Relatório de Algoritmia Avançada

Turma 3DF _ Grupo 025

1200625 – Sérgio André Oliveira Lopes
1200628 – Tiago Francisco da Silva Freitas
1201386 – Rita Ariana de Castro Ribeiro e Pereira Sobral
1202016 – Vasco Filipe Amorim de Azevedo

Data: 04/12/2022

Índice

Base de Conhecimento 3
User Story 1 7
User Story 2 9
User Story 3 11
User Story 4 13
Conclusão 17

Base de Conhecimento

```
%idArmazem(<local>,<codigo>)
idArmazem('Arouca',1).
idArmazem('Espinho',2).
idArmazem('Gondomar',3).
idArmazem('Maia',4).
idArmazem('Matosinhos',5).
idArmazem('Oliveira de Azemeis',6).
idArmazem('Paredes',7).
idArmazem('Porto',8).
idArmazem('Povoa de Varzim',9).
idArmazem('Santa Maria da Feira',10).
idArmazem('Santo Tirso',11).
idArmazem('Sao Joao da Madeira',12).
idArmazem('Trofa',13).
idArmazem('Vale de Cambra',14).
idArmazem('Valongo',15).
idArmazem('Vila do Conde',16).
idArmazem('Vila Nova de Gaia',17).
```

Figura 1 - Armazéns Existentes

A empresa em questão é constituída por 17 armazéns que se encontram identificados na Figura 1. Cada armazém é identificado, entre outros atributos, por uma descrição e por um identificador único.

```
%cidadeArmazem(<codigo>).
cidadeArmazem(5).
```

Figura 2 - Armazém Principal

O facto presente na Figura 2 identifica o armazém principal da empresa, ou seja, o armazém por onde começam e acabam as entregas, sendo neste momento o armazém de Matosinhos.

```
%carateristicasCam(<nome_camiao>,<tara>,<capacidade_carga>,<carga_total_baterias>,<autonomia>,<t_recarr_bat_20a80>).
carateristicasCam(eTruck01,7500,4300,80,100,60).
```

Figura 3 - Características dos Camiões

A distribuição dos produtos entre as empresas é realizada por camiões elétricos, onde as características estão discriminadas na Figura 3.

```
%dadosCam_t_e_ta(<nome_camiao>,<cidade_origem>,<cidade_destino>,<tempo>,<energia>,<tempo_adicional>).
dadosCam_t_e_ta(eTruck01,1,2,122,42,0).
dadosCam_t_e_ta(eTruck01,1,3,122,46,0).
dadosCam_t_e_ta(eTruck01,1,4,151,54,25).
dadosCam_t_e_ta(eTruck01,1,5,147,52,25).
dadosCam_t_e_ta(eTruck01,1,6,74,24,0).
dadosCam_t_e_ta(eTruck01,1,7,116,35,0).
dadosCam_t_e_ta(eTruck01,1,8,141,46,0).
dadosCam_t_e_ta(eTruck01,1,9,185,74,53).
dadosCam_t_e_ta(eTruck01,1,10,97,30,0).
dadosCam_t_e_ta(eTruck01,1,11,164,64,40).
dadosCam_t_e_ta(eTruck01,1,12,76,23,0).
dadosCam_t_e_ta(eTruck01,1,13,174,66,45).
dadosCam_t_e_ta(eTruck01,1,14,59,18,0).
dadosCam_t_e_ta(eTruck01,1,15,132,51,24).
dadosCam_t_e_ta(eTruck01,1,16,181,68,45).
dadosCam_t_e_ta(eTruck01,1,17,128,45,0).
```

Figura 4

```
dadosCam_t_e_ta(eTruck01,2,1,116,42,0).
dadosCam_t_e_ta(eTruck01,2,3,55,22,0).
dadosCam_t_e_ta(eTruck01,2,4,74,25,0).
dadosCam_t_e_ta(eTruck01,2,5,65,22,0).
dadosCam_t_e_ta(eTruck01,2,6,69,27,0).
dadosCam_t_e_ta(eTruck01,2,7,74,38,0).
dadosCam_t_e_ta(eTruck01,2,8,61,18,0).
dadosCam_t_e_ta(eTruck01,2,9,103,44,0).
dadosCam_t_e_ta(eTruck01,2,10,36,14,0).
dadosCam_t_e_ta(eTruck01,2,11,88,41,0).
dadosCam_t_e_ta(eTruck01,2,12,61,19,0).
dadosCam_t_e_ta(eTruck01,2,13,95,42,0).
dadosCam_t_e_ta(eTruck01,2,14,78,34,0).
dadosCam_t_e_ta(eTruck01,2,15,69,30,0).
dadosCam_t_e_ta(eTruck01,2,16,99,38,0).
dadosCam_t_e_ta(eTruck01,2,17,46,14,0).
```

Figura 5

```
dadosCam_t_e_ta(eTruck01,3,1,120,45,0).
dadosCam_t_e_ta(eTruck01,3,2,50,22,0).
dadosCam_t_e_ta(eTruck01,3,4,46,15,0).
dadosCam_t_e_ta(eTruck01,3,5,46,14,0).
dadosCam_t_e_ta(eTruck01,3,6,74,37,0).
dadosCam_t_e_ta(eTruck01,3,7,63,23,0).
dadosCam_t_e_ta(eTruck01,3,8,38,8,0).
dadosCam_t_e_ta(eTruck01,3,9,84,36,0).
dadosCam_t_e_ta(eTruck01,3,10,59,28,0).
dadosCam_t_e_ta(eTruck01,3,11,61,27,0).
dadosCam_t_e_ta(eTruck01,3,12,67,32,0).
dadosCam_t_e_ta(eTruck01,3,13,67,29,0).
dadosCam_t_e_ta(eTruck01,3,14,82,38,0).
dadosCam_t_e_ta(eTruck01,3,15,34,8,0).
dadosCam_t_e_ta(eTruck01,3,16,80,30,0).
dadosCam_t_e_ta(eTruck01,3,17,36,10,0).
```

Figura 6

```
dadosCam_t_e_ta(eTruck01,4,1,149,54,25).
dadosCam_t_e_ta(eTruck01,4,2,65,24,0).
dadosCam_t_e_ta(eTruck01,4,3,46,16,0).
dadosCam_t_e_ta(eTruck01,4,5,27,10,0).
dadosCam_t_e_ta(eTruck01,4,6,103,47,0).
dadosCam_t_e_ta(eTruck01,4,7,55,27,0).
dadosCam_t_e_ta(eTruck01,4,8,36,10,0).
dadosCam_t_e_ta(eTruck01,4,9,50,26,0).
dadosCam_t_e_ta(eTruck01,4,10,78,34,0).
dadosCam_t_e_ta(eTruck01,4,11,42,19,0).
dadosCam_t_e_ta(eTruck01,4,12,97,42,0).
dadosCam_t_e_ta(eTruck01,4,13,44,11,0).
dadosCam_t_e_ta(eTruck01,4,14,111,48,0).
dadosCam_t_e_ta(eTruck01,4,15,32,13,0).
dadosCam_t_e_ta(eTruck01,4,16,53,14,0).
dadosCam_t_e_ta(eTruck01,4,17,38,11,0).
```

Figura 7

```
dadosCam_t_e_ta(eTruck01,5,1,141,51,24).
dadosCam_t_e_ta(eTruck01,5,2,55,20,0).
dadosCam_t_e_ta(eTruck01,5,3,48,14,0).
dadosCam_t_e_ta(eTruck01,5,4,25,9,0).
dadosCam_t_e_ta(eTruck01,5,6,97,44,0).
dadosCam_t_e_ta(eTruck01,5,7,55,28,0).
dadosCam_t_e_ta(eTruck01,5,8,29,7,0).
dadosCam_t_e_ta(eTruck01,5,9,48,24,0).
dadosCam_t_e_ta(eTruck01,5,10,69,30,0).
dadosCam_t_e_ta(eTruck01,5,11,53,26,0).
dadosCam_t_e_ta(eTruck01,5,12,95,36,0).
dadosCam_t_e_ta(eTruck01,5,13,63,20,0).
dadosCam_t_e_ta(eTruck01,5,14,105,45,0).
dadosCam_t_e_ta(eTruck01,5,15,34,14,0).
dadosCam_t_e_ta(eTruck01,5,16,46,18,0).
dadosCam_t_e_ta(eTruck01,5,17,27,7,0).
```

Figura 8

```
dadosCam_t_e_ta(eTruck01,6,1,69,23,0).
dadosCam_t_e_ta(eTruck01,6,2,71,27,0).
dadosCam_t_e_ta(eTruck01,6,3,74,38,0).
dadosCam_t_e_ta(eTruck01,6,4,103,46,0).
dadosCam_t_e_ta(eTruck01,6,5,99,44,0).
dadosCam_t_e_ta(eTruck01,6,7,88,48,0).
dadosCam_t_e_ta(eTruck01,6,8,92,38,0).
dadosCam_t_e_ta(eTruck01,6,9,134,66,45).
dadosCam_t_e_ta(eTruck01,6,10,42,14,0).
dadosCam_t_e_ta(eTruck01,6,11,116,56,30).
dadosCam_t_e_ta(eTruck01,6,12,23,9,0).
dadosCam_t_e_ta(eTruck01,6,13,126,58,33).
dadosCam_t_e_ta(eTruck01,6,14,25,9,0).
dadosCam_t_e_ta(eTruck01,6,15,84,44,0).
dadosCam_t_e_ta(eTruck01,6,16,132,60,35).
dadosCam_t_e_ta(eTruck01,6,17,80,38,0).
```

Figura 9

```
dadosCam_t_e_ta(eTruck01,7,1,116,36,0).
dadosCam_t_e_ta(eTruck01,7,2,71,38,0).
dadosCam_t_e_ta(eTruck01,7,3,61,22,0).
dadosCam_t_e_ta(eTruck01,7,4,53,26,0).
dadosCam_t_e_ta(eTruck01,7,5,53,28,0).
dadosCam_t_e_ta(eTruck01,7,6,88,48,0).
dadosCam_t_e_ta(eTruck01,7,8,59,26,0).
dadosCam_t_e_ta(eTruck01,7,9,88,48,0).
dadosCam_t_e_ta(eTruck01,7,10,84,44,0).
dadosCam_t_e_ta(eTruck01,7,11,74,22,0).
dadosCam_t_e_ta(eTruck01,7,12,82,42,0).
dadosCam_t_e_ta(eTruck01,7,13,76,31,0).
dadosCam_t_e_ta(eTruck01,7,14,97,49,21).
dadosCam_t_e_ta(eTruck01,7,15,29,16,0).
dadosCam_t_e_ta(eTruck01,7,16,84,42,0).
dadosCam_t_e_ta(eTruck01,7,17,69,30,0).
```

Figura 10

```
dadosCam_t_e_ta(eTruck01,8,1,134,46,0).
dadosCam_t_e_ta(eTruck01,8,2,59,18,0).
dadosCam_t_e_ta(eTruck01,8,3,32,6,0).
dadosCam_t_e_ta(eTruck01,8,4,34,10,0).
dadosCam_t_e_ta(eTruck01,8,5,32,7,0).
dadosCam_t_e_ta(eTruck01,8,6,88,38,0).
dadosCam_t_e_ta(eTruck01,8,7,57,26,0).
dadosCam_t_e_ta(eTruck01,8,9,69,30,0).
dadosCam_t_e_ta(eTruck01,8,10,65,26,0).
dadosCam_t_e_ta(eTruck01,8,11,53,22,0).
dadosCam_t_e_ta(eTruck01,8,12,82,34,0).
dadosCam_t_e_ta(eTruck01,8,13,61,24,0).
dadosCam_t_e_ta(eTruck01,8,14,97,40,0).
dadosCam_t_e_ta(eTruck01,8,15,36,12,0).
dadosCam_t_e_ta(eTruck01,8,16,65,23,0).
dadosCam_t_e_ta(eTruck01,8,17,32,6,0).
```

Figura 11

```
dadosCam_t_e_ta(eTruck01,9,1,181,72,50).
dadosCam_t_e_ta(eTruck01,9,2,95,41,0).
dadosCam_t_e_ta(eTruck01,9,3,86,35,0).
dadosCam_t_e_ta(eTruck01,9,4,55,24,0).
dadosCam_t_e_ta(eTruck01,9,5,48,23,0).
dadosCam_t_e_ta(eTruck01,9,6,134,65,42).
dadosCam_t_e_ta(eTruck01,9,7,95,47,0).
dadosCam_t_e_ta(eTruck01,9,8,69,28,0).
dadosCam_t_e_ta(eTruck01,9,10,109,51,24).
dadosCam_t_e_ta(eTruck01,9,11,61,29,0).
dadosCam_t_e_ta(eTruck01,9,12,132,57,31).
dadosCam_t_e_ta(eTruck01,9,13,67,19,0).
dadosCam_t_e_ta(eTruck01,9,14,143,66,45).
dadosCam_t_e_ta(eTruck01,9,15,71,34,0).
dadosCam_t_e_ta(eTruck01,9,16,15,3,0).
dadosCam_t_e_ta(eTruck01,9,17,67,28,0).
```

Figura 12

```
dadosCam_t_e_ta(eTruck01,10,1,97,30,0).
dadosCam_t_e_ta(eTruck01,10,2,34,14,0).
dadosCam_t_e_ta(eTruck01,10,3,59,27,0).
dadosCam_t_e_ta(eTruck01,10,4,78,33,0).
dadosCam_t_e_ta(eTruck01,10,5,71,30,0).
dadosCam_t_e_ta(eTruck01,10,6,40,14,0).
dadosCam_t_e_ta(eTruck01,10,7,82,42,0).
dadosCam_t_e_ta(eTruck01,10,8,65,24,0).
dadosCam_t_e_ta(eTruck01,10,9,109,52,25).
dadosCam_t_e_ta(eTruck01,10,11,92,46,0).
dadosCam_t_e_ta(eTruck01,10,12,32,6,0).
dadosCam_t_e_ta(eTruck01,10,13,99,46,0).
dadosCam_t_e_ta(eTruck01,10,14,63,17,0).
dadosCam_t_e_ta(eTruck01,10,15,74,34,0).
dadosCam_t_e_ta(eTruck01,10,16,105,46,0).
dadosCam_t_e_ta(eTruck01,10,17,53,23,0).
```

Figura 13

```
dadosCam_t_e_ta(eTruck01,11,1,164,65,42).
dadosCam_t_e_ta(eTruck01,11,2,88,41,0).
dadosCam_t_e_ta(eTruck01,11,3,65,28,0).
dadosCam_t_e_ta(eTruck01,11,4,42,18,0).
dadosCam_t_e_ta(eTruck01,11,5,55,25,0).
dadosCam_t_e_ta(eTruck01,11,6,118,57,31).
dadosCam_t_e_ta(eTruck01,11,7,74,23,0).
dadosCam_t_e_ta(eTruck01,11,8,59,23,0).
dadosCam_t_e_ta(eTruck01,11,9,63,28,0).
dadosCam_t_e_ta(eTruck01,11,10,97,46,0).
dadosCam_t_e_ta(eTruck01,11,12,111,52,25).
dadosCam_t_e_ta(eTruck01,11,13,25,7,0).
dadosCam_t_e_ta(eTruck01,11,14,126,58,33).
dadosCam_t_e_ta(eTruck01,11,15,53,25,0).
dadosCam_t_e_ta(eTruck01,11,16,59,27,0).
dadosCam_t_e_ta(eTruck01,11,17,67,27,0).
```

Figura 14

```
dadosCam_t_e_ta(eTruck01,12,1,76,23,0).
dadosCam_t_e_ta(eTruck01,12,2,61,19,0).
dadosCam_t_e_ta(eTruck01,12,3,67,32,0).
dadosCam_t_e_ta(eTruck01,12,4,97,41,0).
dadosCam_t_e_ta(eTruck01,12,5,92,38,0).
dadosCam_t_e_ta(eTruck01,12,6,19,8,0).
dadosCam_t_e_ta(eTruck01,12,7,82,42,0).
dadosCam_t_e_ta(eTruck01,12,8,86,33,0).
dadosCam_t_e_ta(eTruck01,12,9,128,61,37).
dadosCam_t_e_ta(eTruck01,12,10,32,6,0).
dadosCam_t_e_ta(eTruck01,12,11,109,50,23).
dadosCam_t_e_ta(eTruck01,12,13,120,53,26).
dadosCam_t_e_ta(eTruck01,12,14,40,10,0).
dadosCam_t_e_ta(eTruck01,12,15,78,38,0).
dadosCam_t_e_ta(eTruck01,12,16,126,54,28).
dadosCam_t_e_ta(eTruck01,12,17,74,32,0).
```

Figura 15

```
dadosCam_t_e_ta(eTruck01,13,1,174,65,42).
dadosCam_t_e_ta(eTruck01,13,2,107,35,0).
dadosCam_t_e_ta(eTruck01,13,3,74,29,0).
dadosCam_t_e_ta(eTruck01,13,4,46,11,0).
dadosCam_t_e_ta(eTruck01,13,5,67,20,0).
dadosCam_t_e_ta(eTruck01,13,6,128,57,31).
dadosCam_t_e_ta(eTruck01,13,7,80,30,0).
dadosCam_t_e_ta(eTruck01,13,8,76,20,0).
dadosCam_t_e_ta(eTruck01,13,9,67,20,0).
dadosCam_t_e_ta(eTruck01,13,10,105,47,0).
dadosCam_t_e_ta(eTruck01,13,11,27,7,0).
dadosCam_t_e_ta(eTruck01,13,12,122,52,25).
dadosCam_t_e_ta(eTruck01,13,14,137,58,33).
dadosCam_t_e_ta(eTruck01,13,15,67,17,0).
dadosCam_t_e_ta(eTruck01,13,16,59,15,0).
dadosCam_t_e_ta(eTruck01,13,17,78,22,0).
```

Figura 16

```
dadosCam_t_e_ta(eTruck01,14,1,59,18,0).
dadosCam_t_e_ta(eTruck01,14,2,80,35,0).
dadosCam_t_e_ta(eTruck01,14,3,80,38,0).
dadosCam_t_e_ta(eTruck01,14,4,109,46,0).
dadosCam_t_e_ta(eTruck01,14,5,105,45,0).
dadosCam_t_e_ta(eTruck01,14,6,27,9,0).
dadosCam_t_e_ta(eTruck01,14,7,97,48,0).
dadosCam_t_e_ta(eTruck01,14,8,99,38,0).
dadosCam_t_e_ta(eTruck01,14,9,143,66,45).
dadosCam_t_e_ta(eTruck01,14,10,61,17,0).
dadosCam_t_e_ta(eTruck01,14,11,122,57,31)
dadosCam_t_e_ta(eTruck01,14,12,42,10,0).
dadosCam_t_e_ta(eTruck01,14,13,132,58,35)
dadosCam_t_e_ta(eTruck01,14,15,90,44,0).
dadosCam_t_e_ta(eTruck01,14,16,139,61,37)
dadosCam_t_e_ta(eTruck01,14,17,86,38,0).
```

Figura 17

```
dadosCam_t_e_ta(eTruck01,15,1,132,51,24)
dadosCam_t_e_ta(eTruck01,15,2,74,30,0).
dadosCam_t_e_ta(eTruck01,15,3,34,8,0).
dadosCam_t_e_ta(eTruck01,15,4,36,12,0).
dadosCam_t_e_ta(eTruck01,15,5,36,14,0).
dadosCam_t_e_ta(eTruck01,15,6,86,44,0).
dadosCam_t_e_ta(eTruck01,15,7,34,16,0).
dadosCam_t_e_ta(eTruck01,15,8,42,13,0).
dadosCam_t_e_ta(eTruck01,15,9,71,35,0).
dadosCam_t_e_ta(eTruck01,15,10,82,36,0).
dadosCam_t_e_ta(eTruck01,15,11,53,25,0).
dadosCam_t_e_ta(eTruck01,15,12,80,38,0).
dadosCam_t_e_ta(eTruck01,15,13,69,18,0).
dadosCam_t_e_ta(eTruck01,15,14,95,45,0).
dadosCam_t_e_ta(eTruck01,15,16,69,29,0).
dadosCam_t_e_ta(eTruck01,15,17,53,17,0).
```

Figura 18

```
dadosCam_t_e_ta(eTruck01,16,1,179,68,45).
dadosCam_t_e_ta(eTruck01,16,2,92,37,0).
dadosCam_t_e_ta(eTruck01,16,3,84,31,0).
dadosCam_t_e_ta(eTruck01,16,4,57,16,0).
dadosCam_t_e_ta(eTruck01,16,5,46,18,0).
dadosCam_t_e_ta(eTruck01,16,6,132,60,35).
dadosCam_t_e_ta(eTruck01,16,7,92,42,0).
dadosCam_t_e_ta(eTruck01,16,8,67,23,0).
dadosCam_t_e_ta(eTruck01,16,9,15,3,0).
dadosCam_t_e_ta(eTruck01,16,10,105,46,0).
dadosCam_t_e_ta(eTruck01,16,11,57,28,0).
dadosCam_t_e_ta(eTruck01,16,12,130,52,25).
dadosCam_t_e_ta(eTruck01,16,13,61,15,0).
dadosCam_t_e_ta(eTruck01,16,14,141,61,37).
dadosCam_t_e_ta(eTruck01,16,15,69,29,0).
dadosCam_t_e_ta(eTruck01,16,17,65,24,0).
```

Figura 19

```
dadosCam_t_e_ta(eTruck01,17,1,128,46,0).
dadosCam_t_e_ta(eTruck01,17,2,42,14,0).
dadosCam_t_e_ta(eTruck01,17,3,40,11,0).
dadosCam_t_e_ta(eTruck01,17,4,42,13,0).
dadosCam_t_e_ta(eTruck01,17,5,34,10,0).
dadosCam_t_e_ta(eTruck01,17,6,82,38,0).
dadosCam_t_e_ta(eTruck01,17,7,74,30,0).
dadosCam_t_e_ta(eTruck01,17,8,29,6,0).
dadosCam_t_e_ta(eTruck01,17,9,69,31,0).
dadosCam_t_e_ta(eTruck01,17,10,55,24,0).
dadosCam_t_e_ta(eTruck01,17,11,69,29,0).
dadosCam_t_e_ta(eTruck01,17,12,80,30,0).
dadosCam_t_e_ta(eTruck01,17,13,82,23,0).
dadosCam_t_e_ta(eTruck01,17,14,90,38,0).
dadosCam_t_e_ta(eTruck01,17,15,53,18,0).
dadosCam_t_e_ta(eTruck01,17,16,67,25,0).
```

Figura 20

Da Figura 4 à Figura 20 estão representadas as informações sobre os diversos percursos que o camionista, no processo de entrega das encomendas, poderá fazer. A informação de cada percurso corresponde às características da viagem entre a cidade origem e a cidade destino, tais como o tempo e a energia da viagem se o camião estiver totalmente carregado e o tempo extra se for necessário efetuar um carregamento a meio do percurso.

```
%entrega(<idEntrega>,<data>,<massaEntrefa>,<armazemEntrega>,<tempoColoc>,<tempoRet>)
entrega(4439, 20221205, 200, 1, 8, 10).
entrega(4438, 20221205, 150, 9, 7, 9).
entrega(4445, 20221205, 100, 3, 5, 7).
entrega(4443, 20221205, 120, 8, 6, 8).
entrega(4449, 20221205, 300, 11, 15, 20).
entrega(4398, 20221205, 310, 17, 16, 20).
entrega(4432, 20221205, 270, 14, 14, 18).
entrega(4437, 20221205, 180, 12, 9, 11).
entrega(4451, 20221205, 220, 6, 9, 12).
entrega(4452, 20221205, 390, 13, 21, 26).
entrega(4444, 20221205, 380, 2, 20, 25).
entrega(4455, 20221205, 280, 7, 14, 19).
entrega(4399, 20221205, 260, 15, 13, 18).
entrega(4454, 20221205, 350, 10, 18, 22).
entrega(4446, 20221205, 260, 4, 14, 17).
entrega(4456, 20221205, 330, 16, 17, 21).
```

Figura 21

As entregas que a empresa tem de realizar são caracterizadas pela data de entrega, o armazém de entregas e o tempo que demora a ser colocada e retirada do camião, como podemos ver pela Figura 21.

User Story 1

Recebendo os dados das entregas a fazer por 1 camião e dos troços entre armazéns: gerar todas as trajetórias possíveis através de sequências de armazéns onde deverão ser feitas as entregas.

```
armazensViagem(L,Data):-findall(Id, entrega(_,Data,_,Id,_,_),L).

viagens(LT,Data):- armazensViagem(LA,Data), findall(Viagem, permutation(LA, Viagem), LT).

viagensFinal(LF,Data):-viagens(LT,Data), viagensCompleta(LT, LF).

viagensCompleta([],[]).
viagensCompleta([T|LT], [R|LF]):- adicionarMatosinhos(T, R),write(R),nl, viagensCompleta(LT, LF).

adicionarMatosinhos(LI, LF):-cidadeArmazem(Id), append([Id|LI],[Id],LF).
```

Figura 22 – Predicados relativos à User Story 1

O predicado armazensViagens/2 recebe a data correspondente ao dia da viagem a qual pretendemos determinar todos os armazéns onde deverão ser feitas as entregas. Pesquisa todas as entregas existentes e adiciona os identificadores dos armazéns de entrega para a data especificada na lista L.

O predicado viagens/2 invoca o predicado armazensViagens/2 para determinar quais os armazéns a incluir na viagem e de seguida gera todas as viagens possíveis, recorrendo ao findall.

O predicado adicionarMatosinhos/2 recebe uma lista de armazéns e com recurso ao append gera uma nova lista em que o primeiro e o último elemento são o 5, correspondente ao armazém de Matosinhos, e os restantes elementos são os elementos da lista recebida.

O predicado viagemCompleta/2 recebe uma lista que contém as várias viagens geradas pelo predicado viagens/2, viagens que ainda não contêm o armazém de Matosinhos, e invoca o predicado adicionarMatosinhos/2, tantas vezes como as viagens geradas anteriormente, para que o armazém de Matosinhos seja adicionado no início e fim de cada viagem.

Por fim, o predicado viagensFinal/2, predicado que inicia o processo de gerar todas as trajetórias possíveis, invoca primeiramente o predicado viagens/2 para determinar todas as viagens possíveis para o camião realizar as entregas na data indicada e depois invoca o predicado viagemCompleta/2 para que seja adicionado tanto no início como no fim das diferentes viagens o armazém Principal.

```
?- viagensFinal(LF,20221205).
[5,1,9,3,5]
[5,1,3,9,5]
[5,9,1,3,5]
[5,9,3,1,5]
[5,3,1,9,5]
[5,3,9,1,5]
LF = [[5, 1, 9, 3, 5], [5, 1, 3, 9, 5], [5, 9, 1, 3, 5], [5, 9, 3, 1, 5], [5, 3, 1, 9|...], [5, 3, 9|...]].
```

Figura 23

Na Figura 23, temos um exemplo do resultado obtido quando invocado o predicado viagensFinal/2 para a data de 5 de dezembro de 2022 se para esse dia existissem entregas para 3 armazéns, o armazém 1, 9 e 3.

User Story 2

Avaliar as trajetórias geradas na User Story anterior de acordo com o tempo para completar todas as entregas e voltar ao armazém base de Matosinhos e escolher a solução que permite a volta do camião mais cedo.

Para implementar este requisito a subdivisão em pequenas tarefas foi fundamental, por este motivo o problema foi subdividido nas seguintes etapas para determinar o tempo de uma viagem:

1. Obter a carga do Camião
2. Adicionar a tara do Camião
3. Determinar tempo da Viagem

```
obterCargaCamiao(_, [], [], 0):-!.
obterCargaCamiao(Data, [Armazem|LA], [Carga|LC], Carga):-obterCargaCamiao(Data, LA, LC, CargaAux), entrega(_,Data,Massa,Armazem,_,_),
                                                         Carga is Massa + CargaAux.

adicionarTaraCamiao(NomeCamiao, LC, LCT):- carateristicasCam(NomeCamiao,Tara,_,_,_,_), adicionarTara(Tara,LC,LCT).

adicionarTara(Tara,[],[Tara]):-!.
adicionarTara(Tara, [Carga|LC], [CargaTara|LCT]):- adicionarTara(Tara,LC,LCT), CargaTara is Carga + Tara.

obterCapacidadeCargaComTara(Camiao,Capacidade):-carateristicasCam(Camiao,Tara,CapacidadeCarga,_,_,_),
                                                Capacidade is Tara + CapacidadeCarga.

determinarTempo(Data, Camiao, LA, Tempo):-
        obterCargaCamiao(Data, LA, LC,_), adicionarTaraCamiao(Camiao,LC, LCT),
        cidadeArmazem(Id), append([Id|LA],[Id],Viagem),
        obterCapacidadeCargaComTara(Camiao,Capacidade),
        carateristicasCam(Camiao,_,_,CargaBaterias,_,DuracaoCarregamento),
        tempoViagem(Viagem, LCT,Capacidade,CargaBaterias,CargaBaterias,DuracaoCarregamento,Data,Tempo),!.
```

Figura 24

O predicado obterCargaCamiao/4 determina para uma viagem a carga do camião em cada ponto do percurso, tendo no início da rota a soma de todas as massas das entregas que tem de realizar.

O predicado adicionarTaraCamiao/3 invoca o predicado adicionarTara/3 que vai adicionar a cada elemento da lista obtida pelo predicado obterCargaCamiao a tara do camião que irá realizar a viagem adicionando no final da lista um novo elemento só com a tara, que corresponde ao peso do camião depois de ter realizado todas as entregas a regressar ao armazém de Matosinhos.

O predicado determinarTempo/4 invoca os predicados mencionados anteriormente e por fim invoca o predicado tempoViagem/ 8 que determina a duração da Viagem passada no primeiro parâmetro tendo em conta todos os fatores indicados pelo cliente.

```
?-
% c:/Users/arian/eletricGo/lapr5_22-23/Planeamento/SprintB/Algav.pl compiled 0.00 sec, 329 clauses
?- determinarTempo(20221205,eTruck01,[1,9,3,8,11],Tempo).
Tempo = 555.270127118644.
```

Figura 25

Na Figura 25, temos um exemplo do resultado obtido quando invocado o predicado determinarTempo/4 para a data de 5 de dezembro de 2022 para a viagem 5-> 1 -> 9 -> 3 -> 8 -> 11 -> 5.

```
melhorViagem(L,Tempo,Data,Camiao):- get_time(Ti),
                                    (run(Data, Camiao);true),menorTempo(L,Tempo),
                                    get_time(Tf),
                                    TSol is Tf-Ti,
                                    write(TSol),nl.

run(Data, Camiao):- retractall(menorTempo(_,_)), assertz(menorTempo(_,1000000)),
        findall(Id, entrega(_,Data,_,Id,_,_),LF),
        permutation(LF,LFPerm),
        determinarTempo(Data,Camiao,LFPerm,Tempo),
        atualiza(LFPerm,Tempo),
        fail.

atualiza(LFPerm,Tempo):-
        menorTempo(_,TempoMin),
        ((Tempo<TempoMin,!,retract(menorTempo(_,_)),assertz(menorTempo(LFPerm,Tempo)));true).
```

Figura 26

O predicado melhorViagem/4 invoca o predicado run/2 n! vezes que por sua vez invoca o predicado determinarTempo/4. Cada vez que o predicado determinarTempo/4 é invocado o tempo da viagem é comparado com o menor Tempo identificado até ao momento e caso seja menor a viagem e o tempo são guardados para futuras comparações, obtendo no final a rota com menor duração e o respetivo tempo.

```
?- melhorViagem(L,Tempo,20221205,eTruck01).
0.0020928382873535156
L = [8, 1, 3, 11, 9],
Tempo = 412.569279661017.
```

Figura 27

Na Figura 27, temos um exemplo do resultado obtido quando invocado o predicado melhorViagem/4 para a data de 5 de dezembro de 2022 com entregas para 5 armazéns.

User Story 3

Aumentar a dimensão do problema (colocando mais armazéns a visitar) e verificar até que dimensão é viável proceder do modo adotado (com um gerador de todas as soluções) efetuando um estudo de complexidade do problema.

Estudo da Complexidade e Viabilidade

Nº de Armazéns de Entrega	Nº de Soluções	Lista com a sequência de armazéns	Tempo (min) para fazer as entregas	Tempo (s) de geração da solução
5	120	[8, 1, 3, 11, 9]	412,57	0.00209
6	720	[8, 3, 1, 17, 11, 9]	453.60	0.01448
7	5040	[17, 8, 3, 1, 14, 11, 9]	500.45	0.10598
8	40320	[17, 8, 3, 12, 1, 14, 11, 9]	532.44	1.01675
9	362880	[8, 3, 12, 6, 14, 1, 17, 11, 9]	562.66	10.21029
10	3628800	[17, 8, 3, 12, 6, 14, 1, 11, 13, 9]	618.03	107.32605
11	39916800	[9, 13, 11, 17, 2, 12, 6, 14, 1, 3, 8]	676.88	1392.74243
12	479001600	-	-	-
13	6227020800	-	-	-
14	87178291200	-	-	-
15	1307674368000	-	-	-
16	20922789888000	-	-	-

Após uma análise da tabela elaborada durante a realização da User Story 3 conseguimos chegar a algumas conclusões:

- O tempo de geração da solução concebida na User Story 2 encontra-se relacionado com um fatorial, conseguindo assim prever quanto tempo, por exemplo, demorará gerar uma solução que envolva 13 armazéns. O valor previsto seria o último tempo que sabemos multiplicado pelo número de armazéns que ainda nos falta descobrir até ao armazém 13, ou seja, $1392{,}7424 \times 12 \times 13 = 217267{,}8144$. Para comprovar que o resultado é um número próximo do obtido vamos testar para o tempo de geração com 11 armazéns. O valor teórico seria $107{,}32605 \times 11 = 1180{,}6$. 1180,6s que é um valor próximo de 1392,7s.

- Devido ao tempo de geração da solução, a partir de uma certa quantidade de armazéns não é viável optar pela solução concebida na User Story 2. Existem algumas variantes para determinarmos até que quantidade de armazéns é praticável utilizar este processo, como é o caso do tempo de precedência com que estamos a efetuar o planeamento das entregas para um dado dia. A partir de 12 armazéns não é viável, para 11 poderá ser realizável se for feito com antecedência e até 10 armazéns esta é a melhor opção para realizar a elaboração da rota.

- A complexidade do predicado determinarTempo/4 é O(n+1), sendo n a quantidade de armazéns onde temos de efetuar entregas, pois neste sprint por cada armazém só é efetuada uma entrega, logo a quantidade de entregas é igual à quantidade de armazéns a visitar durante a viagem. A complexidade do predicado melhorViagem/4 é O(n!).

User Story 4

Implementar heurísticas que possam rapidamente gerar uma solução (não necessariamente a melhor) e avaliar a qualidades dessas heurísticas (por exemplo, entregar no armazém mais próximo; efetuar de seguida a entrega com maior massa; combinar distância para a entrega com massa entregue).

Estudo da Adequação das Heurísticas

Nº de Armazéns de Entrega	Solução Ótima	Tempo para Entregas Sol. ótima	Tempo Heurística do menor tempo	Tempo Heurística da maior massa	Tempo para Entregas Heurística combinada	Melhor solução pelas 3 heurísticas
5	-	412,57	451.61	574.75	430.33	430.33
6	-	453.60	497.03	622.91	489.64	489.64
7	-	500.45	546.65	653.41	558.33	546.65
8	-	532.44	585.83	640.44	632.85	585.83
9	-	562.66	614.85	696.75	711.52	614.85
10	-	618.03	668.82	845.36	773.97	668.82
11	-	676.88	692.25	944.22	858.96	692.25
12	-	-	728.78	1028.62	891.09	728.78
13	-	-	838.62	1208.38	944.22	838.62
14	-	-	867.39	1277.18	811.41	811.41
15	-	-	962.53	1365.77	863.74	863.74
16	-	-	1016.16	1510.60	912.98	912.98

De maneira a gerar soluções mais rapidamente foram implementadas 3 heurísticas. A primeira heurística determina que o próximo armazém a visitar é, de entre os que ainda não foram visitados, aquele a que demorámos menos tempo a chegar. A segunda heurística determina que o próximo armazém a visitar é aquele onde efetuamos a entrega com mais massa. Por último, a terceira heurística determina que o próximo armazém a visitar é aquele onde existir uma melhor relação entre o tempo de viagem e a massa da entrega.

```
armazemMaisProximo(_, [], 1000000,_):-!.

armazemMaisProximo(Origem, [A1|Armazens], MenorTempo, Armazem):-armazemMaisProximo(Origem,Armazens,MenorTempo1,Armazem1),
                                                                dadosCam_t_e_ta(_,Origem,A1,Tempo,_,_),
                                                                ((Tempo<MenorTempo1,!, MenorTempo is Tempo, Armazem = A1);
                                                                MenorTempo is MenorTempo1, Armazem = Armazem1).


bfsArmazemMaisProximo(_,[],[]):-!.

bfsArmazemMaisProximo(Origem,[A|RestantesArmazens],[ArmazemSeguinte|Viagem]):-
                                        armazemMaisProximo(Origem,[A|RestantesArmazens],_,ArmazemSeguinte),
                                        delete([A|RestantesArmazens],ArmazemSeguinte,ArmazensEmFalta),
                                        bfsArmazemMaisProximo(ArmazemSeguinte,ArmazensEmFalta,Viagem).


melhorViagemArmazemMaisProximo(Data, Camiao, Viagem, Tempo):-armazensViagem(ArmazensVisitar,Data),
                                                            cidadeArmazem(Origem),
                                                            bfsArmazemMaisProximo(Origem,ArmazensVisitar,Viagem),
                                                            determinarTempo(Data, Camiao, Viagem, Tempo), !.
```

Figura 28

O predicado armazemMaisProximo/4 determina qual o armazém mais próximo, dos que ainda não foram percorridos, do armazém enviado no primeiro parâmetro.

O predicado melhorViagemArmazemMaisProximo/4 primeiro determina quais são os armazéns onde existem entregas para efetuar e depois chama o predicado bfsArmazemMaisProximo/3 que determina qual é o armazém mais próximo seguinte e apaga-o da lista de armazéns que faltam visitar.

```
?- melhorViagemArmazemMaisProximo(20221205, eTruck01, Viagem, Tempo).
Viagem = [8, 3, 11, 9, 1],
Tempo = 451.6148305084746.
```

Figura 29

Na Figura 29, temos um exemplo do resultado obtido quando invocado o predicado melhorViagemArmazemMaisProx/4 para a data de 5 de dezembro de 2022 com entregas para 5 armazéns.

```
entregaMaisMassa([],_, 0,_):-!.

entregaMaisMassa([A1|Armazens], Data, MaiorMassa, Armazem):-entregaMaisMassa(Armazens,Data,MaiorMassa1,Armazem1),
                                                            entrega(_,Data,Massa,A1,_,_),
                                                            ((Massa>MaiorMassa1,!, MaiorMassa is Massa, Armazem = A1);
                                                            MaiorMassa is MaiorMassa1, Armazem = Armazem1),!.


bfsArmazemMaisMassa(_,[],[]):-!.

bfsArmazemMaisMassa(Data,[A|RestantesArmazens],[ArmazemSeguinte|Viagem]):-entregaMaisMassa([A|RestantesArmazens],Data,_,ArmazemSeguinte),
                                                                          delete([A|RestantesArmazens],ArmazemSeguinte,ArmazensEmFalta),
                                                                          bfsArmazemMaisMassa(Data, ArmazensEmFalta,Viagem).


melhorViagemArmazemMaisMassa(Data, Camiao, Viagem, Tempo):-armazensViagem(ArmazensVisitar,Data),
                                                           bfsArmazemMaisMassa(Data,ArmazensVisitar,Viagem),
                                                           determinarTempo(Data, Camiao, Viagem, Tempo), !.
```

Figura 30

O predicado entregaMaisMassa/4 determina qual a entrega com mais massa, das que ainda não foram entregues.

O predicado melhorViagemArmazemMaisMassa/4 primeiro determina quais são os armazéns onde existem entregas para efetuar e depois chama o predicado bfsArmazemMaisMassa/3 que determina qual é o armazém onde será efetuada a entrega com maior massa e apaga-o da lista de armazéns que ainda faltam visitar.

```
?- melhorViagemArmazemMaisMassa(20221205, eTruck01, Viagem, Tempo).
Viagem = [11, 1, 9, 8, 3],
Tempo = 574.7485169491525.
```

Figura 31

Na Figura 31, temos um exemplo do resultado obtido quando invocado o predicado melhorViagemArmazemMaisMassa/4 para a data de 5 de dezembro de 2022 com entregas para 5 armazéns.

```
entregaMelhorRelacao(_,[],_, -10000000,_):-!.

entregaMelhorRelacao(Origem,[A1|Armazens], Data, MelhorRelacao, Armazem):-
                entregaMelhorRelacao(Origem, Armazens,Data,MelhorRelacao1,Armazem1),
                dadosCam_t_e_ta(_,Origem,A1,Tempo,_,_),
                entrega(_,Data,Massa,A1,_,_),
                (((Massa/Tempo)>MelhorRelacao1,!, MelhorRelacao is (Massa/Tempo), Armazem = A1);
                MelhorRelacao is MelhorRelacao1, Armazem = Armazem1),!.

bfsMelhorRelacao(_,_,[],[]):-!.

bfsMelhorRelacao(Data,Origem,[A|RestantesArmazens],[ArmazemSeguinte|Viagem]):-
                entregaMelhorRelacao(Origem,[A|RestantesArmazens],Data,_,ArmazemSeguinte),
                delete([A|RestantesArmazens],ArmazemSeguinte,ArmazensEmFalta),
                bfsMelhorRelacao(Data, ArmazemSeguinte, ArmazensEmFalta,Viagem).


melhorViagemMelhorRelacao(Data, Camiao, Viagem, Tempo):-armazensViagem(ArmazensVisitar,Data),
                                                        cidadeArmazem(Origem),
                                                        bfsMelhorRelacao(Data,Origem,ArmazensVisitar,Viagem),
                                                        determinarTempo(Data, Camiao, Viagem, Tempo), !.
```

Figura 32

O predicado entregaMelhorRelacao/4 determina qual a entrega com melhor relação entre distância e massa, das que ainda não foram entregues. Como queremos a melhor relação entre menor distância e maior massa dividimos a massa pelo tempo e escolhemos a que der o valor maior.

O predicado melhorViagemMelhorRelacao/4 primeiro determina quais são os armazéns onde existem entregas para efetuar e depois chama o predicado bfsMelhorRelacao/3 que determina qual é o armazém onde será efetuada a entrega com melhor relação entre a massa e o tempo e apaga-o da lista de armazéns que ainda faltam visitar.

```
?- melhorViagemMelhorRelacao(20221205, eTruck01, Viagem, Tempo).
Viagem = [11, 9, 8, 3, 1],
Tempo = 430.33898305084745.
```

Figura 33

Na Figura 33, temos um exemplo do resultado obtido quando invocado o predicado melhorViagemMelhorRelacao/4 para a data de 5 de dezembro de 2022 com entregas para 5 armazéns.

Depois de analisadas as diferentes heurísticas e a tabela elaborada nesta User Story conseguimos determinar que a melhor heurística é a primeira, pois é a que mais frequentemente gera soluções próximas da melhor solução.

Conclusão

Após a implementação das funcionalidades pedidas no âmbito da unidade curricular de Algoritmia Avançada, do terceiro ano da Licenciatura em Engenharia Informática, do Instituto Superior de Engenharia do Porto, conseguimos concluir que:

- O método findall não é a melhor solução para gerar todas as combinações possíveis de viagem se existirem muitos armazéns para realizar entregas, pois poderemos obter um erro de Stack Limit Exceeded.
- À medida em que aumentamos a quantidade de armazéns os valores temporais aumentam também tornando o método de determinação da melhor viagem através da análise de todas as possibilidades pouco eficiente para grandes quantidades de informação.

Em conclusão, as evidências apresentadas neste relatório demonstram claramente que as heurísticas são uma ferramenta importante para a resolução de problemas e tomada de decisões. Elas fornecem uma estrutura para resolver problemas complexos e ajudam a processar rapidamente grandes quantidades de informações disponíveis e tomar decisões sem ter de considerar todas as opções possíveis.