

INTELIGÊNCIA ARTIFICIAL

Licenciatura engenharia informática
Janeiro 2024

GRUPO23

Ana Margarida Sousa Pimenta A100830

Inês Gonzalez Perdigão Marques A100606

Pedro Miguel da Costa Azevedo A100557

Miguel Tomás Antunes Pinto A100815

Samuel Macieira Ferreira A100654

Avaliação pelos pares:

- PG100830 Ana Margarida Sousa Pimenta DELTA = 0
- PG100606 Inês Gonzalez Perdigão Marques DELTA = 0
- PG100557 Pedro Miguel da Costa Azevedo DELTA = 0
- PG100815 Miguel Tomás Antunes Pinto DELTA = 0
- PG100654 Samuel Macieira Ferreira DELTA = 0

Consideramos que todos os elementos do grupo tiveram um papel importante na realização deste projeto.

Todos os métodos adotados foram conversados entre todos e desde início que foi feita uma divisão de tarefas por entre todos os elementos.

Índice

Introdução	3
Descrição e Formulação do Problema	4
Descrição das Tarefas Desenvolvidas.....	5
Menu do Cliente.....	5
Menu do Estafeta e Gestão dos Estafetas	6
Grafos.....	7
Algoritmos Utilizados	9
Procura Informada	9
Procura Não Informada.....	11
Análise de Resultados.....	12
Conclusão.....	14

Introdução

Para a realização do projeto respetivo à *UC de Inteligência Artificial*, foi-nos proposta a implementação de uma empresa de distribuição de encomendas, a *Health Planet*. Esta empresa visa adotar métodos mais sustentáveis, sendo que para esse fim recorrem a fatores que tentam promover a utilização de meios de transporte menos poluentes.

Quanto ao seu objetivo, o foco principal deste trabalho prático compreende-se no desenvolvimento de algoritmos de procura que permitam otimizar o meio de entrega de encomendas, considerando, como referimos previamente, diferentes meios de transporte com distintos níveis de impacto ambiental.

A nossa rede de distribuição estará inserida na zona sul, sendo que os estafetas irão operar na Freguesia da Misericórdia, em Lisboa.

Para agilizar as entregas e aumentarmos a área sobre as quais estas são realizadas, recorreremos à biblioteca *OSMnx* de modo a criar um grafo que abrange toda a área da freguesia e consequentemente, todas as ruas a esta correspondentes. Sob este grafo são utilizados algoritmos de procura informada de maneira a otimizar o processo de escolha.

Para além disso, para demonstrar e visualizar o comportamento de algoritmos de procura não informada, implementámos um outro grafo, que suporta poucas ruas da freguesia, mas que permite ver a atividade dos algoritmos e assim ter de facto uma perceção mais realista de como estes funcionam.

Assim, de um modo geral, o projeto propõe o desenvolvimento de algoritmos que integrem todos estes elementos, desde a otimização do percurso e escolha do meio de transporte mais sustentável até à gestão de prazos de entrega e satisfação do cliente. A conjugação destes fatores constitui um desafio que será abordado ao longo deste trabalho prático.

1. Descrição e Formulação do Problema

Quanto à descrição do problema, este envolve a otimização do processo de distribuição de encomendas feito pela *Health Planet*.

Os estafetas da empresa têm à sua disposição diferentes meios de transporte, cada um caracterizado por um consumo de energia específico.

O objetivo é encontrar a combinação ideal de entregas, meios de transporte e percursos que minimizem tanto o impacto ambiental quanto o tempo de entrega, garantindo a satisfação dos clientes.

Assim sendo, temos então que:

Estado Inicial: Pedidos para entregas pendentes no armazém na Travessa do Carmo.

Estado Objetivo: Pedidos entregues com sucesso nas respetivas localizações.

Os operadores neste processo de distribuição de encomendas refletem-se em ações que de certa forma afetam de um modo construtivo a concretização da tarefa, pelo que temos:

Operadores:

- Atribuição de estafetas aos pedidos

A atribuição de estafetas aos pedidos que se encontram pendentes é necessária para alcançar o estado objetivo e reduzir o número de pedidos pendentes.

- Atribuição dos meios de transporte

A atribuição do meio de transporte para além de ser necessária para atingir o estado objetivo é também importante no que toca à contribuição para a sustentabilidade.

- Priorização de pedidos que apresentam um menor tempo de entrega

A priorização dos pedidos que apresentam um menor tempo de entrega é um processo importante e necessário de modo a não só garantir uma ordem e gestão nas entregas, mas também para obedecer ao tempo de entrega e contribuir para a satisfação dos clientes.

Tipo do Problema: Este é um problema de estado único pois conhecemos as ações/possíveis estados e o estado global. Quanto às suas características é determinístico, discreto, de informação perfeita, estático, não episódico e acessível.

2. Descrição das Tarefas Desenvolvidas

De modo a descrever as tarefas que desenvolvemos, iremos dividir em tópicos o processo e métodos adotados para os vários intervenientes e variáveis do nosso projeto.

Quanto às informações respetivas tanto aos estafetas como às encomendas, de modo a poder inicializar o estado e posteriormente atualizar as informações, decidimos criar um ficheiro que guarda a representação do estado.

2.1 Menu Cliente

Quanto ao menu do cliente, isto é, as atividades que este pode realizar na plataforma, a partir da análise do enunciado compreendemos que este terá as seguintes funcionalidades:

- **Definir o tempo de uma encomenda:**

Para o cliente definir o tempo em que pretende receber uma encomenda, o programa pede ao cliente para este inserir o *ID* da sua encomenda e o tempo, em horas e/ou minutos, que este pretende que a entrega seja efetuada.

O cliente pode inserir o número de encomendas que pretende e estas serão atribuídas aos estafetas tendo em conta a prioridade de entrega, ou seja, as encomendas que tem menos tempo serão atribuídas aos estafetas pela ordem em que estes se encontram na nossa representação do estado.

As encomendas são armazenadas numa lista que contém as encomendas registadas.

- **Avaliar encomenda:**

No menu do cliente este pode avaliar a sua encomenda atribuindo um valor de 0 a 5. Após essa avaliação o programa pergunta ao cliente se a encomenda chegou dentro do tempo pretendido. Caso o cliente indique que esta não chegou dentro do tempo pretendido, a avaliação é penalizada em 0,2. Após isso são exibidas as informações do estafeta como a avaliação atribuída, o conjunto de avaliações atribuídas por outros clientes, a média dessas avaliações e o número de entregas realizadas.

Quando é efetuada a avaliação da encomenda, o seu estado de entrega é atualizado para “*Entregue*”.

- **Visualizar Encomendas:**

O cliente pode ver as encomendas que tem registadas. A lista das encomendas registadas é verificada e se o cliente tiver encomendas as suas informações serão exibidas.

A informação relativa ao preço é calculada recorrendo ao ficheiro onde o preço é calculado tendo em conta o prazo definido, ou seja, uma taxa é acrescida ao valor base no caso de o tempo definido ser curto.

- **Criar Encomenda:**

O cliente pode criar uma encomenda com novos dados. O programa pede ao cliente para este introduzir o a rua onde pretende que a encomenda seja entregue, o peso e o volume.

Após isso o programa vai atribuir o próximo *ID* de encomenda disponível e o cliente pode definir o tempo de entrega para quando pretender.

- **Ver Ranking de Estafetas com Melhor Avaliação:**

O cliente pode ver os cinco estafetas com melhor avaliação.

Para efetuar esta tarefa, implementámos uma função verifica os estafetas válidos, ou seja, com *ranking* positivo, e ordena-os por ordem decrescente. Após isso obtém os cinco primeiros estafetas da lista e exhibe a média das suas avaliações.

2.2 Menu do Estafeta e Gestão de Estafetas

Neste tópico iremos abordar tanto as funcionalidades do menu do estafeta bem como o modo como está a ser efetuada a gestão dos estafetas, ou seja, como estes são atribuídos às encomendas.

- **Gestão dos Estafetas:**

De maneira a gerir os estafetas, efetuamos a atribuição das encomendas aos mesmos tendo em conta a prioridade de entrega com base no tempo definido pelo cliente.

Por uma questão de teste e visualização mais simplificada de resultados, atribuímos as encomendas aos estafetas sequencialmente. Após a atribuição da encomenda ao estafeta atualizamos a sua disponibilidade para *False*.

Para realizar esta tarefa, priorizamos as encomendas com base no critério de prioridade que é o prazo de entrega. Após isso ordenamos as encomendas com base nesse mesmo critério que ordena enquanto existirem encomendas.

Caso os números de estafetas para associar a uma entrega esteja esgotado, verificamos se o primeiro já tem disponibilidade *True* ou se o prazo de entrega definido é inferior ao tempo de entrega total (isto é, ida e volta, que é o tempo definido pelo cliente mais cinco minutos de tolerância) e, se isso se verificar, atualizamos a disponibilidade para *True*.

- **Processar Encomenda:**

O processamento da encomenda é realizado no menu do estafeta quando este pretende verificar as encomendas associadas, o caminho e o transporte.

O programa pede ao estafeta para inserir o seu *ID* e após isso são exibidas as encomendas a si associadas. De seguida, para exibir o trajeto, o estafeta escolhe o algoritmo que pretende utilizar para procurar e exibir o melhor caminho.

Quando o estafeta seleciona o algoritmo pretendido, é exibido o caminho e apresentado num novo separador o grafo com o caminho a percorrer destacado a vermelho e com o nome da localização de origem e localização de destino.

- **Ver perfil:**

O estafeta pode visualizar no seu perfil as suas informações. Para isso o programa pede ao estafeta para este introduzir o seu *ID*, acede de seguida ao estado associado a esse mesmo *ID* e exibe todas as suas informações.

- **Ver Ranking de Estafetas com Mais Entregas:**

O estafeta pode ver os cinco estafetas com mais entregas efetuadas.

Para efetuar esta tarefa, implementámos uma função verifica os estafetas válidos, ou seja, com um número de entregas positivo, e ordena-os por ordem decrescente. Após isso obtém os cinco primeiros estafetas da lista e exibe, associada a cada um, o número de entregas realizadas.

2.3 Grafos

Relativamente aos grafos utilizados, optámos por recorrer a dois métodos cada um com diferentes fins e aplicações.

Através das aplicações que iremos abordar, pretendemos não só recorrer à aplicação dos conhecimentos obtidos através das aulas, mas também explorar outros métodos que sejam vantajosos para um projeto mais ótimo e completo.

- **Grafo (Inserção de Nós e Custos Manual):**

Numa primeira abordagem, de modo a testar os algoritmos utilizados, que vamos explorar já de seguida, decidimos implementar um grafo desenhado pelo grupo.

Para realizar esta tarefa escolhemos a Freguesia da Misericórdia, em Lisboa, e através da sua visualização num mapa escolhemos sete ruas. Quanto aos custos atribuídos decidimos fazer essa atribuição “a olho”.

Após esse processo, obtivemos este grafo:

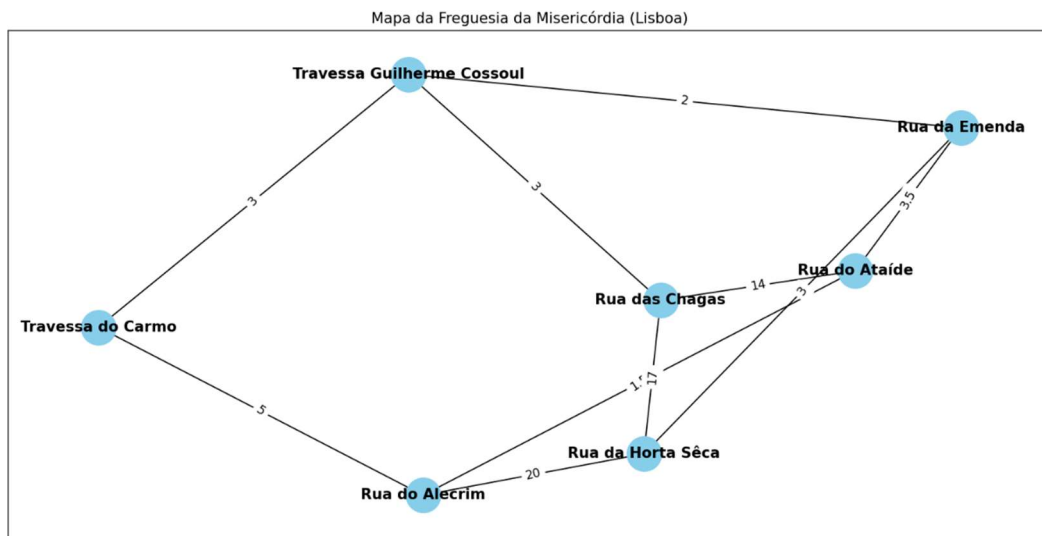


Figura 1 - Grafo Inicial da Freguesia da Misericórdia, Lisboa

Mesmo sendo este um método introdutório para a execução dos algoritmos implementados não nos sentimos satisfeitos nem com a sua aproximação à realidade, nem com o seu tamanho.

Por esse motivo, decidimos explorar um novo método, surgindo a ideia de recorrer à biblioteca *OSMnx*.

- **Grafo (OSMnx)**

Após explorarmos detalhadamente novos métodos de criação de grafos mais ótimos e os mais próximos possíveis à realidade, encontramos então a biblioteca *OSMnx*.

Esta biblioteca usa os dados do *OpenStreetMap* para auxiliar como uma ferramenta para a análise de redes urbanas. Dado isto, decidimos recorrer a este método para termos um grafo mais eficiente e realista, aplicando-o também, à Freguesia da Misericórdia, em Lisboa, sendo este o endereço a partir do qual o grafo é criado.

Esta biblioteca para além de nos permitir abranger a entrega de encomendas para toda a freguesia, e não apenas para algumas ruas, tem acesso a informações reais, como a distância real entre os nós.

Essas informações, são obtidas a partir das arestas, estas apresentam um identificador próprio e informações como a distância, nome das ruas, tipo de estrada, se é ou não de sentido único, entre outras informações. Por sua vez, os nós são identificados pela latitude e longitude, obrigando a algumas conversões no código como por exemplo para o cálculo da heurística.

Durante a execução dos algoritmos conseguimos no grafo, para além de exibir o grafo da freguesia, destacar também o seu caminho.

Após esse processo, obtivemos este grafo de toda freguesia:

Mapa da Freguesia da Misericórdia (Lisboa)



Figura 2 - Grafo OSMnx da Freguesia da Misericórdia, em Lisboa

3. Algoritmos Utilizados

De modo a encontrar o melhor caminho para efetuar as entregas das encomendas, recorreremos a alguns dos algoritmos lecionados, de modo a aprofundar os seus conhecimentos e visualizar de uma forma mais interativa o seu comportamento.

- **Procura Informada -> A* e Greedy:**

O algoritmo A* utiliza uma heurística em cada nó para estimar o custo total do caminho até o destino. Este algoritmo prioriza a exploração dos nós com base na soma do custo real na estimativa heurística do custo do nó até o destino. O A* visita os nós na ordem dessa avaliação feita a partir da soma, focando-se inicialmente nos nós que têm menores estimativas de custo total.

Quanto ao algoritmo Greedy, este procura a solução ideal através de escolhas locais consecutivas, visando assim atingir uma solução global ótima. No entanto, escolhas locais podem não levar a soluções ótimas, mas sim a soluções menos eficientes.

Estes algoritmos de procura informada utilizámos em ambos os grafos, tanto no OSMnx como no grafo que construímos.

No entanto, na procura do grafo OMNSnx, apenas recorreremos a estes algoritmos informados, pois estes são mais eficientes em grafos maiores devido à capacidade de fazer escolhas mais informadas sobre quais caminhos explorar.

Deste modo, a quantidade de exploração desnecessária é reduzida e o desempenho é mais ótimo.

Quanto à heurística utilizada, no nosso projeto decidimos recorrer à distância euclidiana.

Decidimos escolher esta heurística pois esta é admissível, ou seja, não superestima o custo para alcançar o destino e funciona bem em grande parte dos algoritmos de procura informada.

Aqui temos exemplos da procura A^* e *Greedy* no grafo *OSMnx*:

Mapa da Freguesia da Misericórdia (Lisboa) - Caminho de Travessa do Carmo para Rua da Esperança



Figura 3 - Procura A^* no Grafo *OSMnx*

Mapa da Freguesia da Misericórdia (Lisboa) - Caminho de Travessa do Carmo para Rua da Esperança

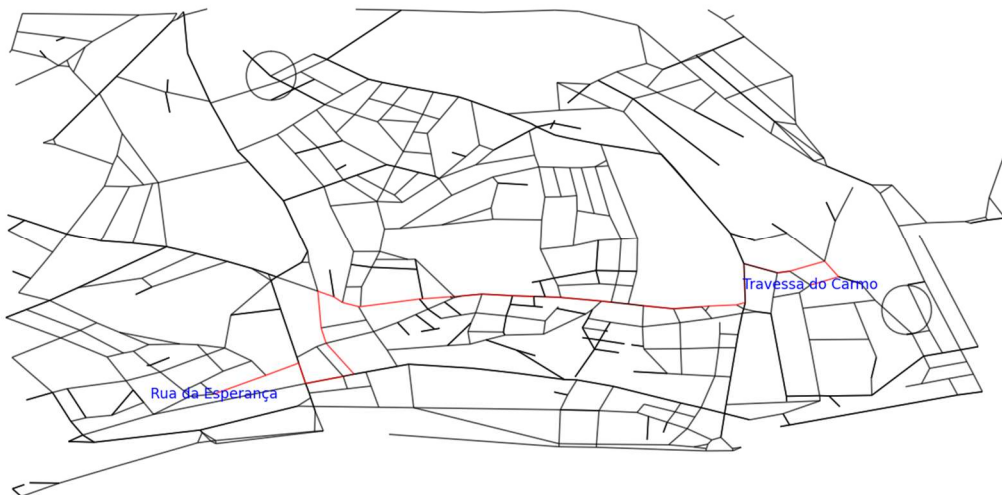


Figura 4 - Procura *Greedy* no Grafo *OSMnx*

Nestes algoritmos, como lidamos com uma maior área e dependendo da localização final, de um possível maior número de nós explorados, incluímos apenas quantos nós foram verificados e não o caminho parcial de modo a não tornar o *output* confuso.

- **Procura Não Informada -> Custo Uniforme, BFS, Dijkstra:**

O algoritmo de Custo Uniforme é um algoritmo que tem sempre em conta o menor custo acumulado, expandindo os seus nós tendo em conta esse critério e, garantindo assim encontrar o caminho mais curto possível.

Relativamente ao algoritmo *BFS*, este é um método de procura não informada, que realiza a procura por níveis e acaba mal encontre o nó final. No entanto não é um algoritmo benéfico para o nosso problema uma vez que nem leva em conta o custo das arestas. Para além disso, em grafos grandes iria consumir muita memória.

Quanto ao *Dijkstra*, este é um algoritmo idêntico ao *BFS*, no entanto é mais otimizado uma vez que tem em conta o peso das arestas.

Estes algoritmos de procura não informada utilizamos apenas no menor grafo, pois em grafos maiores, como o grafo *OSMnx*, estes métodos podem não ser tão eficientes. Isto acontece devido à necessidade de considerar todas as possíveis combinações, o que torna o processo de busca mais demorado e ineficiente à medida que o grafo aumenta em tamanho.

Deste modo, escolhemos apenas recorrer a estes algoritmos no menor grafo para aplicar os conceitos lecionados e interagir de forma mais interativa com os seus comportamentos.

Aqui temos o exemplo da procura não informada no grafo:

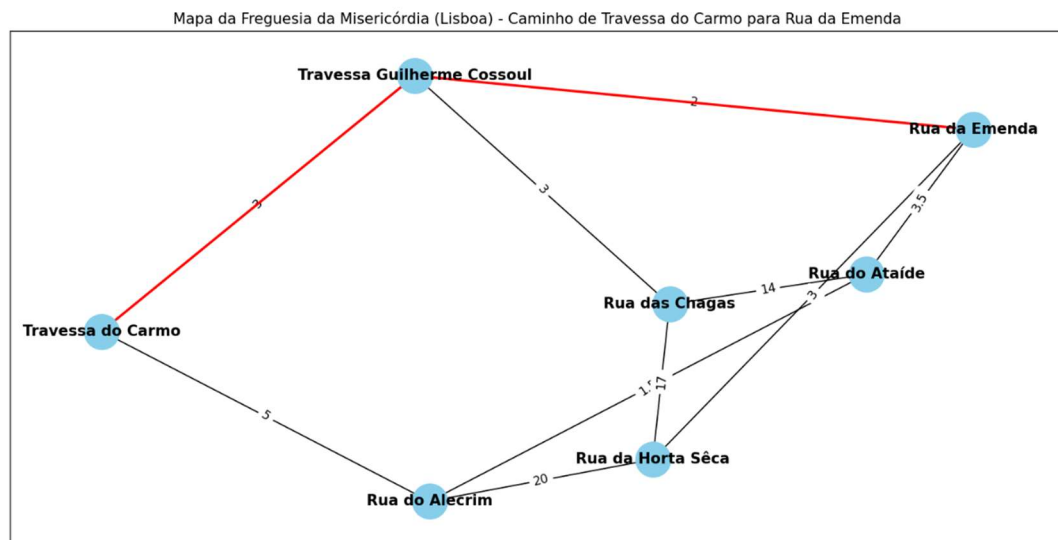


Figura 5 - Procura Custo Uniforme, BFS e Dijkstra

Nestes algoritmos, no *output* exibimos o caminho parcial que estes executam uma vez que se trata de caminhos e procuras menores. Apesar do caminho final ser maioria das vezes o mesmo, a partir do caminho parcial verifica-se que se desenvolvem de maneira diferente.

4. Análise de Resultados

A partir da análise de resultados, conseguimos verificar que quanto à procura não informada, na maioria dos casos, o caminho escolhido é o mesmo. Apesar disso, a sequência de expansão dos nós é diferente, tal como podemos verificar a partir dos caminhos parciais.

Assim, iremos demonstrar, para o caminho da Travessa do Carmo para a Rua da Emenda:

```
Caminho parcial: ['Travessa do Carmo', 'Rua do Alecrim']
Caminho parcial: ['Travessa do Carmo', 'Travessa Guilherme Cossoul']
Caminho parcial: ['Travessa do Carmo', 'Travessa Guilherme Cossoul', 'Rua da Emenda']

Caminho de Travessa do Carmo para Rua da Emenda:
['Travessa do Carmo', 'Travessa Guilherme Cossoul', 'Rua da Emenda']
```

Figura 6 - Procura Custo Uniforme

```
Caminho parcial: ['Travessa do Carmo']
Caminho parcial: ['Travessa do Carmo', 'Rua do Alecrim']
Caminho parcial: ['Travessa do Carmo', 'Travessa Guilherme Cossoul']
Caminho parcial: ['Travessa do Carmo', 'Rua do Alecrim', 'Rua da Horta Sêca']
Caminho parcial: ['Travessa do Carmo', 'Rua do Alecrim', 'Rua do Ataíde']
Caminho parcial: ['Travessa do Carmo', 'Travessa Guilherme Cossoul', 'Rua da Emenda']

Caminho de Travessa do Carmo para Rua da Emenda:
['Travessa do Carmo', 'Travessa Guilherme Cossoul', 'Rua da Emenda']
```

Figura 7 - Procura BSF

```
Caminho parcial: ['Travessa do Carmo', 'Rua do Alecrim']
Caminho parcial: ['Travessa do Carmo', 'Travessa Guilherme Cossoul']
Caminho parcial: ['Travessa do Carmo', 'Travessa Guilherme Cossoul', 'Rua da Emenda']
Caminho parcial: ['Travessa do Carmo', 'Travessa Guilherme Cossoul', 'Rua da Emenda']

Caminho de Travessa do Carmo para Rua da Emenda:
['Travessa do Carmo', 'Travessa Guilherme Cossoul', 'Rua da Emenda']
```

Figura 8 - Dijkstra

Estes foram os caminhos obtidos para este grafo:

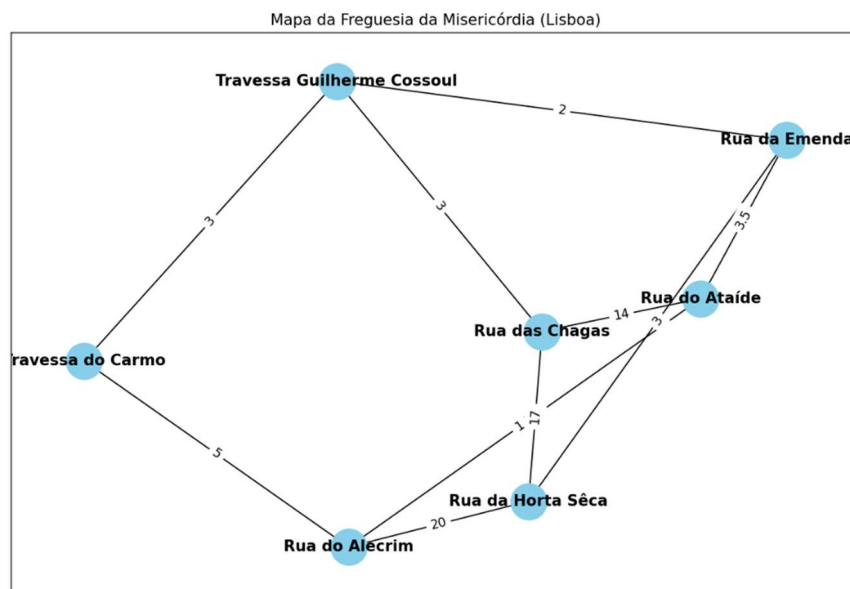


Figura 9 - Grafo Freguesia da Misericórdia (Lisboa)

Quanto aos algoritmos de procura informada, para esses métodos apesar de não termos a sequência dos nós explorados, isto é, o caminho parcial, decidimos introduzir o número de nós expandidos para cada um dos algoritmos.

Através da verificação para várias encomendas com diferentes destinos, conseguimos verificar que na o número de nós explorados é bastante relativo. Mesmo que não seja uma diferença significativa, dependendo do caminho, por vezes o A^* explora mais que a *Greedy*, noutros casos explora menos e até exploram, noutros casos, o mesmo número de nós.

Por esse motivo não conseguimos destacar nenhum comportamento em particular.

Assim, para exemplificar, iremos demonstrar para o mesmo caminho abordado acima, da Travessa do Carmo para a Rua da Emenda, o número de nós explorados:

```
Número de nós expandidos: 22
Caminho de Travessa do Carmo para Rua da Emenda:
['Travessa do Carmo', 'Largo do Carmo', 'Rua da Trindade', 'Rua Nova da Trindade', 'Largo do Chiado', 'Rua da Misericórdia', 'Largo do Barão de Quintela', 'Rua das Flores', 'Travessa de Guilherme Cossoul']
```

Figura 10 - Algoritmo A^*

```
Número de nós expandidos: 20
Caminho de Travessa do Carmo para Rua da Emenda:
['Travessa do Carmo', 'Largo do Carmo', 'Rua da Trindade', 'Rua Nova da Trindade', 'Largo do Chiado', 'Rua da Misericórdia', 'Largo do Barão de Quintela', 'Rua das Flores', 'Travessa de Guilherme Cossoul']
```

Figura 11 - Algoritmo *Greedy*

Através destes exemplos pretendemos demonstrar não só as funcionalidades, mas também o comportamento dos algoritmos perante os grafos implementados.

Neste exemplo de procura informada, apesar de o caminho final ser o mesmo para ambos os processos, em outras encomendas, para cada um dos algoritmos de procura, obtemos diferentes caminhos.

Conclusão

Quanto à realização deste projeto, consideramos que este foi um trabalho bastante interessante e, por esse mesmo motivo, despertou-nos um grande interesse e curiosidade no que toca a explorar novos métodos e meios de resolução dos problemas.

Durante o seu processo de realização encontramos algumas dificuldades, sendo uma delas relativa à heurística que devíamos utilizar.

No trabalho usamos a distância euclidiana como heurística, no entanto, como no grafo que recorre à biblioteca *OMSnx* os nós são identificados pela latitude e longitude, a heurística estava a dar valores extremamente pequenos, o que influenciava o resultado do custo dos algoritmos. No entanto, após uma pesquisa, concluimos que tínhamos de fazer uma conversão de modo a converter a latitude e longitude para metros de maneira a obter uma heurística mais fiável. Após essa conversão, efetuamos *prints* no código que nos permitiram concluir que a heurística tinha valores consistentes.

Outro desafio que tivemos foi quanto à criação de um grafo que abrange um maior número de ruas.

O grafo que construímos inicialmente era demasiado pequeno e para além dos custos serem menos realistas, mesmo a execução dos algoritmos não proporcionava um impacto tão grande como num grafo maior. Desse problema, surgiu-nos então a ideia de recorrer à biblioteca que mencionamos acima, de modo a ter um projeto mais completo e de certo modo, também mais interativo. Para além disso, recorrer a uma nova biblioteca permitiu-nos aumentar o nosso conhecimento, obrigando-nos a realizar pesquisas e encontrar novas informações.

De um modo geral, consideramos que apesar das dificuldades que encontramos conseguimos contornar e arranjar métodos e soluções que resolvessem os nossos problemas. Este foi um trabalho que para além de muito interessante, permitiu-nos também expandir o nosso conhecimento e evoluir a nossa dinâmica em trabalhos de grupo.