



Universidade do Minho

Licenciatura em Engenharia Informática

Inteligência Artificial

Trabalho em grupo – 1^a Fase

Grupo 29



Hugo Brandão
(A93287)

Pedro Ferreira
(A93295)

Gonçalo Freitas
(A93297)

Daniel Faria
(A93187)

2 de dezembro de 2021

Conteúdo

1	Resumo	2
2	Introdução	2
3	Descrição do Sistema	3
3.1	Definições Básicas	3
3.2	Conhecimento Positivo	5
3.2.1	Data/Time	5
3.2.2	Veículo	6
3.2.3	Cliente	6
3.2.4	Estafeta	7
3.2.5	Encomenda	7
3.2.6	Entrega	8
4	Querys	9
4.1	Query1	9
4.2	Query2	10
4.3	Query3	10
4.4	Query4	11
4.5	Query5	11
4.6	Query6	12
4.7	Query7	12
4.8	Query8	12
4.9	Query9	13
4.10	Query10	13
5	Conclusões e Sugestões	14

1 Resumo

Neste relatório vamos demonstrar os conhecimentos obtidos na cadeira de Inteligência Artificial, ao longo de todas as aulas e na realização deste trabalho. Com estes conhecimentos conseguimos concluir o desafio proposto, sendo todo este descrito detalhadamente neste relatório.

2 Introdução

Neste primeiro trabalho da cadeira de Inteligência Artificial, foi-nos proposta a elaboração de um projeto que une a distribuição de encomendas e a ecologia, tendo como objetivo desenvolver um software para a empresa *Green Distribution*. A empresa tem ao seu dispor vários meios de transporte (bicicletas, motos e carros), sendo o objetivo deste software privilegiar o meio de entrega mais sustentável.

3 Descrição do Sistema

Nesta secção explicaremos como procedemos à realização do projeto, descrevendo as componentes básicas do sistema e como estas interagem entre si.

3.1 Definições Básicas

- Veículo: Entidade referente aos veículos utilizados na realização das entregas, guardando os tipos de veículo, as suas respetivas velocidades médias e a sua carga máxima.
- Cliente: Entidade que gera pedidos de serviço, guardando o ID do cliente e o total de encomendas realizadas.
- Estafeta: Entidade que faz a prestação dos serviços pedidos, em diferentes veículos, guardando o ID do estafeta, o total de entregas realizadas e a sua avaliação média consoante a qualidade das entregas.
- Encomenda: Entidade referente ao serviço prestado, onde é guardada a informação sobre o status da encomenda, o seu ID, o ID do cliente que fez o pedido de encomenda, o peso e o volume da encomenda, a freguesia do cliente e a sua morada e por fim o prazo para a realização da encomenda.
- Entrega: Entidade que se refere à informação da entrega realizada pela *Green Distribution*, onde é guardado o ID da entrega, o ID do estafeta que realiza a entrega, o veículo usado na entrega, a data do início da entrega e do fim e por fim a avaliação realizada pelo cliente sobre o estafeta.

Assim sendo, estas entidades representam os componentes mais básicos deste sistema, tendo a seguinte estrutura:

- `veiculo`(Tipo, VelocidadeMedia, CargaMax).
- `cliente`(Id, NrEncomendas).
- `estafeta`(Id, NrEntregas, Avaliacao).
- `encomenda`(Estado, Id, IdCliente, Peso, Volume, Freguesia/Morada, Prazo).
- `entrega`(IdEncomenda, IdEstafeta, Veiculo, DataInicio, DataFim, Avaliacao).

Os atributos *Prazo*, *DataInicio* e *DataFim* são representado por um par *date(Y,M,D)/time(H,M,S)*:

Nos atributos *DataInicio* e *DataFim* escolhemos não generalizar os segundos do tempo, sendo estes sempre igual a 0. No atributo *Prazo*, escolhemos aceitar definir um máximo de 30 dias para o prazo, mantendo sempre Y=0 e M=0.

3.2 Conhecimento Positivo

De modo a poder realizar a inserção de conhecimento na Base de Conhecimento, certas regras têm de ser consideradas. Iremos descrever então as seguintes regras de forma detalhada.

3.2.1 Data/Time

O predicado *date* segue o formato (AAAA, MM, DD). Para tal, consideramos que o mês se encontra entre 1 e 12, e que dependendo de cada mês o número de dias varia(28,29,30,31). Ou seja, tivemos de ter em conta os anos bissextos de modo a conseguir atribuir o número de dias certos correspondente ao ano em questão.

```
verBissexto(Ano, Max) :-  
    (Mod1 is Ano mod 4, Mod2 is Ano mod 100, Mod3 is Ano mod 400) ,  
    (Mod1==0, (Mod2\= 0 ; Mod3==0)) -> Max is 29;  
    Max is 28.  
  
validateDate(A, 2, D) :-  
    D>0, verBissexto(A, Max), D=<Max, !.  
validateDate(_, M, D) :-  
    M>0, M=<12, mes(M, Max), D>0, D=<Max.
```

Figura 1: Predicado date

O predicado *time* segue o formato (HH, MM, SS). Para tal, consideramos que a hora se encontra entre 0 e 24 e os minutos e os segundos entre 0 e 60.

```
validateTime(H, M, S) :-  
    H<24, H>=0, M<60, M>=0, S<60, S>=0.
```

Figura 2: Predicado time

Temos também de fazer verificação se o prazo não ultrapassa os 30 dias.

```
% dizer que prazo máximo é de 30 dias
convertTime(Val, Date/Time) :-
    Val>0, rounding(Val, D, Dec1),
    rounding(Dec1*24, H, Dec2),
    rounding(Dec2*60, M, _),
    Date = date(0, 0, D), Time = time(H, M, 0).
```

Figura 3: convertTime

3.2.2 Veículo

Quanto ao predicado *veiculo* este tem o formato (tipo, velocidadeMedia, cargaMax), restringindo-se aos três tipos de veículos, referidos no enunciado do projeto, e à sua respetiva velocidade e carga máxima. Não podendo adicionar novos veículos à Base de Conhecimento

```
% ----- veiculo(tipo, velocidadeMedia, cargaMax)
veiculo(bicicleta, 10, 5).
veiculo(mota, 35, 20).
veiculo(carro, 25, 100).
```

Figura 4: Predicado veiculo

3.2.3 Cliente

Este predicado tem como formato: (Id, NrEncomendas), tendo cada cliente um ID diferente. Sendo possível adicionar novos clientes à Base de Conhecimento, é necessário verificar se o cliente já se encontra na mesma. Sendo também atualizado o número de entregas e a avaliação a cada entrega realizada.

```
createCliente(Id) :- \+cliente(Id, _), assert(cliente(Id, 0)), write("Cliente adicionado"), !;
                    write("Cliente já existente").
```

Figura 5: createCliente

3.2.4 Estafeta

Quanto ao estafeta, o seu formato é: (Id, NrEntregas, Avaliacao). Para adicionar um estafeta é necessário apenas um ID que ainda não tenha sido utilizado noutra estafeta. Sendo também atualizado o número de entregas e a avaliação a cada entrega realizada.

```
createEstafeta(Id) :- \+estafeta(Id, _, _), assert(estafeta(Id, 0, 0)), write("Estafeta adicionado"), !;
                        write("Estafeta já existente").

addEstafeta(Id,Av) :- estafeta(Id,Nr,Media), NewNr is Nr + 1, arredondar(((Media*Nr + Av)/NewNr), NewMedia, 2),
                        replace_existing_fact(estafeta(Id,Nr,Media),estafeta(Id,NewNr,NewMedia)).
```

Figura 6: createEstafeta

3.2.5 Encomenda

O predicado *Encomenda* tem como formato (Estado, Id, IdClinet, Peso, Volume, Freguesia/Morada, prazo). Para adicionar uma encomenda à Base de Conhecimento, são necessários muitos testes:

- Verificar se o ID da encomenda já foi utilizado noutra encomenda.
- O cliente tem de existir na Base de Conhecimento.
- Não pode ter um peso superior a 100 kg nem inferior a 0 kg.
- Tem de ter um volume válido(superior a 0 kg).
- Tem de ter uma data válida.

```
createEncomenda(Id, IdCliente, Peso, Volume, Freguesia, Morada, Dias, Horas, Minutos, date(A1, M1, D1)/time(H2, M2, S2)) :-
    encomenda(_, Id, _, _, _, _), write("Encomenda já existente"), !;
    \+cliente(IdCliente, _), write("Cliente não existente"), !;
    Peso > 100, write("Nenhum veiculo suporta a entrega da encomenda"), !;
    Peso < 0, write("Peso inválido"), !;
    Volume < 0, write("Volume inválido"), !;
    (Dias < 0; Horas < 0; Minutos < 0), write("Prazo impossível"), !;
    \+validateDate(A1, M1, D1), \+validateTime(H2, M2, S2), write("Data ou horas de Inicio erradas"), !;

    assert(encomenda(registada, Id, IdCliente, Peso, Volume, Freguesia/Morada, date(0,0,Dias)/time(Horas,Minutos,0))),
    assert(entrega(Id,empty,empty,date(A, M, D)/time(H, M, S),empty/empty/empty)),
    addCliente(IdCliente).
```

Figura 7: createEncomenda

De modo a registar as encomendas realizadas na lista de entregas, elaborámos o predicado *entregarEncomenda* que verifica se a encomenda existe, se já foi entregue, se ainda não se encontra em distribuição e se as datas e as avaliações são válidas.

```
entregarEncomenda(IdEncomenda, Dia, Mes, Ano, Horas, Minutos, Avaliacao) :-
    \+encomenda(_, IdEncomenda, _, _, _, _), write("Encomenda não existente"), !;
    encomenda(entregue, IdEncomenda, _, _, _, _), write("A encomenda já foi entregue"), !;
    encomenda(registada, IdEncomenda, _, _, _, _), write("A encomenda ainda não está em distribuição"), !;
    \+validateDate(Ano, Mes, Dia), \+validateTime(Horas, Minutos, 0), write("Data ou horas de Fim erradas"), !;
    (Avaliacao<0; Avaliacao>5), write("Avaliacao invalida"), !;

    replace_existing_fact(encomenda(distribuicao, IdEncomenda, IdCliente, Peso, Volume, ADDRS, Prazo),
                          encomenda(entregue, IdEncomenda, IdCliente, Peso, Volume, ADDRS, Prazo)),
    replace_existing_fact(entrega(IdEncomenda, IdEstafeta, Veiculo1, DataInicio1, _),
                          entrega(IdEncomenda, IdEstafeta, Veiculo1, DataInicio1,
                                  date(Ano, Mes, Dia)/time(Horas, Minutos, 0), Avaliacao)),
    addEstafeta(IdEstafeta, Avaliacao).
```

Figura 8: entregarEncomenda

3.2.6 Entrega

No predicado de *Entrega* o formato é: (idEncomenda, idEstafeta, veiculo, DataInicio, DataFim, avaliacao) e são verificados os seguintes testes:

- Se a encomenda existe.
- Se o estafeta existe.
- Se a encomenda já foi entregue ou se se encontra em distribuição.
- Se o peso da encomenda é válido para o veículo.

```
createEntrega(IdEncomenda, IdEstafeta, Veiculo) :-
    \+encomenda(_, IdEncomenda, _, _, _, _), write("Encomenda não existente"), !;
    \+estafeta(IdEstafeta, _), write("Estafeta não existente"), !;
    encomenda(entregue, IdEncomenda, _, _, _, _), write("A encomenda já foi entregue"), !;
    encomenda(distribuicao, IdEncomenda, _, _, _, _), write("A encomenda já se encontra em distribuição"), !;

    encomenda(_, IdEncomenda, _, _, Peso, _), veiculo(Veiculo, _, Max),
    Max < Peso, write("Veiculo selecionado não suporta carga da encomenda"), !;

    % adicionar info aos outros factos !!!!
    replace_existing_fact(encomenda(registada, IdEncomenda, IdCliente, Peso, Volume, Freguesia/Morada, Date/Time),
                          encomenda(distribuicao, IdEncomenda, IdCliente, Peso, Volume, Freguesia/Morada, Date/Time)),
    replace_existing_fact(entrega(IdEncomenda, _, _, DateI/TimeI, InfoFim, Aval ),
                          entrega(IdEncomenda, IdEstafeta, Veiculo, InfoFim, Aval )),
    % add potencial -> atribuicao
    write("Entrega registada").
```

Figura 9: createEntrega

4 Querys

De maneira a facilitar a utilização das funcionalidades pretendidas, utilizamos predicados que utilizam os predicados da query pretendida, como podemos observar pela figura seguinte.

```
query1(Ans) :- maisEcologico(Ans).
query2(Cliente, Ans) :- trackEncomenda(Cliente, Ans).
query3(Estafeta, Ans) :- findClientesServidosPorEstafeta(Estafeta, Ans).
query4(Dia, Mes, Ano, Ans) :- calcFaturacao(Dia, Mes, Ano, Ans).
query5(Tag, Top1, Top2, Top3) :- bestZonas(Tag, Top1, Top2, Top3). % tag pode ser freguesia ou rua
query6(Estafeta, Ans) :- calcularMediaSatisfacaoEstafeta(Estafeta, Ans).
query7(DiaI, MesI, AnoI, Hi, Mi, DiaF, MesF, AnoF, Hf, Mf, Ans) :- nrEntregasPorTransporte(DiaI/MesI/AnoI, Hi:Mi, DiaF/MesF/AnoF, Hf:Mf, Ans).
query8(DiaI, MesI, AnoI, Hi, Mi, DiaF, MesF, AnoF, Hf, Mf, Ans) :- nrEntregasPorEstafeta(DiaI/MesI/AnoI, Hi:Mi, DiaF/MesF/AnoF, Hf:Mf, Ans).
query9(DiaI, MesI, AnoI, Hi, Mi, DiaF, MesF, AnoF, Hf, Mf, Ans) :- numEncomendas(date(AnoI, MesI, DiaI), time(Hi, Mi, 0),
                                                                    date(AnoF, MesF, DiaF), time(Hf, Mf, 0), Ans).
query10(Estafeta, Dia, Mes, Ano, Ans) :- pesoNumDia(Estafeta, Dia/Mes/Ano, Ans).
```

Figura 10: Querys

Iremos abordar agora cada uma das querys.

4.1 Query1

A query1 tem por base identificar o estafeta que utilizou mais vezes um meio de transporte mais ecológico, a definição de ecológico que utilizamos foi dependente do tipo de veiculo que o estafeta utiliza consoante o peso da encomenda, ou seja, consideramos que uma estafeta que utilizou um carro para entregar uma encomenda de 1Kg menos ecológico do que um estafeta que utilizou uma bicicleta para realizar uma entrega semelhante.

Consoante a base de conhecimento por nós estabelecida, realizamos esta query pelo que o estafeta mais ecológico é o 'amazon', como podemos constatar pela figura abaixo.

```
?- query1(X).
X = amazon ;
false.
```

Figura 11: Query 1 - maisEcologico

4.2 Query2

A query2 tem por base identificar que estafetas entregaram determinada(s) encomenda(s) a um determinado cliente, em suma a query 2 retorna uma lista em que cada elemento dessa lista tem um duplo, a primeira parte do duplo tem o nome do estafeta que entregou encomenda(s) ao cliente em questão, já a segunda parte do duplo possui uma lista das encomendas que o estafeta em questão(primeira parte do duplo) entregou ao cliente em análise.

Consoante a base de conhecimento por nós estabelecida, realizamos esta query pelo que os estafetas entregaram determinadas encomendas ao cliente 'abacao' foram o 'ctt' que entregou a encomenda 'capa' e o estafeta 'amazon' que entregou uma porta, como podemos constatar pela figura abaixo.

```
?- query2(abacao, X).  
X = [(ctt, [capa]), (amazon, [porta])] ;  
false.
```

Figura 12: Query 2 - trackEcomenda

4.3 Query3

A query3 tem por base identificar os clientes servidos por um determinado estafeta, esta query devolve uma lista dos vários clientes que receberam encomendas do estafeta em questão.

Consoante a base de conhecimento por nós estabelecida, realizamos esta query pelo que os clientes a quem o estafeta 'ctt' fez entregas foram o 'joao', a 'claudia', o 'freitas', o 'daniel' e o 'abacao', como podemos constatar pela figura abaixo.

```
?- query3(ctt, X).  
X = [joao, claudia, freitas, daniel, abacao].
```

Figura 13: Query 3 - findClientesServidosPorEstafeta

4.4 Query4

A query4 tem por base calcular o valor faturado pela Green Distribution num determinado dia, esta query devolve a soma de todos os preços de encomendas realizadas na data selecionada.

Consoante a base de conhecimento por nós estabelecida, realizamos esta query pelo que o valor faturado pela Green Distribution no dia 14 de janeiro de 2021 foi de 19,58, para confirmar calculamos o preço individual das encomendas do dia em questão, como podemos constatar pela figura abaixo.

```
?- query4(14,1,2021, X).  
X = 19.58.  
  
?- calcPreco(boqueira, X), calcPreco(flor, Y), Z is X+Y.  
X = 9.98,  
Y = 9.6,  
Z = 19.58.
```

Figura 14: Query 4 - calcFaturacao

4.5 Query5

Esta query utiliza o predicado *bestZonas* que possui a possibilidade de procurar por melhores freguesias ou ruas.

Consoante a base de conhecimento por nós estabelecida, realizamos esta query pelo que escolhemos procurar as ruas e sendo as melhores zonas por ordem "landim/rua_ponte", "escordo/rua_ponte" e "guimaraes/rua_ponte" como podemos constatar pela figura abaixo.

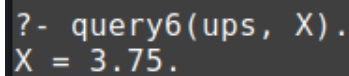
```
?- query5(rua, Top1, Top2, Top3).  
Top1 = landim/rua_ponte,  
Top2 = escordo/rua_ponte,  
Top3 = guimaraes/rua_ponte ;  
false.
```

Figura 15: Query 5 - bestZonas

4.6 Query6

Esta query utiliza o predicado *calcularMediaSatisfacaoEstafeta* que procura por todas as avaliações referentes a um dado estafeta e calculando a média percorrendo uma lista com estes valores.

Consoante a base de conhecimento por nós estabelecida, realizamos esta query pelo que o estafeta *ups* tem uma média de avaliação de 3.75, como podemos constatar pela figura abaixo.



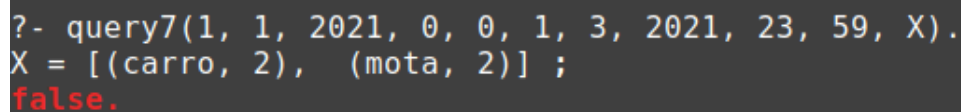
```
?- query6(ups, X).  
X = 3.75.
```

Figura 16: Query 6 - calcularMediaSatisfacaoEstafeta

4.7 Query7

Esta query utiliza o predicado *nrEntregasPorTransporte* que procura todas as encomendas cujas datas de fim se enquadrem no intervalo inserido, criando uma lista com todos os veiculos utilizados, os quais são posteriormente organizados em pares de (Veiculo, Repetições na lista).

Consoante a base de conhecimento por nós estabelecida, realizamos esta query pelo que entre os dias 01/01/2021 às 00h00 e 01/03/2021 às 23:59 foram entregues duas encomendas de *carro* e outras duas de *mota*, como podemos constatar pela figura abaixo.



```
?- query7(1, 1, 2021, 0, 0, 1, 3, 2021, 23, 59, X).  
X = [(carro, 2), (mota, 2)] ;  
false.
```

Figura 17: Query 7 - nrEntregasPorTransporte

4.8 Query8

Esta query utiliza o predicado *nrEntregasPorEstafeta* utilizando um algoritmo similar à query anterior.

Consoante a base de conhecimento por nós estabelecida, realizamos esta query pelo que entre os dias 01/01/2021 às 00h00 e 01/03/2021 às 23:59 foram entregues duas encomendas por *ctt* e outras duas por *ups*, como podemos constatar pela figura abaixo.

```
?- query8(1, 1, 2021, 0, 0, 1, 3, 2021, 23, 59, X).  
X = [(ctt, 2), (ups, 2)] ;  
false.
```

Figura 18: Query 8 - nrEntregasPorEstafeta

4.9 Query9

Esta query utiliza o predicado *numEncomendas* consideramos todas as encomendas não entregues aquelas em que a data de início esteja incluída no intervalo inserido, mas que ainda se encontrem em "distribuição". As encomendas entregues são aquelas em que a data de fim está incluída no intervalo inserido, possuindo a encomenda respetiva o estado de "entregue".

Consoante a base de conhecimento por nós estabelecida, realizamos esta query pelo que entre os dias 01/01/2021 às 00h00 e 01/03/2021 às 23:59 foram entregues quatro encomendas e não entregues três encomendas. como podemos constatar pela figura abaixo.

```
?- query9(1, 1, 2021, 0, 0, 1, 3, 2021, 23, 59, X).  
X = [(entregues, 4), (nao_entregues, 3)].
```

Figura 19: Query 9 - numEncomendas

4.10 Query10

Esta query utiliza o predicado *pesoNumDia* que procura todas as encomendas cujas datas de inicio e datas de fim englobam o dia inserido.

Consoante a base de conhecimento por nós estabelecida, realizamos esta query pelo que foram transportados 2 kg por 'ups' no dia 14/01/2021, como podemos constatar pela figura abaixo.

```
?- query10(ups, 14, 1, 2021, X).  
X = 2.
```

Figura 20: Query 10 - pesoNumDia

5 Conclusões e Sugestões

De um forma geral, consideramos que o nosso trabalho foi bem sucedido. Fomos capazes de criar um sistema capaz de suportar as funcionalidades pedidas. Para além do mais, implementamos, também, funcionalidades secundárias que permitem uma melhor utilização do sistema.