

# Técnicas de Inteligência Artificial 22/23

Universidade do Minho

Fábio da Silva Ferreira  
A92961



João Pedro Araújo Pereira  
A92937



Ricardo Jorge Cruz Oliveira  
A72349



Ana Leonor Lopes Machado  
A96521



# Índice

Projeto 1 .....	1
1.1 - Introdução .....	1
1.1.1 - Enquadramento.....	1
1.1.2 - Objetivos.....	1
1.2 - Execução do Projeto .....	2
1.3 - Tarefa A .....	3
1.4 - Tarefa B .....	5
1.6 - Conclusão .....	8
1.6.1 - Síntese .....	8
1.6.2 - Discussão .....	9
Projeto 2 .....	10
2.1 – Introdução .....	10
2.1.2 - Objetivos.....	10
2.2 - Execução do Projeto .....	10
2.3 - Tarefa A .....	11
2.4 - Tarefa B .....	12
Extra.....	13
2.6 - Conclusão .....	14
2.6.1 - Síntese .....	14
2.6.2 - Discussão .....	14
2.6.3 Funcionamento do Trabalho em Grupo.....	15
Anexo 1.....	15
Parte A .....	15
Parte B: .....	25
Anexo 2:.....	33
Parte A: .....	33
Parte B; .....	35
Contrato de grupo .....	40

# Projeto 1

## 1.1 - Introdução

O projeto proposto, intitulado "Recomendação de Meio de Transporte", tem como objetivo desenvolver um Sistema Baseado em Conhecimento (SBC) capaz de aconselhar sobre o meio de transporte mais adequado para uma determinada viagem. O SBC será implementado em Prolog. A solução deverá ser implementada seguindo a metodologia de desenvolvimento de SBC, bem como as técnicas de aquisição de conhecimento manuais e automáticas mais apropriadas.

### 1.1.1 - Enquadramento

No âmbito da unidade curricular de Técnicas de Inteligência Artificial foi-nos proposta a realização de dois projetos de Sistemas Baseados em Conhecimento (SBC). Estes projetos tinham como objetivo a recomendação do melhor transporte para viajar de um local de origem para um local de destino tendo em conta algumas especificidades tais como custo, tempo e o tipo de viajante.

Para a realização deste projeto utilizamos a linguagem Prolog, para que fosse possível a aquisição manual e automática de conhecimento, que se enquadra no paradigma da programação em lógica matemática que nos auxilie na identificação de padrões, criação de estruturas de dados em árvore, forward e backtracking.

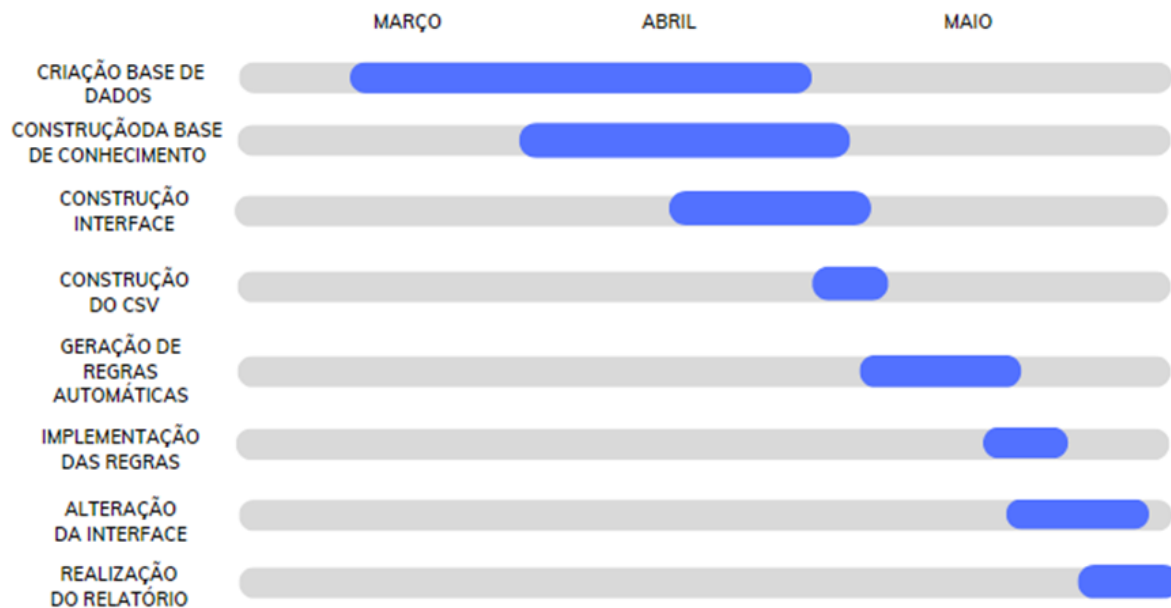
O SBS deste projeto consiste num programa que manipula conhecimento e informações através da inteligência artificial de forma a resolver o problema proposto, envolvendo conhecimento humano e especialização nas matérias e linguagem utilizadas.

Ambos os projetos foram desenvolvidos pela equipa anteriormente identificada, com o auxílio dos docentes desta unidade curricular.

### 1.1.2 - Objetivos

O projecto 1 tem duas partes, na parte A o objectivo é construir um sistema de recomendação de viagens através de um questionário ao utilizador, consoante as respostas dadas ele irá gerar uma solução que mais se adequa às preferências do utilizador. Em relação a parte B, o objectivo é o mesmo, obter uma sistema baseado em conhecimento, com a diferença que os dados são adquiridos automaticamente com auxílio da ferramenta RapidMiner.

## 1.2 - Execução do Projeto



### Contribuição de cada elemento

De seguida iremos descrever algumas tarefas que foram mais repartidas ou individualizadas, apesar de todos os membros terem estado envolvidos na execução do projeto enquanto um todo.

#### Leonor Machado

- Construiu a parte da base de dados e conhecimento que lhe foi atribuída.
- Realizou a construção da interface conjuntamente.
- Realizou as operações do RapidMiner conjuntamente e código subsequente;
- Ajudou na construção do código associado ao “print” do resultado.

#### João Pereira

- Construiu a parte da base de dados e conhecimento que lhe foi atribuída.
- Realizou a construção da interface conjuntamente.
- Realizou as operações do RapidMiner conjuntamente e código subsequente.
- Ajudou na construção do código associado ao “print” do resultado.

## Ricardo Oliveira

- Construiu a parte da base de dados e conhecimento que lhe foi atribuída.
- Realizou a construção da interface conjuntamente.
- Realizou as operações do RapidMiner conjuntamente e código subsequente.
- Ajudou na construção do código associado ao “print” do resultado.

## Fábio Ferreira

- Construiu a parte da base de dados e conhecimento que lhe foi atribuída.
- Realizou a construção da interface conjuntamente.
- Realizou as operações do RapidMiner conjuntamente e código subsequente.
- Ajudou na construção do código associado ao “print” do resultado.

## 1.3 - Tarefa A

Para utilizar a nossa interface primeiro deve escrever “menu.”

```
?- menu.  
  
Sistema de recomendacao de viagens! Descubra a melhor viagem para o seu perfil!  
1 - Entrar no sistema  
2 - Ajuda  
3 - Sair  
|:
```

Depois deve escrever “1.” para entrar no sistema e responder a todas as perguntas do programa.

```

l: 1.
*****
** Em que local esta?
**
** 1 - Guimaraes
** 2 - Chaves
** 3 - Braga
** 4 - Porto
** 5 - Viseu
** 6 - Coimbra
** 7 - Lisboa
** 8 - Faro
l: 1.
*****
** 1 - Guimaraes
** 2 - Chaves
** 3 - Braga
** 4 - Porto
** 5 - Viseu
** 6 - Coimbra
** 7 - Lisboa
** 8 - Faro
l: 4.
*****
** Qual e a sua idade?
** 1 - 0 - 25
** 2 - 25 - 65
** 3 - +65
l: 1.
*****
** Em que estado se encontra?
** 1 - Estudante
** 2 - Trabalhador
** 3 - Reformado
** 4 - Desempregado
l: 1.
*****
** Pertende usar um transporte sustentavel?
**
** 1 - Sim
** 2 - Nao
l: 2.
*****
** Qual pensa ser o seu transporte ideal?
** 1 - autocarro
** 2 - carro
** 3 - comboio
** 4 - aviao
l: 2.
False
*****
**                                     Resultado Obtido
**                                     **
*****

Devia utilizar o transporte: autocarro
A distancia e: 71
A viagem tem um custo de: 14
true .

```

Depois de responder às diferentes questões, o programa vai escolher o transporte ideal e apresentar a distância e o custo. Ainda é possível ver a prova do transporte através da pergunta “Qual pensa ser o seu transporte ideal?” onde vai ser realizado o proof para o transporte que o utilizador escolher (caso não escolha a melhor opção imprime falso).

Exemplo de um proof onde o utilizador “acertou” o transporte:

```

*****
** Qual pensa ser o seu transporte ideal?
** 1 - autocarro
** 2 - carro
** 3 - comboio
** 4 - aviao
l: 1.
autocarro<=(custo_baixo<=jovem and estudante)and(distancia_baixa<=porto and guimaraes)
*****

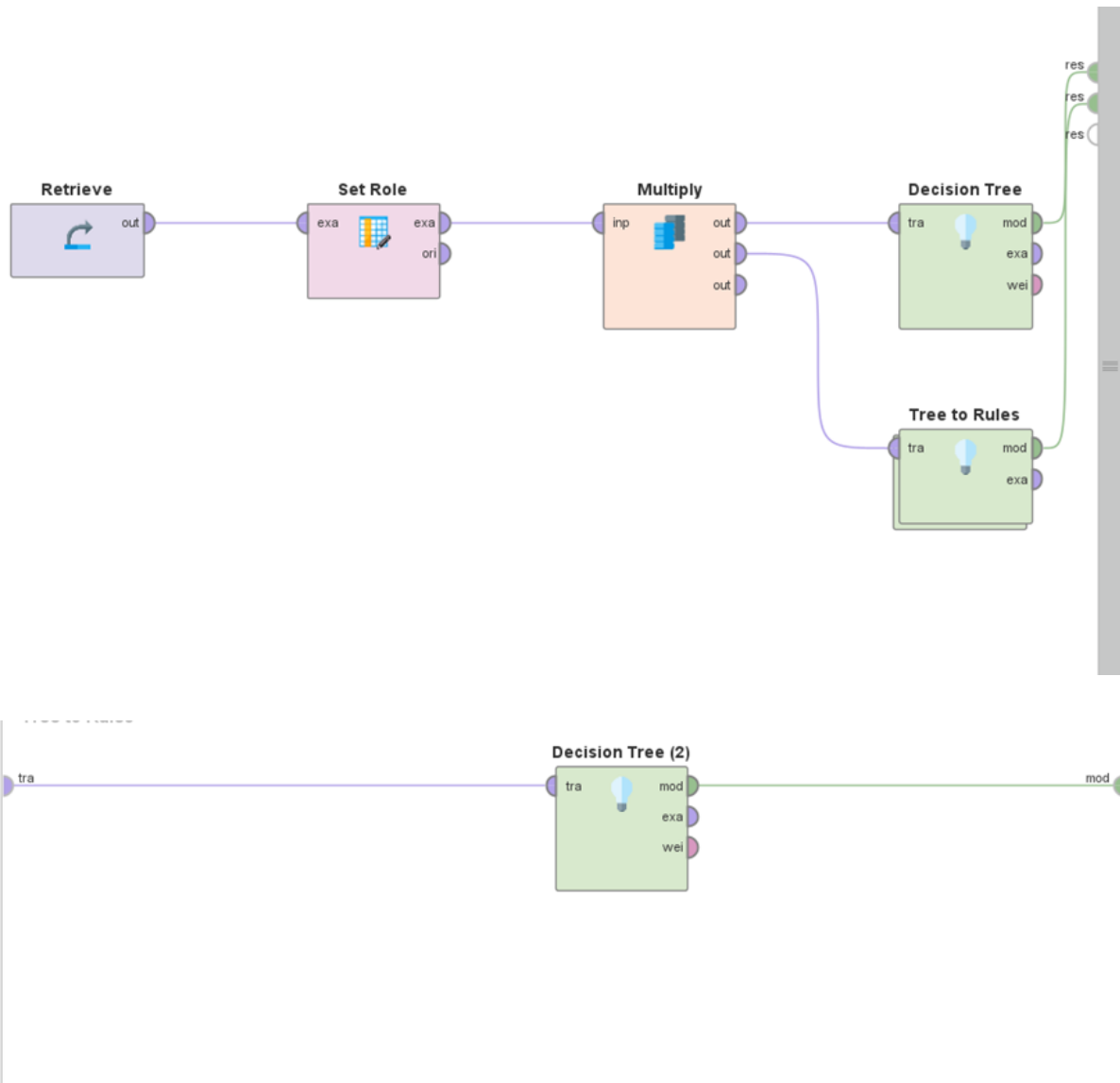
```

Nota: O código desta tarefa encontra-se em anexo.

## 1.4 - Tarefa B

Esta tarefa é muito semelhante à tarefa A, contudo nesta utilizamos o RapidMiner para desenvolver a base de conhecimento. Para isso começamos por desenvolver uma base de dados em Excel para colocar no RapidMiner.

Depois criamos um processo com os operadores “Retrieve”, “Set Role”, Multiply, Tree rules e 2 Decision Tree.



No operador “set role” colocamos o atributo transporte e o target role label.

Depois corremos o processo e o resultado foi o seguinte:

## RuleModel

```

if Preco > 1.500 and Preco > 2.500 and Origem = Braga then Carro (7 / 0 / 0 / 0)
if Preco > 1.500 and Preco > 2.500 and Origem = Coimbra then Autocarro (0 / 1 / 0 / 0)
if Preco > 1.500 and Preco > 2.500 and Origem = Faro then Carro (7 / 0 / 0 / 2)
if Preco > 1.500 and Preco > 2.500 and Origem = Guimaraes then Carro (2 / 2 / 2 / 0)
if Preco > 1.500 and Preco > 2.500 and Origem = Lisboa then Carro (7 / 0 / 0 / 2)
if Preco > 1.500 and Preco > 2.500 and Origem = Porto then Carro (7 / 0 / 0 / 2)
if Preco > 1.500 and Preco > 2.500 and Origem = Viseu then Carro (3 / 3 / 3 / 0)
if Preco > 1.500 and Preco ≤ 2.500 and Origem = Braga then Comboio (0 / 0 / 7 / 0)
if Preco > 1.500 and Preco ≤ 2.500 and Origem = Chaves then Carro (3 / 2 / 0 / 0)
if Preco > 1.500 and Preco ≤ 2.500 and Origem = Faro then Comboio (0 / 0 / 7 / 0)
if Preco > 1.500 and Preco ≤ 2.500 and Origem = Guimaraes then Carro (4 / 4 / 4 / 0)
if Preco > 1.500 and Preco ≤ 2.500 and Origem = Lisboa then Comboio (0 / 0 / 7 / 0)
if Preco > 1.500 and Preco ≤ 2.500 and Origem = Porto then Comboio (0 / 0 / 7 / 0)
if Preco > 1.500 and Preco ≤ 2.500 and Origem = Viseu then Carro (4 / 4 / 4 / 0)
if Preco ≤ 1.500 and Origem = Braga then Autocarro (0 / 7 / 0 / 0)
if Preco ≤ 1.500 and Origem = Chaves then Autocarro (0 / 1 / 0 / 0)
if Preco ≤ 1.500 and Origem = Coimbra then Autocarro (0 / 4 / 1 / 0)
if Preco ≤ 1.500 and Origem = Faro then Autocarro (0 / 7 / 0 / 0)
if Preco ≤ 1.500 and Origem = Guimaraes then Carro (1 / 1 / 1 / 0)
if Preco ≤ 1.500 and Origem = Lisboa then Autocarro (0 / 7 / 0 / 0)
if Preco ≤ 1.500 and Origem = Porto then Autocarro (0 / 7 / 0 / 0)

```

correct: 107 out of 144 training examples.



Para concluir esta parte, criamos uma interface com menos perguntas, uma vez que, agora já não é necessário perguntar a idade, profissão e se o transporte é sustentável. Também criamos uma base de dados nova para esta parte, cuja única diferença é a adição do “o\_” antes de todas as cidades origem (exemplo: viagem('o'o viseu','porto','comboio','77','9')).



?- menu.

Sistema de recomendacao de viagens! Descubra a melhor viagem para o seu perfil!

1 - Entrar no sistema

2 - Ajuda

3 - Sair

|: 1.

\*\*\*\*\*

\*\* Em que local esta?

\*\*

\*\* 1 - Guimaraes

\*\* 2 - Chaves

\*\* 3 - Braga

\*\* 4 - Porto

\*\* 5 - Viseu

\*\* 6 - Coimbra

\*\* 7 - Lisboa

\*\* 8 - Faro

|: 1.

\*\*\*\*\*

\*\*

\*\* 1 - Guimaraes

\*\* 2 - Chaves

\*\* 3 - Braga

\*\* 4 - Porto

\*\* 5 - Viseu

\*\* 6 - Coimbra

\*\* 7 - Lisboa

\*\* 8 - Faro

|: 4.

\*\*\*\*\*

\*\* Qual e a o custo que pretende ter?

\*\* 1 - Custo baixo

\*\* 2 - Custo medio

\*\* 3 - Custo alto

|: 1.

\*\*\*\*\*

\*\* Qual pensa ser o seu transporte ideal?

\*\* 1 - autocarro

\*\* 2 - carro

\*\* 3 - comboio

\*\* 4 - aviao

|: 1.

False

\*\*\*\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

Devia utilizar o transporte: carro

A distancia e: 71

A viagem tem um custo de: 25

true

Nota: O código referente a esta parte encontra-se em anexo.

## 1.6 - Conclusão

### 1.6.1 - Síntese

Na parte A do projeto 1 foi elaborado um SBC para aconselhar o utilizador sobre o meio de transporte mais adequado para uma viagem e para um determinado perfil de viajante ou viagem.

Num primeiro instante, o grupo utilizou o Google Maps para determinar a distância em km das diferentes cidades. Para determinar o custo entre os diferentes transportes, o grupo presumiu alguns custos e recorreu também aos sites de transportes de cada cidade.

Depois a equipa criou a base de dados. Para isso, cada elemento do grupo ficou responsável por estabelecer a “conexão” de duas cidades com as restantes cidades. Desta forma, a base de dados ficou com o seguinte formato: Viagem (origem, destino, transporte, distância, custo). É importante salientar que para todas as viagens entre cidades existe a possibilidade de escolher 3 transportes (carro, autocarro e comboio), exceto para viagens entre cidades que possuem aeroportos onde também foi considerado transporte de avião.

Para a base de conhecimento o grupo considerou que dependendo da idade e da profissão do utilizador o custo da viagem ia ser catalogado como custo baixo, custo médio ou custo alto. Depois, dependendo da distância entre a cidade origem e destino, a viagem pode ser considerada distância baixa, distância média ou distância alta. Para terminar a base de conhecimento é então definido o meio de transporte através da análise do custo e da distância. Caso o utilizador queira um transporte sustentável apenas serão utilizados os transportes de autocarro e comboio.

Por fim, foi implementado o Proof e uma interface amigável via Prolog que auxilia as pessoas a escolherem o transporte que satisfaz os seus requisitos, tendo apenas de responder a algumas perguntas.

Na parte B do projeto 1 foi usado o RapidMiner para criar a base de conhecimento (as regras) através da base de dados.

Num primeiro instante a equipa criou um Excel com a cidade de origem, cidade destino, transporte e custo. É importante salientar que a distância e a sustentabilidade não foram consideradas uma vez que não afetam em nada o resultado, a equipa realizou vários testes utilizando essas variáveis e não utilizando e o resultado era o mesmo.

Por fim, o grupo utilizou os operadores “Retrice”, “Set Role”, Multiply, Tree rules e 2 Decision Tree para obter as regras “escritas” e em forma de árvore.

### 1.6.2 - Discussão

O grupo considera que o Projeto 1 foi realizado com sucesso e que a elaboração do mesmo permitiu ampliar o conhecimento dos elementos relativamente ao tratamento de dados com a linguagem Prolog.

Contudo, o grupo apresentou algumas dificuldades ao longo do projeto, pois nenhum elemento estava familiarizado com a linguagem Prolog. Deste modo, foi necessário despende algum tempo para estudar a linguagem.

Relativamente à parte A, o grupo apresentou algumas dificuldades para representar a base de dados, uma vez que, não encontrou preços exatos entre as viagens, tendo assim de presumir alguns custos. Para além disso, ainda surgiram algumas complicações para elaborar a função que era responsável pela conexão entre a base de dados e a base de conhecimento.

No que diz respeito à parte B, a equipa teve dificuldades a extrair as regras através do RapidMiner.

Para terminar, o grupo autoavalia-se com a nota de 16 para o projeto 1.

### 1.6.3 Funcionamento do Trabalho em Grupo

O grupo considera que conseguiu realizar e dividir as tarefas do projeto de forma eficiente.

	Nota
João Pereira a92937	16
Ricardo Oliveira a72349	16
Fábio Ferreira a92961	16
Leonor Machado a96521	16

# Projeto 2

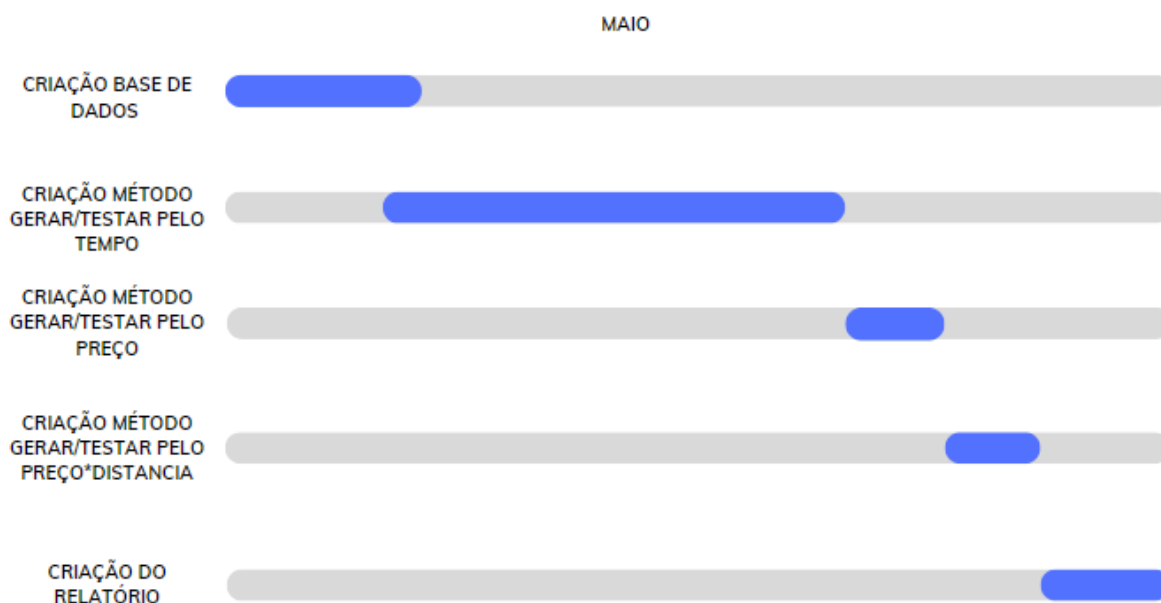
## 2.1 – Introdução

O projeto visa desenvolver um sistema inteligente capaz de fornecer itinerários eficientes, levando em consideração o custo ou o tempo de viagem, utilizando técnicas de inteligência artificial e linguagem Prolog.

### 2.1.2 - Objetivos

O projecto 2 tem duas partes, na parte A o objectivo é obter as viagens mais curtas, para um utilizador que pretenda partir de uma determinada cidade para uma cidade destino através de um algoritmo gerar-testar. Em relação a parte B, o objetivo é obter as viagens com o menor custo/distância de uma dada cidade e cidade destino fornecidas pelo utilizador.

## 2.2 - Execução do Projeto



## **Contribuição de cada membro**

### **Leonor Machado**

- Realizou a construção da Base de Dados;
- Realizou a construção da função seguindo o método de gerar e testar pelo preço.
- Realizou a construção da função seguindo o método de gerar e testar pelo tempo.
- Realizou a construção da função seguindo o método de gerar e testar pelo distancia \* preço.

### **João Pereira**

- Realizou a construção da Base de Dados;
- Realizou a construção da Base de Conhecimento seguindo o método de gerar e testar pelo preço.
- Realizou a construção da Base de Conhecimento seguindo o método de gerar e testar pelo tempo.
- Realizou a construção da Base de Conhecimento seguindo o método de gerar e testar pelo distancia \* preço.

### **Ricardo Oliveira**

- Realizou a construção da Base de Dados;
- Realizou a construção da Base de Conhecimento seguindo o método de gerar e testar pelo preço.
- Realizou a construção da Base de Conhecimento seguindo o método de gerar e testar pelo tempo.
- Realizou a construção da Base de Conhecimento seguindo o método de gerar e testar pelo distancia \* preço.

### **Fábio Ferreira**

- Realizou a construção da Base de Dados;
- Realizou a construção da Base de Conhecimento seguindo o método de gerar e testar pelo preço.
- Realizou a construção da Base de Conhecimento seguindo o método de gerar e testar pelo tempo.
- Realizou a construção da Base de Conhecimento seguindo o método de gerar e testar pelo distancia \* preço.

## **2.3 - Tarefa A**

Nesta etapa a equipa utilizou o código “gerar e testar” fornecido pelo professor nas aulas teóricas.

Numa primeira etapa a equipa criou vários arcos com a estrutura arco(origem, destino). Depois criou a estrutura dist(origem,destino,transporte,tempo,custo) para representar os transportes, preços e tempos entre as diferentes viagens.

Por fim a equipa criou uma interface e alterou a função custo para passar a armazenar os transportes utilizados no caminho mais curto.

```
?- menu.

Sistema de recomendacao de transportes!
1 - Entrar no sistema
2 - Ajuda
3 - Sair
! : 1.
De onde vai partir?

1 - braga
2 - porto
3 - lisboa
4 - guimaraes
5 - viseu
6 - coimbra
! : 1.
Para onde vai?

1 - lisboa
2 - porto
3 - viseu
4 - faro
5 - guimaraes
6 - coimbra
7 - braga
! : 1.
O caminho ideal e: [braga,porto,lisboa]
Os Transportes sao: [comboio,autocarro]
Custo: 28
true .

?-
```

Nota: O código referente a esta parte encontra-se em anexo.

## 2.4 - Tarefa B

Esta etapa é muito semelhante à anterior. A única diferença encontra-se na função custo que agora passa a fazer a multiplicação do tempo pelo custo.

Nota: O código referente a esta parte encontra-se em anexo.

```

?-
% c:/Users/perei/OneDrive/Ambiente de Traba:
?- menu.

Sistema de recomendacao de transportes!
1 - Entrar no sistema
2 - Ajuda
3 - Sair
|: 1.
De onde vai partir?

1 - braga
2 - porto
3 - lisboa
4 - guimaraes
5 - viseu
6 - coimbra
|: 1.
Para onde vai?

1 - lisboa
2 - porto
3 - viseu
4 - faro
5 - guimaraes
6 - coimbra
7 - braga
|: 1.
O caminho ideal e: [braga,porto,lisboa]
Os Transportes sao: [comboio,autocarro]
Custo e Tempo: 3900
true .

?-

```

## Extra

Esta etapa é muito semelhante à parte A. A única diferença é que a parte A procura o caminho mais curto tendo em consideração o custo e esta parte faz o mesmo mas tendo em consideração o tempo.

```

% c:/Users/perei/OneDrive/Ambiente de Traba:
?- menu.

Sistema de recomendacao de transportes!
1 - Entrar no sistema
2 - Ajuda
3 - Sair
|: 1.
De onde vai partir?

1 - braga
2 - porto
3 - lisboa
4 - guimaraes
5 - viseu
6 - coimbra
|: 1.
Para onde vai?

1 - lisboa
2 - porto
3 - viseu
4 - faro
5 - guimaraes
6 - coimbra
7 - braga
|: 1.
O caminho ideal e: [braga,porto,lisboa]
Os Transportes sao: [automovel,aviao]
Tempo: 160
true .

?-

```

Nota: O código referente a esta parte encontra-se em anexo.

## 2.6 - Conclusão

### 2.6.1 - Síntese

Para resolver o projeto 2 parte A o grupo começou por construir arcos (exemplo: arco(braga, porto)) entre as diferentes cidades. Com os arcos já elaborados o grupo associou a cada arco um possível transporte, custo e tempo (exemplo: dist(braga, porto, comboio, 50, 3)). É importante referir que podem existir várias estruturas distantes com a cidades origem e destino iguais, mas com o transporte, custo e tempo diferentes.

Em seguida, o grupo adaptou o código da estratégia “gerar e testar” fornecida pelo professor nas aulas teóricas para calcular o caminho mais curto tendo em consideração o custo e apresentar ao utilizador os transportes usados nesse caminho, o custo total e o caminho escolhido pelo algoritmo.

Na parte B o grupo apenas alterou o código anterior para fazer a multiplicação do tempo pelo custo, permitindo assim calcular qual o caminho que tem a menor relação tempo/custo.

Para finalizar, o grupo ainda realizou uma forma extra de calcular o caminho mais curto tendo consideração o tempo.

### 2.6.2 - Discussão

O grupo considera que o Projeto 2 foi realizado com sucesso e que a elaboração do mesmo permitiu ampliar o conhecimento dos elementos relativamente à escolha do caminho ótimo utilizando a estratégia “gerar e testar”.

O grupo sentiu dificuldades para alterar a função custo de modo a conseguir imprimir os transportes do caminho ótimo. Depois de superado esse problema o grupo conseguiu adaptar rapidamente o código para a parte B do projeto.

Para terminar, o grupo autoavalia-se com a nota de 15 para o projeto 2.



### 2.6.3 Funcionamento do Trabalho em Grupo

O grupo considera que conseguiu realizar e dividir as tarefas do projeto de forma eficiente.

	Nota
João Pereira a92937	15
Ricardo Oliveira a72349	15
Fábio Ferreira a92961	15
Leonor Machado a96521	15

## Anexo 1

### Parte A

Base de dados:

```
:- dynamic(fact/1).
viagem('viseu','porto','comboio','77','9').
viagem('viseu','porto','autocarro','77','14').
viagem('viseu','porto','carro','77','38').
viagem('viseu','braga','comboio','95','14').
viagem('viseu','braga','autocarro','95','10').
viagem('viseu','braga','carro','95','56').
viagem('viseu','guimaraes','comboio','80','11').
viagem('viseu','guimaraes','autocarro','80','10').
viagem('viseu','guimaraes','carro','80','40').
viagem('viseu','lisboa','comboio','260','10').
viagem('viseu','lisboa','autocarro','260','11').
viagem('viseu','lisboa','carro','260','110').
viagem('viseu','faro','comboio','400','20').
viagem('viseu','faro','autocarro','400','15').
viagem('viseu','faro','carro','400','150').
viagem('viseu','coimbra','comboio','80','9').
viagem('viseu','coimbra','autocarro','80','14').
viagem('viseu','coimbra','carro','80','30').
viagem('viseu','chaves','comboio','140','8').
viagem('viseu','chaves','autocarro','140','17').
viagem('viseu','chaves','carro','140','50').
viagem('guimaraes','braga','comboio','50','3').
```

```

viagem('guimaraes','braga','autocarro','60','14').
viagem('guimaraes','braga','carro','40','15').
viagem('guimaraes','viseu','comboio','100','10').
viagem('guimaraes','viseu','autocarro','100','7').
viagem('guimaraes','viseu','carro','100','50').
viagem('guimaraes','porto','comboio','71','3').
viagem('guimaraes','porto','autocarro','71','14').
viagem('guimaraes','porto','carro','71','25').
viagem('guimaraes','lisboa','comboio','713','12').
viagem('guimaraes','lisboa','autocarro','713','10').
viagem('guimaraes','lisboa','carro','713','100').
viagem('guimaraes','faro','comboio','930','30').
viagem('guimaraes','faro','autocarro','930','14').
viagem('guimaraes','faro','carro','930','170').
viagem('guimaraes','coimbra','comboio','100','5').
viagem('guimaraes','coimbra','autocarro','100','10').
viagem('guimaraes','coimbra','carro','100','50').
viagem('guimaraes','chaves','comboio','110','17').
viagem('guimaraes','chaves','autocarro','110','10').
viagem('guimaraes','chaves','carro','110','57').
viagem('lisboa','braga','carro','230','65').
viagem('lisboa','braga','autocarro','240','25').
viagem('lisboa','braga','comboio','220','30').
viagem('lisboa','porto','carro','180','50').
viagem('lisboa','porto','autocarro','210','20').
viagem('lisboa','porto','comboio','160','25').
viagem('lisboa','porto','aviao','60','80').
viagem('lisboa','viseu','carro','190','55').
viagem('lisboa','viseu','autocarro','230','20').
viagem('lisboa','viseu','comboio','240','30').
viagem('lisboa','guimaraes','carro','210','60').
viagem('lisboa','guimaraes','autocarro','320','30').
viagem('lisboa','guimaraes','comboio','260','40').
viagem('lisboa','faro','carro','160','60').
viagem('lisboa','faro','autocarro','200','20').
viagem('lisboa','faro','comboio','170','25').
viagem('lisboa','faro','aviao','170','60').
viagem('lisboa','coimbra','carro','120','40').
viagem('lisboa','coimbra','autocarro','140','15').
viagem('lisboa','coimbra','comboio','100','20').
viagem('lisboa','chaves','carro','270','100').
viagem('lisboa','chaves','autocarro','420','20').
viagem('lisboa','chaves','comboio','380','40').
viagem('faro','braga','carro','340','120').
viagem('faro','braga','autocarro','540','55').
viagem('faro','braga','comboio','420','60').
viagem('faro','porto','carro','310','110').

```

```

viagem('faro', 'porto', 'autocarro', '500', '50').
viagem('faro', 'porto', 'comboio', '350', '60').
viagem('faro', 'porto', 'aviao', '190', '50').
viagem('faro', 'viseu', 'carro', '320', '115').
viagem('faro', 'viseu', 'autocarro', '530', '50').
viagem('faro', 'viseu', 'comboio', '420', '60').
viagem('faro', 'guimaraes', 'carro', '340', '120').
viagem('faro', 'guimaraes', 'autocarro', '600', '30').
viagem('faro', 'guimaraes', 'comboio', '500', '50').
viagem('faro', 'lisboa', 'carro', '160', '60').
viagem('faro', 'lisboa', 'autocarro', '200', '20').
viagem('faro', 'lisboa', 'comboio', '170', '25').
viagem('faro', 'lisboa', 'aviao', '170', '60').
viagem('faro', 'coimbra', 'carro', '240', '70').
viagem('faro', 'coimbra', 'autocarro', '360', '40').
viagem('faro', 'coimbra', 'comboio', '310', '55').
viagem('faro', 'chaves', 'carro', '380', '100').
viagem('faro', 'chaves', 'autocarro', '660', '40').
viagem('faro', 'chaves', 'comboio', '540', '60').
viagem('braga', 'porto', 'comboio', '50', '3').
viagem('braga', 'porto', 'autocarro', '70', '14').
viagem('braga', 'porto', 'carro', '40', '15').
viagem('braga', 'viseu', 'comboio', '200', '14').
viagem('braga', 'viseu', 'autocarro', '200', '5').
viagem('braga', 'viseu', 'carro', '200', '90').
viagem('braga', 'guimaraes', 'comboio', '20', '7').
viagem('braga', 'guimaraes', 'autocarro', '20', '6').
viagem('braga', 'guimaraes', 'carro', '20', '10').
viagem('braga', 'lisboa', 'comboio', '360', '10').
viagem('braga', 'lisboa', 'autocarro', '360', '9').
viagem('braga', 'lisboa', 'carro', '360', '150').
viagem('braga', 'faro', 'comboio', '600', '22').
viagem('braga', 'faro', 'autocarro', '600', '15').
viagem('braga', 'faro', 'carro', '600', '250').
viagem('braga', 'coimbra', 'comboio', '180', '9').
viagem('braga', 'coimbra', 'autocarro', '180', '14').
viagem('braga', 'coimbra', 'carro', '180', '130').
viagem('braga', 'chaves', 'comboio', '100', '8').
viagem('braga', 'chaves', 'autocarro', '100', '13').
viagem('braga', 'chaves', 'carro', '100', '30').
viagem('porto', 'braga', 'comboio', '50', '3').
viagem('porto', 'braga', 'autocarro', '70', '14').
viagem('porto', 'braga', 'carro', '40', '15').
viagem('porto', 'viseu', 'comboio', '110', '10').
viagem('porto', 'viseu', 'autocarro', '110', '7').
viagem('porto', 'viseu', 'carro', '110', '50').
viagem('porto', 'guimaraes', 'comboio', '45', '3').

```

```

viagem('porto','guimaraes','autocarro','65','14').
viagem('porto','guimaraes','carro','35','15').
viagem('porto','lisboa','comboio','320','12').
viagem('porto','lisboa','autocarro','320','10').
viagem('porto','lisboa','carro','320','60').
viagem('porto','lisboa','aviao','320','80').
viagem('porto','faro','comboio','550','20').
viagem('porto','faro','autocarro','550','14').
viagem('porto','faro','carro','550','100').
viagem('porto','faro','aviao','550','130').
viagem('porto','coimbra','comboio','110','5').
viagem('porto','coimbra','autocarro','110','4').
viagem('porto','coimbra','carro','110','50').
viagem('porto','chaves','comboio','50','3').
viagem('porto','chaves','autocarro','50','4').
viagem('porto','chaves','carro','50','17').
viagem('coimbra','faro','autocarro','356','18').
viagem('coimbra','braga','autocarro','149','4').
viagem('coimbra','braga','comboio','173','7').
viagem('coimbra','guimaraes','autocarro','171','5').
viagem('coimbra','porto','autocarro','108','4').
viagem('coimbra','viseu','autocarro','91','4').
viagem('chaves','braga','carro','102','24').
viagem('chaves','braga','autocarro','82','15').
viagem('chaves','guimaraes','autocarro','108','24').
viagem('chaves','porto','carro','150','29').
viagem('chaves','lisboa','autocarro','364','10').
viagem('chaves','faro','carro','683','38').
viagem('coimbra','faro','carro','441','84').
viagem('coimbra','faro','comboio','357','44').
viagem('coimbra','braga','carro','171','30').
viagem('coimbra','guimaraes','carro','167','29').
viagem('chaves','braga','carro','122','24').
viagem('chaves','guimaraes','carro','104','11').
viagem('chaves','lisboa','carro','439','70').
viagem('chaves','faro','autocarro','527','49').
viagem('chaves','braga','comboio','82','25').
viagem('chaves','guimaraes','comboio','108','25').
viagem('chaves','porto','comboio','140','15').
viagem('chaves','lisboa','comboio','350','13').
viagem('chaves','faro','comboio','630','35').
viagem('chaves','coimbra','autocarro','188','17').
viagem('chaves','coimbra','carro','236','34').
viagem('chaves','coimbra','comboio','180','19').
viagem('coimbra','chaves','carro','236','34').
viagem('coimbra','chaves','autocarro','188','17').
viagem('coimbra','chaves','comboio','180','19').

```

## Base de conhecimento:

```
:- op( 800, fx, if).
:- op( 700, xfx, then).
:- op( 300, xfy, or).
:- op( 200, xfy, and).

:- dynamic(fact/1).
if jovem and estudante then custo_baixo.
if jovem and trabalhador then custo_alto.
if adulto and estudante then custo_medio.
if adulto and trabalhador then custo_alto.
if adulto and desempregado then custo_baixo.
if idoso and trabalhador then custo_alto.
if idoso and reformado then custo_baixo.

if coimbra and faro then distancia_alta.
if coimbra and braga then distancia_alta.
if coimbra and guimaraes then distancia_alta.
if coimbra and porto then distancia_media.
if coimbra and viseu then distancia_media.

if chaves and braga then distancia_media.
if chaves and guimaraes then distancia_media.
if chaves and porto then distancia_alta.
if chaves and faro then distancia_alta.

if braga and guimaraes then distancia_baixa.
if braga and porto then distancia_baixa.
if braga and guimaraes then distancia_baixa.
if braga and lisboa then distancia_alta.
if braga and faro then distancia_alta.
if braga and chaves then distancia_media.
if braga and viseu then distancia_alta.
if braga and coimbra then distancia_alta.

if porto and viseu then distancia_baixa.
if porto and guimaraes then distancia_baixa.
if porto and lisboa then distancia_alta.
if porto and faro then distancia_alta.
if porto and coimbra then distancia_media.
if porto and chaves then distancia_baixa.

if lisboa and braga then distancia_alta.
```

if lisboa and porto then distancia\_alta.  
if lisboa and viseu then distancia\_alta.  
if lisboa and guimaraes then distancia\_alta.  
if lisboa and faro then distancia\_alta.  
if lisboa and coimbra then distancia\_media.  
if lisboa and chaves then distancia\_alta.

if faro and braga then distancia\_alta.  
if faro and porto then distancia\_alta.  
if faro and viseu then distancia\_alta.  
if faro and guimaraes then distancia\_alta.  
if faro and lisboa then distancia\_alta.  
if faro and coimbra then distancia\_alta.  
if faro and chaves then distancia\_alta.

if guimaraes and braga then distancia\_baixa.  
if guimaraes and porto then distancia\_media.  
if guimaraes and lisboa then distancia\_alta.  
if guimaraes and faro then distancia\_alta.  
if guimaraes and chaves then distancia\_media.  
if guimaraes and viseu then distancia\_media.  
if guimaraes and coimbra then distancia\_media.

if viseu and porto then distancia\_media.  
if viseu and guimaraes then distancia\_media.  
if viseu and lisboa then distancia\_alta.  
if viseu and faro then distancia\_alta.  
if viseu and coimbra then distancia\_media.  
if viseu and chaves then distancia\_alta.  
if viseu and braga then distancia\_media.

if sustentavel and custo\_baixo then 'autocarro'.  
if sustentavel and custo\_medio then 'comboio'.  
if sustentavel and custo\_alto then 'comboio'.

if custo\_baixo and distancia\_baixa then 'autocarro'.  
if custo\_baixo and distancia\_media then 'autocarro'.  
if custo\_baixo and distancia\_alta then 'comboio'.  
if custo\_medio and distancia\_baixa then 'autocarro'.  
if custo\_medio and distancia\_media then 'comboio'.  
if custo\_medio and distancia\_alta then 'comboio'.  
if custo\_alto and distancia\_baixa then 'carro'.  
if custo\_alto and distancia\_media then 'carro'.  
if custo\_alto and distancia\_alta and porto and lisboa then 'aviao'.  
if custo\_alto and distancia\_alta and porto and faro then 'aviao'.  
if custo\_alto and distancia\_alta then 'carro'.

## Forward:

```
:- op( 800, fx, if).
:- op( 700, xfx, then).
:- op( 500, xfy, and).
:- op( 300, xfy, or).

% Simple forward chaining in Prolog
demo:- new_derived_fact(P), !, % A new fact

% write('De acordo com as opcoes por si escolhidas encontramos o seguinte
% Transporte: '),resultadowrite(P), nl,
assert(fact(P)),

demo. % Continue

demo:- %write( 'Nao existem outros transportes...'),
nl. % All facts derived
new_derived_fact( Concl) :-
if Cond then Concl, % A rule
\+ fact( Concl), % Rule's conclusion not yet a fact
composed_fact( Cond). % Condition true?
composed_fact( Cond) :-
fact( Cond). % Simple fact
composed_fact( Cond1 and Cond2) :-
composed_fact( Cond1),
composed_fact( Cond2). % Both conjuncts true
composed_fact( Cond1 or Cond2) :-
composed_fact( Cond1);
composed_fact( Cond2).
```

## Proof:

```
:- op( 800, fx, if).
:- op( 700, xfx, then).
:- op( 300, xfy, or).
:- op( 200, xfy, and).

% demo( P, Proof) Proof is a proof that P is true
:- op( 800, xfx, <=).
demo( P, P) :-
fact( P).
demo( P, P <= CondProof) :-
```

```

if Cond then P,
demo( Cond, CondProof).
demo( P1 and P2, Proof1 and Proof2) :-
demo( P1, Proof1),
demo( P2, Proof2).
demo( P1 or P2, Proof) :-
demo( P1, Proof);
demo( P2, Proof).

```

## Interface:

```

:- dynamic(fact/1), [forward,proof,bdpartea,sbcpartea].

:- op( 800, fx, if).
:- op( 700, xfx, then).
:- op( 300, xfy, or).
:- op( 200, xfy, and).

%INTERFACE

menu :- nl, nl, write('Sistema de recomendacao de viagens! Descubra a
melhor viagem para o seu perfil!'),nl,
        write('1 - Entrar no sistema'),nl,
        write('2 - Ajuda'),nl,
        write('3 - Sair'),nl,
        read(A),
        avaliar(A).

avaliar(1):- questao1.
avaliar(2):- write('Para começar escreva menu.').
avaliar(3):- write('A sair!'), halt.

resultadowrite:- demo,
fact(Nome),fact(A),fact(B),viagem(A,B,Nome,Distancia,Custo),
        %demo(Nome,Proof),
        write('Devia utilizar o transporte: '),write(Nome),nl,write('A
distancia e: '),write(Distancia),nl,write('A viagem tem um custo de:
'),write(Custo).

questao1:-
        write('*****
*****'), nl,
        write('** Em que local esta?'), nl,

```



```

write('**'),nl,
write('** 1 - Guimaraes '), nl,
write('** 2 - Chaves'), nl,
    write('** 3 - Braga'), nl,
    write('** 4 - Porto'), nl,
    write('** 5 - Viseu'), nl,
    write('** 6 - Coimbra'), nl,
    write('** 7 - Lisboa'), nl,
    write('** 8 - Faro'), nl,
read(A1),
(
    (A1 == 1),assert(fact('guimaraes')), questao2;
(A1 == 2),assert(fact('chaves')), questao2;
(A1 == 3),assert(fact('braga')), questao2;
(A1 == 4),assert(fact('porto')), questao2;
(A1 == 5),assert(fact('viseu')), questao2;
(A1 == 6),assert(fact('coimbra')), questao2;
(A1 == 7),assert(fact('lisboa')), questao2;
(A1 == 8),assert(fact('faro')), questao2).

```

questao2:-

```

write('*****
*****'), nl,
    write('**'),nl,
write('** 1 - Guimaraes '), nl,
write('** 2 - Chaves'), nl,
    write('** 3 - Braga'), nl,
    write('** 4 - Porto'), nl,
    write('** 5 - Viseu'), nl,
    write('** 6 - Coimbra'), nl,
    write('** 7 - Lisboa'), nl,
    write('** 8 - Faro'), nl,
read(A2),
(
    (A2 == 1),assert(fact('guimaraes')), questao3;
(A2 == 2),assert(fact('chaves')), questao3;
(A2 == 3),assert(fact('braga')), questao3;
(A2 == 4),assert(fact('porto')), questao3;
(A2 == 5),assert(fact('viseu')), questao3;
(A2 == 6),assert(fact('coimbra')), questao3;
(A2 == 7),assert(fact('lisboa')), questao3;
(A2 == 8),assert(fact('faro')), questao3).

```

questao3:-

```
    write('*****
*****'), nl,
        write('** Qual e a sua idade?'), nl,

    write('** 1 - 0 - 25'),nl,
    write('** 2 - 25 - 65'),nl,
    write('** 3 - +65'),nl,

    read(A3),
    (
        (A3 == 1),assert(fact('jovem')), questao4;
        (A3 == 2),assert(fact('adulto')), questao4;
        (A3 == 3),assert(fact('idoso')), questao4).
```

questao4:-

```
    write('*****
*****'), nl,
        write('** Em que estado se encontra?'), nl,
        write('** 1 - Estudante'),nl,
    write('** 2 - Trabalhador'),nl,
    write('** 3 - Reformado'),nl,
    write('** 4 - Desempregado'),nl,

    read(A4),
    (
        (A4 == 1),assert(fact('estudante')), questao5;
        (A4 == 2),assert(fact('trabalhador')), questao5;
        (A4 == 3),assert(fact('reformado')), questao5;
        (A4 == 4),assert(fact('desempregado')), questao5).
```

questao5:-

```
    write('*****
*****'), nl,
        write('** Pertende usar um transporte sustentavel?'),
nl,
        write('**'),nl,
        write('** 1 - Sim'),nl,
        write('** 2 - Nao' ),nl, nl,
    read(A5),nl,
    (
        ((A5 == 1), assert(fact('sustentavel')), questao6);
        (
            (A5 == 2), assert(fact('naoSustentavel')),
questao6)).
```

```

questao6:-
write('*****
*****'), nl,
    write('** Qual pensa ser o seu transporte ideal?'), nl,
    write('** 1 - autocarro'),nl,
    write('** 2 - carro'),nl,
    write('** 3 - comboio'),nl,
    write('** 4 - aviao'),nl,

    read(A6),
    ((A6==1), (demo('autocarro',Proof)-
>write(Proof);write('False')));
    (A6==2), (demo('carro',Proof)-
>write(Proof);write('False')));
    (A6==3), (demo('comboio',Proof)-
>write(Proof);write('False')));
    (A6==4), (demo('aviao',Proof)-
>write(Proof);write('False'))),nl,resultado
.

resultado :-
    write('*****
*****'), nl,
        write('**
**'), nl,
        write('**
Resultado Obtido
**'), nl,
        write('**
**'), nl,
        write('*****
*****'),
            demo,resultadowrite,retractall(fact(_)).

%:-menu.

```

## Parte B:

Base de dados:

```

:- dynamic(fact/1).
viagem('o_viseu','porto','comboio','77','9').
viagem('o_viseu','porto','autocarro','77','14').
viagem('o_viseu','porto','carro','77','38').
viagem('o_viseu','braga','comboio','95','14').

```

```

viagem('o_viseu','braga','autocarro','95','10').
viagem('o_viseu','braga','carro','95','56').
viagem('o_viseu','guimaraes','comboio','80','11').
viagem('o_viseu','guimaraes','autocarro','80','10').
viagem('o_viseu','guimaraes','carro','80','40').
viagem('o_viseu','lisboa','comboio','260','10').
viagem('o_viseu','lisboa','autocarro','260','11').
viagem('o_viseu','lisboa','carro','260','110').
viagem('o_viseu','faro','comboio','400','20').
viagem('o_viseu','faro','autocarro','400','15').
viagem('o_viseu','faro','carro','400','150').
viagem('o_viseu','coimbra','comboio','80','9').
viagem('o_viseu','coimbra','autocarro','80','14').
viagem('o_viseu','coimbra','carro','80','30').
viagem('o_viseu','chaves','comboio','140','8').
viagem('o_viseu','chaves','autocarro','140','17').
viagem('o_viseu','chaves','carro','140','50').
viagem('o_guimaraes','braga','comboio','50','3').
viagem('o_guimaraes','braga','autocarro','60','14').
viagem('o_guimaraes','braga','carro','40','15').
viagem('o_guimaraes','viseu','comboio','100','10').
viagem('o_guimaraes','viseu','autocarro','100','7').
viagem('o_guimaraes','viseu','carro','100','50').
viagem('o_guimaraes','porto','comboio','71','3').
viagem('o_guimaraes','porto','autocarro','71','14').
viagem('o_guimaraes','porto','carro','71','25').
viagem('o_guimaraes','lisboa','comboio','713','12').
viagem('o_guimaraes','lisboa','autocarro','713','10').
viagem('o_guimaraes','lisboa','carro','713','100').
viagem('o_guimaraes','faro','comboio','930','30').
viagem('o_guimaraes','faro','autocarro','930','14').
viagem('o_guimaraes','faro','carro','930','170').
viagem('o_guimaraes','coimbra','comboio','100','5').
viagem('o_guimaraes','coimbra','autocarro','100','10').
viagem('o_guimaraes','coimbra','carro','100','50').
viagem('o_guimaraes','chaves','comboio','110','17').
viagem('o_guimaraes','chaves','autocarro','110','10').
viagem('o_guimaraes','chaves','carro','110','57').
viagem('o_lisboa','braga','carro','230','65').
viagem('o_lisboa','braga','autocarro','240','25').
viagem('o_lisboa','braga','comboio','220','30').
viagem('o_lisboa','porto','carro','180','50').
viagem('o_lisboa','porto','autocarro','210','20').
viagem('o_lisboa','porto','comboio','160','25').
viagem('o_lisboa','porto','aviao','60','80').
viagem('o_lisboa','viseu','carro','190','55').
viagem('o_lisboa','viseu','autocarro','230','20').

```

```

viagem('o_lisboa', 'viseu', 'comboio', '240', '30').
viagem('o_lisboa', 'guimaraes', 'carro', '210', '60').
viagem('o_lisboa', 'guimaraes', 'autocarro', '320', '30').
viagem('o_lisboa', 'guimaraes', 'comboio', '260', '40').
viagem('o_lisboa', 'faro', 'carro', '160', '60').
viagem('o_lisboa', 'faro', 'autocarro', '200', '20').
viagem('o_lisboa', 'faro', 'comboio', '170', '25').
viagem('o_lisboa', 'faro', 'aviao', '170', '60').
viagem('o_lisboa', 'coimbra', 'carro', '120', '40').
viagem('o_lisboa', 'coimbra', 'autocarro', '140', '15').
viagem('o_lisboa', 'coimbra', 'comboio', '100', '20').
viagem('o_lisboa', 'chaves', 'carro', '270', '100').
viagem('o_lisboa', 'chaves', 'autocarro', '420', '20').
viagem('o_lisboa', 'chaves', 'comboio', '380', '40').
viagem('o_faro', 'braga', 'carro', '340', '120').
viagem('o_faro', 'braga', 'autocarro', '540', '55').
viagem('o_faro', 'braga', 'comboio', '420', '60').
viagem('o_faro', 'porto', 'carro', '310', '110').
viagem('o_faro', 'porto', 'autocarro', '500', '50').
viagem('o_faro', 'porto', 'comboio', '350', '60').
viagem('o_faro', 'porto', 'aviao', '190', '50').
viagem('o_faro', 'viseu', 'carro', '320', '115').
viagem('o_faro', 'viseu', 'autocarro', '530', '50').
viagem('o_faro', 'viseu', 'comboio', '420', '60').
viagem('o_faro', 'guimaraes', 'carro', '340', '120').
viagem('o_faro', 'guimaraes', 'autocarro', '600', '30').
viagem('o_faro', 'guimaraes', 'comboio', '500', '50').
viagem('o_faro', 'lisboa', 'carro', '160', '60').
viagem('o_faro', 'lisboa', 'autocarro', '200', '20').
viagem('o_faro', 'lisboa', 'comboio', '170', '25').
viagem('o_faro', 'lisboa', 'aviao', '170', '60').
viagem('o_faro', 'coimbra', 'carro', '240', '70').
viagem('o_faro', 'coimbra', 'autocarro', '360', '40').
viagem('o_faro', 'coimbra', 'comboio', '310', '55').
viagem('o_faro', 'chaves', 'carro', '380', '100').
viagem('o_faro', 'chaves', 'autocarro', '660', '40').
viagem('o_faro', 'chaves', 'comboio', '540', '60').
viagem('o_braga', 'porto', 'comboio', '50', '3').
viagem('o_braga', 'porto', 'autocarro', '70', '14').
viagem('o_braga', 'porto', 'carro', '40', '15').
viagem('o_braga', 'viseu', 'comboio', '200', '14').
viagem('o_braga', 'viseu', 'autocarro', '200', '5').
viagem('o_braga', 'viseu', 'carro', '200', '90').
viagem('o_braga', 'guimaraes', 'comboio', '20', '7').
viagem('o_braga', 'guimaraes', 'autocarro', '20', '6').
viagem('o_braga', 'guimaraes', 'carro', '20', '10').
viagem('o_braga', 'lisboa', 'comboio', '360', '10').

```

```

viagem('o_braga','lisboa','autocarro','360','9').
viagem('o_braga','lisboa','carro','360','150').
viagem('o_braga','faro','comboio','600','22').
viagem('o_braga','faro','autocarro','600','15').
viagem('o_braga','faro','carro','600','250').
viagem('o_braga','coimbra','comboio','180','9').
viagem('o_braga','coimbra','autocarro','180','14').
viagem('o_braga','coimbra','carro','180','130').
viagem('o_braga','chaves','comboio','100','8').
viagem('o_braga','chaves','autocarro','100','13').
viagem('o_braga','chaves','carro','100','30').
viagem('o_porto','braga','comboio','50','3').
viagem('o_porto','braga','autocarro','70','14').
viagem('o_porto','braga','carro','40','15').
viagem('o_porto','viseu','comboio','110','10').
viagem('o_porto','viseu','autocarro','110','7').
viagem('o_porto','viseu','carro','110','50').
viagem('o_porto','guimaraes','comboio','45','3').
viagem('o_porto','guimaraes','autocarro','65','14').
viagem('o_porto','guimaraes','carro','35','15').
viagem('o_porto','lisboa','comboio','320','12').
viagem('o_porto','lisboa','autocarro','320','10').
viagem('o_porto','lisboa','carro','320','60').
viagem('o_porto','lisboa','aviao','320','80').
viagem('o_porto','faro','comboio','550','20').
viagem('o_porto','faro','autocarro','550','14').
viagem('o_porto','faro','carro','550','100').
viagem('o_porto','faro','aviao','550','130').
viagem('o_porto','coimbra','comboio','110','5').
viagem('o_porto','coimbra','autocarro','110','4').
viagem('o_porto','coimbra','carro','110','50').
viagem('o_porto','chaves','comboio','50','3').
viagem('o_porto','chaves','autocarro','50','4').
viagem('o_porto','chaves','carro','50','17').
viagem('o_coimbra','faro','autocarro','356','18').
viagem('o_coimbra','braga','autocarro','149','4').
viagem('o_coimbra','braga','comboio','173','7').
viagem('o_coimbra','guimaraes','autocarro','171','5').
viagem('o_coimbra','porto','autocarro','108','4').
viagem('o_coimbra','viseu','autocarro','91','4').
viagem('o_chaves','braga','carro','102','24').
viagem('o_chaves','braga','autocarro','82','15').
viagem('o_chaves','guimaraes','autocarro','108','24').
viagem('o_chaves','porto','carro','150','29').
viagem('o_chaves','lisboa','autocarro','364','10').
viagem('o_chaves','faro','carro','683','38').
viagem('o_coimbra','faro','carro','441','84').

```

```

viagem('o_coimbra','faro','comboio','357','44').
viagem('o_coimbra','braga','carro','171','30').
viagem('o_coimbra','guimaraes','carro','167','29').
viagem('o_chaves','braga','carro','122','24').
viagem('o_chaves','guimaraes','carro','104','11').
viagem('o_chaves','lisboa','carro','439','70').
viagem('o_chaves','faro','autocarro','527','49').
viagem('o_chaves','braga','comboio','82','25').
viagem('o_chaves','guimaraes','comboio','108','25').
viagem('o_chaves','porto','comboio','140','15').
viagem('o_chaves','lisboa','comboio','350','13').
viagem('o_chaves','faro','comboio','630','35').
viagem('o_chaves','coimbra','autocarro','188','17').
viagem('o_chaves','coimbra','carro','236','34').
viagem('o_chaves','coimbra','comboio','180','19').
viagem('o_coimbra','chaves','carro','236','34').
viagem('o_coimbra','chaves','autocarro','188','17').
viagem('o_coimbra','chaves','comboio','180','19').

```

## Base de conhecimento:

```

:- op( 800, fx, if).
:- op( 700, xfx, then).
:- op( 300, xfy, or).
:- op( 200, xfy, and).

:- dynamic(fact/1).
if custo_alto and o_braga then carro.
if custo_alto and o_coimbra then autocarro.
if custo_alto and o_faro then carro.
if custo_alto and o_guimaraes then carro.
if custo_alto and o_lisboa then carro.
if custo_alto and o_porto then carro.
if custo_alto and o_viseu then carro.
if custo_medio and o_braga then comboio.
if custo_medio and o_chaves then carro.
if custo_medio and o_faro then comboio.
if custo_medio and o_guimaraes then carro.
if custo_medio and o_lisboa then comboio.
if custo_medio and o_porto then comboio.
if custo_medio and o_viseu then carro.
if custo_baixo and o_braga then autocarro.
if custo_baixo and o_chaves then autocarro.
if custo_baixo and o_coimbra then autocarro.
if custo_baixo and o_faro then autocarro.

```

```

if custo_baixo and o_guimaraes then carro.
if custo_baixo and o_lisboa then autocarro.
if custo_baixo and o_porto then autocarro.

```

## Interface:

```
:- dynamic(fact/1), [forward,proof,bdparteb,sbcparteb].
```

```
%Base de conhecimento
```

```

:- op( 800, fx, if).
:- op( 700, xfx, then).
:- op( 300, xfy, or).
:- op( 200, xfy, and).

```

```
%INTERFACE
```

```

menu :- nl, nl, write('Sistema de recomendacao de viagens! Descubra a
melhor viagem para o seu perfil!'),nl,
       write('1 - Entrar no sistema'),nl,
       write('2 - Ajuda'),nl,
       write('3 - Sair'),nl,
       read(A),
       avaliar(A).

```

```

avaliar(1):- questao1.
avaliar(2):- write('Para começar escreva menu.').
avaliar(3):- write('A sair!'), halt.

```

```

resultadowrite:- demo,
fact(Nome),fact(A),fact(B),viagem(A,B,Nome,Distancia,Custo),
    %demo(Nome,Proof),
    write('Devia utilizar o transporte: '),write(Nome),nl,write('A
distancia e: '),write(Distancia),nl,write('A viagem tem um custo de:
'),write(Custo).

```

```
questao1:-
```

```

    write('*****
*****'), nl,
        write('** Em que local esta?'), nl,
        write('**'),nl,
        write('** 1 - Guimaraes '), nl,
        write('** 2 - Chaves'), nl,

```



```

        write('** 3 - Braga'), nl,
        write('** 4 - Porto'), nl,
        write('** 5 - Viseu'), nl,
        write('** 6 - Coimbra'), nl,
        write('** 7 - Lisboa'), nl,
        write('** 8 - Faro'), nl,
    read(A1),
    (
        (A1 == 1),assert(fact('o_guimaraes')), questao2;
    (A1 == 2),assert(fact('o_chaves')), questao2;
    (A1 == 3),assert(fact('o_braga')), questao2;
    (A1 == 4),assert(fact('o_porto')), questao2;
    (A1 == 5),assert(fact('o_viseu')), questao2;
    (A1 == 6),assert(fact('o_coimbra')), questao2;
    (A1 == 7),assert(fact('o_lisboa')), questao2;
    (A1 == 8),assert(fact('o_faro')), questao2).

```

questao2:-

```

    write('*****
*****'), nl,
        write('**'),nl,
    write('** 1 - Guimaraes '), nl,
    write('** 2 - Chaves'), nl,
        write('** 3 - Braga'), nl,
        write('** 4 - Porto'), nl,
        write('** 5 - Viseu'), nl,
        write('** 6 - Coimbra'), nl,
        write('** 7 - Lisboa'), nl,
        write('** 8 - Faro'), nl,
    read(A2),
    (
        (A2 == 1),assert(fact('guimaraes')), questao3;
    (A2 == 2),assert(fact('chaves')), questao3;
    (A2 == 3),assert(fact('braga')), questao3;
    (A2 == 4),assert(fact('porto')), questao3;
    (A2 == 5),assert(fact('viseu')), questao3;
    (A2 == 6),assert(fact('coimbra')), questao3;
    (A2 == 7),assert(fact('lisboa')), questao3;
    (A2 == 8),assert(fact('faro')), questao3).

```

questao3:-

```

    write('*****
*****'), nl,
        write('** Qual e a o custo que pretende ter?'), nl,

```

```

write('** 1 - Custo baixo'),nl,
write('** 2 - Custo medio'),nl,
write('** 3 - Custo alto'),nl,

    read(A3),
    (
        (A3 == 1),assert(fact('custo_baixo')), questao4;
        (A3 == 2),assert(fact('custo_medio')), questao4;
        (A3 == 3),assert(fact('custo_alto')), questao4).

questao4:-
write('*****
*****'), nl,
    write('** Qual pensa ser o seu transporte ideal?'), nl,
    write('** 1 - autocarro'),nl,
    write('** 2 - carro'),nl,
    write('** 3 - comboio'),nl,
    write('** 4 - aviao'),nl,

    read(A4),
    ((A4==1), (demo('autocarro',Proof)-
>write(Proof);write('False')));
    (A4==2), (demo('carro',Proof)-
>write(Proof);write('False')));
    (A4==3), (demo('comboio',Proof)-
>write(Proof);write('False')));
    (A4==4), (demo('aviao',Proof)-
>write(Proof);write('False'))),nl,resultado
.

resultado :-
    write('*****
*****'), nl,
        write('**
**'), nl,
        write('**
Resultado Obtido
**'), nl,
        write('**
*****
*****'),
        demo,resultadowrite,retractall(fact(_)).

%:-menu.

```

**Nota:** o proof e o forward são iguais para a parte A e B.

## Anexo 2:

### Parte A:

```
arco(braga,porto).
arco(braga,viseu).
arco(porto,lisboa).
arco(lisboa,faro).
arco(porto,viseu).
arco(braga,guimaraes).
arco(guimaraes,porto).
arco(viseu,coimbra).
arco(coimbra,braga).
arco(coimbra,lisboa).
```

```
dist(braga,porto,comboio,50,3).
dist(braga,porto,autocarro,70,14).
dist(braga,porto,automovel,40,15).
dist(braga,viseu,automovel,200,90).
dist(porto,lisboa,comboio,180,60).
dist(porto,lisboa,automovel,150,90).
dist(porto,lisboa,aviao,120,130).
dist(porto,lisboa,autocarro,150,25).
dist(lisboa,faro,comboio,240,80).
dist(lisboa,faro,automovel,200,150).
dist(lisboa,faro,aviao,120,140).
dist(lisboa,faro,autocarro,240,70).
dist(porto,viseu,automovel,200,90).
dist(porto,viseu,comboio,260,70).
dist(porto,viseu,autocarro,250,60).
dist(braga,guimaraes,automovel,20,10).
dist(braga,guimaraes,autocarro,30,15).
dist(guimaraes,porto,automovel,50,14).
dist(guimaraes,porto,comboio,70,3).
dist(guimaraes,porto,autocarro,60,12).
dist(viseu,coimbra,automovel,100,40).
dist(coimbra,braga,automovel,90,60).
dist(coimbra,lisboa,comboio,120,30).
```

```
caminhocurto(X,Y,C,NC,LTR):-caminho(X,Y,C), % gerar
    maiscurto(X,Y,C,NC,LTR).% testar
maiscurto(X,Y,C,NC,LTR):- custo(C,LTR,NC),
```

```

\+ menor(X,Y,NC),reverse(C,CR),write('O caminho ideal e:
'),write(CR),nl,reverse(LTR,LR),write('Os Transportes sao: '),write(LR).
menor(X,Y,NC):- caminho(X,Y,C1),
    custo(C1,_,NC1),%nl,write(NC1),nl,write(T),
    NC1<NC.

% caminho.pl
caminho(X,Z,Caminho):-caminho(X,Z,[X],Caminho).
caminho(X,X,Caminho,Caminho).
caminho(X,Z,Visitado,Caminho)
:- arco(X,Y),
    \+ member(Y,Visitado),
    caminho(Y,Z,[Y | Visitado],Caminho).

custo([_],[_],0).
custo([A,B|C],[T|T2],Cust):- dist(B,A,T,_,Custo), custo([B|C],T2,Cust2),
    Cust is Custo + Cust2.

menu :- nl, nl, write('Sistema de recomendacao de transportes!'),nl,
    write('1 - Entrar no sistema'),nl,
    write('2 - Ajuda'),nl,
    write('3 - Sair'),nl,
    read(A),
    avaliar(A).

avaliar(1):- questao1.
avaliar(2):- write('Para comecar escreva menu.').
avaliar(3):- write('A sair!'), halt.

questao1:- write("De onde vai partir?"),nl,nl,
    write("1 - braga"),nl,
    write("2 - porto"),nl,
    write("3 - lisboa"),nl,
    write("4 - guimaraes"),nl,
    write("5 - viseu"),nl,
    write("6 - coimbra"),nl,
    read(A1),
    (
        (A1 == 1),questao2(braga);
        (A1 == 2),questao2(porto);
        (A1 == 3),questao2(lisboa);
        (A1 == 4),questao2(guimaraes);
        (A1 == 5),questao2(viseu);
        (A1 == 6),questao2(coimbra)).

questao2(A1):- write("Para onde vai?"),nl,nl,
    write("1 - lisboa"),nl,

```

```

write("2 - porto"),nl,
write("3 - viseu"),nl,
write("4 - faro"),nl,
write("5 - guimaraes"),nl,
write("6 - coimbra"),nl,
write("7 - braga"),nl,
read(A2),
(
    (A2 == 1),caminhocurto(A1,lisboa,_,C,_);
    (A2 == 2),caminhocurto(A1,porto,_,C,_);
    (A2 == 3),caminhocurto(A1,viseu,_,C,_);
    (A2 == 4),caminhocurto(A1,faro,_,C,_);
    (A2 == 5),caminhocurto(A1,guimaraes,_,C,_);
    (A2 == 6),caminhocurto(A1,coimbra,_,C,_);
    (A2 == 7),caminhocurto(A1,braga,_,C,_),nl,write('Custo: '),
write(C).

```

## Parte B;

```

arco(braga,porto).
arco(braga,viseu).
arco(porto,lisboa).
arco(lisboa,faro).
arco(porto,viseu).
arco(braga,guimaraes).
arco(guimaraes,porto).
arco(viseu,coimbra).
arco(coimbra,braga).
arco(coimbra,lisboa).

dist(braga,porto,comboio,50,3).
dist(braga,porto,autocarro,70,14).
dist(braga,porto,automovel,40,15).
dist(braga,viseu,automovel,200,90).
dist(porto,lisboa,comboio,180,60).
dist(porto,lisboa,automovel,150,90).
dist(porto,lisboa,aviao,120,130).
dist(porto,lisboa,autocarro,150,25).
dist(lisboa,faro,comboio,240,80).
dist(lisboa,faro,automovel,200,150).
dist(lisboa,faro,aviao,120,140).
dist(lisboa,faro,autocarro,240,70).
dist(porto,viseu,automovel,200,90).
dist(porto,viseu,comboio,260,70).
dist(porto,viseu,autocarro,250,60).
dist(braga,guimaraes,automovel,20,10).

```

```

dist(braga,guimaraes,autocarro,30,15).
dist(guimaraes,porto,automovel,50,14).
dist(guimaraes,porto,comboio,70,3).
dist(guimaraes,porto,autocarro,60,12).
dist(viseu,coimbra,automovel,100,40).
dist(coimbra,braga,automovel,90,60).
dist(coimbra,lisboa,comboio,120,30).

caminhocurto(X,Y,C,NC,LTR):-caminho(X,Y,C), % gerar
    maiscurto(X,Y,C,NC,LTR).% testar
maiscurto(X,Y,C,NC,LTR):- custo(C,LTR,NC),
    \+ menor(X,Y,NC),reverse(C,CR),write('O caminho ideal e:
'),write(CR),nl,reverse(LTR,LR),write('Os Transportes sao: '),write(LR).
menor(X,Y,NC):- caminho(X,Y,C1),
    custo(C1,_,NC1),%nl,write(NC1),nl,write(T),
    NC1<NC.

% caminho.pl
caminho(X,Z,Caminho):-caminho(X,Z,[X],Caminho).
caminho(X,X,Caminho,Caminho).
caminho(X,Z,Visitado,Caminho)
    :- arco(X,Y),
    \+ member(Y,Visitado),
    caminho(Y,Z,[Y | Visitado],Caminho).

custo([_],[_],0).
custo([A,B|C],[T|T2],Cust):- dist(B,A,T,Dist1,Temp1),
    custo([B|C],T2,Cust2), Cust is Dist1*Temp1 + Cust2.

menu :- nl, nl, write('Sistema de recomendacao de transportes!'),nl,
    write('1 - Entrar no sistema'),nl,
    write('2 - Ajuda'),nl,
    write('3 - Sair'),nl,
    read(A),
    avaliar(A).

avaliar(1):- questao1.
avaliar(2):- write('Para começar escreva menu.').
avaliar(3):- write('A sair!'), halt.

questao1:- write("De onde vai partir?"),nl,nl,
    write("1 - braga"),nl,
    write("2 - porto"),nl,
    write("3 - lisboa"),nl,
    write("4 - guimaraes"),nl,
    write("5 - viseu"),nl,
    write("6 - coimbra"),nl,

```

```

read(A1),
(
    (A1 == 1),questao2(braga);
    (A1 == 2),questao2(porto);
    (A1 == 3),questao2(lisboa);
    (A1 == 4),questao2(guimaraes);
    (A1 == 5),questao2(viseu);
    (A1 == 6),questao2(coimbra)).

questao2(A1):- write("Para onde vai?"),nl,nl,
    write("1 - lisboa"),nl,
    write("2 - porto"),nl,
    write("3 - viseu"),nl,
    write("4 - faro"),nl,
    write("5 - guimaraes"),nl,
    write("6 - coimbra"),nl,
    write("7 - braga"),nl,
    read(A2),
    (
        (A2 == 1),caminhocurto(A1,lisboa,_,C,_);
        (A2 == 2),caminhocurto(A1,porto,_,C,_);
        (A2 == 3),caminhocurto(A1,viseu,_,C,_);
        (A2 == 4),caminhocurto(A1,faro,_,C,_);
        (A2 == 5),caminhocurto(A1,guimaraes,_,C,_);
        (A2 == 6),caminhocurto(A1,coimbra,_,C,_);
        (A2 == 7),caminhocurto(A1,braga,_,C,_),nl,write('Custo e
Tempo: '), write(C).

```

### Extra:

```

arco(braga,porto).
arco(braga,viseu).
arco(porto,lisboa).
arco(lisboa,faro).
arco(porto,viseu).
arco(braga,guimaraes).
arco(guimaraes,porto).
arco(viseu,coimbra).
arco(coimbra,braga).
arco(coimbra,lisboa).

dist(braga,porto,comboio,50,3).
dist(braga,porto,autocarro,70,14).
dist(braga,porto,automovel,40,15).
dist(braga,viseu,automovel,200,90).
dist(porto,lisboa,comboio,180,60).
dist(porto,lisboa,automovel,150,90).
dist(porto,lisboa,aviao,120,130).

```

```

dist(porto, lisboa, autocarro, 150, 25).
dist(lisboa, faro, comboio, 240, 80).
dist(lisboa, faro, automovel, 200, 150).
dist(lisboa, faro, aviao, 120, 140).
dist(lisboa, faro, autocarro, 240, 70).
dist(porto, viseu, automovel, 200, 90).
dist(porto, viseu, comboio, 260, 70).
dist(porto, viseu, autocarro, 250, 60).
dist(braga, guimaraes, automovel, 20, 10).
dist(braga, guimaraes, autocarro, 30, 15).
dist(guimaraes, porto, automovel, 50, 14).
dist(guimaraes, porto, comboio, 70, 3).
dist(guimaraes, porto, autocarro, 60, 12).
dist(viseu, coimbra, automovel, 100, 40).
dist(coimbra, braga, automovel, 90, 60).
dist(coimbra, lisboa, comboio, 120, 30).

caminhocurto(X,Y,C,NC,LTR):-caminho(X,Y,C), % gerar
    maiscurto(X,Y,C,NC,LTR).% testar
maiscurto(X,Y,C,NC,LTR):- custo(C,LTR,NC),
    \+ menor(X,Y,NC),reverse(C,CR),write('O caminho ideal e:
'),write(CR),nl,reverse(LTR,LR),write('Os Transportes sao: '),write(LR).
menor(X,Y,NC):- caminho(X,Y,C1),
    custo(C1,_,NC1),%nl,write(NC1),nl,write(T),
    NC1<NC.

% caminho.pl
caminho(X,Z,Caminho):-caminho(X,Z,[X],Caminho).
caminho(X,X,Caminho,Caminho).
caminho(X,Z,Visitado,Caminho)
    :- arco(X,Y),
    \+ member(Y,Visitado),
    caminho(Y,Z,[Y | Visitado],Caminho).

custo([_],[_],0).
custo([A,B|C],[T|T2],Temp):- dist(B,A,T,Tempo,_), custo([B|C],T2,Temp2),
Temp is Tempo + Temp2.

menu :- nl, nl, write('Sistema de recomendacao de transportes!'),nl,
    write('1 - Entrar no sistema'),nl,
    write('2 - Ajuda'),nl,
    write('3 - Sair'),nl,
    read(A),
    avaliar(A).

avaliar(1):- questao1.
avaliar(2):- write('Para começar escreva menu.').

```



```

avaliar(3):- write('A sair!'), halt.

questao1:- write("De onde vai partir?"),nl,nl,
    write("1 - braga"),nl,
    write("2 - porto"),nl,
    write("3 - lisboa"),nl,
    write("4 - guimaraes"),nl,
    write("5 - viseu"),nl,
    write("6 - coimbra"),nl,
    read(A1),
    (
        (A1 == 1),questao2(braga);
        (A1 == 2),questao2(porto);
        (A1 == 3),questao2(lisboa);
        (A1 == 4),questao2(guimaraes);
        (A1 == 5),questao2(viseu);
        (A1 == 6),questao2(coimbra)).

questao2(A1):- write("Para onde vai?"),nl,nl,
    write("1 - lisboa"),nl,
    write("2 - porto"),nl,
    write("3 - viseu"),nl,
    write("4 - faro"),nl,
    write("5 - guimaraes"),nl,
    write("6 - coimbra"),nl,
    write("7 - braga"),nl,
    read(A2),
    (
        (A2 == 1),caminhocurto(A1,lisboa,_,C,_);
        (A2 == 2),caminhocurto(A1,porto,_,C,_);
        (A2 == 3),caminhocurto(A1,viseu,_,C,_);
        (A2 == 4),caminhocurto(A1,faro,_,C,_);
        (A2 == 5),caminhocurto(A1,guimaraes,_,C,_);
        (A2 == 6),caminhocurto(A1,coimbra,_,C,_);
        (A2 == 7),caminhocurto(A1,braga,_,C,_),nl,write('Tempo: '),
write(C).

```

# Contrato de grupo

Neste contrato são apresentadas cláusulas que os elementos do grupo terão obrigatoriamente de cumprir ao longo do desenvolvimento dos projetos, sendo sujeitos a penalizações caso não respeitem o mesmo.

## 1ª cláusula

As reuniões de trabalho são obrigatórias para todos os elementos do grupo. Caso algum elemento não possa comparecer a alguma reunião deve avisar com pelo menos 5 horas de antecedência.

## 2ª cláusula

Todos os elementos da equipa comprometem-se a desenvolver as tarefas pelas quais são responsáveis no prazo estipulado pela equipa.

## 3ª cláusula

Todos os elementos da equipa comprometem-se a comparecer nas aulas T e TP. Caso algum elemento não possa comparecer deve justificar a sua falta com os restantes elementos da equipa.

## 4ª cláusula

No caso de alguma dúvida referente ao trabalho a ser realizado, o elemento em questão deverá expor a sua questão para o restante do grupo. Caso seja necessário, o grupo deverá marcar uma reunião para esclarecer essa dúvida.

## 5ª cláusula

Todos os elementos do grupo comprometem-se a realizar uma autoavaliação justa e honesta.

Todos os elementos do grupo concordam com as normas e cláusulas estipuladas neste anteriormente.

Guimarães, 27 de maio de 2023

João Pereira

Fábio Ferreira.

Ricardo Jorge Cruz Oliveira

Ana Leonor Itachaco