



Universidade do Minho
Escola de Engenharia

TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL

Licenciatura em Engenharia e Gestão de Sistemas de Informação

2022/2023



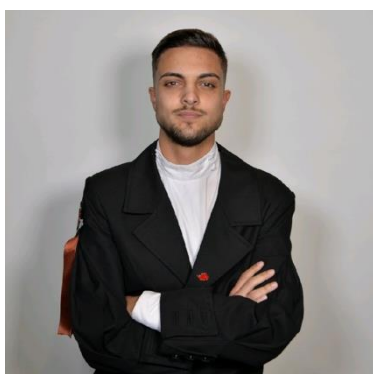
Ana João Monteiro Mendonça
de Assunção da Silva

A96712



Maria Inês Monteiro de Lima

A96271



Gonçalo Mendes Rodrigues

A95212



João Pedro de Carvalho
Mendes

A97991



Índice

1. INTRODUÇÃO	3
1.1 Enquadramento	3
1.2 Objetivos	3
2. EXECUÇÃO DO PROJETO	4
2.1 Diagrama de Gantt	4
2.2 Distribuição do trabalho por cada elemento.....	5
3. TAREFA A- RECOMENDAÇÃO DE MEIO DE TRANSPORTE	9
3.1 Parte A: Recomendação de Meio de Transporte (aquisição manual de conhecimento).9	
3.2 Parte B: Recomendação de Meio de Transporte (aquisição automática de conhecimento).....	11
4. TAREFA B – ITENERÁRIO E MEIOS DE TRANSPORTES	16
4.1 - Parte A.....	16
Parte B.....	17
5. CONCLUSÕES	18
5.1 Síntese	18
5.2 Discussão	18
5.3 Funcionamento do Trabalho em Grupo	19
6. ANEXOS	20
6.1 ANEXO A	20
PROJETO 1.....	20
PROJETO 2.....	28
6.2 ANEXO B	33
Contrato do grupo	33
BIBLIOGRAFIA	35

Índice de Figuras

Figura 1- Planeamento do projeto na ferramenta Project Libre	4
Figura 2- Diagrama de Gantt	4
Figura 3- Ficheiro Base de Dados Excel	11
Figura 4- Processo desenvolvido no RapidMiner	12
Figura 5- Configuração Set Role	13
Figura 6- Configuração Tree to Rules.....	13
Figura 7- Parâmetros da Decision Tree.....	14
Figura 8- Nova Base de Conhecimento	14



1. INTRODUÇÃO

1.1 Enquadramento

No âmbito da unidade curricular de Técnicas de Inteligência Artificial foi-nos proposto a realização de dois projetos SBC distintos divididos em duas partes, designadas de Parte A e Parte B, relacionadas com a recomendação de meios de transporte.

Para a elaboração destes projetos é essencial ter conhecimento do que é um Sistema Baseado em Conhecimento. Estes sistemas são ferramentas computacionais que utilizam e manipulam conhecimentos de forma estruturada e explícita para ajudar a resolver problemas complexos, utilizando conceitos de inteligência artificial. São usados em procedimentos de solução de problemas e para apoiar a aprendizagem humana, tomada de decisão e ações.

O projeto foi desenvolvido por 4 elementos: Ana Mendonça, Gonçalo Rodrigues, João Mendes e Maria Inês Lima.

1.2 Objetivos

Neste projeto, o grupo constituído por quatro elementos terá de ser capaz de desenvolver um conjunto de objetivos para realizar o projeto com sucesso. Desta forma, os objetivos a alcançar são: desenvolver um Sistema Baseado em Conhecimento utilizando a linguagem Prolog, a exploração dos métodos de extração automática via ferramentas como o RapidMiner e a exploração de técnicas de Aquisição e Representação de Conhecimento.

Isto permitirá ao grupo desenvolver conhecimentos sobre técnicas de inteligência artificial, assim como desenvolver novos conhecimentos ao nível de programação e apresentar uma solução para os projetos propostos.



2.2 Distribuição do trabalho por cada elemento

O grupo decidiu que o projeto seria realizado em conjunto, elaborando reuniões semanais, de forma a perceber e discutir em conjunto as próximas etapas. Esta abordagem permitiu que todos os membros entendessem o projeto, compartilhassem ideias uns com os outros e atingissem os objetivos propostos para a unidade curricular.

Contributo do elemento Maria Inês Lima(a96271)

Numa fase inicial, realizamos uma reunião onde todos os elementos do grupo estavam presentes e elaboramos o contrato de trabalho, no qual todos tivemos participação.

Em seguida, fiquei responsável pelo desenvolvimento do planeamento do projeto utilizando a ferramenta Project Libre. Depois de realizado o planeamento do projeto, fiz uma análise do enunciado do projeto juntamente com os elementos de grupo e cada um ficou responsável por efetuar pesquisas e adquirir conhecimentos.

No projeto 1 em conjunto com a Ana realizei a base de dados que contém o conjunto de transportes disponíveis, além disso também ajudei a desenvolver a interface. Na elaboração da interface com a ajuda dos restantes elementos, colocamos todas as questões para o utilizador responder, e depois devolver o meio de transporte mais adequado relativamente às respostas fornecidas.

Preenchi a folha de Excel com os dados necessários para o RapidMiner, com a ajuda da Ana. Além disso, realizei testes para encontrar erros no programa.

No projeto 2, com a ajuda da Ana desenvolvemos o predicado ‘transportesUtilizados’ que encontra os transportes utilizados numa viagem.

Por último, relativamente ao relatório fiquei responsável por efetuar o enquadramento do projeto e os objetivos do mesmo. Fiquei também responsável de ajudar a explicar o que foi desenvolvido no Projeto 2 e pela elaboração do relatório final nomeadamente da sua estrutura e de juntar todas as partes num só documento.



Contributo do elemento Gonalo Mendes Rodrigues (a95212)

Enquanto l der do grupo, fiquei respons vel por organizar reuni es de trabalho e garantir que ao longo das semanas o projeto era discutido e entendido por todos.

Primeiro comecei por analisar o enunciado e fazer uma pesquisa sobre os conhecimentos que precisava de adquirir para realizar o projeto com sucesso, em conjunto com o resto do grupo

Relativamente ao projeto 1, participei no desenvolvimento da interface e realizei a base de conhecimento em conjunto com o Jo o, ou seja, as regras de produ  o onde atrav s das v rias respostas do utilizador o programa tira conclus es interm dias sobre o seu perfil e no final recomenda um transporte. Inicialmente a nossa abordagem consistia na constru  o de perfis, por m, ap s mostrar o trabalho ao professor das aulas PL, ele aconselhou a n o seguir essa abordagem, uma vez que sempre que adicionasse uma pergunta, seria muito mais trabalhoso adicionar os novos factos na nossa base de conhecimento. Al m disso, o programa n o tinha qualquer tipo de conclus es interm dias relativas ao perfil do viajante e   viagem.

Contribu  tamb m na realiza  o da parte B do projeto, ajudando a entender o funcionamento do RapidMiner e elaborei uma s rie de combina  es no software onde conseguimos obter novas regras e a obter a nova base de conhecimento.

No projeto 2, analisei o c digo disponibilizado pelo professor na Blackboard e nas aulas te ricas e adaptei-o com a ajuda do meu colega Jo o para o contexto do nosso problema.

Por  ltimo ajudei na realiza  o do relat rio, fiquei respons vel por explicar a parte A do projeto 1 com a Ana, em que consiste e os v rios ficheiros que a constituem e o mesmo para a fase B do Projeto 2 e fiz a revis o/verifica  o final para ter a certeza que o trabalho estava de acordo com o que foi pedido.



Contributo do elemento João Mendes (a97991)

O grupo dividiu as tarefas de forma igual, por isso todos os elementos do grupo fizeram maioritariamente o mesmo. Todos os elementos participaram na execução do projeto e todos realizaram o relatório. O trabalho foi todo realizado em conjunto em chamadas online.

Primeiro começamos todos por analisar o enunciado e por escolher os critérios que iríamos usar no nosso trabalho.

Depois, junto com o Gonçalo, procedemos à realização do sistema baseado em conhecimento, onde mais uma vez o trabalho foi feito em conjunto online, sendo que foi um processo bastante demorado e complexo. Após algum feedback dos docentes tivemos que refazer a nossa base de conhecimento duas vezes, pois estávamos a usar perfis, em vez de fazer uma base de conhecimento com conclusões intermédias.

Por fim, procedi à elaboração da interface com a ajuda dos restantes elementos, colocando todas as questões para o utilizador responder, e depois devolver o meio de transporte mais adequado relativamente às respostas fornecidas.

Utilizei a ferramenta RapidMiner onde, após elaborarmos uma série de combinações no software, obtivemos uma série de novas regras que correspondem à base de conhecimento. Depois disso a equipa procedeu à substituição do primeiro ficheiro por este.

No projeto 2, analisei o código disponibilizado pelo professor na BlackBoard e nas aulas teóricas e adaptei-o com a ajuda do meu colega Gonçalo ao contexto do nosso projeto.

Quanto ao relatório, desenvolvi a explicação da parte B do Projeto 1, ajudei na explicação do código do Projeto 2 e fui responsável por fazer a síntese na conclusão.



Contributo do elemento Ana João Silva (96712)

Primeiro comecei por fazer uma análise do enunciado do projeto juntamente com os elementos do meu grupo e cada um ficou responsável por efetuar pesquisas e adquirir conhecimentos.

Relativamente ao primeiro projeto, em conjunto com a Inês desenvolvi a base de dados do projeto com os vários transportes disponíveis e ajudei a realizar a interface do projeto. Na elaboração da interface com a ajuda dos restantes elementos, colocamos todas as questões para o utilizador responder, e depois devolver o meio de transporte mais adequado relativamente às respostas fornecidas.

Com a ajuda da Inês, preenchi a folha de Excel com os dados necessários para o RapidMiner. Também realizei testes onde encontrei alguns erros no programa.

No segundo projeto realizei com a ajuda da Inês o predicado ‘transportesUtilizados’ que encontra os transportes utilizados numa viagem.

Por último, também fiquei responsável pela elaboração do relatório final. Ajudei na explicação do projeto 1 parte A e fui responsável por fazer a discussão e o funcionamento do grupo na conclusão.



3. TAREFA A- RECOMENDAÇÃO DE MEIO DE TRANSPORTE

3.1 Parte A: Recomendação de Meio de Transporte (aquisição manual de conhecimento)

Na parte A do P1, a equipa elaborou um sistema baseado em conhecimento que orienta e aconselha o utilizador sobre o meio de transporte mais adequado para uma viagem. Após responder a algumas questões, conseguimos obter informação sobre a viagem e sobre o utilizador. A viagem é caracterizada pelo seu transporte, origem, destino, custo e pela sua duração. No final, será sugerido o transporte mais adequado para a viagem do utilizador, tendo em conta o seu perfil.

O projeto está dividido em vários ficheiros: base de conhecimento, base de dados, sistema de inferência e interface.

Na base de conhecimento foram elaboradas um conjunto de regras de produção que podem ser vistas como conclusões intermédias realizadas pelo programa, com base nas respostas dadas pelo utilizador sobre a viagem e o viajante. Estas regras estão definidas através de um conjunto de condições if/then.

Para realizar a base de conhecimento tivemos de definir quais as perguntas que íamos realizar ao utilizador para sabermos quais as conclusões que o programa deve atingir sobre o utilizador e a sua viagem. Consoante a “origem” e o “destino” escolhidos pelo viajante, o programa irá concluir se a viagem será “curta” ou “longa”.

De seguida, e após perguntar ao utilizador se pretende realizar a viagem de forma rápida ou demorada, e com base nos factos concluídos anteriormente, o programa tira conclusões sobre o perfil do viajante, por exemplo, caso a viagem seja “curta” e “demorada” o facto derivado será “turista”. Os outros factos que podem ser derivados após saber se a viagem é demorada ou rápida são: “apressado”, “aventureiro” e “trabalho”, sendo esta última uma característica da viagem e não do viajante.

A questão seguinte consiste em entender se o viajante tem ou não poderio financeiro. Dessa questão, os factos derivados são: “conforto”, “económico”, “conveniente”, “restrito”, “explorador”, “desafio”, “diretor” e “funcionário”. Como exemplo, caso a viagem seja de “trabalho” e o viajante tenha “poderio financeiro”, concluímos o facto “diretor”, caso seja “nao poderio financeiro”, concluímos que ele é “funcionário”.

A última questão consiste em entender a preocupação ambiental do viajante. Desse modo, e tendo em conta todas as outras questões realizadas e factos derivados, concluímos qual o transporte mais adequado para o utilizador.

De seguida, desenvolvemos uma base de dados que contém os dados das várias viagens, incluindo o transporte, origem, destino, duração e custo.

No final, realizamos uma interface que disponibiliza ao utilizador as várias questões e permite definir os vários factos para, no final da sua execução, recomendar o meio de transporte mais adequado, presente na base de dados do programa, recomendado pela base de conhecimento, consoante o seu perfil e a sua viagem.



Recorremos ainda ao código de forward chaining disponibilizado na Blackboard, sendo esse o nosso sistema de inferência, permitindo ao sistema fazer inferências ou tirar conclusões com base nos factos e regras existentes.

Requisitos mínimos	Cumprido	Não cumprido
A solução (gerada pelo SBC) terá de ser implementada na linguagem Prolog via regras de produção geradas via: aquisição de conhecimento manual.	X	
O SBC deverá aconselhar e orientar o utilizador através de um processo interativo que lhe permitirá, após responder a algumas questões, chegar a um (ou vários) cenário(s) possível(eis) para aconselhamento.	X	
Deverá ser seguida a metodologia de desenvolvimento de SBC estudada (4 fases) e as técnicas de aquisição de conhecimento manuais e automáticas mais convenientes.	X	
Separar (em ficheiros distintos ou identificar claramente por comentários): base de conhecimento, sistema de inferência, base de dados e interface.	X	

Requisitos Opcionais	Cumprido	Não cumprido
Sistema de inferência considerando incerteza		X
Sistema de inferência considerando explicação		X
Interface amigável para o utilizador		X



3.2 Parte B: Recomendação de Meio de Transporte (aquisição automática de conhecimento)

Depois de terminada a parte A, passamos para a parte B do Projeto 1, que tem como base o projeto já desenvolvido, porém a aquisição de conhecimento será agora realizada de forma automática.

Para esta etapa, começamos por preencher uma folha Excel com dados relativos ao nosso programa realizado até ao momento, onde colocamos as respostas às questões da parte A do projeto disponibilizadas ao utilizador na interface, e o meio de transporte que foi recomendado.

1	Origem	Destino	Duracao	PoderioFinanceiro	ViagemAmbiental	Transporte
2	braga	porto	rapida	poderiofinanceiro	ambiente	mota
3	braga	porto	demorada	poderiofinanceiro	ambiente	mota
4	braga	porto	rapida	naopoderiofinanceiro	naoambiente	autocarro
5	braga	porto	demorada	naopoderiofinanceiro	ambiente	bicicleta
6	braga	porto	rapida	poderiofinanceiro	naoambiente	carro
7	porto	braga	rapida	naopoderiofinanceiro	ambiente	comboio
8	porto	braga	demorada	naopoderiofinanceiro	naoambiente	autocarro
9	porto	braga	demorada	poderiofinanceiro	naoambiente	carro
10	braga	viseu	rapida	poderiofinanceiro	ambiente	aviao
11	braga	viseu	demorada	naopoderiofinanceiro	ambiente	comboio
12	braga	viseu	rapida	naopoderiofinanceiro	naoambiente	carro
13	braga	viseu	demorada	naopoderiofinanceiro	naoambiente	autocarro
14	viseu	braga	rapida	poderiofinanceiro	naoambiente	carro
15	porto	viseu	rapida	poderiofinanceiro	ambiente	mota
16	porto	viseu	rapida	poderiofinanceiro	naoambiente	carro
17	porto	viseu	demorada	naopoderiofinanceiro	ambiente	bicicleta
18	porto	viseu	demorada	poderiofinanceiro	ambiente	mota
19	porto	viseu	rapida	naopoderiofinanceiro	ambiente	comboio
20	porto	viseu	rapida	naopoderiofinanceiro	naoambiente	autocarro
21	braga	lisboa	rapida	poderiofinanceiro	ambiente	aviao
22	braga	lisboa	rapida	naopoderiofinanceiro	ambiente	mota
23	braga	lisboa	rapida	poderiofinanceiro	naoambiente	carro
24	braga	lisboa	rapida	naopoderiofinanceiro	naoambiente	carro
25	braga	lisboa	demorada	poderiofinanceiro	ambiente	aviao
26	braga	lisboa	demorada	naopoderiofinanceiro	ambiente	comboio
27	braga	lisboa	demorada	poderiofinanceiro	naoambiente	carro
28	braga	lisboa	demorada	naopoderiofinanceiro	naoambiente	autocarro
29	lisboa	porto	demorada	poderiofinanceiro	ambiente	mota
30	lisboa	porto	rapida	poderiofinanceiro	ambiente	aviao
31	lisboa	porto	demorada	naopoderiofinanceiro	ambiente	comboio
32	lisboa	porto	rapida	naopoderiofinanceiro	naoambiente	carro
33	coimbra	braga	rapida	poderiofinanceiro	ambiente	mota
34	coimbra	braga	rapida	naopoderiofinanceiro	ambiente	comboio
35	coimbra	braga	demorada	naopoderiofinanceiro	ambiente	bicicleta
36	viseu	coimbra	rapida	poderiofinanceiro	ambiente	aviao
37	viseu	coimbra	rapida	naopoderiofinanceiro	ambiente	mota
38	viseu	coimbra	demorada	naopoderiofinanceiro	ambiente	comboio

Figura 3- Ficheiro Base de Dados Excel

Após a recolha dos dados, inserimos os mesmos no Rapid Miner. O Rapid Miner é um software de data science, que fornece um ambiente integrado para a preparação de dados e implementação de modelo preditivo. A combinação no Rapid Miner foi a seguinte:

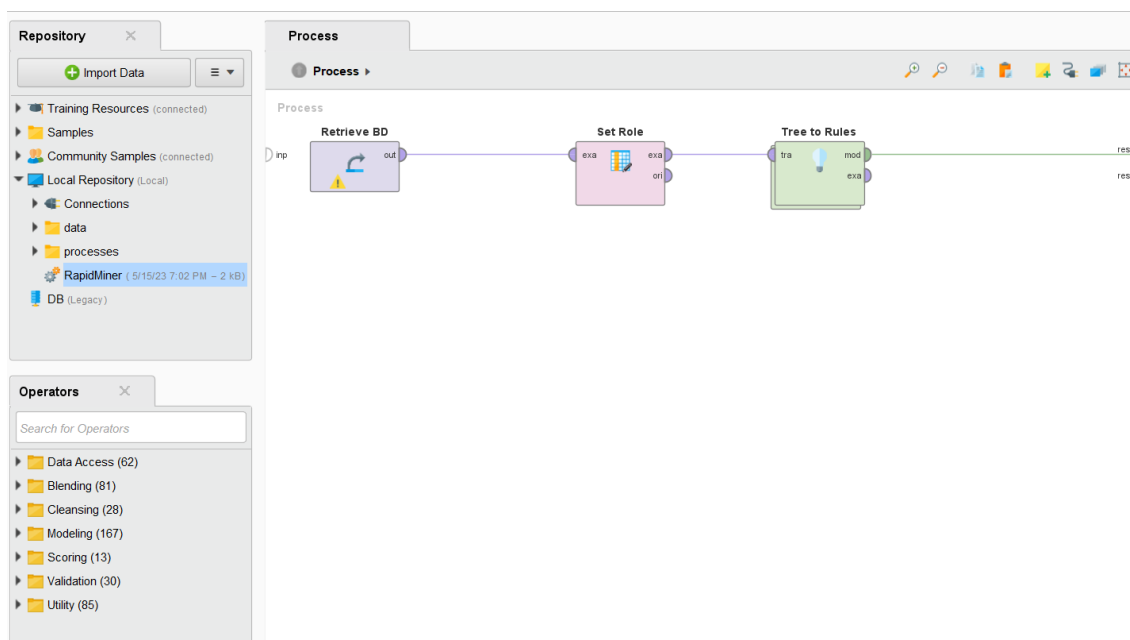
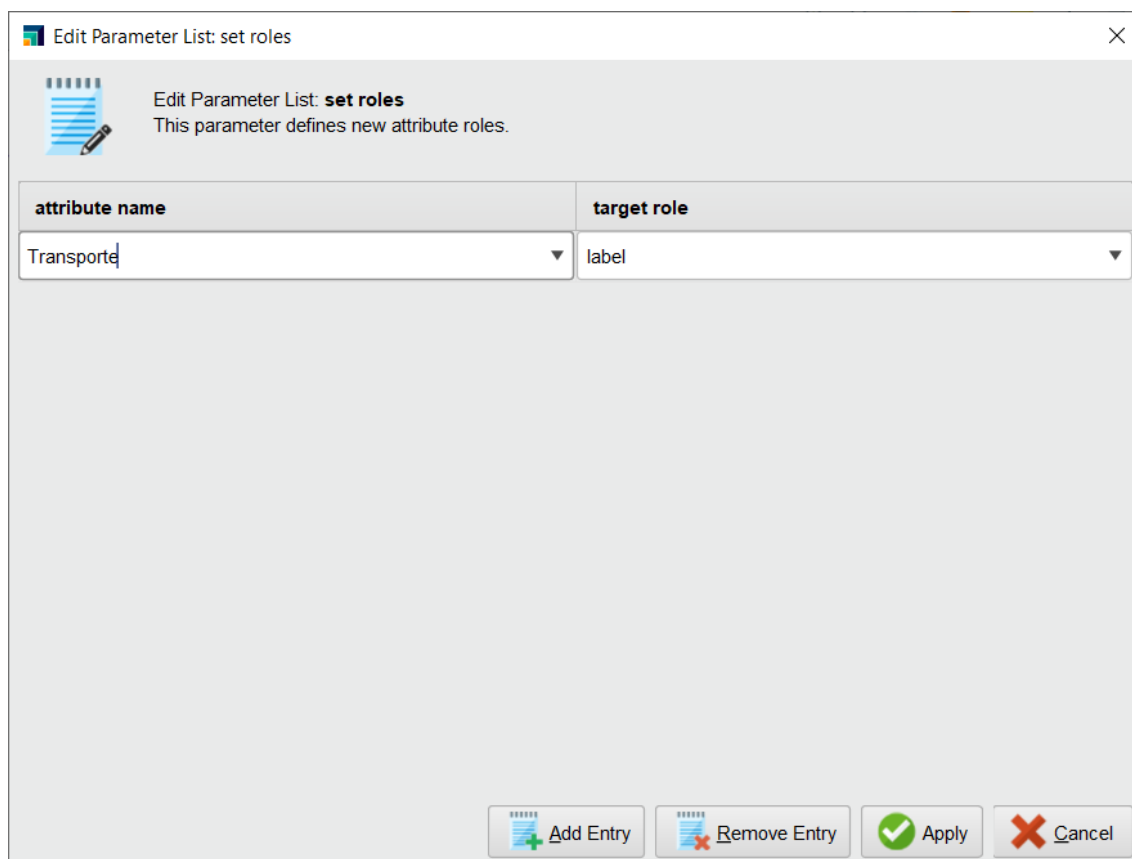


Figura 4- Processo desenvolvido no RapidMiner

Configuração Set Role:



Edit Parameter List: **set roles**

Edit Parameter List: **set roles**
This parameter defines new attribute roles.

attribute name	target role
Transporte	label

Add Entry Remove Entry Apply Cancel

Figura 5- Configuração Set Role

Configuração Tree to Rules:

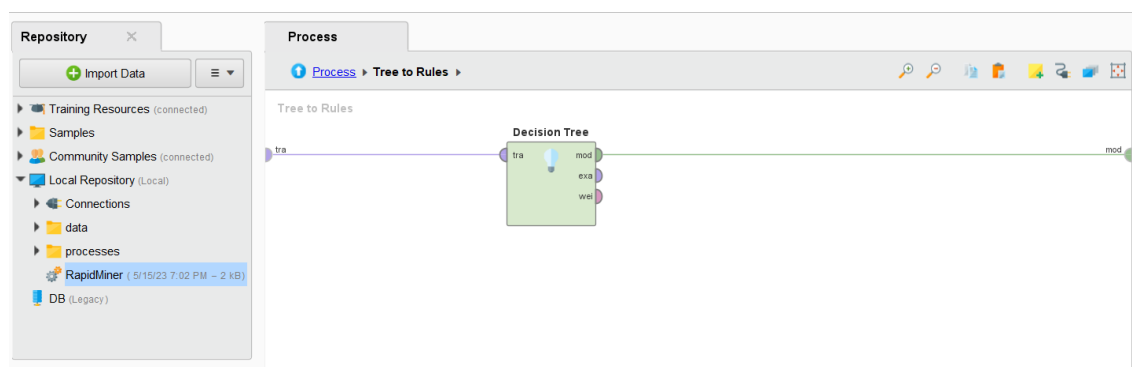


Figura 6- Configuração Tree to Rules

Parâmetros da Decision Tree:

Parameters

Decision Tree

criterion
gain_ratio

maximal depth
10

☒ apply pruning

confidence
0.1

☒ apply prepruning

minimal gain
0.01

minimal leaf size
1

minimal size for split
1

number of prepruning alternatives
3

Figura 7- Parâmetros da Decision Tree

Depois de fazermos as combinações de componentes acima obtivemos uma nova base de conhecimento:

Result History
RuleModel (Tree to Rules)

RuleModel

Description
Annotations

```

if ViagemAmbiental = ambiente and PoderioFinanceiro = naopoderiofinanceiro and Duracao = demorada and Origem = braga and Destino = lisboa then comboio (0 / 0 / 0 / 0 / 1 / 0)
if ViagemAmbiental = ambiente and PoderioFinanceiro = naopoderiofinanceiro and Duracao = demorada and Origem = braga and Destino = porto then bicicleta (0 / 0 / 1 / 0 / 0 / 0)
if ViagemAmbiental = ambiente and PoderioFinanceiro = naopoderiofinanceiro and Duracao = demorada and Origem = braga and Destino = viseu then comboio (0 / 0 / 0 / 0 / 1 / 1)
if ViagemAmbiental = ambiente and PoderioFinanceiro = naopoderiofinanceiro and Duracao = demorada and Origem = coimbra then bicicleta (0 / 0 / 1 / 0 / 0 / 0)
if ViagemAmbiental = ambiente and PoderioFinanceiro = naopoderiofinanceiro and Duracao = demorada and Origem = lisboa then comboio (0 / 0 / 0 / 0 / 1 / 0)
if ViagemAmbiental = ambiente and PoderioFinanceiro = naopoderiofinanceiro and Duracao = demorada and Origem = porto then bicicleta (0 / 0 / 1 / 0 / 0 / 0)
if ViagemAmbiental = ambiente and PoderioFinanceiro = naopoderiofinanceiro and Duracao = demorada and Origem = viseu then comboio (0 / 0 / 0 / 0 / 1 / 0)
if ViagemAmbiental = ambiente and PoderioFinanceiro = naopoderiofinanceiro and Duracao = rapida and Origem = braga then moto (1 / 0 / 0 / 0 / 0 / 0)
if ViagemAmbiental = ambiente and PoderioFinanceiro = naopoderiofinanceiro and Duracao = rapida and Origem = coimbra then comboio (0 / 0 / 0 / 0 / 1 / 0)
if ViagemAmbiental = ambiente and PoderioFinanceiro = naopoderiofinanceiro and Duracao = rapida and Origem = porto then comboio (0 / 0 / 0 / 0 / 2 / 0)
if ViagemAmbiental = ambiente and PoderioFinanceiro = naopoderiofinanceiro and Duracao = rapida and Origem = viseu then moto (1 / 0 / 0 / 0 / 0 / 0)
if ViagemAmbiental = ambiente and PoderioFinanceiro = poderiofinanceiro and Destino = braga then moto (1 / 0 / 0 / 0 / 0 / 0)
if ViagemAmbiental = ambiente and PoderioFinanceiro = poderiofinanceiro and Destino = coimbra then aviao (0 / 0 / 0 / 0 / 0 / 1)
if ViagemAmbiental = ambiente and PoderioFinanceiro = poderiofinanceiro and Destino = lisboa then aviao (0 / 0 / 0 / 0 / 0 / 2)
if ViagemAmbiental = ambiente and PoderioFinanceiro = poderiofinanceiro and Destino = porto and Origem = braga then moto (2 / 0 / 0 / 0 / 0 / 0)
if ViagemAmbiental = ambiente and PoderioFinanceiro = poderiofinanceiro and Destino = porto and Origem = lisboa and Duracao = demorada then moto (1 / 0 / 0 / 0 / 0 / 0)
if ViagemAmbiental = ambiente and PoderioFinanceiro = poderiofinanceiro and Destino = porto and Origem = lisboa and Duracao = rapida then aviao (0 / 0 / 0 / 0 / 0 / 1)
if ViagemAmbiental = ambiente and PoderioFinanceiro = poderiofinanceiro and Destino = viseu and Origem = braga then aviao (0 / 0 / 0 / 0 / 0 / 1)
if ViagemAmbiental = ambiente and PoderioFinanceiro = poderiofinanceiro and Destino = viseu and Origem = porto then moto (2 / 0 / 0 / 0 / 0 / 0)
if ViagemAmbiental = naooambiente and PoderioFinanceiro = naopoderiofinanceiro and Duracao = demorada then autocarro (0 / 3 / 0 / 0 / 0 / 0)
if ViagemAmbiental = naooambiente and PoderioFinanceiro = naopoderiofinanceiro and Duracao = rapida and Origem = braga and Destino = lisboa then carro (0 / 0 / 0 / 1 / 0 / 0)
if ViagemAmbiental = naooambiente and PoderioFinanceiro = naopoderiofinanceiro and Duracao = rapida and Origem = braga and Destino = porto then autocarro (0 / 1 / 0 / 0 / 0 / 0)
if ViagemAmbiental = naooambiente and PoderioFinanceiro = naopoderiofinanceiro and Duracao = rapida and Origem = braga and Destino = viseu then carro (0 / 0 / 0 / 1 / 0 / 0)
if ViagemAmbiental = naooambiente and PoderioFinanceiro = naopoderiofinanceiro and Duracao = rapida and Origem = lisboa then carro (0 / 0 / 0 / 1 / 0 / 0)
if ViagemAmbiental = naooambiente and PoderioFinanceiro = poderiofinanceiro and Destino = porto then autocarro (0 / 1 / 0 / 0 / 0 / 0)
correct: 37 out of 37 training examples.

```

Figura 8- Nova Base de Conhecimento



Ao contrário da parte A do P1, agora a nossa base de conhecimento é obtida automaticamente e não de forma manual, tornando o processo de obter resultados mais ágil e eficaz.

Requisitos	Cumprido	Não cumprido
A solução (gerada pelo SBC) terá de ser implementada na linguagem Prolog via regras de produção geradas. As regras devem ser geradas via uma aquisição de conhecimento automática (Data Mining).	X	
Os métodos de extração automática podem ser obtidos via Prolog e/ou via ferramentas de Data Mining (e.g. R, Python, Rapidminer, Weka). Podem ainda processar/preparar os dados com quaisquer ferramentas, incluindo o MS Excel.	X	
O conhecimento extraído deverá ser implementado em Prolog via regras de produção, por exemplo via esforço de codificação manual.	X	



4. TAREFA B – ITENERÁRIO E MEIOS DE TRANSPORTES

4.1- Parte A

Neste projeto, o grupo desenvolveu um sistema baseado em conhecimento que tem como objetivo criar um itinerário entre dois locais com meios de transporte intercalares. O objetivo é recomendar uma viagem com o menor custo possível, que tem de partir de Local1 e terminar a viagem no Local2, de acordo com os dados da tabela presente no enunciado do projeto.

A equipa baseou-se nos exercícios que o professor desenvolveu nas aulas teóricas e os códigos que estavam presentes nos slides da aula teórica, usando a estratégia “gerar e testar”. Esta estratégia percorre todo o espaço de procura até encontrar uma solução, ou seja, o itinerário com o menor custo possível.

Começamos por criar os vários arcos que definem as ligações entre cada cidade, por um grafo, da seguinte forma: arco(braga,porto). Para garantir a simetria também iremos apresentar o arco inverso arco:(porto,braga).

De seguida criamos o conjunto de dados relativos às várias viagens que contém a origem, destino, transporte, tempo e custo, por exemplo: transporte(braga, porto, autocarro, 70, 14), todos esses dados estão contidos na tabela do enunciado.

Começamos pelo primeiro predicado, que nos vai permitir verificar o caminho entre dois pontos, ou seja se dois pontos estão interligados por um arco. Este predicado usa recursão para explorar as conexões entre as cidades. A cada chamada recursiva, verifica se a cidade atual possui uma conexão com outra cidade que ainda não foi visitada. Se encontrar uma conexão válida, adiciona a cidade atual à lista de visitadas e continua a busca pela cidade de destino. Quando encontra o caminho entre as cidades, unifica a lista de cidades percorridas com o argumento Caminho.

O predicado “caminhomaisbarato” é o predicado principal que recebe duas cidades como argumentos e encontra o caminho mais barato entre elas. Ele chama os outros predicados para realizar essa tarefa. Primeiro, ele chama o predicado “caminho” para encontrar o caminho entre as cidades. Em seguida, chama o predicado “maisbarato” para verificar se o caminho encontrado é o mais barato. Se for, imprime o trajeto, o preço total da viagem e os transportes utilizados.

O predicado “maisbarato” verifica se o caminho atual é o mais barato entre duas cidades. Ele usa o predicado “barato” para verificar se existem caminhos mais baratos entre as mesmas cidades. Se não existir nenhum caminho mais barato, unifica o custo total do caminho atual com o argumento ‘Custo’ e a lista de transportes utilizados com o argumento ‘Transportes’.

O predicado “barato” verifica se existe um caminho mais barato entre duas cidades. Ele chama o predicado “caminho” para encontrar outros caminhos entre as mesmas cidades. Em seguida, chama o predicado “custoTotal” para calcular o custo total de cada caminho encontrado. Se encontrar um caminho com custo total menor que o custo atual, retorna verdadeiro.

O predicado “custoTotal” calcula o custo total de um caminho, somando os custos de transporte entre as cidades percorridas. Ele usa recursão para percorrer o caminho. A cada chamada recursiva, verifica o meio de transporte entre as duas cidades e adiciona o custo do transporte ao custo total. Unifica o custo total com o argumento ‘Custo’ e a lista de transportes utilizados com o argumento ‘Transportes’.



O predicado **“transportesUtilizados”** exibe os transportes utilizados ao longo do caminho. Ele usa recursão para percorrer o caminho e imprime numa interface as informações de cada transporte utilizado, incluindo a cidade de partida, a cidade de destino, o meio de transporte, o custo e a sua duração.

Parte B

Nesta segunda fase do projeto, o objetivo passa novamente por encontrar o caminho mais barato e, além disso, esse mesmo caminho deve ser percorrido o mais rápido possível.

Para tal, utilizando o código da Parte A, alteramos o nome do predicado ‘caminhomaisbarato’ para ‘melhorCaminho’, para se enquadrar melhor no contexto do problema.

Alteramos o predicado ‘transporte’ que se encontra no predicado ‘custoTotal’. Este passa a receber também como parâmetro o Tempo da viagem (‘transporte(B,A,Transporte, T, X)’). Ainda dentro do predicado ‘custoTotal’, alteramos o cálculo do Custo de ‘Custo is X + Y’ para ‘Custo is X * T + Y’, onde X * T corresponde ao produto do tempo pelo preço da viagem atual e Y corresponde ao preço das restantes viagens obtidas na chamada recursiva ‘custoTotal([B | C], Y, Transportes)’.

As últimas alterações foram feitas no predicado ‘transportesUtilizados’, onde voltamos a alterar o predicado ‘transporte’ para passar a receber como parâmetro o Tempo da viagem (‘transporte(A,B,Transporte,T,Custo)’ e adicionamos uma linha para imprimir o tempo de cada uma das viagens realizadas (‘write(‘ | Tempo: ’), write(T), write(‘ minutos’), nl.)).

Deste modo, é apresentado ao utilizador a melhor viagem, tendo em conta o seu custo e duração, e também o custo e duração de cada uma das viagens que o utilizador vai realizar até chegar ao seu destino.

Requisitos Mínimos	Cumprido	Não Cumprido
A solução (gerada pelo SBC) terá de ser implementada na linguagem Prolog via uma abordagem de Procura via Transição de Estados ou Otimização (por Hill Climbing).	*	
O SBC consegue atingir o objetivo 1.	X	

* O método utilizado foi o de gerar-testar

Requisitos Opcionais	Cumprido	Não Cumprido
Definir outras instâncias do problema, para qualquer configuração de partida e chegada;		X
Permitir que o utilizador escolha o método de resolução		X



5. CONCLUSÕES

5.1 Síntese

Este relatório é o reflexo do trabalho realizado pelo grupo. Nele apresentamos, não só, todo o planeamento e organização do projeto, como toda a nossa proposta de resolução e passos que seguimos para a realização dos sistemas baseados em conhecimento.

Na parte A do Projeto 1, como já foi referido, foi produzido um Sistema Baseado em Conhecimento, que mediante um conjunto de questões iterativas ao utilizador, irá atribuir o meio de transporte mais aconselhável para o mesmo, de acordo com as suas respostas na interface. Importante referir que o sistema de inferência utilizado tem como base a técnica Forward Chaining que nos permite através de um conjunto de factos, chegar a uma determinada conclusão. Para a parte B, elaboramos uma base de conhecimento através de técnicas de aquisição automática de conhecimento, utilizando o software RapidMiner.

No Projeto 2, foi criado novamente um Sistema Baseado em Conhecimento que permitisse a otimização da viagem do utilizador, de modo a obter uma solução que respondesse a dois objetivos essenciais, nomeadamente a minimização do custo da viagem, devolvendo sempre o itinerário mais barato para a viagem do utilizador, e o segundo objetivo que consiste em devolver sempre o itinerário que trará melhor relação entre a duração da viagem e o seu custo ao utilizador. Neste caso o grupo recorreu à técnica “Gerar e Testar” disponibilizada pelo docente da UC.

5.2 Discussão

A equipa, após uma revisão do trabalho feito, considera que o trabalho vai de encontro aos objetivos inicialmente definidos, e que estes foram cumpridos com sucesso pois conseguem dar uma resposta ao utilizador, de acordo com os critérios pretendidos.

Sendo assim, o grupo encontra-se bastante satisfeito com o trabalho realizado, apesar das várias dificuldades que fomos encontrando. Dificuldades essas a nível de compreensão do que nos era pedido no primeiro projeto, mas de resto manipulamos facilmente as ferramentas e os materiais que nos foram disponibilizados.

Relativamente aos temas escolhidos para os dois projetos, a equipa reconheceu que os temas eram bastante atuais e pertinentes à criação de um Sistema Baseado em Conhecimento, pois para além de podermos sobrepor os nossos conceitos da UC, levou-nos também a realizar várias pesquisas, de forma a tornar a solução o mais real possível.

Em função do trabalho realizado, a equipa autoavaliou o projeto para 15 valores.



5.3 Funcionamento do Trabalho em Grupo

As tarefas foram divididas de forma a garantir a contribuição de todos os elementos para a realização do projeto. A equipa demonstrou vontade de aprender e de colaborar para cumprir os prazos estabelecidos, além de apresentar uma comunicação eficiente e um espírito de equipa positivo.

Para garantir o progresso do projeto, foram realizadas reuniões semanais, nas quais todos os membros estavam presentes. Estas reuniões proporcionaram um espaço para discutir e analisar em que fase do projeto nos encontrávamos, compartilhar ideias, resolver problemas e tomar decisões em conjunto.

O contrato foi estabelecido e cumprido por todos, motivando o grupo a chegar ao objetivo final. Abaixo encontra-se a autoavaliação de cada elemento do grupo.

Nomes	Autoavaliação
Maria Ines Lima	14
João Mendes	15
Ana Silva	14
Gonçalo Rodrigues	15



6. ANEXOS

6.1 ANEXO A

PROJETO 1

Parte A

- **Basedeconhecimento.pl**

if origembraga and porto then curta.
if origembraga and viseu then longa.
if origembraga and lisboa then longa.
if origembraga and coimbra then longa.
if origemporto and braga then curta.
if origemporto and lisboa then longa.
if origemporto and viseu then curta.
if origemporto and coimbra then longa.
if origemlisboa and braga then curta.
if origemlisboa and porto then longa.
if origemlisboa and viseu then curta.
if origemlisboa and coimbra then longa.
if origemviseu and braga then curta.
if origemviseu and lisboa then longa.
if origemviseu and viseu then curta.
if origemviseu and coimbra then longa.
if origemcoimbra and braga then curta.
if origemcoimbra and lisboa then longa.
if origemcoimbra and viseu then curta.
if origemcoimbra and coimbra then longa.

if curta and demorada then turista.
if curta and rapida then apressado.
if longa and demorada then aventureiro.
if longa and rapida then trabalho.

if turista and poderiofinanceiro then conforto.
if turista and naopoderiofinanceiro then economico.
if apressado and poderiofinanceiro then conveniente.
if apressado and naopoderiofinanceiro then restrito.
if aventureiro and poderiofinanceiro then explorador.
if aventureiro and naopoderiofinanceiro then desafio.
if trabalho and poderiofinanceiro then diretor.
if trabalho and naopoderiofinanceiro then funcionario.



if conforto and ambiente then mota.
if conforto and naoambiente then carro.
if economico and ambiente then bicicleta.
if economico and naoambiente then autocarro.
if conveniente and ambiente then mota.
if conveniente and naoambiente then carro.
if restrito and ambiente then comboio.
if restrito and naoambiente then autocarro.
if explorador and ambiente then aviao.
if explorador and naoambiente then carro.
if desafio and ambiente then comboio.
if desafio and naoambiente then autocarro.
if diretor and ambiente then aviao.
if diretor and naoambiente then carro.
if funcionario and ambiente then mota.
if funcionario and naoambiente then carro.

- **Basededados.pl:**

transporte(comboio, origembraga, porto, 50, 8).
transporte(carro, origembraga, porto, 20, 12).
transporte(autocarro, origembraga, porto, 60, 7).
transporte(aviao, origembraga, porto, 10, 30).
transporte(bicicleta, origembraga, porto, 150, 0).
transporte(mota, origembraga, porto, 25, 11).

transporte(comboio, origemporto, braga, 50, 8).
transporte(carro, origemporto, braga, 20, 12).
transporte(autocarro, origemporto, braga, 60, 7).
transporte(aviao, origemporto, braga, 10, 30).
transporte(bicicleta, origemporto, braga, 150, 0).
transporte(mota, origemporto, braga, 25, 11).

transporte(comboio, origembraga, viseu, 180, 13).
transporte(carro, origembraga, viseu, 130, 30).
transporte(autocarro, origembraga, viseu, 190, 12).
transporte(aviao, origembraga, viseu, 50, 40).
transporte(bicicleta, origembraga, viseu, 300, 0).
transporte(mota, origembraga, viseu, 140, 28).

transporte(comboio, origemviseu, braga, 180, 13).
transporte(carro, origemviseu, braga, 130, 30).
transporte(autocarro, origemviseu, braga, 190, 12).
transporte(aviao, origemviseu, braga, 50, 40).
transporte(bicicleta, origemviseu, braga, 300, 0).



transporte(mota, origemviseu, braga, 140, 28).

transporte(comboio, origemlisboa, porto, 240, 20).
transporte(carro, origemlisboa, porto, 200, 40).
transporte(autocarro, origemlisboa, porto, 250, 18).
transporte(aviao, origemlisboa, porto, 100, 50).
transporte(bicicleta, origemlisboa, porto, 500, 0).
transporte(mota, origemlisboa, porto, 210, 38).

transporte(comboio, origemporto, lisboa, 240, 20).
transporte(carro, origemporto, lisboa, 200, 40).
transporte(autocarro, origemporto, lisboa, 250, 18).
transporte(aviao, origemporto, lisboa, 100, 50).
transporte(bicicleta, origemporto, lisboa, 500, 0).
transporte(mota, origemporto, lisboa, 210, 38).

transporte(comboio, origembraga, lisboa, 280, 25).
transporte(carro, origembraga, lisboa, 220, 45).
transporte(autocarro, origembraga, lisboa, 290, 23).
transporte(aviao, origembraga, lisboa, 110, 60).
transporte(bicicleta, origembraga, lisboa, 600, 0).
transporte(mota, origembraga, lisboa, 230, 42).

transporte(comboio, origemlisboa, braga, 280, 25).
transporte(carro, origemlisboa, braga, 220, 45).
transporte(autocarro, origemlisboa, braga, 290, 23).
transporte(aviao, origemlisboa, braga, 110, 60).
transporte(bicicleta, origemlisboa, braga, 600, 0).
transporte(mota, origemlisboa, braga, 230, 42).

transporte(comboio, origemporto, viseu, 80, 12).
transporte(carro, origemporto, viseu, 50, 19).
transporte(autocarro, origemporto, viseu, 90, 11).
transporte(aviao, origemporto, viseu, 20, 35).
transporte(bicicleta, origemporto, viseu, 300, 0).
transporte(mota, origemporto, viseu, 55, 17).

transporte(comboio, origemviseu, porto, 80, 12).
transporte(carro, origemviseu, porto, 50, 19).
transporte(autocarro, origemviseu, porto, 90, 11).
transporte(aviao, origemviseu, porto, 20, 35).
transporte(bicicleta, origemviseu, porto, 300, 0).
transporte(mota, origemviseu, porto, 55, 17).

transporte(comboio, origemlisboa, viseu, 220, 22).
transporte(carro, origemlisboa, viseu, 150, 28).
transporte(autocarro, origemlisboa, viseu, 230, 21).



transporte(aviao, origemlisboa, viseu, 40, 40).
transporte(bicicleta, origemlisboa, viseu, 500, 0).
transporte(mota, origemlisboa, viseu, 160, 26).

transporte(comboio, origemviseu, lisboa, 220, 22).
transporte(carro, origemviseu, lisboa, 150, 29).
transporte(autocarro, origemviseu, lisboa, 230, 21).
transporte(aviao, origemviseu, lisboa, 40, 40).
transporte(bicicleta, origemviseu, lisboa, 500, 0).
transporte(mota, origemviseu, lisboa, 160, 26).

transporte(comboio, origemporto, coimbra, 120, 14).
transporte(carro, origemporto, coimbra, 80, 24).
transporte(autocarro, origemporto, coimbra, 130, 13).
transporte(aviao, origemporto, coimbra, 20, 35).
transporte(bicicleta, origemporto, coimbra, 300, 0).
transporte(mota, origemporto, coimbra, 90, 22).

transporte(comboio, origemcoimbra, porto, 120, 14).
transporte(carro, origemcoimbra, porto, 80, 24).
transporte(autocarro, origemcoimbra, porto, 130, 13).
transporte(aviao, origemcoimbra, porto, 20, 35).
transporte(bicicleta, origemcoimbra, porto, 300, 0).
transporte(mota, origemcoimbra, porto, 90, 22).

transporte(comboio, origemviseu, coimbra, 110, 13).
transporte(carro, origemviseu, coimbra, 70, 20).
transporte(autocarro, origemviseu, coimbra, 120, 11).
transporte(aviao, origemviseu, coimbra, 15, 30).
transporte(bicicleta, origemviseu, coimbra, 250, 0).
transporte(mota, origemviseu, coimbra, 80, 18).

transporte(comboio, origemcoimbra, viseu, 110, 13).
transporte(carro, origemcoimbra, viseu, 70, 20).
transporte(autocarro, origemcoimbra, viseu, 120, 11).
transporte(aviao, origemcoimbra, viseu, 15, 30).
transporte(bicicleta, origemcoimbra, viseu, 250, 0).
transporte(mota, origemcoimbra, viseu, 80, 18).

transporte(comboio, origemcoimbra, braga, 140, 20).
transporte(carro, origemcoimbra, braga, 100, 30).
transporte(autocarro, origemcoimbra, braga, 150, 18).
transporte(aviao, origemcoimbra, braga, 25, 35).
transporte(bicicleta, origemcoimbra, braga, 400, 0).
transporte(mota, origemcoimbra, braga, 110, 28).

transporte(comboio, origembraga, coimbra, 140, 20).



```
transporte(carro, origembraga, coimbra, 100, 30).
transporte(autocarro, origembraga, coimbra, 150, 18).
transporte(aviao, origembraga, coimbra, 25, 35).
transporte(bicicleta, origembraga, coimbra, 400, 0).
transporte(mota, origembraga, coimbra, 110, 28).
```

```
transporte(comboio, origemcoimbra, lisboa, 150, 22).
transporte(carro, origemcoimbra, lisboa, 120, 35).
transporte(autocarro, origemcoimbra, lisboa, 160, 20).
transporte(aviao, origemcoimbra, lisboa, 30, 40).
transporte(bicicleta, origemcoimbra, lisboa, 500, 0).
transporte(mota, origemcoimbra, lisboa, 130, 33).
```

```
transporte(comboio, origemlisboa, coimbra, 150, 22).
transporte(carro, origemlisboa, coimbra, 120, 35).
transporte(autocarro, origemlisboa, coimbra, 160, 20).
transporte(aviao, origemlisboa, coimbra, 30, 40).
transporte(bicicleta, origemlisboa, coimbra, 500, 0).
transporte(mota, origemlisboa, coimbra, 130, 33).
```

- **forward.pl:**

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

```
demo:- new_derived_fact(P),
    !,

    assert(fact(P)),
    demo.
demo:- write( '').
```

```
new_derived_fact( Concl) :-
    if Cond then Concl,
    \+ fact( Concl),
    composed_fact( Cond).
```

```
composed_fact( Cond) :-
    fact( Cond).
```

```
composed_fact( Cond1 and Cond2) :-
    composed_fact( Cond1),
    composed_fact( Cond2).
```




```
composed_fact( Cond1 or Cond2) :-  
    composed_fact( Cond1)  
;  
composed_fact( Cond2).
```

- **Interface.pl:**

```
:- dynamic(fact/1),  
[forward, basededados, basedeconhecimento].
```

```
menu :- nl,nl ,  
write('*****  
*****'), nl,  
        write('Hora de viajar! Antes de iniciar, qual o seu nome?'),nl,  
        read(Nome) ,nl,  
  
        write('Prazer ajuda-lo/a Sr./a '), write(Nome) ,nl,nl,nl,  
  
        write('Vamos agora descobrir qual o meio de transporte mais  
recomendado para a sua viagem!'), nl,  
        write('Faremos algumas perguntas para o/a ajudar na sua  
escolha!'), nl,  
  
        write(' Menu:'), nl,  
  
        write(' 1- Iniciar'), nl,  
        write(' 2- Sair'), nl,nl,  
        read(Opcao),  
        avaliarEscolha(Opcao).
```

```
avaliarEscolha(1):- questao1.
```

```
avaliarEscolha(2):- write('Terminado'), halt.
```

```
avaliarEscolha(other):- write('Opcao invalida. Volte a escolher.'), menu.
```

```
questao1 :-
```

```
    write('*****  
*****'), nl,  
    write('Qual a origem da sua viagem? Introduza o numero correspondente:'), nl,  
  
    write(' 1 - Braga'), nl,  
    write(' 2 - Porto'), nl,  
    write(' 3 - Viseu'), nl,
```



```
write(' 4 - Lisboa'), nl,
write(' 5 - Coimbra'), nl,

read(Q1),
(
(Q1 == 1), assert(fact(origembraga)), questao2;
(Q1 == 2), assert(fact(origemporto)), questao2;
(Q1 == 3), assert(fact(origemviseu)), questao2;
(Q1 == 4), assert(fact(origemlisboa)), questao2;
(Q1 == 5), assert(fact(origemcoimbra)), questao2).

questao2 :-

write('*****
*****'), nl,
write(' Qual e o seu destino? Destas opções, introduza o numero correspondente:'),
nl,
write(' 1 - Braga'), nl,
write(' 2 - Porto'), nl,
write(' 3 - Viseu'), nl,
write(' 4 - Lisboa'), nl,
write(' 5 - Coimbra'), nl,
read(Q2),
(
(fact(origembraga), Q2 == 1) -> (write('Destino não pode ser igual a origem. Tente
novamente.'), questao2);
(fact(origemporto), Q2 == 2) -> (write('Destino não pode ser igual a origem. Tente
novamente.'), questao2);
(fact(origemviseu), Q2 == 3) -> (write('Destino não pode ser igual a origem. Tente
novamente.'), questao2);
(fact(origemlisboa), Q2 == 4) -> (write('Destino não pode ser igual a origem. Tente
novamente.'), questao2);
(fact(origemcoimbra), Q2 == 5) -> (write('Destino não pode ser igual a origem.
Tente novamente.'), questao2);
(Q2 == 1) -> assert(fact(braga)), questao3;
(Q2 == 2) -> assert(fact(porto)), questao3;
(Q2 == 3) -> assert(fact(viseu)), questao3;
(Q2 == 4) -> assert(fact(lisboa)), questao3;
(Q2 == 5) -> assert(fact(coimbra)), questao3
).
```



questao3 :-

```
write('*****')
*****'), nl,
write('Pretende realizar uma viagem rapida ou demorada?:'), nl,
write(' 1 - Rapida'), nl,
write(' 2 - Demorada'),nl,
read(Q3),
(
(Q3 == 1), assert(fact(rapida)), questao4;
(Q3 == 2), assert(fact(demorada)), questao4).
```

questao4 :-

```
write('*****')
*****'), nl,
write('Tem poderio financeiro? Escolha uma das opções:'), nl,

write(' 1 - Sim'), nl,
write(' 2 - Não'),nl,
read(Q4),
(
(Q4 == 1), assert(fact(poderiofinanceiro)), questao5;
(Q4 == 2), assert(fact(naopoderiofinanceiro)), questao5).
```

questao5:-

```
write('*****')
*****'), nl,
write(' Pretende optar por um transporte que polua menos?:'), nl,
write(' 1 - Sim'), nl,
write(' 2 - Não'),nl,
read(Q5),
(
(Q5 == 1), assert(fact(ambiente)), consulta;
(Q5 == 2), assert(fact(naoambiente)), consulta).
```

consulta :- demo, resultado.

resultado:- fact(Transp), fact(Origem), fact(Destino), transporte(Transp, Origem, Destino, Duracao, Custo),
write(' Transporte recomendado: '), write(Transp), nl,
write(' Duracao (em minutos): '), write(Duracao), nl,
write(' Custo (em euros): '), write(Custo).



Parte B

- **basedeconhecimento.pl**

if ambiente and naopoderiofinanceiro and demorada and braga and lisboa then comboio.
if ambiente and naopoderiofinanceiro and demorada and braga and porto then bicicleta.
if ambiente and naopoderiofinanceiro and demorada and braga and viseu then comboio.
if ambiente and naopoderiofinanceiro and demorada and coimbra then bicicleta.
if ambiente and naopoderiofinanceiro and demorada and lisboa then comboio.
if ambiente and naopoderiofinanceiro and demorada and porto then bicicleta.
if ambiente and naopoderiofinanceiro and demorada and viseu then comboio.
if ambiente and naopoderiofinanceiro and rapida and braga then mota.
if ambiente and naopoderiofinanceiro and rapida and coimbra then comboio.
if ambiente and naopoderiofinanceiro and rapida and porto then comboio.
if ambiente and naopoderiofinanceiro and rapida and viseu then mota.
if ambiente and poderiofinanceiro and braga then mota.
if ambiente and poderiofinanceiro and coimbra then aviao.
if ambiente and poderiofinanceiro and lisboa then aviao.
if ambiente and poderiofinanceiro and porto and braga then mota.
if ambiente and poderiofinanceiro and porto and lisboa and demorada then mota.
if ambiente and poderiofinanceiro and porto and lisboa and rapida then aviao.
if ambiente and poderiofinanceiro and viseu and braga then aviao.
if ambiente and poderiofinanceiro and viseu and porto then mota.
if ambiente and poderiofinanceiro and rapida then mota.
if naoambiente and naopoderiofinanceiro and demorada then autocarro.
if naoambiente and naopoderiofinanceiro and rapida and braga and lisboa then carro.
if naoambiente and naopoderiofinanceiro and rapida and braga and porto then autocarro.
if naoambiente and naopoderiofinanceiro and rapida and braga and viseu then carro.
if naoambiente and naopoderiofinanceiro and rapida and lisboa then carro.
if naoambiente and naopoderiofinanceiro and rapida and porto then autocarro.
if naoambiente and poderiofinanceiro then carro.

PROJETO 2

Parte A:

arco(braga, porto).
arco(porto, braga).
arco(braga, viseu).
arco(viseu, braga).



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

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



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

```
caminhomaisbarato(X,Y):-
    caminho(X,Y,Caminho), % gerar
    maisbarato(X,Y,Caminho,Custo,Transportes),
    write('Trajeto: '), reverse(Caminho, C) , write(C),nl,
    transportesUtilizados(Caminho, Transportes),
    write('Preco Viagem: '), write(Custo), write(' euros').
```

```
maisbarato(X,Y,Caminho, Custo, Transportes):-
    custoTotal(Caminho,Custo, Transportes),
    \+ barato(X,Y,Custo).
```

```
barato(X,Y,Custo):-
    caminho(X,Y,C1),
    custoTotal(C1,NC,_),
    NC < Custo.
```

```
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).
```

```
custoTotal([],0,[]).
custoTotal([A,B | C], Custo, [Transporte | Transportes]) :-
    transporte(B,A,Transporte,_,X),
    custoTotal([B | C], Y, Transportes),
    Custo is X + Y.
```

```
transportesUtilizados([], []).
transportesUtilizados([A,B | C], [Transporte | Transportes]) :-
    transporte(A,B,Transporte,_,Custo),
```



```
transportesUtilizados([B|C], Transportes),  
write('De '), write(B), write(' para '), write(A),  
write(' | Transporte: '), write(Transporte),  
write(' | Custo: '), write(Custo), write(' euros'), nl.
```

Parte B

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

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



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

melhorCaminho(X,Y):-

```
    caminho(X,Y,Caminho), % gerar
    maisbarato(X,Y,Caminho,Custo,Transportes),
    write('Trajeto: '), reverse(Caminho, C) , write(C),nl,
    transportesUtilizados(Caminho, Transportes),
    write('Preco*Duracao: '), write(Custo).
```

maisbarato(X,Y,Caminho, Custo, Transportes):-

```
    custoTotal(Caminho,Custo, Transportes),
    \+ barato(X,Y,Custo).
```

barato(X,Y,Custo):-

```
    caminho(X,Y,C1),
    custoTotal(C1,NC,_),
```




NC < Custo.

```
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).
```

```
custoTotal([],0,[]).
```

```
custoTotal([A,B | C], Custo, [Transporte | Transportes]) :-
```

```
transporte(B,A,Transporte,T,X),
```

```
custoTotal([B | C], Y, Transportes),
```

```
Custo is X * T + Y.
```

```
transportesUtilizados([], []).
```

```
transportesUtilizados([A,B | C], [Transporte | Transportes]) :-
```

```
transporte(A,B,Transporte,T,Custo),
```

```
transportesUtilizados([B | C], Transportes),
```

```
write('De '), write(B), write(' para '), write(A),
```

```
write(' | Transporte: '), write(Transporte),
```

```
write(' | Custo: '), write(Custo), write(' euros'),
```

```
write(' | Tempo: '), write(T), write(' minutos'), nl.
```

6.2 ANEXO B

Contrato do grupo

Este contrato de trabalho é celebrado entre o grupo constituído por Maria Ines Lima (96271), Gonçalo Rodrigues (95212), Ana Silva (96712) e João Mendes (97991) e contém as regras que todos os membros do grupo se comprometem a cumprir ao longo do projeto, que se regerá pelas seguintes:

- É esperado que todos os elementos pertencentes à equipa estejam presentes em todas as reuniões marcadas, exceto casos inesperados;
- É esperada a ajuda entre os elementos;
- Não poderá existir qualquer plágio de outros projetos e/ou conteúdos de outros autores, exceto caso este esteja devidamente identificado;
- Cada membro do grupo compromete-se a participar no desenvolvimento do projeto. Para tal, compromete-se a contribuir para a análise do problema, adquirir conhecimento e desenvolvimento do projeto
- Os elementos devem ser proativos nas reuniões;
- O grupo compromete-se a ser justo quanto à autoavaliação do projeto;
- Os elementos têm responsabilidade pelo trabalho desenvolvido;



- Quanto a tomada de decisões, estas devem ser discutidas e aprovadas por todos os elementos da equipa;
- Os elementos devem se tratar de forma que não haja qualquer mau ambiente entre a equipa;
- A equipa deverá avisar um elemento caso este não esteja a cumprir com as suas tarefas
- Cada elemento deve realizar o seu trabalho de forma a não comprometer o trabalho dos outros elementos;
- Cada elemento deve cumprir as suas tarefas dentro do tempo estipulado

O contrato foi reconhecido e validado por todos os elementos da equipa, e as suas assinaturas são evidência da sua validade e veracidade:

Azurém, 27 de Março 2023

Maria Inês Monteiro Lima
a96271@alunos.uminho.pt

Gonçalo Mendes Rodrigues
a95212@alunos.uminho.pt

João Pedro Mendes
a97991@alunos.uminho.pt

Ana João Silva
a96712@alunos.uminho.pt



BIBLIOGRAFIA

- Código e material disponibilizado pelo docente da unidade curricular.
- <https://www.swi-prolog.org/>