

**MIEIC - Concepção e Análise de Algoritmos**

**Planeamento de itinerários multimodais**

**Relatório Intercalar**

**realizado por:**

**2MIEIC01 - Grupo B - Tema 4**

# Página de Rosto

**Tema do Projeto:** 4;

**Curso:** Mestrado Integrado em Engenharia Informática e Computação;

**Unidade Curricular:** Cencepção e Análise de Algoritmos.

**Ano letivo:** 2014/15;

**Grupo B:**

* João Nogueira, nº 201303882 up201303882@fe.up.pt;
* António Pedro Fraga nº 201303095 up201303095@fe.up.pt;
* Filipa Barroso nº 201307852 up201307852@fe.up.pt.

Turma 1;

**Data de Entrega:** 27 de Abril de 2015.

**Índice**

[Página de Rosto 2](#_Toc417779162)

[Introdução 4](#_Toc417779163)

[Explicação do Problema 5](#_Toc417779164)

[Descrição do Problema 6](#_Toc417779165)

[Input 6](#_Toc417779166)

[Introdução de dados 6](#_Toc417779167)

[Output 6](#_Toc417779168)

[Objetivo 6](#_Toc417779169)

[Restrição 6](#_Toc417779170)

[Formalização do Problema 7](#_Toc417779171)

[Input 7](#_Toc417779172)

[Output 7](#_Toc417779173)

[Objetivo 7](#_Toc417779174)

[Restrição 7](#_Toc417779175)

[Solução 8](#_Toc417779176)

[Algoritmos 8](#_Toc417779177)

[Esquema do programa 9](#_Toc417779178)

[9](#_Toc417779179)

[Casos de Utilização 10](#_Toc417779180)

[Métricas de avaliação 10](#_Toc417779181)

[Avaliação empírica do seu desempenho 10](#_Toc417779182)

[Complexidade temporal 10](#_Toc417779183)

[Diagrama de Classes 11](#_Toc417779184)

[Principais dificuldades 12](#_Toc417779185)

[Esforço por elemento 12](#_Toc417779186)

[Código 12](#_Toc417779187)

[Introdução de Dados 12](#_Toc417779188)

[Implementação Gráfica 12](#_Toc417779189)

[Implementação de algoritmos 12](#_Toc417779190)

[Relatório 12](#_Toc417779191)

[Conclusão 13](#_Toc417779192)

# Introdução

Foi no âmbito da Unidade Curricular de Concepção e Análise de Algoritmos do 2º ano do MIEIC que nos foi apresentado um problema que nos foi pedido para resolver da forma mais eficiente possível.

Neste relatório está explicado o problema em si, por forma a que se entenda exatamente qual o nosso objetivo e como o pretendemos atingir.

Depois de explicitar qual o problema que procuraremos resolver, explicamos neste relatório os métodos que usaremos para a resolução do problema em questão.

# Explicação do Problema

É uma preocupação cada vez maior, o facto de nas grandes cidades o transporte individual ser demasiado utilizado, em detrimento dos transportes comunitários. Um problema presente na maior parte das cidades com vários serviços de transportes públicos é o facto de não haverem plataformas que facilitem ao utilizador a escolha do melhor itinerário tendo em conta os vários tipos de transporte e as várias alternativas de chegar a um destino. Estas plataformas teriam em conta os transportes com paragens em comum e os horários dos vários transportes e linhas diferentes.

Nesta plataforma, caberia ao utilizador apenas introdroduzir a origem e o destino do percurso. A plataforma calcularia os vários itinerários possíveis e mostraria ao utilizador o itenerário multimodal mais curto/rápido.

Para avaliar o itinerário podem ser utilizados diversos critérios tal como o tempo de viagem e distância.

# Descrição do Problema

De uma forma geral, pretendemos resolver o problema de itinerários intermodais através de uma aplicação que, apenas recebendo a estação de origem e a de destino, consiga devolver o itinerário mais rápido, curto ou barato dependendo das preferências do utilizador.

## Input

Construção de um grafo, G = (V, E), de estações e linhas no qual:

* V – vértices – representam todas as paragens/estações dos meios de transportes 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 das estações com respetivas linhas de transportes que nelas passam, id e coordenadas de cada um deles.

## Output

Todas as estações em que o utilizador passou por forma a otimizar o seu trajeto e o valor final da distância, duração da viagem dependendo da preferência do utilizador.

## Objetivo

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

## Restrição

No mesmo caminho não se passa pela mesma estação mais do que uma vez.

# Formalização do Problema

Formalizamos agora o problema, de acordo com aquela que achamos ser a melhor forma para resolver aquilo a que nos propusemos.

## Input

*;*

*.*

## Output

## Objetivo

## Restrição

# Solução

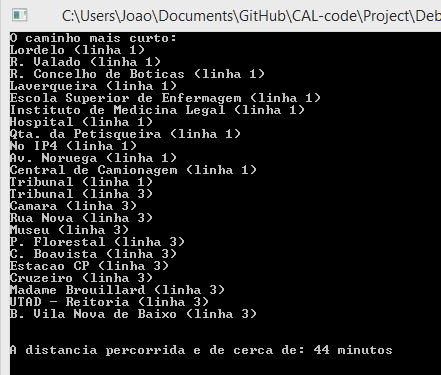
## Algoritmos

Pensamos que o melhor algoritmo a ser usado neste trabalho é o **algoritmo de Dijkstra,** que encontra o caminho mais curto num grafo dirigido ou não dirigido em **tempo computacional O([arestas + vértices]log (vértices)),** este algoritmo não pode ser usado para grafos com pesos negativos, mas visto que este não é o caso, o algoritmo é perfeitamente aplicável.

Este algoritmo é um algoritmo ganancioso, tomando decisões que parecem óptimas no momento, determinando assim o conjunto de melhores caminhos intermediários. O valor de cada aresta está associado ou à distância, calculada através das coordenadas de cada estação, ou ao tempo, calculado com uma velocidade média. A única diferença entre estes dois pesos, são as trocas de linha, estas apenas têm peso quando o tempo for tido em conta.

## Esquema do programa

Começamos por introduzir dados de graphos, vértices (estações) e arestas (linhas). De seguida, agrupamos os graphos em dois tipos de estruturas de dados. Uma das estruturas é utilizada para representação gráfica e outra delas é utilizada para cálculos de rotas ideais, cálculos esses efetuados pelo algoritmo descrito em cima. As duas estruturas de dados apresentam ligeiras diferenças, de forma a tornar a representação gráfica mais “*user friendly*”.

Depois de carregar todos os dados, é altura do utilizador escolher o tipo de optimização de viagem, a estação de origem e a estação de destino, seguido da indicação de rota (diferente cor de arestas) na representação gráfica e uma lista das estações que constituem a rota otimizada.

# 

# Casos de Utilização

* Leitura de dados de ficheiros relativos a redes de transportres;
* Escolha do melhor percurso em termos de tempo de viagem;
* Escolha do melhor percurso em termos de distância percorrida;
* Visualização através do GraphViewer de toda a rede de transportes;
* Visualização do melhor percurso através do GraphViewer.

# Métricas de avaliação

## Avaliação empírica do seu desempenho

Para avaliar e testar a complexidade temporal dos algoritmos utilizamos diferentes dados de entrada.

## Complexidade temporal

Como já foi referido anteriormente, a complexidade temporal do algoritmo que estamos a implementar é de **O([arestas + vértices]log (vértices)).**

# Diagrama de Classes

# Principais dificuldades

Ao longo da realização deste trabalho encontramos algumas dificuldades, no entanto, aquela que se demonstrou mais difícil de contornar foi a obtenção de informação acerca das várias linhas e redes de transportes a partir dos websites disponíveis, fosse o website da STCP, fosse a partir de *Open Street Maps*.

Optámos então por introduzir manualmente em ficheiros *.txt* a informação relativa à rede de transportes.

# Esforço por elemento

## Código

### Introdução de Dados

No que diz respeito à procura e forma de introdução dos dados da rede de transportes o João Nogueira foi quem supervisionou. Porém contudo, esta foi feita também pelos outros elementos.

### Implementação Gráfica

Em relação à implementação da parte Gráfica, todos os elementos participaram de igual forma e ativamente, tendo esta sido feita paralelamente à implementação do algoritmo.

### Implementação de algoritmos

Em relação à implementação do algoritmo, todos participaram na implementação estando o Pedro Fraga responsável por esta componente.

## Relatório

No que diz respeito ao relatório, todos os elementos do grupo se esforçaram igualmente para a sua realização.

# Conclusão

Desde o relatório perliminar houve algumas pequenas alterações ao trabalho, tendo em conta os problemas encontrados durante o desenvolvimento do programa. O utilizador pode apenas escolher o percurso otimizado por tempo de viagem ou por distância percorrida. Os graphos são apenas preenchidos com informação após a escolha por parte do utilizador da forma de otimização do percurso. Estas são algumas das alterações que nos vimos obrigados a fazer por forma a implementar o trabalho da melhor forma.

Toda a nossa rede de transportes foi baseada na cidade de Vila Real. Não vimos portanto a necessidade de implementar mais do que um tipo de transporte, tendo em conta que a alteração de uma linha para a outra (transbordo) implica o tempo de deslocação entre estações e tempo de espera.

A realização deste trabalho serviu para ficarmos mais inteirados da matéria em questão, particularmente com o modo de funcionamento do Algoritmo de Dijkstra.