

```
int main() {  
    BaseDados bd;  
    bd.loadAll();  
}
```

```
/// Loader  
/// Função que chama todas as outras funções que carregam as informações para a base de dados  
void BaseDados::loadAll() {  
    loadServicos();  
    loadAirports();  
    loadVoos();  
    loadAvioes();  
}  
  
/// Saver  
/// Função que chama todas as outras funções que guardam as informações da base de dados nos ficheiros  
void BaseDados::saveAll() {  
    saveServicos();  
    saveAirports();  
    saveVoos();  
    saveAvioes();  
}
```

# Dificuldades Encontradas e Contributos

## Dificuldades

Problemas relacionados com o git e o cmake.

Transporte da Bagagem para o avião: Não foi totalmente implementada.

Listagens Parciais.

Problemas ao ler ficheiros que não acabavam com '\n'.

Erro ao ler ficheiros de Aeroportos sem transportes, por isso é obrigatório todos os aeroportos terem pelo menos um transporte.

Falta de tempo.

## Contributos



# FIM

