Easy Pilot: Sistema de Navegação

Conceção e Análise de Algoritmos



27 de Maio de 2016

**Turma 6, Grupo C**

José Aleixo Peralta da Cruz

[up201403526@fe.up.pt](mailto:up201403526@fe.up.pt)

José Carlos Alves Vieira

[up201404446@fe.up.pt](mailto:up201404446@fe.up.pt)

Renato Sampaio Abreu

[up201403377@fe.up.pt](mailto:up201403377@fe.up.pt)

Índice

[1. Descrição do tema 2](#_Toc449373081)

[2. Formalização do problema 2](#_Toc449373082)

[2.1. Dados de entrada 2](#_Toc449373083)

[2.2. Limites da aplicação 2](#_Toc449373084)

[2.3. Situações de contorno 2](#_Toc449373085)

[2.4. Resultados esperados 3](#_Toc449373086)

[3. Descrição da Solução 4](#_Toc449373087)

# Descrição do tema

No âmbito da unidade curricular Conceção e Análise de Algoritmos, foi atribuída ao nosso grupo a tarefa de elaborarmos um programa em C++ que, através da implementação de grafos, funcionasse de forma semelhante a uma interface de GPS.

Neste sistema de navegação, apelidado “*Easy Pilot*”, o utilizador tem à sua disposição um mapa, no qual pode selecionar os pontos de início e de final de uma viagem. O programa encarrega-se de encontrar o caminho mais curto possível entre esses dois pontos, que deve evitar zonas inacessíveis. Fica ainda ao alcance do utilizador adicionar pontos de interesse que queira visitar durante essa viagem, sendo que o programa adapta o trajeto para incluir esses locais.

# Formalização do problema

## Dados de entrada

Como dados de entrada no programa são utilizados mapas que representam localizações reais, obtidos no OpenStreetMaps (OSM - [www.openstreetmaps.org](http://www.openstreetmaps.org)). Os ficheiros exportados a partir do OSM estão escritos em XML, pelo que se usa um *parser* para transformar essa informação em texto, de forma a facilitar a leitura dos dados.

O *parser* origina três ficheiros de texto, cada um contendo informações necessárias para a elaboração de um grafo que representa a área geográfica extraída, segundo a seguinte tabela.

|  |  |  |
| --- | --- | --- |
| Ficheiro | Informação do grafo (estrutura) | Equivalência real |
| FicheiroA.txt | Nós (id, latitude, longitude) | Locais |
| FicheiroB.txt | Arestas (id, nome, sentido) | Estradas |
| FicheiroC.txt | Conexões (idAresta, idNó1, idNó2) | Estradas entre locais |

O grafo G obtido pode ser declarado da seguinte forma: . é o conjunto dos pontos geográficos relevantes ao mapa e é o conjunto das secções de estrada que interligam esses pontos.

## Limites da aplicação

No programa são usados mapas que representam uma área geográfica relativamente pequena, pois o grafo torna-se exponencialmente mais complexo quanto maior for o mapa usado.

Assim, uma limitação desta aplicação é que nem sempre será encontrado um caminho que siga todas as condições estabelecidas. Isto acontece porque a rota que cumpriria essas restrições está fora da área coberta pelo programa.

## Situações de contorno

Para permitir alguma liberdade no percurso que o utilizador faz, a aplicação possibilita a adição de “pontos de interesse” ao trajeto que foi definido segundo a origem e o destino final. Neste caso, o programa reajusta o trajeto calculado segundo uma determinada condição, para que seja possível visitar todos os pontos de interesse adicionados.

Com o objetivo de simular um cenário real, em que uma traço de uma estrada pode se encontrar em obras, inclui-se um conjunto de “zonas inacessíveis” ao programa, que o obrigam a delinear rotas que evitem essas zonas.

## Resultados esperados

A função desta aplicação está em deixar o utilizador selecionar um mapa dos que estarão disponíveis, escolhendo um ponto de origem e um ponto de chegada de uma viagem, calculando a melhor trajetória a seguir. A essa viagem podem ou não ser adicionados pontos de interesse, que deverão fazer parte do trajeto. Além disso, é de esperar que o programa evite zonas inacessíveis, indicando uma rota rápida e correta.

# Descrição da Solução