



Universidade do Minho
Escola de Engenharia

Mestrado em Engenharia de Telecomunicações e Informática



Eco Delivery – Sistema de representação de conhecimento e raciocínio
Inteligência Artificial para as Telecomunicações

Ano Letivo 2021/2022
Guimarães, dezembro de 2021

Hugo Miguel Miranda Reynolds, a83924@alunos.uminho.pt

Pedro Miguel Fonseca Sampaio, a79668@alunos.uminho.pt

Rui Filipe Ribeiro Freitas, pg47639@alunos.uminho.pt

Tiago João Pereira Ferreira, pg47692@alunos.uminho.pt



Índice

Índice de figuras	3
Introdução	4
Desenvolvimento.....	5
1. Especificação do projeto	5
1.1. Característica do Centro de Distribuição Eco Delivery	6
1.1.1. Circuito de Entrega	6
1.2. Estratégias	7
1.2.1. Algoritmo Profundidade (DFS-Depth-First Search)	7
1.2.2. Algoritmo A*(Estrela).....	8
2. Descrição da solução	9
2.1. Base de conhecimento	9
2.1.1. Cliente	9
2.1.2. Estafeta	9
2.1.3. Encomendas entregues	10
2.1.4. Encomendas.....	10
2.1.5. Cidades.....	10
2.2. Interface do utilizador	11
2.2.1. Registo de um cliente	11
2.2.2. Realização de uma encomenda	12
2.2.3. Entrega de uma encomenda.....	13
2.2.4. Estafeta que utilizou o transporte mais ecológico	13
2.2.5. Número de entregas num período de tempo	13
Testes e discussão de resultados.....	14
Conclusão.....	15



Índice de figuras

Figura 1 - Diagrama DFS.	7
Figura 2 - Diagrama A*.	8
Figura 3 - Base de conhecimento (Cliente).	9
Figura 4 - Base de conhecimento (Estafeta).	9
Figura 5 - Base de conhecimento (Encomendas entregues).	10
Figura 6 - Base de conhecimento (Encomendas).	10
Figura 7 - Base de conhecimento (Cidades).	10
Figura 8 - Interface do utilizador.	11
Figura 9 - Método para o registo de um cliente.	11
Figura 10 - Método realização de uma encomenda.	12
Figura 11 - Método de confirmação de uma encomenda.	13
Figura 12 - Método qual o estafeta que mais vezes utilizou a bicicleta.	13
Figura 13 - Método que devolve o número de entregas num período de tempo.	13
Figura 14 - Teste da solução final.	14

Introdução

O presente relatório tem como objetivo estimular o uso de técnicas de formulação de problemas, bem como a aplicação de diversas estratégias para a resolução de problemas com a utilização de algoritmos de procura e o desenvolvimento de mecanismos de raciocínio com recurso à linguagem PROLOG, para o problema apresentado pelos docentes na unidade curricular de Inteligência Artificial para as Telecomunicações.

Este relatório apresenta as tarefas desenvolvidas no contexto do trabalho prático. O trabalho consiste no desenvolvimento de um sistema de representação de conhecimento e raciocínio com a capacidade de utilizar um universo de discurso na área de distribuição de encomendas, entre outras tarefas, para isso foi desenvolvida uma empresa *Eco Delivery* que tem como principal objetivo privilegiar sempre o meio de entrega mais ecológico.

Este projeto foi dividido em 3 fases principais. A primeira que passou pela leitura e interpretação do enunciado começando pela realização de um grafo tendo em conta o nosso local escolhido. Este local corresponde então ao distrito de Braga pelo que decidimos realizar uma análise às distâncias entre os vários concelhos deste distrito. Após isso passamos para a elaboração da base de conhecimento do PROLOG onde começamos por colocar os predicados que iríamos utilizar como por exemplo os clientes, os estafetas assim como as encomendas e os veículos. Com estas 2 fases bem implementadas passamos para a realização de métodos que nos permitisse ver certos atributos do nosso sistema como quais os melhores estafetas assim como quais veículos estes utilizaram, entre outras.

De modo a sermos capazes de cumprir com os objetivos propostos tivemos de realizar um estudo bastante aprofundado da linguagem de programação PROLOG visto não ser de todo uma linguagem utilizada no nosso curso. Isto tirou-nos algum tempo pois tivemos de perder bastantes horas a corrigir erros e procurar documentação por ser uma linguagem nova.

Desenvolvimento

Neste capítulo serão demonstrados os vários passos que nos levaram a chegar à nossa solução que apesar de não acharmos ser a solução perfeita foi o melhor que conseguimos alcançar com o tempo que tivemos.

1. Especificação do projeto

A empresa *Eco Delivery* pratica entregas entre todos os concelhos de Braga, para isso dispõem de diversos meios de transporte: bicicletas, motos e carros, que são classificados consoante os seus níveis de ecologia. A empresa para além de possuir vários tipos de transporte também possui vários estafetas. Um estafeta da *Eco Delivery*, no contexto de um concelho, tem associado um conjunto de entregas a efetuar em determinados pontos de uma cidade.

Um cliente terá a possibilidade de indicar o tempo máximo em que pretende que a sua encomenda seja entregue, para isso existem vários prazos de entrega. Consoante o desempenho do estafeta, este terá uma classificação associada de 0 a 5 estrelas, que é definida pelo cliente. Caso um estafeta não cumpra com o prazo estipulado para a entrega da encomenda, este sofrerá uma penalização na sua classificação, bem como uma diminuição de entregas a ele associadas.

Cada encomenda é classificada pelo seu peso e volume. O preço final do serviço de entrega tem em conta, para além da encomenda, a distância, o prazo estipulado de entrega e o meio de transporte utilizado.

Os vários transportes utilizados pela empresa têm as seguintes características:

- As bicicletas podem transportar encomendas no máximo até 5 Kg, tendo em conta uma velocidade média de 10km/h;
- No caso das motos, este meio poderá transportar encomendas com um limite máximo de 20 Kg e com uma velocidade média de cerca de 35km/h;
- Relativamente ao carro, este terá uma velocidade média de aproximadamente 25km/h, com um peso máximo de transporte de 100kg.

1.1. Característica do Centro de Distribuição Eco Delivery

É importante referir que todas as encomendas se iniciam num centro de distribuição, neste caso existe apenas um, portanto, todas as encomendas têm o mesmo ponto de partida.

Sendo assim, o centro de distribuição deverá cumprir os seguintes requisitos:

- Gerar os circuitos de entrega;
- Apresentar pontos de entrega;
- Identificar circuitos com maior volume de entregas;
- Comparar circuitos em relação ao seu desempenho;
- Identificar e escolher o circuito mais rápido para cada caso (encomenda), utilizando a distância;
- Selecionar o trajeto mais ecológico, com base no tempo que cada entrega demora a ser concretizada.

1.1.1. Circuito de Entrega

O Circuito de entrega mencionado no ponto anterior é importante referir porque grande parte do projeto assenta no circuito, sendo que se algum erro acabar por ocorrer neste processo as informações transmitidas pelo programa não serão as mais corretas.

O circuito de entrega está dividido em três fases:

1. Estado Inicial -> nesta primeira etapa os estafetas efetuam a recolha da encomenda, consoante a distância e o tempo necessário para efetuar a tarefa;
2. Ponto de Entrega -> o percurso que o estafeta realiza entre o ponto inicial e o ponto final (consumidor);
3. Estado Final -> Retorno do estafeta ao centro de distribuição *Eco Delivery*.

1.2. Estratégias

Existem dois tipos de estratégia, a procura não informada e a informada. Para cada uma delas existem algoritmos que conseguem resolver certos problemas. Para a procura não informada existem algoritmos como a Profundidade (DFS - *Depth-First Search*) e a Largura (BFS - *Breadth-First Search*), em relação ao algoritmo de procura informada existe o algoritmo Gulosa e a A* (Estrela).

No caso particular, o grupo procurou estabelecer um melhor rendimento do programa e evitar erros durante a sua utilização, dito isto, implementou-se o algoritmo de profundidade, relacionado com a procura não informada e o algoritmo A* (Estrela) na procura informada.

1.2.1. Algoritmo Profundidade (DFS-Depth-First Search)

Este algoritmo consiste na procura do destino sem conhecer o melhor caminho, ou seja, a procura inicia-se num determinado nó e continua até que o destino seja encontrado, nó após nó, caso o destino não seja encontrado, o processo retrocede e volta ao nó anterior para realizar novamente a procura e assim sucessivamente até encontrar o destino pretendido. Com este tipo de procura não é garantido que o "caminho" será o mais rentável, o mais acessível apenas é assegurado que o destino final é encontrado.

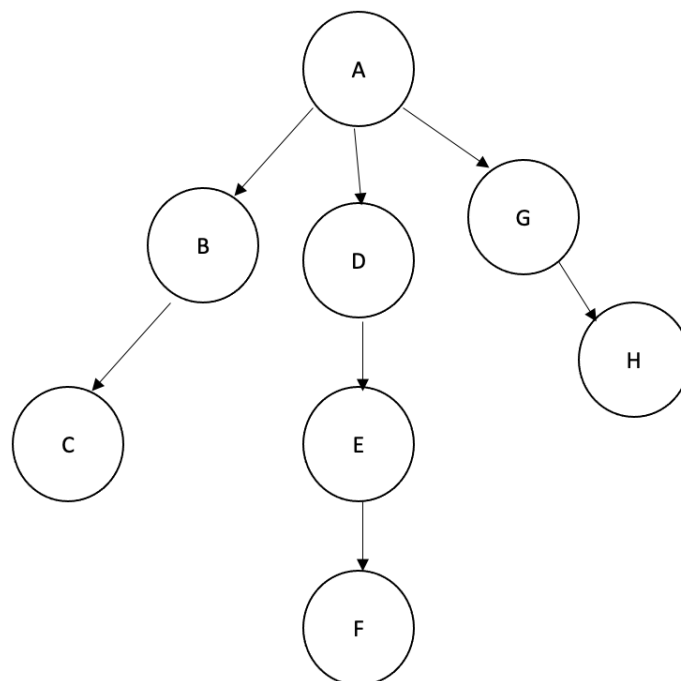


Figura 1 - Diagrama DFS.

1.2.2. Algoritmo A*(Estrela)

Ao contrário do que se verifica no algoritmo anterior, este é míope, por outras palavras, vai escolhendo certos caminhos com base nas informações existentes procurando sempre escolher o melhor caminho que resolva o problema. É melhor opção que o anterior porque procura sempre o melhor caminho, mas nem sempre é o melhor caminho pois está sempre dependente da informação existente na base de dados.

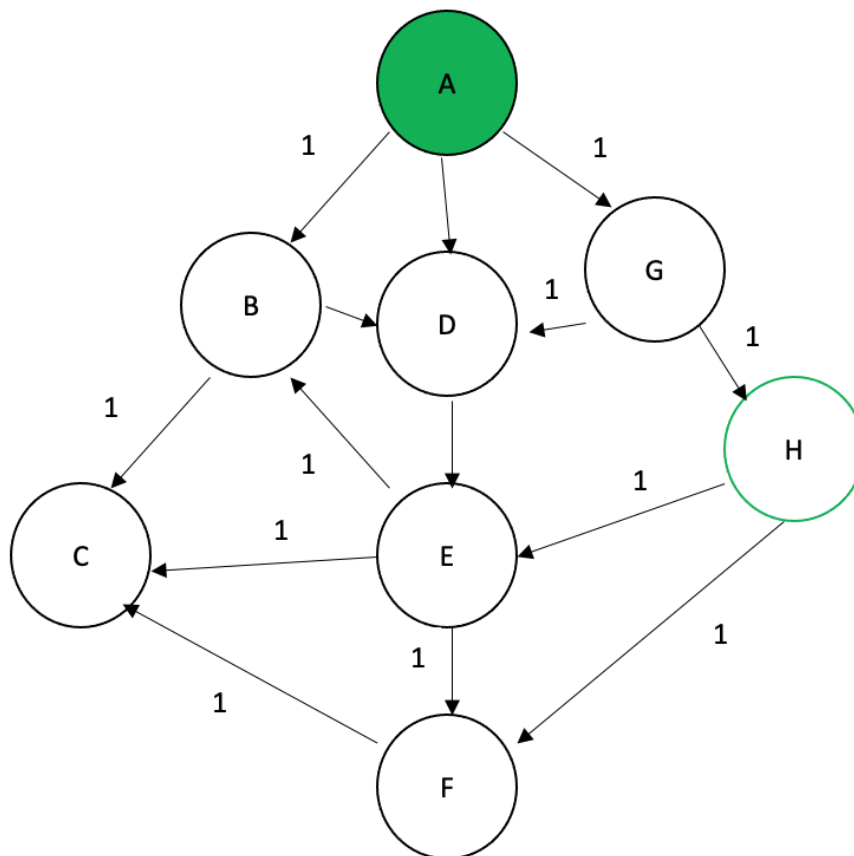


Figura 2 - Diagrama A*.

Através da figura apresentada, podemos concluir que o algoritmo A* escolhe sempre o melhor caminho com base na informação obtida.

Existem vários caminhos que podiam chegar ao local final (H), mas com base na informação disponibilizada, estes não eram os mais rentáveis apenas o caminho A-G-H é o que tem melhor desempenho.

2. Descrição da solução

Neste subcapítulo vamos apresentar os pontos fulcrais do código PROLOG onde iremos retratar a base de conhecimento e os métodos usados para a solução final.

2.1. Base de conhecimento

Relativamente à base de conhecimento do nosso projeto e com o objetivo de retratar o conhecimento necessário à implementação do problema foi analisado problema e foi necessária a criação de 5 entidades principais. Foram realizadas outras entidades, mas achamos estas as mais fundamentais ao funcionamento eficiente do nosso programa: Client, Driver, Order, Delivery e City. De seguida apresentamos um excerto da base de conhecimento de cada uma delas.

2.1.1. Cliente

Quanto ao cliente este é guardado na base de conhecimento com um código que o caracteriza, o seu nome e o concelho onde este habita. Na figura 3 é apresentada a nossa base de conhecimento do **cliente**.

```
%%% CLIENTS (ClientID,ClientName,City)
client(cli1,'Rui','Fafe').
client(cli2,'Pedro','Braga').
client(cli3,'Hugo','Vila Verde').
client(cli4,'Marco','Barcelos').
client(cli5,'Ana','Esposende').
client(cli6,'Maria','Amares').
client(cli7,'Paulo','Guimaraes').
client(cli8,'David','Vizela').
client(cli9,'Joana','Terras de Bouro').
client(cli10,'Afonso','Fafe').
```

Figura 3 - Base de conhecimento (Cliente).

2.1.2. Estafeta

Quanto ao estafeta este é guardado na base de conhecimento com o seu nome, o estado deste, ou seja, se este se encontra a realizar uma entrega ou se está disponível e as encomendas que já realizou. Na figura 4 é apresentada a nossa base de conhecimento do **estafeta**.

```
%%% DRIVERS (driver,state,[order])
driver('Tiago','Available',[ord1,ord8,ord12,ord19,ord20,ord21,ord34,ord35]).
driver('Filipe','Unavailable',[ord2,ord6,ord7,ord18,ord23,ord29,ord30,ord39]).
driver('Joao','Unavailable',[ord3,ord11,ord15,ord17,ord22,ord28,ord36,ord37]).
driver('Ines','Available',[ord4,ord10,ord14,ord16,ord26,ord27,ord33,ord38]).
driver('Daniel','Available',[ord5,ord9,ord13,ord24,ord25,ord31,ord32,ord40]).
```

Figura 4 - Base de conhecimento (Estafeta).

2.1.3. Encomendas entregues

Quanto às encomendas estas estão guardadas em 2 bases distintas, uma com as encomendas já realizadas de modo a ser mais prático aceder a estas assim como às classificações dos estafetas e data de quando foram entregues e outra com todas as encomendas, tanto as entregues como por entregar com um argumento que as identifica. Quanto às encomendas entregues estas são guardadas na base de conhecimento com o nome do estafeta que realizou a entrega, a classificação dada pelo cliente, o número da encomenda, a data de entrega e o veículo utilizado. Na figura 5 é apresentada a base de conhecimento das **encomendas entregues**.

```
%%%%%%%% DELIVERYS (driver,classification,orderId,date,vehicle)

delivery('Tiago', 5, ord1, "2021-12-05", bike).
delivery('Filipe', 4, ord2, "2021-12-05", car).
delivery('Joao', 5, ord3, "2021-12-05", car).
delivery('Ines', 4, ord4, "2021-12-05", motorcycle).
delivery('Daniel', 4, ord5, "2021-12-05", bike).
delivery('Filipe', 3, ord6, "2021-12-06", car).
delivery('Filipe', 5, ord7, "2021-12-06", motorcycle).
delivery('Tiago', 5, ord8, "2021-12-06", car).
delivery('Daniel', 5, ord9, "2021-12-06", car).
delivery('Ines', 5, ord10, "2021-12-06", bike).
delivery('Joao', 3, ord11, "2021-12-06", motorcycle).
```

Figura 5 - Base de conhecimento (Encomendas entregues).

2.1.4. Encomendas

A outra base de conhecimento das entregas onde estão presentes todas as encomendas entregues e não entregues é guardada na base de conhecimento com o número da encomenda, o identificador do cliente, o peso da encomenda, o volume da encomenda, o estado da encomenda (se foi entregue ou não), o custo monetário, a data de quando foi realizada a encomenda e a data prevista de entrega. Na figura 6 é apresentada a base de conhecimento de todas as **encomendas**.

```
%%%%%%%% ORDERS (orderId, clientID, weight,volume,state,price,orderDate,deliveryDate)

order(ord1, cli1, 2, 10, "Delivered", 5, "2021-12-02", "2021-12-05").
order(ord2, cli2, 8, 20, "Delivered", 10, "2021-12-02", "2021-12-05").
order(ord3, cli3, 5, 20, "Delivered", 10, "2021-12-02", "2021-12-06").
order(ord4, cli4, 10, 15, "Delivered", 20, "2021-12-02", "2021-12-06").
order(ord5, cli5, 2, 6, "Delivered", 5, "2021-12-02", "2021-12-06").
order(ord6, cli6, 16, 10, "Delivered", 20, "2021-12-03", "2021-12-06").
order(ord7, cli7, 22, 25, "Delivered", 25, "2021-12-03", "2021-12-06").
order(ord8, cli8, 6, 12, "Delivered", 10, "2021-12-03", "2021-12-06").
```

Figura 6 - Base de conhecimento (Encomendas).

2.1.5. Cidades

Quanto à última base de conhecimento que vamos apresentar neste relatório esta corresponde às cidades que fazem o nosso grafo. Estão guardadas na base de conhecimento da seguinte maneira: cidade, coordenada X e coordenada Y. Apesar de acharmos esta a mais importante também convém salientar a outra base de conhecimento que trata das distâncias entre as cidades. Na figura 7 é apresentada a base de conhecimento das **cidades**. Na figura apenas está presente um excerto.

```
%%% CITY(city,x,y)

city('Centro',0,0).
city('Esposende',5,30).
city('Vila Verde',40,50).
city('Barcelos',20,30).
city('Vila Nova Famalicao',25,10).
city('Terras de Bouro',65,60).
city('Amares',58,40).
```

Figura 7 - Base de conhecimento (Cidades).

2.2. Interface do utilizador

Quanto à solução final decidimos implementar um menu em que quando o utilizador corre o programa este pode chamar de modo a ser mais prática a execução de certos métodos. Na figura seguinte é apresentado o menu com o qual o utilizador se depara ao correr o programa.

```
% c:/Users/rui_f/Desktop/MIETI_2122/Inteligencia_Artificial/Inteligencia_Artificial/main.pl compiled 0.00 sec, 201 clauses
?- menu().

----- ECO DELIVERY -----

- 1 - Registrar um cliente.
- 2 - Efetuar uma encomenda.
- 3 - Completar uma encomenda.
- 4 - Ver estado de uma encomenda.
- 5 - Estafeta que utilizou mais vezes o meio de transporte mais ecologico(bicicleta).
- 6 - Estafeta que entregou determinada encomenda.
- 7 - Identificar os Clientes servidos por um determinado Estafeta.
- 8 - Calcular o Valor Faturado pela Eco Delivery num determinado dia.
- 9 - Identificar qual o concelho do distrito de Braga com maior volume de encomendas por parte da Eco Delivery.
- 10 - Identificar qual o estafeta com maior volume de encomendas por parte da Eco Delivery.
- 11 - Calcular a Classificacao Media de um determinado estafeta.
- 12 - Numero de Encomendas Entregues.
- 13 - Numero de Encomendas por entregar e encomendas a ser entregues.
- 14 - Numero de entregas por parte da Eco Delivery num determinado periodo.
- 15 - Sair.

|:
```

Figura 8 - Interface do utilizador.

O utilizador é apresentado com um conjunto de opções que dependendo do número escolhido por este realiza o método correspondente no código. Para a implementação deste menu utilizamos uma função 'goto' que tendo em conta o número escolhido pelo utilizador corre o método pretendido. Relativamente aos métodos por nós implementados como são vários apenas iremos ilustrar neste relatório os mais importantes como é o caso do registo de um cliente, da realização de uma encomenda, da entrega de uma encomenda e métodos de procura de conhecimento como qual o estafeta que mais vezes utilizou o meio de transporte mais ecológico e o número de entregas por parte da empresa num determinado período. Métodos estes considerados obrigatórios no enunciado do projeto. Os métodos apresentados não foram os únicos implementados apenas achamos que estes ilustrem bem o que era pretendido e se for oportuna a revisão dos restantes métodos é necessário consultar o código-fonte.

2.2.1. Registo de um cliente

Quanto ao primeiro método, o de registar um cliente este é realizado com base na função assertz do PROLOG que o que faz é adicionar no final da base de conhecimento o predicado por nós passado. Este método é ilustrado na figura seguinte.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Method 1 - Client registration

method1(ClientID,ClientName,City) :-
    assertz(client(ClientID,ClientName,City)).
```

Figura 9 - Método para o registo de um cliente.

2.2.2. Realização de uma encomenda

O segundo método é o da realização de uma encomenda. Este método é mais complexo que o primeiro visto ser necessário várias operações. Primeiro é necessário utilizar o algoritmo *A Estrela* para encontrar o caminho mais curto entre o centro de entregas e a localização da encomenda e entre a localização da encomenda e o destino. Após obtermos a distância deste percurso escolhemos qual o veículo de acordo com o peso e o volume da encomenda e após isso calculamos o seu custo. Com estas características procuramos um estafeta que esteja disponível para realizar a encomenda e atribuímos-lhe esta adicionando-a à base de conhecimento com o estado “Em distribuição”. Na figura seguinte apresentamos este método assim como as funções que o complementam.

```
method2(OrderID,ClientID,Weight,Volume,Start_location,Destination,Deadline,OrderType) :-
    add_order(OrderID,ClientID,Weight,Volume,Start_location,Destination,Deadline,OrderType).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
add_order(OrderID, ClientID, Weight, Volume, Start_location, Destination, Deadline,OrderType):-
    aStar('Centro',Start_location,Caminho1,Distance1),
    aStar(Start_location,Destination,Caminho2,Distance2),
    Distance is (Distance1 + Distance2),
    choosevehicle(Weight,Distance,Volume,Vehicle),
    calculate_price(Distance, Vehicle, Deadline, Price),
    selectDriver(OrderID),
    assertz(order(OrderID,ClientID,Weight,Volume,'Delivering',Price,"2021-12-10",Deadline)) -> true.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Choose vehicle %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
choosevehicle(Weight,Distance,Volume,Vehicle):-
    vehicle(bike,_,_,MaxWeight),
    Weight <= MaxWeight,Distance <= 20,Volume <= 5, Vehicle=bike;
    vehicle(motorcycle,_,_,MaxWeight),
    Weight <= MaxWeight,Distance <= 50,Volume <= 10, Vehicle=motorcycle;
    vehicle(car,_,_,MaxWeight),
    Weight <= MaxWeight, Vehicle=car.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% calculate price %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
calculate_price(Distance, Vehicle, Deadline, Price):-
    Price is (Distance*(0.5)).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% select driver %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
selectDriver(OrderID) :-
    findall(Driver, driver(Driver,'Available',Lista), ListDrivers),
    [H|T] = ListDrivers,
    findall(Orders, driver(H,_,Orders), Lista),
    nth0(0,Lista,Encomendas),
    add_tail(Encomendas,OrderID,L),
    retract(driver(H,_,_)),
    asserta(driver(H,'Unavailable',L)) -> true.

add_tail([],X,[X]).
add_tail([H|T],X,[H|L]):-add_tail(T,X,L).
```

Figura 10 - Método realização de uma encomenda.

2.2.3. Entrega de uma encomenda

Quanto ao terceiro método que iremos abordar é o método de entrega de uma encomenda. É um método menos complexo que o anterior, no entanto não foi muito simples a sua implementação até porque ainda contém um erro. Este erro é o de não conseguirmos obter o veículo utilizado para a entrega da encomenda. Uma das maneiras de resolver isto era através da inserção do método de entrega na base de conhecimento das encomendas não entregues, no entanto não conseguimos resolver isso por falta de tempo e por não acharmos ser a solução ótima pois o veículo não necessita de ser indicado aquando da realização da encomenda. Na figura seguinte encontra-se o método utilizado para a confirmação de uma encomenda onde é utilizado o método retract (retira um predicado da base de conhecimento) e asserta (insere um predicado no início da base de conhecimento).

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Method 3 - Complete Order

method3(OrderID, Classification, Date) :-
    findall(Driver, driver(Driver, 'Unavailable', Lista), ListDrivers),
    [H|T] = ListDrivers,
    retract(driver(H, _, L)),
    asserta(driver(H, 'Available', L)),
    retract(order(OrderID, _, _, _, _)),
    asserta(order(OrderID, _, _, "Delivered", _)),
    assertz(delivery(H, Classification, OrderID, Date, car)).
```

Figura 11 - Método de confirmação de uma encomenda.

2.2.4. Estafeta que utilizou o transporte mais ecológico

Relativamente aos métodos de procura de conhecimento vamos apresentar 2 exemplos começando por procurar qual o estafeta que utilizou o meio de transporte mais ecológico, a bicicleta. Este método está presente na figura seguinte.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Method 5 - Driver who used the most ecologic vehicle (bike)

method5(Driver) :-
    getDriversBike(Drivers),
    sort(Drivers, Uniq),
    findall([Freq, X], (member(X, Uniq), include(=(X), Drivers, Y), length(Y, Freq)), Freqs),
    sort(Freqs, SFreqs),
    last(SFreqs, [Freq, Driver]).

getDriversBike(Drivers) :-
    findall(DriverName, (delivery(DriverName, _, _, bike)), Drivers).
```

Figura 12 - Método qual o estafeta que mais vezes utilizou a bicicleta.

2.2.5. Número de entregas num período de tempo

Quanto ao segundo método de procura de conhecimento é o número de entregas por parte da empresa num determinado período, apresentado na figura seguinte.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Method 14 - Number of orders between 2 dates

method14(Init, Final, Number) :-
    filterTime(Init, Final, Orders),
    length(Orders, Number).

filterTime(Init, Final, Orders) :-
    findall(Delivery, filt(Init, Final, Delivery), Orders).

filt(Init, Final, Delivery) :-
    delivery(_, _, X, _),
    Delivery = delivery(_, _, X, _),
    parse_time(Init, Ini),
    parse_time(Final, Fin),
    parse_time(X, Y),
    Y >= Ini,
    Y <= Fin.
```

Figura 13 - Método que devolve o número de entregas num período de tempo.

Testes e discussão de resultados

Após o desenvolvimento e implementação da nossa solução decidimos realizar vários testes para verificar se tudo funcionava como era pretendido, ou seja, se os métodos por nós realizados davam os resultados que era suposto.

Um dos testes que realizamos e decidimos ilustrar no relatório do projeto foi o de registar um novo cliente e este cliente realizar uma encomenda. Depois de confirmada a receção desta encomenda este pode ver qual o custo da encomenda e ver o estado desta. Na figura seguinte podemos observar o registo de um cliente com sucesso, a realização de uma encomenda através do método 2 do menu e após isso a confirmação da entrega da encomenda com o método 3. Finalmente utilizamos o método 4 para observar que a encomenda *ord41* realizada pelo cliente *cli11* custou 81.5 euros.

```
% c:/Users/ruif/Desktop/MIETI_2122/Inteligencia_Artificial/Inteligencia_Artificial/main.pl compiled 0.02 sec, 201 clauses
?- menu().

----- ECO DELIVERY -----

- 1 - Registar um cliente.
- 2 - Efetuar uma encomenda.
- 3 - Completar uma encomenda.
- 4 - Ver custo e estado de uma encomenda.
- 5 - Estafeta que utilizou mais vezes o meio de transporte mais ecologico(bicicleta).
- 6 - Estafeta que entregou determinada encomenda.
- 7 - Identificar os Clientes servidos por um determinado Estafeta.
- 8 - Calcular o Valor Faturado pela Eco Delivery num determinado dia.
- 9 - Identificar qual o concelho do distrito de Braga com maior volume de encomendas por parte da Eco Delivery.
- 10 - Identificar qual o estafeta com maior volume de encomendas por parte da Eco Delivery.
- 11 - Calcular a Classificacao Media de um determinado estafeta.
- 12 - Numero de Encomendas Entregues.
- 13 - Numero de Encomendas por entregar e encomendas a ser entregues.
- 14 - Numero de entregas por parte da Eco Delivery num determinado periodo.
- 15 - Sair.

|: 1.
ID: |: cli11.
Nome : |: 'Rui'.
Cidade: |: 'Fafe'.
Cliente registado .

|: 2.
ID Encomenda : |: ord41.
ID Cliente : |: cli11.
Peso : |: 2.
Volume : |: 7.
Concelho onde se encontra : |: 'Fafe'.
Concelho para onde quer enviar : |: 'Esposende'.
Prazo : |: "2021-12-25".
Tipo (0-Food,1-Package): |: 1.
Encomenda realizada .

|: 3.
Qual o ID da encomenda que quer completar? |: ord41.
Classificacao? |: 5.
Data de entrega? |: "2021-12-21".
Encomenda registada .

|: 4.
Encomenda que deseja analisar: |: ord41.
O estado da encomenda : ord41 realizada pelo cliente :cli11 encontra se Delivered e o seu custo foi de 81.5 euros
```

Figura 14 - Teste da solução final.

Com isto conseguimos comprovar que a solução por nós criada e implementada funcionava como era pretendido pelo que achamos ter realizado um trabalho dentro dos padrões pedidos no enunciado. Apesar de acharmos que a solução poderia ser melhor do que a atingida pensamos ter realizado um bom trabalho.



Conclusão

Na realização deste trabalho foi importante a boa compreensão do projeto para que a sua planificação fosse eficaz e cuidada, de forma que as tarefas fossem bem distribuídas. A nossa organização permitiu-nos um trabalho mais organizado e mais completo.

Relativamente às dificuldades que foram surgindo ao longo da realização deste trabalho estas foram ultrapassadas com sucesso apesar do desconhecimento inicial da linguagem de programação e da sua documentação.

No término deste relatório, entende-se que a teoria e a prática complementam-se. No decorrer deste trabalho o grupo procurou sempre encontrar a melhor solução para os problemas, sendo o principal objetivo deste trabalho melhorar o desempenho do programa.

Em suma, com a realização deste projeto o grupo conseguiu melhorar a sua performance no manuseamento da linguagem de programação PROLOG, lecionada ao longo da cadeira de Inteligência Artificial para as Telecomunicações, bem como a matéria teórica dada nas aulas da UC.