

Universidade do Minho
Escola de Engenharia

Inteligência Artificial

Relatório Trabalho Prático
Grupo 32

Licenciatura em Engenharia Informática
3.º Ano — 1.º Semestre

Realizado por:

A98695 Lucas Gabriel Ribeiro Oliveira
A89292 Mike Stephane Melendes Pinto
A96208 Rafael João Ferreira Gomes

Braga,
18 de fevereiro de 2025

Avaliação pelos pares:

A98695 Lucas Gabriel Ribeiro Oliveira: $\delta = 0.34$

A89292 Mike Stephane Melendes Pinto: $\delta = 0.33$

A96208 Rafael João Ferreira Gomes: $\delta = 0.33$

Conteúdo

1	Introdução	3
2	Descrição do Problema	3
3	Formulação do Problema	3
4	Arquitetura da Solução	4
5	Criação do Mapa (Grafo)	4
6	Execução do Programa	5
6.1	Inicialização do Dataset	5
6.2	Menu Inicial	5
6.3	Menu Cliente	6
6.4	Lista de Encomendas	7
6.5	Lista de Entregas	9
6.6	Menu Administrador	10
6.7	Lançamento de Encomendas Pendentes	13
7	Estratégias de Procura Utilizadas	14
7.1	Procura não informada	14
7.1.1	Procura em Largura (BFS)	14
7.1.2	Exemplo de uma Procura em Largura (BFS)	15
7.1.3	Procura em Profundidade (DFS)	16
7.1.4	Exemplo de Procura em Profundidade (DFS)	17
7.1.5	Procura do Custo Uniforme (Uniform)	18
7.1.6	Exemplo de Procura do Custo Uniforme (Uniform)	19
7.2	Procura Informada	20
7.2.1	Heurística	20
7.2.2	Procura A-estrela (A*)	21
7.2.3	Exemplo de procura A*	22
7.2.4	Procura Gulosa (Greedy)	23
7.2.5	Exemplo de Procura Greedy	24
8	Conclusão e Trabalho futuro	25

Lista de Figuras

1	Arquitetura proposta da solução.	4
2	Mapa utilizado pela <i>Health Planet</i> para realizar as suas entregas.	4
3	Menu inicial do utilizador.	5
4	Menu do Cliente.	6
5	Lista de encomendas de um cliente.	7
6	Percorso percorrido de uma entrega.	8
7	Informações impressas no terminal sobre a entrega.	8
8	Lista de entregas de um cliente.	9
9	Menu do Administrador.	10
10	Opções de Entregas Concluídas.	10
11	Ranking dos Estafetas Melhor Classificados.	11
12	Ranking dos estafetas mais ecológicos.	12
13	Menu Auxiliar	12
14	Entrega realizada com o Algoritmo BFS.	15
15	Informações da entrega realizada com o Algoritmo BFS.	15
16	Entrega realizada com o Algoritmo DFS.	17
17	Informações da entrega realizada com o Algoritmo DFS.	17
18	Entrega realizada com o Algoritmo Uniform.	19
19	Informações da entrega realizada com o Algoritmo Uniform.	19
20	Entrega realizada com o Algoritmo A*.	22
21	Informações da entrega realizada com o Algoritmo A*	22
22	Entrega realizada com o Algoritmo Greedy.	24
23	Informações da entrega realizada com o Algoritmo Greedy.	24

1 Introdução

Este trabalho foi realizado no contexto da unidade curricular de **Inteligência Artificial**, tendo sido proposto como objetivo a resolução de problemas através da formalização de problemas, implementação e conceção dos algoritmos de procura.

2 Descrição do Problema

Sendo a sustentabilidade, um dos tópicos mais relevantes da nossa sociedade, a *Health Planet*, uma empresa de distribuição de encomendas, procura uma forma mais ecológica e sustentável de fazer as entregas aos seus clientes. Os estafetas da empresa possuem ao seu dispor vários **tipos de transporte**, com vários **níveis de consumo de energia**. Esses estafetas possuem um conjunto de entregas a efetuar, na qual os clientes introduzem um tempo limite para a entrega da sua encomenda. Todas as encomendas podem ser avaliadas de 0 a 5 estrelas pelos próprios clientes. No final, o custo do serviço da entrega tem em conta a encomenda, o transporte utilizado e o seu prazo de entrega.

3 Formulação do Problema

O problema proposto encontra-se num **ambiente determinístico totalmente observável**, onde um estafeta sabe exatamente o estado em que estará. A solução do problema é uma sequência, sendo assim um problema de **estado único**.

- **Representação do Estado:** Grafo não orientado, onde cada nodo representa uma freguesia e cada aresta representa a distância entre duas freguesias.
- **Estado Inicial:** A posição inicial é representada pela localização do nodo da empresa (*Health Planet*), onde todos os estafetas partem.
- **Estado/Teste Objetivo:** Alcançar o nodo da freguesia de destino da encomenda do cliente.
- **Operadores:** Deslocação entre freguesias adjacentes.
- **Custo da Solução:** Distância percorrida na entrega realizada por um estafeta.

4 Arquitetura da Solução

Após a formulação e análise do problema, começamos por sugerir a seguinte arquitetura para a solução:

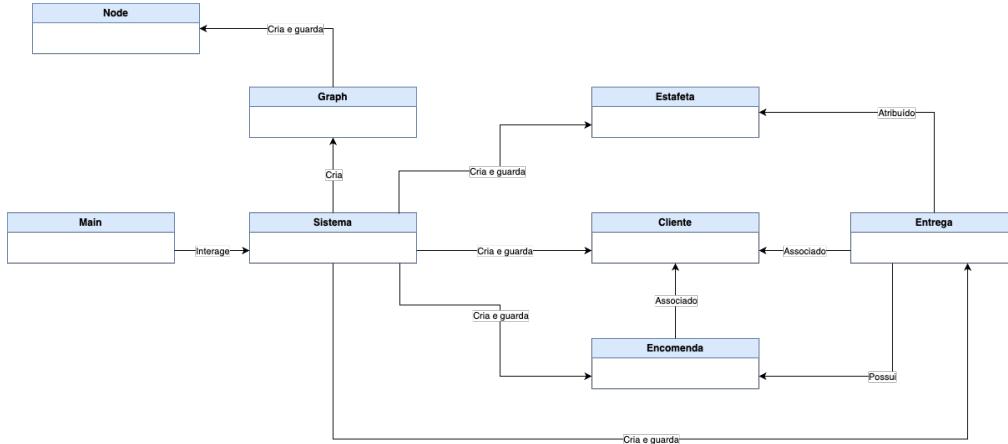


Figura 1: Arquitetura proposta da solução.

5 Criação do Mapa (Grafo)

O mapa é criado ao executar o sistema, sendo representado da seguinte maneira:

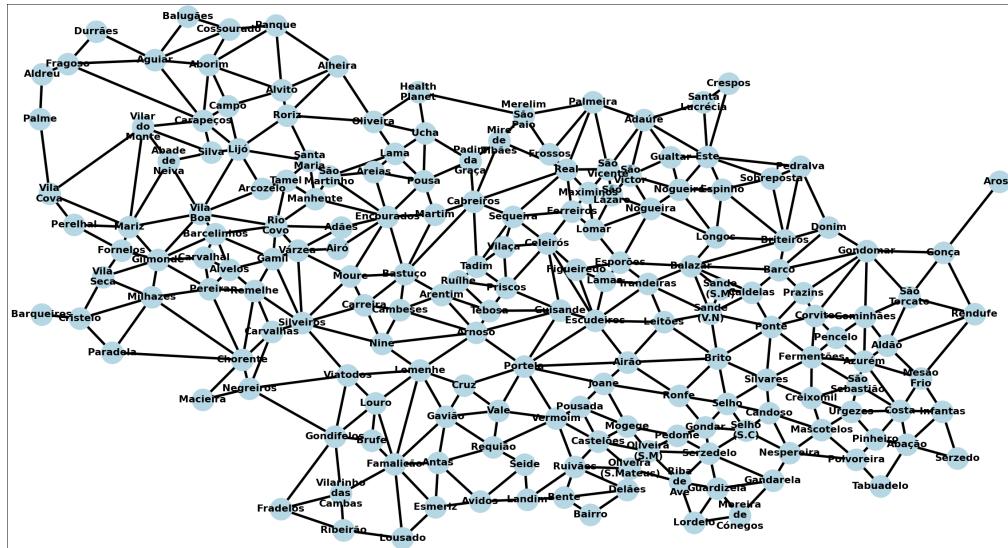


Figura 2: Mapa utilizado pela *Health Planet* para realizar as suas entregas.

O mapa na figura 2 representa os municípios de *Braga*, *Barcelos*, *Guimarães* e *Vila Nova de Famalicão*. Ele foi gerado com base nas coordenadas geográficas (latitude e longitude) de cada freguesia, onde as arestas refletem o custo da distância entre elas. Todas as informações dos nós são armazenadas na classe *Node*, enquanto a classe *Graph*, armazena as informações das arestas e os próprios nós. O ponto de origem de cada estafeta é o nó ***Health Planet***.

6 Execução do Programa

6.1 Inicialização do Dataset

Ao iniciar o sistema, para além de iniciar a criação do grafo, também carrega as informações guardadas em ficheiros de *Clientes*, *Encomendas*, *EncomendasPendentes*, *Veículos*, *Estafetas* e *Entregas*, e no caso de qualquer um dos ficheiros ou pasta não existir, estas são criadas automaticamente pelo sistema.

6.2 Menu Inicial

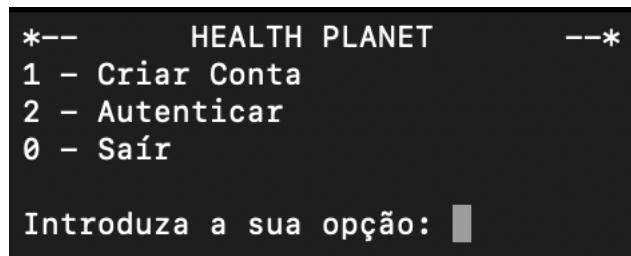


Figura 3: Menu inicial do utilizador.

Ao iniciar é apresentado o menu da figura 3 e tem as seguintes opções:

- **1 - Criar Conta:** O utilizador cria a sua conta, introduzindo o seu *Id*, *Nome*, *Freguesia* e *Password*.
- **2 - Autenticar:** O utilizador autentica-se, introduzindo o seu *Id* e *Password*.
- **0 - Sair:** O utilizador sai do programa.

6.3 Menu Cliente

```
*--      HEALTH PLANET      --*
1 - Ver Perfil
2 - Criar Encomenda
3 - Lista de Encomendas
4 - Lista de Entregas
0 - Voltar Atras

Introduza a sua opção-> █
```

Figura 4: Menu do Cliente.

Após autenticar-se, é apresentado o menu da figura 4 e tem as seguintes opções:

- **1 - Ver Perfil:** O utilizador visualiza as suas informações.
- **2 - Criar Encomenda:** O utilizador cria uma encomenda, introduzindo o seu *Peso, Volume e Prazo de Entrega*.
- **3 - Lista de Encomendas:** Apresenta uma lista de encomendas que o utilizador possui.
- **4 - Lista de Entregas:** Apresenta uma lista de entregas que o utilizador possui.
- **0 - Voltar Atrás:** O utilizador volta para o menu inicial.

6.4 Lista de Encomendas

```
*-- HEALTH PLANET --*
Id do Cliente: maria85; Id da Encomenda: 2; Peso: 14.3; Volume: 21.45; Prazo de Entrega: 7.9; Estado: 0
Id do Cliente: maria85; Id da Encomenda: 75; Peso: 26.0; Volume: 39.0; Prazo de Entrega: 5.2; Estado: 0
Id do Cliente: maria85; Id da Encomenda: 107; Peso: 25.9; Volume: 38.85; Prazo de Entrega: 7.6; Estado: 0
Id do Cliente: maria85; Id da Encomenda: 131; Peso: 15.0; Volume: 5.0; Prazo de Entrega: 6.0; Estado: 1
Deseja escolher alguma encomenda para entrega?
1 - SIM
2 - NÃO
Introduza a sua opção: ■
```

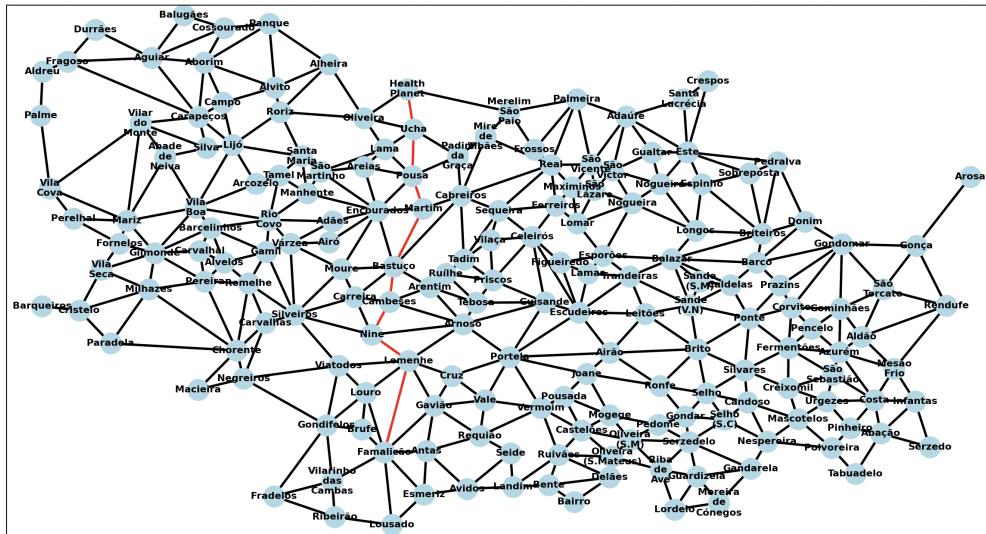
Figura 5: Lista de encomendas de um cliente.

Um utilizador ao selecionar a opção de listar as suas encomendas, é apresentado a lista de todas as suas encomendas com as suas informações e também com o seu estado, se for 0 então a encomenda foi entregue e caso for 1 significa que a encomenda ainda não foi entregue, tal como mostrado na figura 5. Além disso, é solicitado ao utilizador se este pretende escolher alguma encomenda para entregar, tendo as seguintes opções:

- **1 - Sim:** É questionado ao utilizador qual a urgência da entrega das encomendas, sendo apresentado duas opções:
 - **1 - Imediato:** O utilizador introduz os *ids* das encomendas que deseja, e a seguir é questionado se deseja escolher o algoritmo de procura:
 - * **1 - Sim:** São lhe apresentados 3 algoritmos de procura não informada e 2 algoritmos de procura informada, sendo elas as seguintes:
 - 1 - DFS
 - 2 - BFS
 - 3 - Uniform
 - 4 - A*
 - 5 - Greedy
 - * **2 - Não:** O sistema calcula o melhor caminho usando todos os algoritmos de procura disponíveis, escolhendo o que tiver menor custo.

No final é apresentado um desenho do grafo com o caminho percorrido, como podemos constatar, na figura 6. Após fechar o desenho, é apresentado ao utilizador toda a informação relacionada sobre a entrega como, por exemplo, a ordem de expansão dos nós do algoritmo utilizado, como demonstrado na figura 7:

Entrega Realizada



[Fechar](#)

Figura 6: Percurso percorrido de uma entrega.

Ordem de Expansão: ['Arentim', 'Louro', 'Carreira', 'Airo', 'Encourados', 'Tamel', 'Merelim São Paio', 'Frossos', 'Adães', 'Health Planet', 'Bastuço', 'Alheira', 'Várzea', 'Areias', 'Martin', 'Moura', 'Sequeria', 'Padim da Graca', 'Priscos', 'Nine', 'Tadim', 'Maximinos', 'Ucha', 'Vilaça', 'Oliveira', 'Mire de Tibães', 'Arnoso', 'Cabreiros', 'Santa Maria', 'Manhente', 'Pousa', 'Lama', 'Lomenhe', 'Cambezes', 'São Martinho', 'Rio Covo', 'Ferreiros', 'Real']
 Entrega da encomenda: 132 foi entregue com sucesso!
 Id da Entrega: 132; Id do Cliente: vito03; Id do Estafeta: Jmpe; Classificação: None; Preço: 36.48€ ; Veículo: Moto; Caminho: [Health Planet, Ucha, Pousa, Martin, Bastuço, Cambezes, Nine, Lomenhe, Famalicão]; Distância: 33.7KM ; Tempo Previsto: 1h17m44s; Id da Encomenda: 132; Algoritmo: A*
 prima enter para continuar

Figura 7: Informações impressas no terminal sobre a entrega.

- **2 - Não Imediato:** O utilizador introduz os *ids* das encomendas que deseja e estas são guardadas numa lista de encomendas pendentes no sistema.
- **2 - Não:** O utilizador volta para o menu do cliente.

6.5 Lista de Entregas

```

<-- HEALTH PLANET -->
Id da Entrega: 31; Id do Cliente: rita93; Id do Estafeta: francisco; Classificação: None★ ; Preço: 13.16€ ; Veículo: Bicicleta; Caminho: ['Health Planet', 'Ucha', 'Pousa', 'Martim', 'Bastuço', 'Cambeses', 'Nine', 'Leme', 'Leme', 'Famalicense']; Distância: 33.7KM ; Tempo Previsto: 4H:29m:46s; Id da Encomenda: 29; Algoritmo: A*
Id da Entrega: 32; Id do Cliente: rita93; Id do Estafeta: alex; Classificação: None★ ; Preço: 26.69€ ; Veículo: Mota; Caminho: ['Health Planet', 'Ucha', 'Pousa', 'Martim', 'Bastuço', 'Cambeses', 'Nine', 'Leme', 'Leme', 'Famalicense']; Distância: 33.7KM ; Tempo Previsto: 1H:53m:31s; Id da Encomenda: 131; Algoritmo: A*
Id da Entrega: 42; Id do Cliente: rita93; Id do Estafeta: ines; Classificação: None★ ; Preço: 26.38€ ; Veículo: Mota; Caminho: ['Health Planet', 'Ucha', 'Pousa', 'Martim', 'Bastuço', 'Cambeses', 'Nine', 'Leme', 'Leme', 'Famalicense']; Distância: 33.7KM ; Tempo Previsto: 1H:53m:31s; Id da Encomenda: 131; Algoritmo: A*
Id da Entrega: 43; Id do Cliente: rita93; Id do Estafeta: ines; Classificação: None★ ; Preço: 26.69€ ; Veículo: Mota; Caminho: ['Health Planet', 'Ucha', 'Pousa', 'Martim', 'Bastuço', 'Cambeses', 'Nine', 'Leme', 'Leme', 'Famalicense']; Distância: 33.7KM ; Tempo Previsto: 1H:57m:46s; Id da Encomenda: 132; Algoritmo: A*
Id da Entrega: 44; Id do Cliente: rita93; Id do Estafeta: andré; Classificação: None★ ; Preço: 26.69€ ; Veículo: Mota; Caminho: ['Health Planet', 'Ucha', 'Pousa', 'Martim', 'Bastuço', 'Cambeses', 'Nine', 'Leme', 'Leme', 'Famalicense']; Distância: 33.7KM ; Tempo Previsto: 1H:57m:46s; Id da Encomenda: 133; Algoritmo: A*
Deseja classificar alguma Entrega?
1 - SIM
2 - NÃO
Introduza a sua opção: 

```

Figura 8: Lista de entregas de um cliente.

Quando um utilizador seleciona visualizar as entregas das suas encomendas, é apresentado a lista de todas as entregas das suas encomendas já realizadas, com toda a informação de cada entrega, tal como apresentado na figura 8. É solicitado também, ao utilizador, se este deseja classificar alguma entrega, dando as seguintes opções:

- **1 - Sim:** O utilizador introduz o *id* da entrega que deseja classificar, dando uma avaliação entre 0 a 5 estrelas.
- **2 - Não:** O utilizador volta para o menu do cliente.

6.6 Menu Administrador

Para aceder ao menu do administrador, introduzimos o **id** e a **password** de um administrador, apresentando o menu da figura 9:

```
*--      HEALTH PLANET      --*
1 - Ver Clientes registados no Sistema
2 - Ver Estafetas registados no Sistema
3 - Ver Veiculos registados no Sistema
4 - Adicionar Estafeta
5 - Adicionar Veículo
6 - Ver Encomendas do Sistema
7 - Ver Entregas Concluídas do Sistema
8 - Lançar Entregas Pendentes
9 - Ranking dos Estafetas Classificação
10 - Ranking dos Estafetas Ecológicos
11 - Menu Auxiliar
0 - Voltar Atrás

Introduza a sua opção-> █
```

Figura 9: Menu do Administrador.

São apresentadas as seguintes opções:

- **1 - Ver Clientes no Sistema:** apresenta uma lista com toda a informação dos clientes registados.
- **2 - Ver Estafetas no Sistema:** apresenta uma lista com toda a informação dos estafetas registados.
- **3 - Ver Veículos no Sistema:** apresenta uma lista com a quantidade de cada tipo de veículo registrado no sistema.
- **4 - Adicionar Estafeta:** adiciona um estafeta ao sistema, introduzindo o *id* e o nome.
- **5 - Adicionar Veículo:** adiciona um veículo ao sistema, escolhendo entre bicicleta, mota ou carro.
- **6 - Ver Encomendas no Sistema:** o administrador é questionado qual o cliente que pretende ver as encomendas, no qual é apresentado uma lista de todos os clientes que possuem encomendas, após selecionar um, é apresentado uma lista das encomendas do cliente selecionado.
- **7 - Ver Entregas Concluídas no Sistema:** é apresentado ao administrador as seguintes opções, como mostra a figura 10:

```
*--      HEALTH PLANET      --*
1 - Visualizar Entregas de um Cliente
2 - Visualizar Entregas de um Estafeta
0 - Voltar Atrás

Introduza a sua opção -> █
```

Figura 10: Opções de Entregas Concluídas.

- **1 - Visualizar Entregas de Clientes:** apresenta uma lista de clientes com entregas realizadas, após selecionar um, é apresentado todas as entregas desse cliente que foram realizadas.
- **2 - Visualizar Entregas de Estafetas:** apresenta uma lista de estafetas com entregas realizadas, após selecionar um, é apresentado todas as entregas realizadas por esse estafeta.
- **0 - 0 - Voltar Atrás:** O administrador volta para o menu inicial.
- **8 - Lançar Encomendas Pendentes:** O administrador envia todas as encomendas guardadas na lista de encomendas pendentes no sistema.
- **9 - Ranking dos Estafetas por Classificação:** apresenta uma lista de todos os estafetas com entregas classificadas, ordenados da melhor média de classificação até à pior, em caso de empate, é ordenado pelo seu *Id* como, por exemplo, na figura 11:

```
*--      HEALTH PLANET      --*
RANKING:
1 -> hugo, 5.0★
2 -> tiago, 5.0★
3 -> ricardo, 4.8★
4 -> diogo, 4.5★
5 -> francisco, 3.8★
6 -> mariana, 2.85★
7 -> rui, 2.3★
prima enter para continuar
```

Figura 11: Ranking dos Estafetas Melhor Classificados.

- **10 - Ranking dos Estafetas Ecológicos:** apresenta uma lista de todos os estafetas com entregas realizadas, ordenados pelo tipo de veículos usados nas entregas, em que para fazer o *ranking*, consideramos que uso de uma bicicleta tem um custo 0, uma mota um custo 1 e um carro tem um custo 2, e então percorremos todas as entregas dos estafetas, somamos a quantidade de cada tipo de veículo usado e no final fazemos uma soma da *quantidade de cada tipo de veículo * o custo desse veículo*, sendo o que tiver menor valor fica melhor classificado e o que tiver o valor maior, fica pior classificado, em caso de empate, é ordenado pelo nome como, por exemplo, na figura 12:

```

*--      HEALTH PLANET      --*
Ranking Estafetas Ecológicos

1 -> carolina | Veiculos: Bicicletas: 0 | Motas: 1 | Carros: 0
2 -> diogo | Veiculos: Bicicletas: 3 | Motas: 1 | Carros: 0
3 -> sara | Veiculos: Bicicletas: 1 | Motas: 1 | Carros: 0
4 -> alex | Veiculos: Bicicletas: 0 | Motas: 2 | Carros: 0
5 -> francisco | Veiculos: Bicicletas: 1 | Motas: 2 | Carros: 0
6 -> ricardo | Veiculos: Bicicletas: 0 | Motas: 2 | Carros: 0
7 -> andre | Veiculos: Bicicletas: 0 | Motas: 3 | Carros: 0
8 -> ines | Veiculos: Bicicletas: 1 | Motas: 3 | Carros: 0
9 -> tiago | Veiculos: Bicicletas: 0 | Motas: 3 | Carros: 0
10 -> hugo | Veiculos: Bicicletas: 0 | Motas: 4 | Carros: 0
11 -> beatriz | Veiculos: Bicicletas: 0 | Motas: 1 | Carros: 2
12 -> mariana | Veiculos: Bicicletas: 0 | Motas: 1 | Carros: 3
13 -> marta | Veiculos: Bicicletas: 0 | Motas: 1 | Carros: 3
14 -> rui | Veiculos: Bicicletas: 0 | Motas: 0 | Carros: 5

prima enter para continuar

```

Figura 12: Ranking dos estafetas mais ecológicos.

- **11 - Menu Auxiliar:** é apresentado ao administrador uma série de funcionalidades bónus, como mostra a figura 13.

```

*--      HEALTH PLANET      --*
1 - Desenhar Grafo
2 - Algoritmos
3 - Heuristica
0 - Voltar Atras
introduza a sua opcao-> 

```

Figura 13: Menu Auxiliar

- **1 - Desenhar Grafo:** desenha o mapa de todo o grafo.
- **2 - Algoritmos:** possibilita aplicar um algoritmo à escolha do administrador entre uma freguesia origem e destino.
- **3 - Heurística:** desenha o mapa de todo grafo com as heurísticas atribuídas.
- **0 - Voltar Atrás:** O administrador volta para o menu administrador.
- **0 - Voltar atrás:** O administrador volta para o menu inicial

6.7 Lançamento de Encomendas Pendentes

Para implementar esta funcionalidade, inicialmente reorganizamos a lista de encomendas pendentes no sistema com base na heurística de ordem crescente. Isso implica que ao calcular a heurística no nó de origem (Health Planet), as encomendas com valores menores estarão mais próximas da base. Essa abordagem facilita a escolha de veículos mais ecológicos, como bicicletas, uma vez que as opções mais sustentáveis tendem a ser mais acessíveis quando as encomendas estão organizadas dessa maneira.

Após essa reorganização, percorremos a lista e adicionamos cada encomenda a uma lista de entregas. Em seguida, com base no peso e volume, selecionamos o veículo mais adequado para cada entrega. Posteriormente, reorganizamos a lista de entregas com base nos prazos de entrega das encomendas, priorizando as com prazos mais curtos. Além disso, agrupamos as encomendas de um mesmo cliente em listas, garantindo que cada posição na lista de entregas contém encomendas ordenadas para atender ao cliente com o prazo mais curto.

Seguindo isso, realizamos uma verificação para cada encomenda na lista, determinando se é possível realizar a entrega com o peso atual da lista de encomendas no momento da entrega. As encomendas que podem ser entregues permanecem na lista de entregas, enquanto as demais são transferidas para uma lista de encomendas removidas.

Em seguida, recalculamos o tempo de entrega para cada encomenda na lista de entregas (aqueelas que não foram removidas), fornecendo o tempo exato que o estafeta levará para cada entrega. Posteriormente, tentamos adicionar as encomendas da lista de removidas, verificando novamente a viabilidade com base no peso e volume. As encomendas possíveis de serem entregues são adicionadas à lista de entregas finais, enquanto aquelas que não são viáveis permanecem na lista de encomendas pendentes no sistema.

Por fim, ao termos a lista final de entregas para um estafeta, selecionamos um estafeta aleatoriamente e atribuímos a ele a lista de entregas final. Repetimos esse processo até que a lista de encomendas pendentes esteja vazia, ou não haja estafetas disponíveis, ou não haja veículos disponíveis.

7 Estratégias de Procura Utilizadas

7.1 Procura não informada

As estratégias de procura não informada implementadas neste projeto foram **Procura em Largura (BFS)**, **Procura em Profundidade (DFS)** e a **Procura do Custo Uniforme (Uniform)**.

7.1.1 Procura em Largura (BFS)

O Algoritmo *BFS* é um método de procura em grafos que explora todos os vértices do grafo a partir de uma origem específica, visitando primeiro todos os vizinhos antes de avançar para os vértices mais distantes, ou seja, realiza a sua busca por níveis de camadas no grafo.

- **Vantagens:**

- Eficiente em grafos não pesados.
- Encontra a solução ótima, caso o custo de cada aresta seja 1.

- **Desvantagens:**

- Pode consumir uma quantidade significativa de memória, especialmente em grafos grandes, devido à necessidade de armazenar todos os vértices na fila.
- Pouco eficiente em grafos densos.
- Não adequado em grafos pesados.

Os critérios utilizados para avaliar este algoritmo foram:

- **Completude:** Se existir um caminho entre o vértice de origem e o vértice destino, ele encontrará um dos caminhos mais curto.
- **Otimização:** Não encontra a melhor solução, como podemos ver na figura 14, pois não é um bom algoritmo quando se trata de grafos com peso por não considerar o peso das arestas.
- **Complexidade no Tempo:** $O(b^d)$
- **Complexidade no Espaço:** $O(b^d)$

Onde:

- **b**: o máximo fator de ramificação (o número máximo de sucessores de um nó) da árvore de procura.
- **d**: a profundidade da melhor solução.

7.1.2 Exemplo de uma Procura em Largura (BFS)

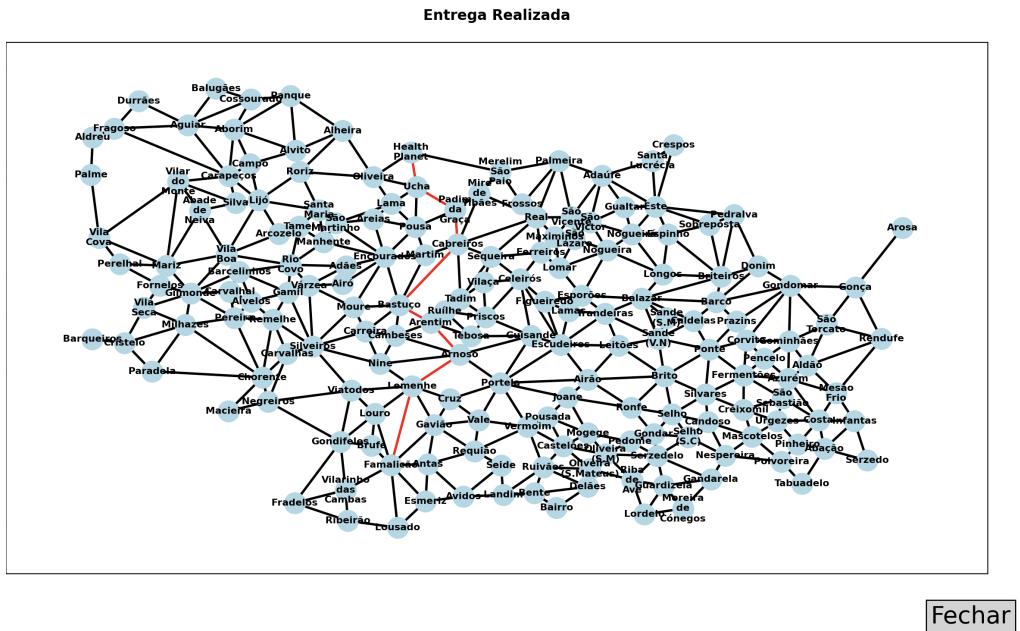


Figura 14: Entrega realizada com o Algoritmo BFS.

Figura 15: Informações da entrega realizada com o Algoritmo BFS.

Como podemos ver na figura 15, o Algoritmo de Procura em Largura obteve um caminho curto, mas não o ideal para o nosso tipo de problema, como veremos em outros algoritmos, e como podemos ver pela ordem de expansão, percorreu todos aqueles nodos até encontrar esta solução, levando um tempo considerável a executar comparado a outros algoritmos. Por fim, é um algoritmo bom para encontrar soluções boas, mas não para encontrar a melhor solução neste tipo de problema.

7.1.3 Procura em Profundidade (DFS)

O Algoritmo *DFS* é um método de procura em grafos que explora o máximo possível ao longo de cada ramificação antes de retroceder. Ele percorre um ramo profundamente antes de retornar para explorar outros ramos.

- **Vantagens:**

- Pode ser mais eficiente que o *BFS* em grafos pesados.
- Pouco espaço de memória utilizado.

- **Desvantagens:**

- Não adequado para encontrar o menor caminho.
- Dependência na ordem que visita.

Os critérios utilizados para avaliar este algoritmo foram:

- **Completude:** Não, falha em espaços de profundidade infinita, com repetições.
- **Otimização:** Não, como podemos ver na figura 16 pois devolve a 1^a solução que encontra.
- **Complexidade no Tempo:** $O(b^m)$
- **Complexidade no Espaço:** $O(bm)$

Onde:

- **b:** o máximo fator de ramificação (o número máximo de sucessores de um nó) da árvore de procura.
- **m:** a máxima profundidade do espaço de estados

7.1.4 Exemplo de Procura em Profundidade (DFS)

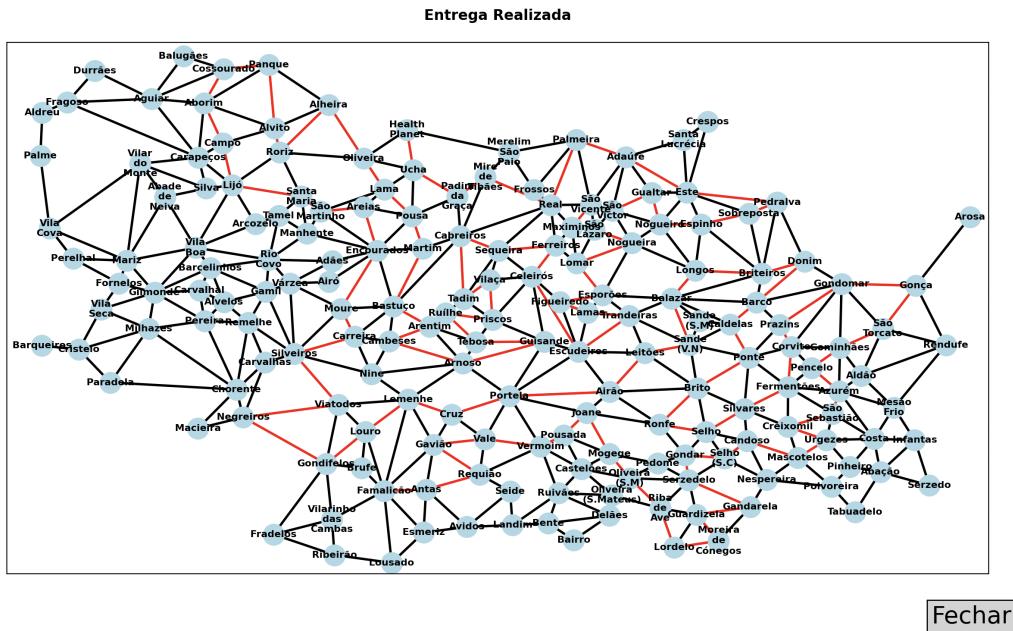


Figura 16: Entrega realizada com o Algoritmo DFS.

Figura 17: Informações da entrega realizada com o Algoritmo DFS.

Como podemos ver na figura 17, apesar deste algoritmo utilizar menos memória que o *BFS*, acaba por ser pior para o nosso tipo de problema por devolver a primeira solução que encontra e não a solução com o caminho mais curto.

7.1.5 Procura do Custo Uniforme (Uniform)

O Algoritmo *Uniform* é um método de procura em grafos, que expande o nó de menor custo até encontrar o objetivo desejado. Ao contrário do *BFS* e do *DFS*, este considera o custo associado a cada aresta, permitindo encontrar o caminho de menor custo em grafos pesados.

- **Vantagens:**

- Ótimo para grafos pesados.

- **Desvantagens:**

- Tempo de execução, pode ser mais lento em comparação com algoritmos menos informados.

Os critérios utilizados para avaliar este algoritmo foram:

- **Completude:** Sim, se o custo da etapa for maior que alguma constante positiva.
- **Otimização:** Sim, como podemos ver na figura 18.
- **Complexidade no Tempo:** $O(b^{(C^*/\epsilon)})$
- **Complexidade no Espaço:** $O(b^{(C^*/\epsilon)})$

Onde:

- **b**: o máximo fator de ramificação (o número máximo de sucessores de um nó) da árvore de procura.
 - C^* : custo da solução ideal.
 - ϵ : constante positiva.

7.1.6 Exemplo de Procura do Custo Uniforme (Uniform)

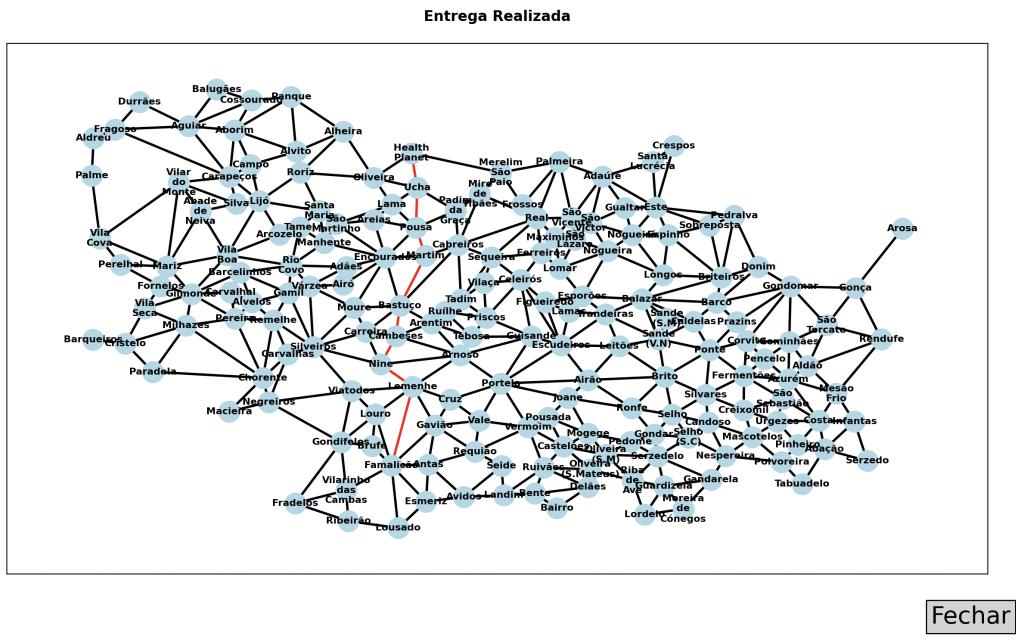


Figura 18: Entrega realizada com o Algoritmo Uniform.

Figura 19: Informações da entrega realizada com o Algoritmo Uniform.

Como podemos ver na figura 19, o Algoritmo de Uniform obteve um caminho ótimo, o que é a melhor circunstância possível para este problema, uma vez que nos encontramos num grafo pesado e ele encontra sempre o caminho mais curto para o destino desejado.

7.2 Procura Informada

As estratégias de procura informada implementadas neste projeto foram **Procura A-estrela (A*)** e a **Procura Gulosa (Greedy)**

7.2.1 Heurística

Para este tipo de problema e para o desenvolvimento dos métodos de procura informada, optamos por usar a heurística estática da distância em linha reta entre os nodos do grafo e os nodos finais. As heurísticas para todos os nodos são calculadas a cada criação de uma entrega e não são alteradas durante o percurso do estafeta no grafo. A distância em linha reta entre dois nós é calculado usando as coordenadas guardadas de cada nó e com isso usamos a **Fórmula de Harsevine** da seguinte maneira:

Nodo atual: (lat1,long1)
Nodo final: (lat2,long2)

Começamos por converter ambas as coordenadas para radianos, e de seguida calculamos a diferença:

$$\begin{aligned} \text{dlat} &= \text{lat2} - \text{lat1} \\ \text{dlong} &= \text{long2} - \text{long1} \end{aligned}$$

Aplicamos a **Fórmula de Harsevine**:

$$R = 6371.0$$

$$\begin{aligned} a &= \sin^2\left(\frac{\text{dlat}}{2}\right) + \cos(\text{lat1}) \cdot \cos(\text{lat2}) \cdot \sin^2\left(\frac{\text{dlong}}{2}\right) \\ c &= 2 \cdot \text{atan2}\left(\sqrt{a}, \sqrt{1-a}\right) \end{aligned}$$

$$h(n) = R \cdot c$$

Com isto, obtemos a distância real em linha reta entre dois nós. Com esta heurística, permitiu-nos testar e implementar as estratégias de Procura Informada que implementamos.

7.2.2 Procura A-estrela (A^*)

O Algoritmo A^* é um método de procura em grafos, que evita expandir os caminhos dispendiosos, usando a combinação dos benefícios dos Algoritmos da Procura Gulosa e do Algoritmo de Procura do Custo Uniforme, sendo amplamente usado para encontrar o caminho de menor custo entre um ponto de origem e um ponto de destino.

Utiliza esta função para escolher o próximo nodo a ser explorado:

$$f(n) = g(n) + h(n)$$

Onde:

- **$g(n)$** = custo total, até ao momento, para chegar ao estado n (custo do percurso).
- **$h(n)$** = custo estimado para chegar ao objetivo (não deve sobreestimar o custo para chegar à solução (heurística)).
- **$f(n)$** = custo estimado da solução menos dispendiosa que passa pelo nó n.
- **Vantagens:**
 - Eficiência em encontrar o caminho de menor custo, especialmente em grafos grandes.
 - Adaptabilidade.
- **Desvantagens:**
 - Requer cuidado na heurística.
 - Espaço de memória.

Os critérios utilizados para avaliar este algoritmo foram:

- **Completude:** Sim.
- **Otimização:** Sim, como podemos ver na figura 20.
- **Complexidade no Tempo:** $O(b^d)$
- **Complexidade no Espaço:** $O(b^d)$

Onde:

- **b:** o máximo fator de ramificação (o número máximo de sucessores de um nó) da árvore de procura.
- **d:** a profundidade da solução mais curta.

7.2.3 Exemplo de procura A*

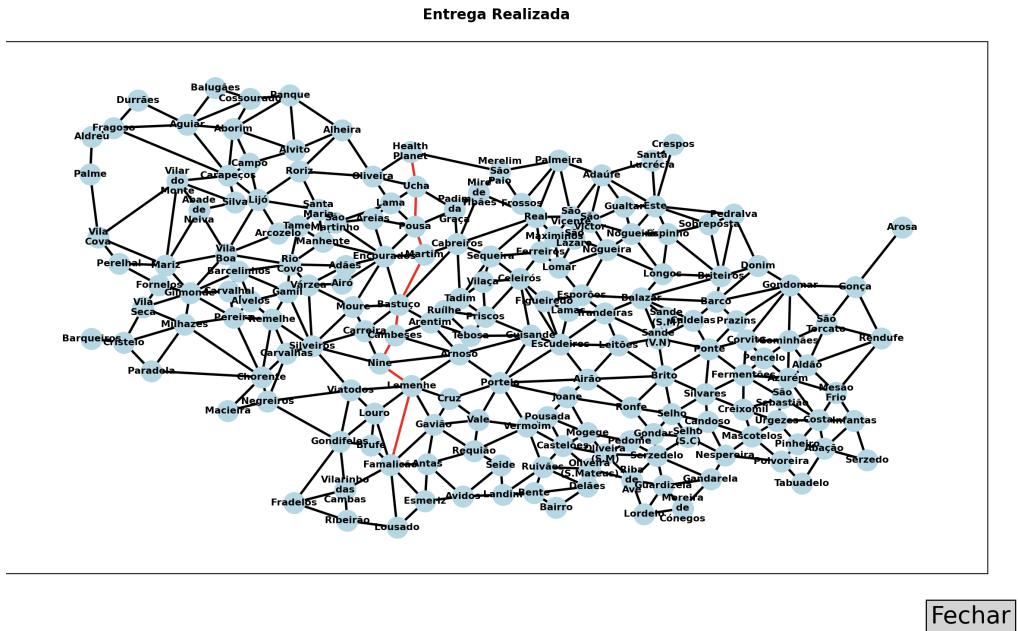


Figura 20: Entrega realizada com o Algoritmo A*.

```
Ordem de Expansão: ['Sequeira', 'Moure', 'Merelim São João', 'Aizó', 'Maximinos', 'Martim', 'Health Planet', 'Vilaça', 'Ferreiros', 'Manhente', 'Mire de Tibães', 'Areias', 'Frossos', 'Real', 'Várzea', 'Tamei', 'Louro', 'Arentim', 'Encourados', 'Cambeses', 'Lemehe', 'São Martinho', 'Arnos', 'Lama', 'Ucha', 'Santa Maria', 'Priscos', 'Carreira', 'Rio Covo', 'Padim da Graca', 'Tadim', 'Cabeiros', 'Pousa', 'Ades', 'Bastuço', 'Oliveira', 'Nine', 'Alheira']
Entrega de encomenda: 137 foi entregue com sucesso!
Id da Entrega: 51; Id do Cliente: ritu93; Id do Estafeta: alex; Classificação: None★ ; Preço: 25.09€ ; Veículo: Moto; Caminho: ['Health Planet', 'Ucha', 'Pousa', 'Martim', 'Bastuço', 'Cambeses', 'Nine', 'Lemehe', 'Famalicão', 'Antas', 'Esmerto', 'Avôs', 'Landim', 'Pente', 'Bairro', 'Guardizela', 'Mareixa', 'Lordelo', 'Côegos', 'Tabuadelo', 'Sarzedo', 'Pinhel', 'Faria', 'Gondar', 'Selha', 'Lameiro', 'Castelouço', 'Livra', 'Mogre', 'Pouzona', 'Gondaseijo', 'S.C.', 'Serzedelo', 'Nespera', 'Povoreira', 'Tabuadelo']
prima enter para continuar
```

Figura 21: Informações da entrega realizada com o Algoritmo A*

Como podemos ver na figura 21, o algoritmo obteve a solução ótima, tal como o Algoritmo do Custo Uniforme, sendo a melhor circunstância possível para este problema, sendo que estamos num grafo grande e pesado, ele encontra sempre o caminho mais curto para o destino desejado, acabando sempre por ser o utilizado no programa para escolher o caminho que o estafeta deve percorrer.

7.2.4 Procura Gulosa (Greedy)

O Algoritmo *Greedy* é um método de procura em grafos, que toma decisões com base apenas na heurística, escolhendo assim o seu próximo passo na qual pareça ser mais promissora, normalmente com o objetivo de maximizar algum critério.

- **Vantagens:**

- Eficiência local, por poder ser eficaz quando a busca é guiada por uma heurística bem ajustada para o problema em questão.
- Simplicidade

- **Desvantagens:**

- A solução pode não ser a solução óptima.

Os critérios utilizados para avaliar este algoritmo foram:

- **Completude:** Não, pois não garante encontrar uma solução, mesmo que ela exista
- **Otimização:** Não, como podemos ver na figura 22.
- **Complexidade no Tempo:** $O(b^m)$, podendo diminuir significativamente com uma boa heurística.
- **Complexidade no Espaço:** $O(b^m)$

Onde:

- **b:** o máximo fator de ramificação (o número máximo de sucessores de um nó) da árvore de procura.
- **m:** a máxima profundidade do espaço de estados

7.2.5 Exemplo de Procura Greedy

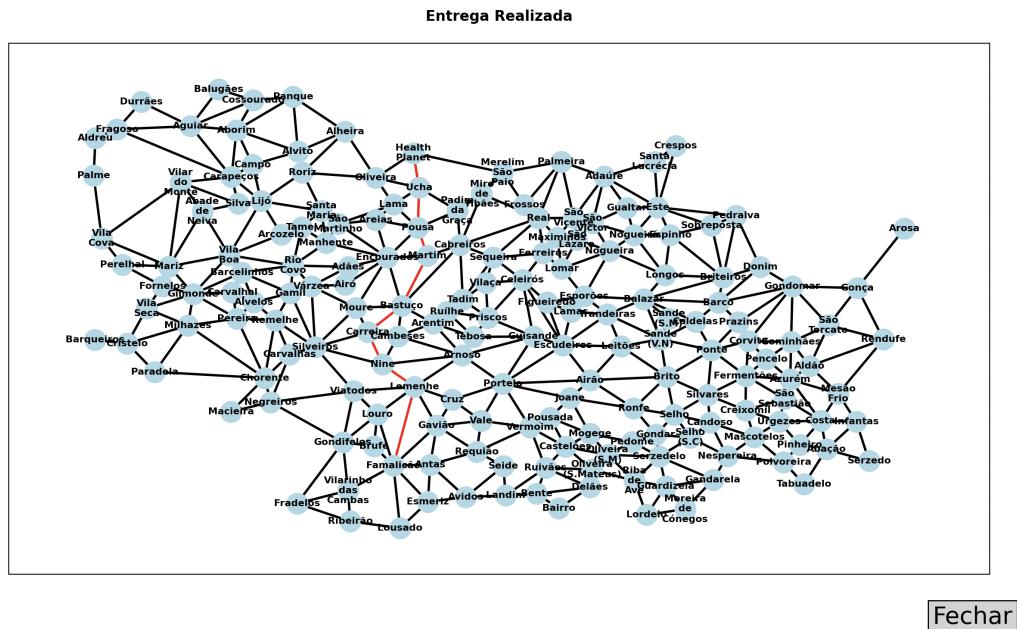


Figura 22: Entrega realizada com o Algoritmo Greedy.

```
Ordem de Expansão: ['Lemenhe', 'Pousa', 'Bastuço', 'Uche', 'Nine', 'Carreira', 'Martim', 'Health Planet']
Entrega da encomenda: 135 foi entregue com sucesso!
Id da Entrega: 62; Id do Cliente: ritav93; Id do Estafeta: beatriz; Classificação: None★ ; Preço: 26.34€ ; Veículo: Moto; Caminho: ['Health Planet', 'Uche', 'Pousa', 'Martim', 'Bastuço', 'Carreira', 'Nine', 'Lemenhe', 'Famalicão'], Distância: 34.4KM ; Tempo Previsto: 1H18m26s; Id da Encomenda: 135; Algoritmo: GREEDY
Clique enter para continuar
```

Figura 23: Informações da entrega realizada com o Algoritmo Greedy.

Como podemos ver na figura 23, apesar de encontrar uma solução boa, não foi a solução ótima que pretendemos alcançar para este tipo de problema, apesar de ser um bom algoritmo em termos de tempo de procura.

8 Conclusão e Trabalho futuro

A realização deste projeto permitiu-nos aplicar os conhecimentos adquiridos ao longo da unidade curricular, explorando os algoritmos de procura em grafos, tanto informados quanto não informados. Desenvolvemos todas as funcionalidades necessárias para a gestão de clientes, estafetas, encomendas e entregas, sendo que todas as informações são armazenadas em ficheiros de texto para utilização posterior do programa ao ser reaberto. Além da implementação da interface principal para a interação entre utilizador e sistema, dedicamos esforços ao desenvolvimento de funcionalidades de desenho, proporcionando uma melhor visualização dos percursos percorridos pelos estafetas no grafo.

No que diz respeito a futuras melhorias, planeamos a implementação de cortes nas estradas, o que implica a possibilidade de interromper arestas no grafo. Este ajuste pode requerer a modificação da heurística estática para uma heurística dinâmica, possivelmente incorporando o tempo como um fator crucial. Adicionalmente, consideramos introduzir penalizações nos rankings dos estafetas, uma vez que a recalibração do caminho ao longo do tempo pode resultar em atrasos que impactam os prazos de entrega. Por fim, gostaríamos de abordar uma última implementação que exploramos sem sucesso no decorrer do trabalho. Consiste na verificação, durante a atribuição de uma entrega de encomenda a um estafeta, se o caminho escolhido pelo mesmo poderia incluir outras encomendas ao longo do percurso. Este aprimoramento proporcionaria uma otimização adicional na gestão das entregas.