

**AirShuttle: transfer em aeroportos**

CONCEPÇÃO E ANÁLISE DE ALGORITMOS

**Turma 1, Grupo G**

Daniel Garrido, up201403060

Edgar Passos, up201404131

Nuno Ramos, up201405498

**Índice**

Conteúdo

[Introdução 2](#_Toc449547284)

[Formalização do Problema 3](#_Toc449547285)

[Descrição da solução implementada 4](#_Toc449547286)

[Consola 6](#_Toc449547287)

[Interface Gráfica 9](#_Toc449547288)

Divisao de Trabalho10

Diagrama UML10

# 

# 

# Introdução

Foi no âmbito da unidade curricular de Conceção e Análise de Algoritmos, que nos foi atribuído o tema “AirShutle: transfer em aeroportos”, e foi pedido para resolver o problema da forma mais eficiente possível.

Nas próximas linha esta explicado qual a forma que encontramos para resolver o problema, de forma eficiente e estruturada.

# Formalização do Problema

Dada uma empresa que se especializa no movimento de passageiros que chegam ao aeroporto Francisco Sá Carneiro para hotéis na zona do porto (Porto, Matosinhos, Maia, Gaia, etc.), para o transporte dos passageiros, a empresa dispõe de carrinhas.

A empresa funciona da seguinte maneira. Os clientes requisitam o serviço pela internet, indicando hora de chegada do voo, e destino para onde desejam ir. A empresa com base nestes dados trata de fazer o plano de deslocações para o dia, tendo em conta o número de clientes, o destino desses clientes (hotéis) e a capacidade do meio de transporte, de forma a tornar eficiente o número de carrinhas utilizadas e os percursos feitos pelas carrinhas.

# Descrição da solução implementada

Na resolução começamos por criar classes que são fundamentais para a criação do grafo. As classes são as seguintes:

* Coordinates – classe que contém latitude e longitude, coordenadas de um node no mapa-mundo;
* Point – classes que contém coordenadas x e y, em radianos, localização de um node no mapa-mundo;
* Graph – classe responsável pela criação do grafo e respetivos cálculos;
* Node – os vértices do nosso grafo;
* Edge – ligação entre os Nodes do grafo;
* Road – representa uma rua no mapa, poderá ter nele contigo, várias Edges e vários Nodes, uma Road pode ser de dois sentidos uma Edge não pode;

De seguida definimos classes para lidar com os clientes, as reservas e as carrinhas:

* Passenger – o cliente;
* Reservation – representa uma reserva do cliente;
* Van – simula uma carrinha;

A classe responsável pela interação do utilizador com o programa é a classe UserInterface, nestas classes simulamos a empresa a prestar o serviço.

Relativamente à solução implementada para o problema apresentado, resolvemos começar pelas reservas, cada reserva é feita por um cliente indicando o nome, o destino (id do Node) e o horário de chegada, as reservas são colocadas numa fila de prioridade, sendo ordenadas pelo horário de chegada ao aeroporto. Em seguida, decidimos atribuir zonas de hotéis às carrinhas, cada zona é definida tendo em conta a distância entre os hotéis tornando a distância mínima e tornando a viagem de cada carrinha o mais eficiente possível. Passamos para a atribuição dos passageiros às carrinhas, nesta parte simplesmente distribuímos os passageiros pelas carrinhas tendo em conta o seu destino, colocando o passageiro numa carrinha que tenha o seu destino na zona de hotéis atribuída. Tendo sempre cuidado que a carrinha têm um limite de passageiros e caso esse limite seja ultrapassado e não exista mais nenhuma carrinha para essa zona o cliente fica na fila de prioridades e irá na viagem seguinte. Na demonstração do projeto será possível simular a empresa com várias carrinhas ou só com uma. Os hotéis e o aeroporto são Nodes escolhidos de forma aleatória.

O percurso é o mais curto possível, esse percurso é calculado utilizando o algoritmo de Dijkstra.

# Consola

Quando o programa é iniciado o menu apresentado é o seguinte:

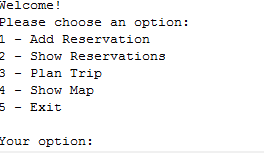


Figura 1 - Menu Principal

A opção 1 é para fazer uma reserva, sendo pedido o nome, o nif , o destino e a hora de chegada.

A opção 2 mostra as reservas todas da empresa.

A opção 3 executa as viagens de todas as carrinhas existentes, escrevendo na consola o seu percurso (ver figura 2) e em seguida abre uma janela do graph viewer onde mostra a vermelho o percurso realizado pela carrinha.



Figura 2 - Percurso seguido pela carrinha



Figura 3- Percurso feito por uma carrinha

A opção 4 mostra o mapa, onde tem os Nodes a amarelo e os Nodes dos hotéis têm um pequeno ícone com a imagem de uma casa.

A opção 5 serve para sair do programa.

# 

# Interface Gráfica

A interface gráfica apenas mostra os Nodes e Edges, na posição correta. Os Nodes dos hotéis têm um pequeno ícone com uma imagem de uma casa e têm o nome respetivo na legenda do Node.

Quando a carrinha efetua o seu percurso aparece no mapa, a cor vermelha, o percurso que realizou.

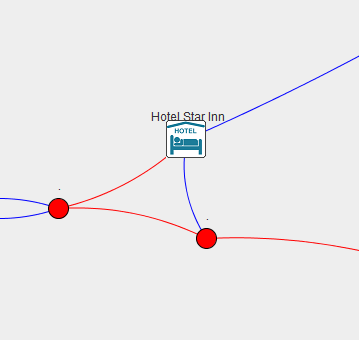
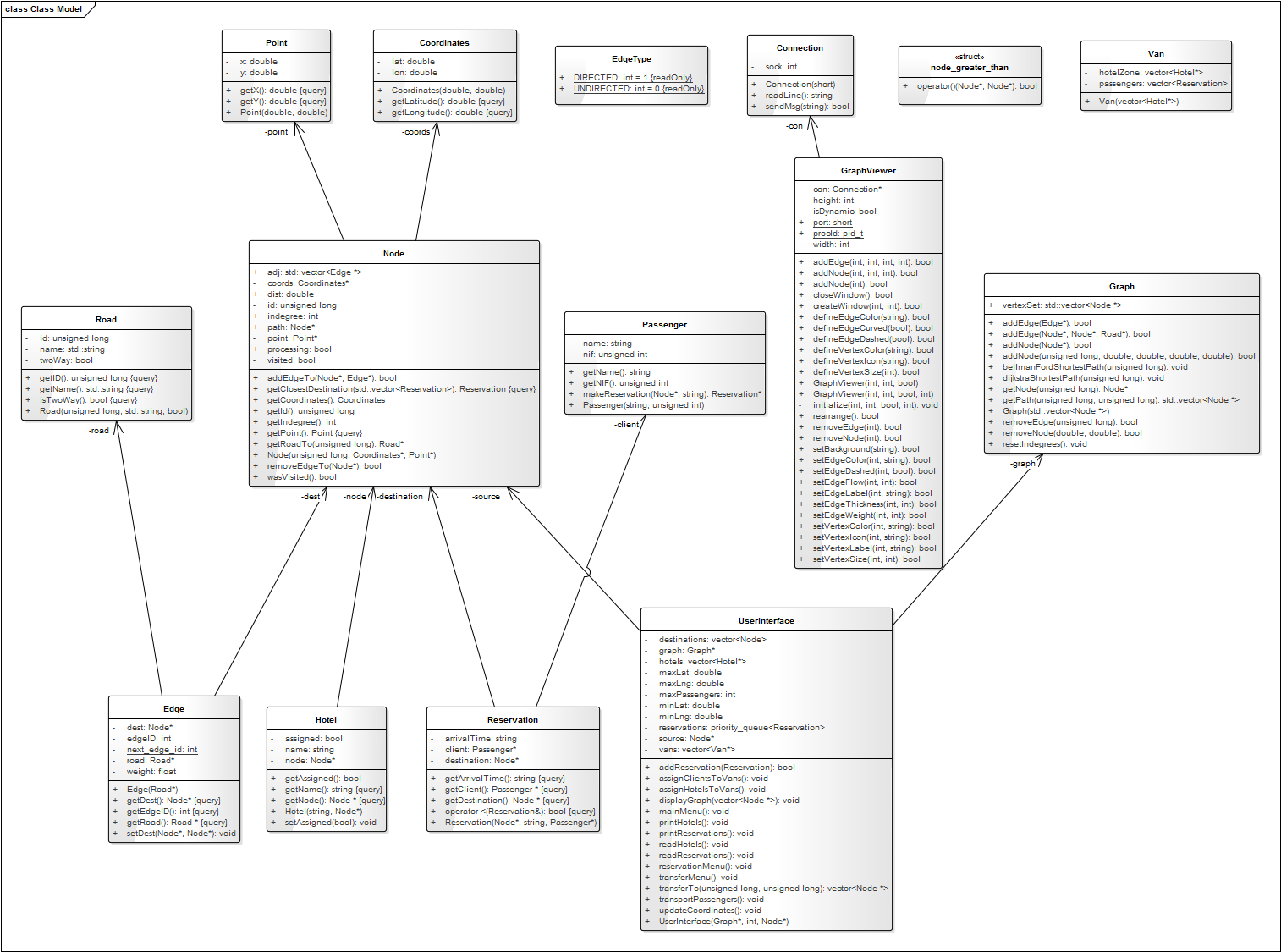


Figura 4 - Ícone de um hotel

# Divisão de Trabalho

# Os membros do grupo trabalharam de forma colaborativa em todas as componentes do trabalho.

# **Diagrama UML**

  
Figura 5 - Diagrama UML (disponivel em na pasta docs)