



Universidade do Porto

**FEUP** Faculdade de  
Engenharia

# T3 - Reserva de Voos (II)

Bruno Sousa –201604145– up201604145@fe.up.pt

João Gonçalves – 201604245 – up201604245@fe.up.pt

Joaquim Santos– 201604602– up201604602@fe.up.pt

03 de Janeiro de 2018

Turma 2 – Grupo 3 – MIEIC

# Índice

Índice	1
Companhia Aérea	2
Solução Implementada	3
Diagrama de classes (UML)	4
Casos de Utilização	5
Dificuldades	7
Distribuição do trabalho pelos elementos do grupo	7

# Companhia Aérea

Os aviões efetuam uma manutenção periódica, em que os aviões existentes são ordenados por ordem crescente por data da próxima manutenção, através de uma árvore binária de pesquisa. O programa vai permitir trabalhar com a periodicidade da manutenção, bem como a data da manutenção.

Os técnicos responsáveis pela manutenção dos aviões são qualificados em modelos de avião específicos, podendo um técnico ser qualificado em mais que um modelo de avião. Estes técnicos vão ser ordenados pela sua disponibilidade. Vamos considerar que o tempo de realizar a manutenção é igual para todos os aviões.

Os passageiros que não compram bilhetes há mais de um certo período (p.ex. 1 ano), são considerados inativos, retirados do sistema (usado na primeira parte do trabalho) e guardados numa tabela de dispersão sendo possível consultar os passageiros presentes na tabela de dispersão e atualizar os seus dados (nº telefone, p.ex). Se um passageiro inativo efetua a compra de um bilhete, é retirado da tabela de dispersão e passa novamente para o sistema.

## Solução Implementada

Para a solução implementada foi, primeiramente, criada uma árvore binária de pesquisa para os aviões existentes, estes que vão ser ordenados por ordem crescente por data da próxima manutenção.

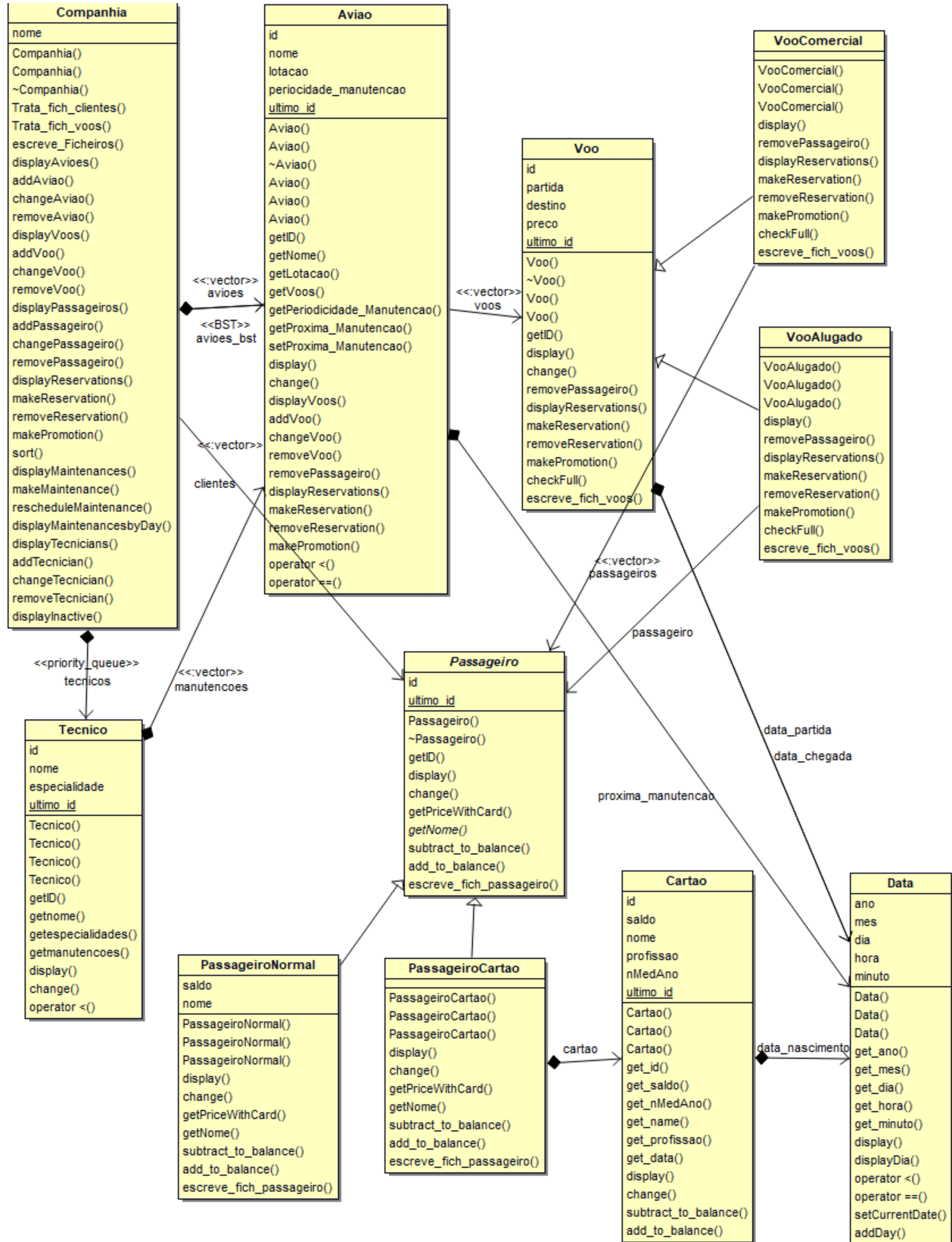
O programa vai permitir efetuar a manutenção de um avião, remarcar a manutenção de um avião e, por fim, fazer uma listagem das próximas manutenções num certo período. A data da manutenção vai estar no mesmo formato que a definida na classe “Data”.

Para fazer a marcação vamos também necessitar da informação relativa aos técnicos responsáveis por essa manutenção. Os técnicos de manutenção vão ser guardados numa fila de prioridade, sendo que estes vão estar ordenados pela sua disponibilidade, ou seja o técnico que está no topo da fila será o que estará disponível mais cedo, e por isso, vai ser este a que lhe vai ser atribuída a função de fazer a manutenção do avião.

Por fim, vamos também usar uma tabela de dispersão de forma a guardar todos os passageiros inativos (passageiros que não comprem bilhetes há mais de um certo período). Vamos poder trabalhar nesta tabela de dispersão, sendo que o programa vai permitir consultar os passageiros que estão na tabela de dispersão; vai permitir alterar os dados dos passageiros presentes na tabela de dispersão, e por fim, se um passageiro inativo fizer a compra de um bilhete, este é automaticamente retirado desta tabela de dispersão.

Relativamente às exceções, será lançada uma quando nós tentamos fazer a manutenção de um avião que não tem marcado uma manutenção para esse dia; quando tentamos remarcar a manutenção ou fazer a listagem mas o ficheiro dos aviões está vazio; quando não encontra o avião com o Id dado pelo utilizador, para as funções que fazem e remarcam as manutenções e, por fim, quando há fail no input.

# Diagrama de classes (UML)



## Casos de utilização

- Visualizar aviões: Listagem dos aviões guardados no ficheiro Avioes.txt
  - Adicionar(criar) avião: Pede o nome e a lotação do avião a adicionar para o ficheiro
  - Mudar avião: Inserindo o Id desse avião, pode-se mudar o nome e/ou lotação desse avião
  - Remover avião: Pede o Id do avião que queremos remover
- 

- Visualizar voos: Listagem dos voos guardados, no ficheiro Voo.txt
  - Adicionar(criar) voo: Pede o Id do avião e pergunta se o voo é comercial ou é um voo alugado, depois, pede o Id do voo, o lugar de partida, o lugar de chegada, a data de partida e a data de chegada (dd / mm /aaaa hora:min) e, por fim, pede o preço do bilhete
  - Mudar voo: Inserindo o Id desse voo, pode-se mudar o lugar de partida, o lugar de chegada, a data de partida, a data de chegada e/ou o preço do bilhete
  - Remover voo: Pede o Id do voo que queremos remover
- 

- Visualizar passageiros: Listagem dos passageiros ativos, guardados no ficheiro Clientes.txt
  - Adicionar(criar) passageiro: Pergunta se se trata de um passageiro normal ou de um passageiro com cartão e pede o nome do passageiro a adicionar para o ficheiro e o saldo deste se se tratar de um passageiro normal, enquanto que se for um passageiro com cartão pergunta a informação do cartão desse passageiro. Adiciona à hashtable de passageiros inativos
  - Mudar passageiro: Inserindo o Id desse passageiro, pode-se mudar o nome deste e/ou o saldo deste para um passageiro normal. Para um passageiro com cartão pode-se mudar o nome, o saldo, a profissão, a data de nascimento e/ou o número médio de voos por ano
  - Remover passageiro: Pede o Id do passageiro que queremos remover
- 

- Visualizar reservas: Listagem das reservas, guardados no ficheiro Voo.txt
  - Fazer(criar) reserva: Pede o Id do passageiro e o Id do voo em que se vai fazer a reserva •  
Remover reserva: Pede o Id do passageiro e o Id do voo de que queremos remover a reserva.
- 

- Fazer promoção: O preço do bilhete fica 10% mais barato se este cumprir as condições definidas (tempo < 48h, lotação < (lotação\_máxima /2)
-

- Sort: Ordena os aviões por lotação não crescente
- 

- Visualizar manutenções: Listagem das manutenções a realizar
  - Fazer manutenção: Pede o Id do avião e verifica se a data da manutenção corresponde com a data atual. Se sim, faz a manutenção do avião com esse Id
  - Remarcar manutenção: Primeiro pede o Id do avião e depois o ano, o mês e o dia da sua nova data de manutenção.
  - Visualizar manutenções por dia: Listagem das manutenções a realizar num determinado dia
- 

- Visualizar técnicos: Listagem dos técnicos inativos, guardados no ficheiro Tecnicos.txt
  - Adicionar(criar) técnico: Pergunta o nome, o numero de especialidades e as especialidade ao utilizador, cria um técnico com essa informação e adiciona-o à fila de prioridade de técnicos
  - Mudar passageiro: Inserindo o Id desse técnico, pode-se mudar o nome do técnico e o numero e a lista de especialidades
  - Remover passageiro: Pede o Id do técnico que queremos remover
- 

- Visualizar inativos: Listagem dos passageiros inativos, guardados no ficheiro Clientes.txt

## **Dificuldades**

Não houve grandes dificuldades quer no planeamento do projecto, quer na sua implementação, sendo que o mais trabalhoso deste trabalho foram as exceções, devido ao seu elevado número, e abrir e guardar os dados nos ficheiros de texto.

## **Distribuição de trabalho pelos elementos do grupo**

O nosso grupo tentou distribuir o trabalho pelos três elementos do grupo.

Todos os elementos participaram ativamente, contribuindo quer na implementação das estruturas de dados quer na sua aplicação e na parte mais teórica.

Concluindo, pensamos que o resultado final foi positivo, pois todos os membros se empenharam e trabalharam quer individualmente quer em prol do grupo, dando assim origem a um entendimento e uma boa cooperação entre os elementos do grupo.