

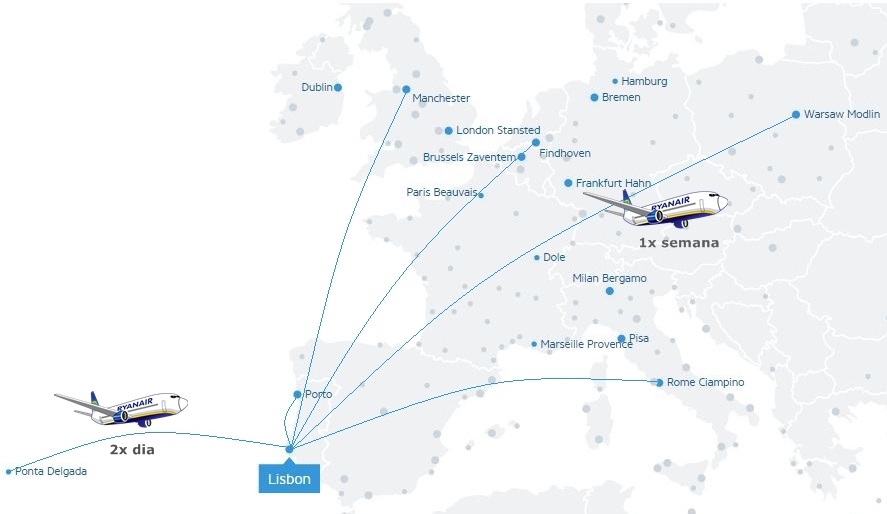
**2ºano – MIEIC – 2016/2017**

**Agência de Viagens**

**Parte 2**

**Conceção e Análise de Algoritmos**

**Turma 6 - Grupo A**

****

**Beatriz de Henriques Martins**

**João Carlos Almeida**

**João Miguel Monteiro**

**Porto, 9 de abril de 2017**

Índice

[Introdução 3](#_Toc479527692)

[Enunciado 4](#_Toc479527693)

[Descrição do Problema 6](#_Toc479527694)

[Formalização do Problema 7](#_Toc479527695)

[Solução 8](#_Toc479527697)

Algoritmos .........................................................................................................................7

Estrutura do Programa ......................................................................................................8

[Casos de Utilização 10](#_Toc479527698)

[Métricas de Avaliação 12](#_Toc479527699)

[Diagrama de Classes 13](#_Toc479527700)

[Principais Dificuldades 14](#_Toc479527701)

[Conclusão 15](#_Toc479527702)

# Introdução

Nos dias que correm, cada vez mais, queremos viajar da melhor forma possível, mais rápida e barata. Existem as mais diversas aplicações e sites para aconselhamento. Estes executam pesquisas por datas e melhor preço e melhor trajecto.

No âmbito da disciplina de Conceção de Análise de Algoritmos do curso Mestrado Integrado em Engenharia Informática e Computação foi-nos proposto a realização de uma Agência de Viagens. O nosso programa permite ao utilizador criar novos clientes e associar aos clientes uma viagem com deslocação e estadia. Fazendo uma pesquisa por melhor preço ou por um conjunto de pontos de interesse.

# Enunciado

**Parte I**

Uma agência de viagens auxilia os seus clientes na pesquisa das melhores soluções para a marcação de viagens e respetivo alojamento.

Neste trabalho, pretende-se implementar um sistema que permita à agência de viagens conseguir uma seleção mais criteriosa de uma solução de acordo com as necessidades do seu cliente, procurando sempre minimizar o seu custo. Considere que o custo das viagens (bilhetes de avião, por ex.) é fixo para o período a considerar. Mas o preço do alojamento varia de acordo com o dia da semana ou período específico (festividade local, por ex.).

O cliente especifica:

* origem e destino para a sua viagem, ou - conjunto de locais que pretende visitar na viagem
* O cliente pode ainda especificar um tempo limite para realizar a viagem.

Avalie a conectividade do grafo, a fim de evitar que locais a visitar se encontrem em zonas inacessíveis.

**Parte II**

Para a segunda parte deste trabalho, considere que o cliente oferece uma lista de nomes de lugares e pontos de interesse a serem incluídos no seu itinerário. Os pontos de interesses podem ser monumentos ou lugares numa cidade. A agência de viagem deverá pesquisar, na lista dos seus destinos, aqueles que estão incluídos na lista do cliente e/ou possuem os pontos de interesses que o cliente deseja visitar durante a sua viagem. Por exemplo, o cliente pode desejar visitar a “Torre dos Clérigos” e a agência deve ser capaz de lhe incluir a cidade do Porto, como destino no seu itinerário. Implemente esta funcionalidade, considerando tanto pesquisa exata, assim como pesquisa aproximada, das strings identificativas dos nomes de cidades e pontos de interesse. O resultado da pesquisa exata deverá indicar se existe ou não o lugar desejado. Para a pesquisa aproximada, deverá retornar uma lista com os resultados mais próximos daquilo que se procura.

Estas novas funcionalidades deverão ser integradas no trabalho já realizado para a primeira parte. Avalie a complexidade (teórica e empiricamente) dos algoritmos implementados em função dos dados de input usados.

# Descrição do Problema

De uma forma geral, pretendemos resolver o problema de transportes entre países através de uma aplicação que, apenas recebendo o local de partida e chegada e/ou uma lista de pontos de interesse associado a datas, consiga devolver o transporte mais rápido ou barato e o melhor trajeto possível. Dependendo das preferências do utilizador.

**Input**

Construção de um grafo, G = (V, E) de transportes disponíveis no qual:

* V – vértices

Representam todas as cidades dos meios de transporte disponíveis;

* E – arestas

Representam todas as distâncias, tempos de viagens e custos;

* Nó de início de viagem e nó de destino.

**Introdução de dados**

Um ficheiro com as cidades e pontos de interesse mais relevantes com as respetivas coordenadas.

Um ficheiro com os dados dos transportes disponíveis.

**Output**

O transporte com os melhores preços e percurso, associado às datas desejadas.

**Objetivo**

Facilitar aos utilizadores a escolha dos melhores trajetos consoantes os critérios por estes preferidos.

Otimizar o percurso segundo os ponto de interesse.

# Formalização do Problema

## Formalizamos agora o problema, de acordo com a solução que achamos mais vantajosa

**Input**

G<V, E>

V: cidades

E: ligações entre as cidades

I: cidade inicial

F: cidade final

**Output**

Caminho={Vi},I=1...n

Valor

**Objetivo**

Min(valor):

# Solução

**Algoritmos**

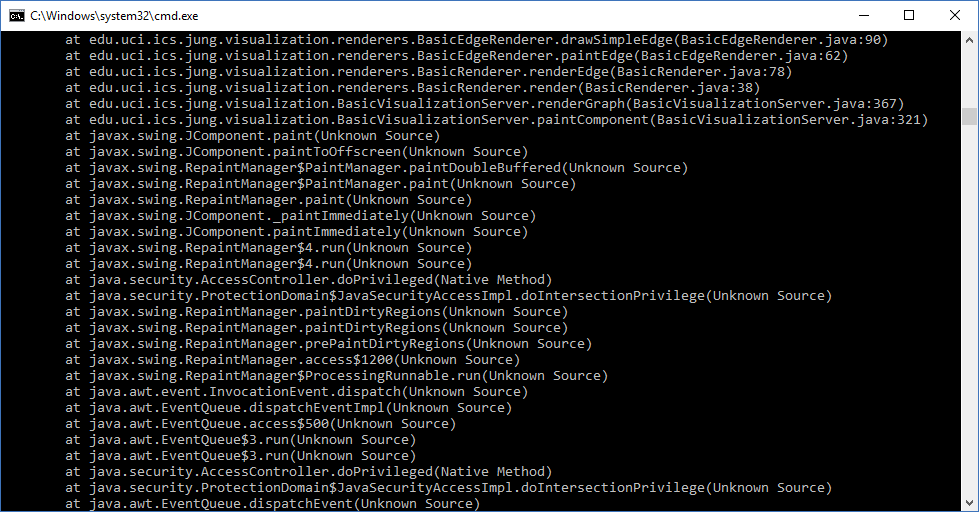
Depois de analisar muito bem o problema, chegamos à conclusão de que o melhor algoritmo a implementar seria o ***Algoritmo de Dijkstra***. Este encontra o menor caminho possível entre dois vértices de um grafo, dirigido ou não, e em tempo computacional

***O( [arestas + vértices] log (vértices))***

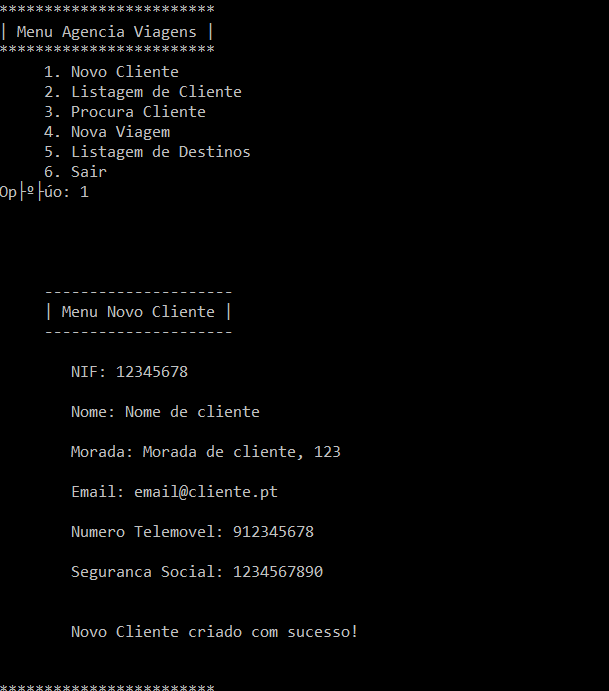
O ***Algoritmo de Dijkstra*** é um algoritmo ganancioso, ou seja, toma decisões que parecem ótimas no momento, determinando assim os conjuntos de melhores caminhos intermediários possíveis. O valor de cada aresta está associado à distância entre duas cidades, calculada através das coordenadas (latitude e longitude) de cada cidade. O algoritmo em questão não pode ser usado em grafos com peso negativo, dado que a distância entre duas cidades nunca pode ser negativa, é perfeitamente aplicável ao problema em questão.

**Estrutura do Programa**

Ao iniciar o programa, começamos por construir o grafo. Introduzindo as Cidades com as respetivas coordenadas para construir os caminhos entre si (vértices) e o hotel correspondente a cada cidade. De seguida, carrega a informação dos clientes.



Depois, é altura de o utilizador poder interagir com o programa em si. O usuário é obrigado a ter um ficha com os seus dados quando pretende agendar uma viagem.



O Cliente escolhe uma data e um local de partida seguido de um local de chegada e a data de regresso.

(IMAGEM NOVA VIAGEM)

De seguida, é sugerido ao utilizador uma viagem com tudo o que este escolheu.

(IMAGEM Grafo com viagem e estadia)

# Casos de Utilização

* Leitura e interpretação de dados de ficheiro relativos a um mapa;
* Escolha do melhor percurso em termos de preço ou tempo por viagem;
* Escolha do melhor percurso segundo os pontos de interesse do cliente;
* Visualização de todo o mapa através do GraphViewer;
* Visualização do percurso escolhido através do GraphViewer.

# Métricas de Avaliação

**Avaliação Empírica do Desempenho**

Para testar e avaliar a complexidade temporal do programa, utilizamos diferentes dados de entrada.

**Complexidade Temporal**

Como referido anteriormente, o ***Algoritmo de Dijkstra*** tem um tempo computacional de

***O( [arestas + vértices] log (vértices))***

# ../Downloads/Sem%20nome.jpgDiagrama de Classes

# Principais Dificuldades

Durante a realização deste projeto, deparamo-nos com alguns problemas. Um dos maiores problemas foi a introdução de dados fidedignos sobre viagens entre países. Inicialmente, íamos usar uma ***API*** gratuita e muito completa. Até que percebemos que fazer a leitura do ficheiro seria uma muito difícil e atrasaria em muito o projeto. Por esse motivo, decidimos usar um ficheiro de texto com toda a informação que necessitávamos para a construção do grafo.

Também tivemos alguns problemas de leitura de dados de ficheiros, devíamos às diferentes gamas implementadas nos diferentes sistemas operativos usados pelos membros do grupo.

Relativamente à parte essencial do projeto, sentimos algumas dificuldades na implementação da pesquisa em grafos.

# Conclusão

A realização deste projeto desempenhou um papel crucial no auxilio à interpretação de grafos e dos algoritmos que lhes são relacionados. Nomeadamente, o ***Algoritmo de Dijkstra***.

A ponto de minimizar os erros, decidimos criar uma “Agência de Viagens” apenas para a Europa. Assim, conseguimos que os nossos dados fossem o mais fidedignos possíveis.