

UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

ELABORAÇÃO DE PLANOS DE TRANSPORTE DE PESSOAS

MARISOL DE BRITO CORREIA
(LICENCIADA)

Dissertação para a obtenção do grau de mestre em:
Engenharia Electrotécnica e de Computadores

Orientador:

Doutor Nuno João Neves Mamede

Júri:

Doutor Rogério dos Santos Carapuça

Doutor Hélder Manuel Ferreira Coelho

Doutor Nuno João Neves Mamede

LISBOA, MARÇO DE 1996

Marisol de Brito Correia

Departamento de: **Engenharia Electrotécnica e de Computadores**

Mestrado em: **Engenharia Electrotécnica e de Computadores**

Orientador: **Doutor Nuno João Neves Mamede**

Data: **Março de 1996**

Título:

ELABORAÇÃO DE PLANOS DE TRANSPORTE DE PESSOAS

RESUMO

O objectivo deste estudo é definir e apresentar as etapas necessárias para elaborar planos de transporte de pessoas e, de entre estes, escolher os melhores em função dos critérios indicados pelo utilizador.

Este estudo apresenta, em primeiro lugar, o conhecimento adquirido sobre os transportes e uma possível representação deste utilizando *os enquadramentos* como método de representação. Para se estabelecerem planos de transportes, foi necessário definir uma base de dados, de modo a se poder guardar e gerir a informação adquirida sobre estes. A base de dados é relacional, tendo-se utilizado como Sistema de Gestão de Base de Dados Relacional (SGBDR) o Microsoft Access 2.0.

Analisou-se o universo dos transportes utilizando a abordagem Entidade-Associação. Os diagramas obtidos foram posteriormente convertidos em relações, de modo a poderem ser suportados pelo SGBDR.

Os planos são construídos com base em algoritmos e heurísticas que foram desenvolvidos tendo presente as restrições e situações que ocorrem na realidade. Os algoritmos permitem construir planos com transportes directos e planos constituídos por vários transportes, incluindo o táxi. Para determinar os melhores planos, foram utilizadas as técnicas de decisão multi-atributo, especificamente o método da Pesagem Simples Aditiva. Os algoritmos foram desenvolvidos no próprio SGBDR, utilizando a linguagem Microsoft Access Basic 2.0.

Title:

SETTING UP PLANS FOR THE TRANSPORT OF PEOPLE

ABSTRACT

The aim of this essay is to define and present some necessary procedures to build people transport plans and among these plans to choose the best ones according to the criteria provided by the user.

This study presents the acquired knowledge about transport and its possible representation, using *frames* as a representation method.

To establish transport plans it was necessary to define a database in order to save and manage the obtained information about them. Once it is a relational database, the Microsoft Access 2.0 was used as Relational Database Management System (RDBMS).

The approach Entity-Relationship was used to analyse the universe of the transport system. The obtained diagrams were converted later on into relations so that they could be supported by RDBMS.

The plans are based on algorithms and heuristics which were developed keeping in mind the restrictions and events that may occur. The algorithms allow the setting of plans for direct transport and for various means of transport, including the taxi.

To find the best plans, the multi-attribute method was used, more specifically the Weighted Average Method. The algorithms were developed within the SGBDR, using the language of the Microsoft Access Basic 2.0.

Nome:

Marisol de Brito Correia

Título:

ELABORAÇÃO DE PLANOS DE TRANSPORTE DE PESSOAS

Palavras chave:

planos de transportes

representação do conhecimento

enquadramentos

base de dados relacional

algoritmos

método da pesagem simples aditiva.

Key words:

transport plans

knowledge representation

frames

relational database

algorithms

weighted average method.

Quero agradecer ao Prof. Nuno João Neves Mamede, pela orientação e conhecimentos transmitidos durante todo o mestrado; à Escola Superior de Gestão, Hotelaria e Turismo, da Universidade do Algarve, por apoiar e incentivar a formação dos seus docentes; aos meus pais e ao meu marido, pela compreensão e apoio demonstrados.

ÍNDICE GERAL

1. INTRODUÇÃO	10
1.1 Objectivos	10
1.2 Estrutura da tese	11
2. REPRESENTAÇÃO DO CONHECIMENTO	14
2.1 Introdução	14
2.2 Enquadramentos	14
2.2.1 Conceitos básicos	15
2.2.2 Representação dos enquadramentos utilizando o formalismo BNF	16
2.3 Significado dos valores dos atributos das classes (semântica)	30
2.4 Classificação dos transportes e definição das heranças entre as classes	31
2.5 Definição das estruturas de informação	40
3. DEFINIÇÃO E REALIZAÇÃO DA BASE DE DADOS	63
3.1 Introdução	63
3.2 Abordagem Entidade-Associação	64
3.2.1 Transportes Inter-urbanos	69
3.2.2 Transportes Urbanos	75
3.2.3 Táxis	80
3.3 Conversão dos diagramas E-R em relações	84
3.4 Comparação das relações obtidas com as classes definidas na representação do conhecimento	91
3.5 Realização da base de dados	92
3.5.1 Informação contida na base de dados	92
3.5.2 Aspectos de realização	93
4. ALGORITMOS E HEURÍSTICAS	95
4.1 Introdução	95
4.2 Algoritmos que determinam os planos de transportes	96
4.2.1 Algoritmos que determinam os planos de transportes para todos os tipos de transportes, excepto os táxis.	97
4.2.2 Algoritmos que determinam o plano de transporte para os táxis	123

4.3 Heurísticas _____	125
4.4 Técnica de decisão multi-atributo _____	128
4.4.1 Introdução _____	128
4.4.2 Caso prático _____	130
4.5 Algoritmos que determinam os melhores planos de transportes _____	135
4.6 Realização do programa desenvolvido _____	138
4.7 Exemplo de uma simulação do programa _____	139
5. CONCLUSÕES _____	143
ANEXOS	
BIBLIOGRAFIA _____	
A. RELAÇÕES-NOTAÇÃO LINEAR _____	151
B. INTRODUÇÃO DA INFORMAÇÃO NA BASE DE DADOS _____	155
C. SIMULAÇÃO DO PROGRAMA _____	171

ÍNDICE DE FIGURAS

FIG. 2.1. CLASSIFICAÇÃO DOS TRANSPORTES PÚBLICOS EM PORTUGAL _____	32
FIG. 2.2. ESQUEMA DE HERANÇAS PARA AS CLASSES URBANO E INTER_URBANO _____	33
FIG. 2.3. ESQUEMA DE HERANÇAS PARA AS CLASSES OPERADOR_URBANO , OPERADOR_INTER_URBANO E OPERADOR_TÁXI _____	35
FIG. 2.4. ESQUEMA DE HERANÇAS PARA AS CLASSES LOCAL_AQUISIÇÃO_BILHETES_URBANOS , LOCAL_AQUISIÇÃO_BILHETES_INTER_URBANOS , PARAGEM_E_LOCAL_AQUISIÇÃO_BILHETES_URBANOS E PARAGEM_E_LOCAL_AQUISIÇÃO_BILHETES_INTER_URBANOS _____	37
FIG. 2.5. ESQUEMA DE HERANÇAS PARA AS CLASSES BILHETES_URBANOS_PASSE E BILHETES_URBANOS_OUTROS _____	38
FIG. 2.6. ESQUEMA DE HERANÇA PARA A CLASSE CARACTERÍSTICAS_TRANSPORTE_INTER_URBANO _____	39
FIG. 3.1. CONJUNTO DE ENTIDADES: URBANO _____	64
FIG. 3.2. . CONJUNTO DE ASSOCIAÇÕES: TEM_7 _____	66
FIG. 3.3. ASSOCIAÇÃO DE MUITOS-PARA-UM: TEM_7 _____	67
FIG. 3.4. ASSOCIAÇÃO DE UM-PARA-UM: TEM_8 _____	67
FIG. 3.5. ASSOCIAÇÃO DE MUITOS-PARA-MUITOS: CATEGORIA_TEM_DESCONTOS _____	68
FIG. 3.6. CONJUNTO DE ENTIDADES COM ATRIBUTOS: MORADA_URBANO _____	69
FIG. 3.7. DIAGRAMA E-R PARA OS TRANSPORTES INTER-URBANOS _____	76
FIG. 3.8. DIAGRAMA E-R PARA OS TRANSPORTES URBANOS _____	81
FIG. 3.9. DIAGRAMA E-R PARA OS TÁXIS _____	83
FIG. 3.10. COMBINAÇÃO DO CONJUNTO DE ASSOCIAÇÕES TEM_4 COM O CONJUNTO DE ENTIDADES LOCAL_AQUISIÇÃO_BILHETES_INTER_URBANOS _____	85
FIG. 3.11. RELAÇÕES: OPERADOR_INTER_URBANO E LOCAL_AQUISIÇÃO_BILHETES_INTER_URBANOS _____	86
FIG. 3.12. COMBINAÇÃO DO CONJUNTO DE ASSOCIAÇÕES TEM_1 COM OS CONJUNTOS DE ENTIDADES TIPO_CATEGORIA E OUTRAS_CARACTERÍSTICAS _____	88
FIG. 3.13. RELAÇÃO: TIPO_CATEGORIA _____	88
FIG. 3.14. DIAGRAMA DE RELAÇÕES PARA OS TRANSPORTES INTER-URBANOS _____	90
FIG. 3.15. DIAGRAMA DE RELAÇÕES PARA OS TRANSPORTES URBANOS _____	90
FIG. 3.16. DIAGRAMA DE RELAÇÕES PARA OS TÁXIS _____	91

ÍNDICE DE TABELAS

TAB. 3.1. RELAÇÕES RESULTANTES DA COMBINAÇÃO DE CONJUNTOS DE ENTIDADES COM CONJUNTOS DE ASSOCIAÇÕES DE MULTIPLICIDADE MUITOS-PARA-UM _____	87
TAB. 3.2. CASO ESPECIAL: RELAÇÕES RESULTANTES DA COMBINAÇÃO DE CONJUNTOS DE ENTIDADES COM CONJUNTOS DE ASSOCIAÇÕES _____	89
TAB. 4.1. PLANOS DE TRANSPORTES POSSÍVEIS _____	140
TAB. 4.2. MELHORES PLANOS DE TRANSPORTES QUANDO O CRITÉRIO SELECIONADO CORRESPONDE AO TEMPO COM PRIORIDADE MÁXIMA _____	141
TAB. 4.3. MELHORES PLANOS DE TRANSPORTES QUANDO O CRITÉRIO SELECIONADO CORRESPONDE AO PREÇO COM PRIORIDADE MÁXIMA _____	141

1. INTRODUÇÃO

1

1.1 Objectivos

Este estudo visa a obtenção de procedimentos que permitam elaborar planos de transporte de pessoas e a escolha dos melhores em função dos critérios indicados pelo utilizador.

O utilizador, para além de indicar a localidade de onde pretende partir (a que chamaremos Origem) e a localidade onde pretende chegar (a que chamaremos Destino), indica também o dia e a hora a partir da qual pretende sair de Origem ou o dia e a hora antes da qual pretende chegar a Destino. Por outro lado, deve especificar também os critérios a considerar na determinação dos melhores planos de transportes. Estes critérios consistem na importância que atribui ao preço dos bilhetes, ao tempo de duração da viagem e à qualidade dos transportes.

O trabalho desenvolvido pode ser utilizado como um sistema de informação, que poderá estar disponível em terminais de transportes (aeroportos, terminais rodoviários, ferroviários, etc.), em agências de viagens ou em quiosques de informação espalhados pela cidade.

Uma utilização diferente das atrás mencionadas, que exigiria a introdução de outros critérios e da indicação de vários locais destino em vez de um só, seria o planeamento de circuitos turísticos, que indicaria como paragens destino os locais a visitar (poderia ser utilizado na EXPO-98 para planear o percurso a ser seguido pelo visitante, em função dos pavilhões que este pretende visitar).

A primeira abordagem ao problema foi definir a estrutura de informação, ou seja, representar o conhecimento sobre os transportes utilizando como método de representação os *enquadramentos*. De seguida e utilizando esta representação, definiu-se uma base de dados

que foi desenvolvida num Sistema de Gestão de Base de Dados Relacional (SGBDR), concretamente no Microsoft Access 2.0. A metodologia utilizada começa por analisar o sistema num nível abstracto, utilizando para tal a abordagem Entidade-Associação. Os diagramas obtidos foram convertidos em relações, por forma a serem suportados pelo SGBDR, tendo a preocupação de não introduzir redundância na base dados, de se manterem as restrições de integridade de entidades e de integridade referencial e de se obter um número mínimo de relações. Por último, desenvolveram-se os algoritmos e as heurísticas necessários para elaborar os planos de transportes possíveis e para determinar os melhores planos. Os planos tanto podem ser constituídos por transportes directos como por vários transportes (planos com transbordos). Como o objectivo é determinar os melhores planos de transportes em função dos critérios indicados pelo utilizador, recorreu-se às técnicas de decisão multi-atributo, utilizando o método da Pesagem Simples Aditiva. O programa que utiliza estes algoritmos e heurísticas, foi desenvolvido no próprio SGBDR, recorrendo-se ao Microsoft Access Basic 2.0.

1.2 Estrutura da tese

Este estudo está estruturado em cinco capítulos.

No segundo está definida a estrutura de informação utilizada para elaborar planos de transportes. O método de representação escolhido e apresentado neste capítulo são os *enquadramentos*. Este capítulo apresenta também a representação das classes-enquadramento, das instâncias enquadramento e dos valores dos atributos utilizando o formalismo BNF, indicando também o significado (semântica) destes valores. Posteriormente, são apresentadas as classes-enquadramento definidas, a descrição de cada um dos seus atributos, dos valores destes atributos e as relações de herança entre elas.

No terceiro capítulo, especificam-se as várias etapas percorridas para passar da representação do conhecimento sobre os transportes à base de

dados relacional. A primeira fase apresentada é a análise realizada ao universo dos transportes, utilizando a abordagem Entidade-Associação. A segunda, é a conversão dos diagramas obtidos em relações, por forma a garantir a não introdução de redundância na base de dados e a obtenção de um número mínimo de relações, mantendo as restrições de integridade.

No quarto capítulo são apresentados os algoritmos e heurísticas desenvolvidos para elaborar os planos de transportes. Os algoritmos desenvolvidos permitem determinar os planos de transportes possíveis para ir de um sítio a outro, utilizando todos os tipos de transportes excepto os táxis, determinar planos de táxis e por último, determinar os melhores planos de entre todos os possíveis. Neste capítulo é também explicado o método da Pesagem Simples Aditiva que permite determinar e seriar os melhores planos.

No quinto, são apresentadas as conclusões, indicadas possíveis melhorias e sugeridas outras abordagens ao problema.

Por último, acrescentaram-se três anexos, em que no primeiro, listam-se as relações obtidas no terceiro capítulo utilizando outra notação; no segundo, apresenta-se o manual do operador da base de dados, que explica a forma como se deve introduzir a informação na base de dados; no quarto e último anexo, é apresentada uma simulação do programa desenvolvido.

2. REPRESENTAÇÃO DO CONHECIMENTO

2

2.1 Introdução

O objectivo deste capítulo é definir a estrutura de informação associada à elaboração de planos de transporte de pessoas.

Começou-se por analisar os meios de transportes existentes no nosso país, as suas características, os tipos de serviços que oferecem e as condições de pagamento. Verificou-se o que havia de comum e de diferente entre eles.

Para representar este conhecimento, decidiu-se utilizar os *enquadramentos* ([1] e [2]) como método de representação do conhecimento.

A organização deste capítulo é a seguinte; na secção 2.2 são apresentados os conceitos básicos relativos *aos enquadramentos*, método de representação do conhecimento adoptado, assim como a representação das classes-enquadramento, das instâncias enquadramento e dos valores dos atributos utilizando o formalismo BNF. Na secção 2.3 é apresentado o significado dos valores dos atributos das classes-enquadramentos. Na secção 2.4 é feita a classificação dos meios de transportes, bem como a descrição das classes-enquadramento definidas e das relações de herança entre elas. Na secção 2.5 é apresentada a constituição de cada classe-enquadramento, a descrição de cada um dos seus atributos e dos valores destes atributos.

2.2 Enquadramentos

2.2.1 Conceitos básicos

Os enquadramentos são estruturas que permitem representar o conhecimento sobre um aspecto limitado do mundo, ou seja, são estruturas que permitem representar conjuntamente o conhecimento declarativo e o conhecimento procedimental sobre uma situação ou objecto.

Na maior parte dos sistemas, é possível distinguir dois tipos diferentes de enquadramentos: classe-enquadramento e instância enquadramento.

Uma classe-enquadramento corresponde a uma descrição de uma classe de entidades no mundo, enquanto que uma instância enquadramento é uma descrição de uma entidade individual nesse mesmo mundo. Por outras palavras, uma classe-enquadramento representa o conhecimento geral sobre determinada situação ou objecto, por exemplo, o curso universitário enquadramento, que representa o conhecimento geral sobre os cursos universitários, enquanto que uma instância enquadramento é uma particularização de uma classe-enquadramento. Um exemplo de uma instância enquadramento da classe curso universitário enquadramento seria um enquadramento que representasse o conhecimento acerca do curso de Engenharia Informática.

Para simplificar a terminologia utilizada, decidiu-se retirar a palavra enquadramento tanto nos termos já utilizados como nos que irão ser apresentados ao longo deste capítulo. Assim, a palavra classe substitui classe-enquadramento, instância substitui instância enquadramento, etc..

A informação é guardada nas classes através dos atributos. Um atributo é constituído por duas partes: o nome do atributo, que o identifica e o valor desse atributo.

Por outro lado, a partir de uma classe é possível criar outras classes, a que se dá o nome de subclasses, sendo estas utilizadas como especializações das classes que estão no topo da hierarquia. Por exemplo, no caso em estudo, existe uma classe chamada **Urbano**, que é uma subclasse (está hierarquicamente abaixo) de uma outra classe

chamada **Transporte** (que contém as descrições comuns aos transportes urbanos e aos transportes inter-urbanos).

No entanto, uma classe pode estar hierarquicamente dependente não apenas de outra classe, mas de várias (neste caso diz-se que se está perante um caso de herança múltipla).

Concluindo, uma classe pode não herdar atributos de nenhuma outra, (estar no topo da hierarquia), herdá-los apenas de outra e neste caso é uma herança simples, de duas ou mais classes e neste caso diz-se uma herança múltipla. Quando uma classe herda os atributos de outra(s) classe(s), herda-os na totalidade, podendo posteriormente acrescentar ou não outros atributos.

As instâncias são constituídas também por atributos, cujos valores descrevem um objecto ou uma situação particular. Nos valores desses atributos podem surgir vários tipos de dados, como por exemplo, cadeias de caracteres, números, identificadores de outras instâncias, etc..

Uma instância só pode ser instância de apenas uma classe.

2.2.2 Representação dos enquadramentos utilizando o formalismo BNF

2.2.2.1 Formalismo BNF

Para descrever a sintaxe, decidiu-se utilizar o formalismo BNF (Backus-Naur Form) ([3] e [4]), que foi inventado por John Backus, sendo posteriormente desenvolvido por Peter Naur (1960). A sua primeira aplicação foi a definição da sintaxe do ALGOL.

Para definir uma linguagem é necessário definir a sua sintaxe e a sua semântica. De uma forma simplificada pode-se dizer que a sintaxe é a forma da linguagem, enquanto que a semântica é o significado associado a essas formas.

Para descrever formalmente a sintaxe de uma linguagem, é necessário explicitar os seus componentes. Estes são:

1. Símbolos terminais; conjunto de símbolos que aparecem na linguagem.

2. Símbolos não terminais; conjunto de símbolos que não aparecem na linguagem, mas que são utilizados para descrever os componentes dessa linguagem.
3. Regras de produção; conjunto de regras que descrevem a estrutura dos diferentes componentes dessa linguagem.

As regras de produção do formalismo BNF consistem num componente da linguagem (símbolo não terminal) seguido do símbolo "::<=". Estes símbolos não terminais são colocados entre o sinal de menor "<" e o sinal de maior ">", enquanto que os símbolos terminais são escritos sem estes sinais. Por outro lado, o símbolo "|" representa possíveis alternativas.

Por uma questão de legibilidade decidiu-se representar os símbolos terminais e os não terminais com letras diferentes. Assim, os símbolos terminais ficam com o seguinte aspecto "símbolos terminais", enquanto que os não terminais ficam com a seguinte forma "<símbolos não terminais>".

2.2.2.2 Representação das classes

Uma classe é representada por um nome, pela especificação do(s) nome(s) da(s) classe(s) da qual herda os atributos, pela indicação de ser ou não uma classe abstracta e por último pelos seus atributos e pelos valores desses atributos.

Exemplo de uma classe:

```
{Transporte
  {Superclasse Ø}
  {Classe abstracta}
  {Operador: instância_id (Operador)}
  {Validade: intervalo data}
  {Horas_partida: lista de horas}
  {Tempo_percurso: lista de horas}
```

{Dias_efectua_percurso: dias da semana}}

A primeira especificação que surge numa classe é o seu nome, que na classe atrás apresentada é **Transporte**, seguido pela palavra **Superclasse** e do nome ou nomes das classes (no caso da herança ser simples ou múltipla, respectivamente), da qual esta classe herda os atributos. Pode acontecer também, que uma classe não herde nenhum atributo de outra classe; tal ocorrência é representada pelo símbolo \emptyset . No exemplo apresentado, a classe **Transporte** não herda os atributos de qualquer outra classe.

Quando a herança é múltipla, os nomes das classes herdadas devem ser separadas por uma vírgula.

Quando uma classe tem a indicação de **{Classe abstracta}**, que é o caso da classe **Transporte**, nunca são criadas instâncias dessa classe, ou seja, essa classe é apenas utilizada para ajudar a estruturar o conhecimento, possibilitando que os seus atributos sejam herdados por outras classes. De uma forma geral, esta classe será herdada por mais do que uma classe, porquanto não faria sentido estar a criar uma classe que não tem instâncias para poder ser herdada por apenas uma outra. Quando uma classe não tem indicação de **{Classe abstracta}** é porque esta classe pode originar instâncias.

Por último, uma classe pode ter vários atributos. Na definição da classe para além do nome do atributo, deverá estar especificado o tipo de dados que podem surgir nesses atributos, como por exemplo, um número, uma lista de números, uma cadeia de caracteres, etc..

Na classe **Transporte**, os nomes **Operador**, **Validade**, **Características_transporte**, **Horas_partida**, **Tempo_percurso** e **Dias_efectua_percurso** correspondem aos nomes dos atributos, enquanto que **instância_id (Operador)**, **intervalo data**, **instância_id (Características_transporte)**, **lista de horas** e **dias da semana** correspondem ao tipo de dados que podem surgir nesses atributos.

A seguir são apresentados os símbolos terminais, não terminais e as regras de produção que são utilizados tanto na definição das classes como na definição das instâncias.

Palavra

Utiliza-se quando queremos especificar apenas um nome.

Exemplos: Aeroporto, Saldanha

```
<carácter> ::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q |
R|S|T|U|V|W|X|Y|Z|a|b|c|d|e|f|g|h |i|j|k|l|m
|n|o|p|q|r|s|t|u|v|w|x|y|z |0|1|2|3|4|5|6|7|8|
9|!|@|§|#|$|%|^|& |*|_|-|+|^|^|^|=|;|:|'|"|<|>|
(|)|[|]|.|/| \|?|~|^|«|»|" |Ø|Ç|ç|á|é|í|ó|ú|Á|É
|Í| Ó|Ú|à|À|ã|õ|Ã|Õ|â|ê|ô|Â|Ê|Ô
<palavra> ::= <carácter> | <carácter><palavra>
```

Nome classe

Utiliza-se para especificar o nome de uma classe.

Exemplos: Transporte, Operador

```
<nome classe> ::= <palavra>
```

Nome atributo

Utiliza-se para especificar o nome de um atributo, tanto das classes como das instâncias.

Exemplos: Validade, Horas_partida

```
<nome atributo> ::= <palavra>
```

A seguir são apresentados os símbolos terminais, não terminais e as regras de produção que definem apenas as classes.

Lista nomes classes

Utiliza-se para especificar vários nomes de classes.

Exemplo: Paragem,Local_aquisição_bilhetes_inter_urbanos

```
<lista nomes classes> ::= <nome classe> | <nome classe>, <lista nomes classes>
```

Tipo atributo

Utiliza-se para especificar a informação que surge nos atributos das classes.

Exemplos: Número, Preço

```
<tipo atributo> ::= palavras |  
    número |  
    preço |  
    coordenada |  
    lista de números telefone_fax |  
    hora |  
    lista de horas |  
    horário |  
    intervalo data |  
    booleano |  
    intervalo idade |  
    dias da semana |  
    instância_id (<nome classe>) |  
    lista de instâncias_id (<nome classe>) |  
    lista de instâncias_id (<nome classe> ou <nome classe>) |  
    instância_id (<nome classe>) ou  
        instância_id (<nome classe>) |  
    lista de instâncias_id (<nome classe>) ou preço |  
    tabela
```

Repare-se que na regra de produção anterior aparecem símbolos terminais e não terminais.

Atributo classe

Permite especificar a representação de um atributo nas classes.

Exemplo: {Tempo_percurso: lista de horas}

```
<atributo classe> ::= {<nome atributo>: <tipo atributo>}
```

Lista atributos classe

É utilizada para definir vários atributos numa classe.

Exemplo:

{Operador: instância_id (Operador)}

{Validade: intervalo data}

{Horas_partida: lista de horas}

{Tempo_percurso: lista de horas}

{Dias_efectua_percurso: dias da semana}

```
<lista atributos classe> ::=  
<atributo classe> |  
<atributo classe><lista atributos classe>
```

Classe

Permite definir uma classe.

Exemplo:

{Transporte

{Superclasse \emptyset }

{Classe abstracta}

{Operador: instância_id (Operador)}

{Validade: intervalo data}

{Horas_partida: lista de horas}

{Tempo_percurso: lista de horas}

{Dias_efectua_percurso: dias da semana}}

```
<classe> ::=
  {<nome classe>
    {Superclasse <lista nomes classes>}
    {Classe abstracta}
    <lista atributos classe>} |

  {<nome classe>
    {Superclasse <lista nomes classes>}
    <lista atributos classe>} |

  {<nome classe>
    {Superclasse <lista nomes classes>}}
```

Uma classe pode ser representada como na primeira parte da regra de produção e neste caso está-se perante uma classe abstracta na qual são definidos atributos; ou como na segunda parte, e neste caso está-se perante uma classe que não é abstracta mas na qual são definidos atributos ou, por último, como na terceira parte, onde se define uma classe que não é abstracta e na qual não são acrescentados outros atributos.

Em relação à terceira parte da regra de produção, convém salientar, embora pareça óbvio, que não faz sentido criar uma classe sem atributos, no entanto pode acontecer que se crie uma classe que herda o(s) atributo(s) de outra(s) classe(s), mas na qual não são definidos outros atributos. Como exemplo disso temos as classes **Paragem_e_local_aquisição_bilhetes_urbanos** e **Paragem_e_local_aquisição_bilhetes_inter_urbanos**, definidas na secção 2.5.

De notar que as classes são representadas entre chavetas.

2.2.2.3 Representação das instâncias

Como já foi referido, as instâncias são utilizadas para representar uma entidade particular, ou seja, uma entidade única do universo em estudo.

Uma instância pode representar, por exemplo, um determinado comboio intercity, que faz a ligação Faro-Lisboa, ou pode representar uma determinada paragem, que está localizada numa cidade específica, em determinada rua/avenida/praceta, etc..

Uma instância é representada por um identificador, pela especificação de qual a classe de que é instância e dos atributos que a constituem.

Exemplo de uma instância:

{P_555

{Membro_de Paragem}

{Nome: Saldanha}

{Morada: MA2}

**{Transportes_que_param_nessa_paragem: Autocarro_44,
Autocarro_11}}**

O identificador de uma instância é o que identifica essa instância, ou seja, não poderá haver duas instâncias com o mesmo identificador. No exemplo atrás apresentado o identificador da instância é **P_555**.

A seguir surge a palavra **Membro_de** seguida do nome da classe de que é instância. No exemplo em análise, a instância **P_555** é uma instância da classe **Paragem**. Como é óbvio, uma instância só pode ser instância de apenas uma classe.

Por último, uma instância pode ter um ou vários atributos (não faz sentido ter uma instância sem atributos). Para além do nome de cada atributo surge também o seu valor. Os valores desses atributos podem ser números, identificadores de instâncias de outras classes, etc.. No exemplo em questão, a instância tem três atributos, **Nome**, **Morada** e **Transportes_que_param_nessa_paragem**, cujos valores são **Saldanha**, **MA2** e **Autocarro_44**, **Autocarro_11**, respectivamente. Neste estudo quando não é possível dar um valor a um atributo deixa-se em branco.

Os símbolos terminais, não terminais e as regras de produção que definem as instâncias são apresentados de seguida.

Instância_id

Define o identificador de uma instância.

Exemplos: P_555, Autocarro_11

```
<instância_id> ::= <palavra>
```

Valor atributo

Define os valores dos atributos nas instâncias.

Exemplos: Autocarro_44, Autocarro_11, Saldanha

```
<valor atributo> ::= <palavras> |  
    <número> |  
    <preço> |  
    <coordenada> |  
    <lista de números telefone_fax> |  
    <hora> |  
    <lista de horas> |  
    <horário> |  
    <intervalo data> |  
    <booleano> |  
    <intervalo idade> |  
    <dias da semana> |  
    <tabela> |  
    <instância_id> |  
    <lista de instâncias_id>
```

Repare-se na diferença entre Tipo atributo e Valor atributo, que à primeira vista podem causar confusão. Enquanto que em Tipo atributo surgem símbolos terminais que correspondem aos nomes dos vários tipos de dados que podem surgir nos atributos aquando da definição das classes, em Valor atributo, surgem símbolos não terminais, que indicam os tipos de dados que podem surgir nos atributos das instâncias.

Os símbolos não terminais apresentados nesta regra de produção, são explicados posteriormente nesta secção.

Atributo instância

Permite especificar a representação de um atributo nas instâncias.

Exemplo: {Nome: Saldanha}

```
<atributo instância> ::= {<nome atributo>: <valor atributo>}
```

Lista atributos instância

É utilizada para definir vários atributos numa instância.

Exemplo:

{Nome: Saldanha}

{Morada: MA2}

{Transportes_que_param_nessa_paragem: Autocarro_44,
Autocarro_11}}

```
<lista atributos instância> ::=  
  <atributo instância> |  
  <atributo instância><lista atributos instância>
```

Instância

Permite definir uma instância.

Exemplo:

{P_555

 {Membro_de Paragem}

 {Nome: Saldanha}

 {Morada: MA2}

 {Transportes_que_param_nessa_paragem: Autocarro_44,
 Autocarro_11}}

```
<instância> ::=
```

```
{<instância_id>  
  {Membro_de <nome classe>}  
  <lista atributos instância>}
```

Repare-se que as instâncias são apresentadas entre chavetas.

Representação dos valores dos atributos das instâncias

Indicam-se a seguir, os símbolos terminais, não terminais e as regras de produção que descrevem os valores dos atributos das instâncias.

Palavras

Utiliza-se quando queremos especificar um nome, descrever o nome de uma rua, etc..

Exemplos: Lisboa, Rua Vasco da Gama, 15

```
<palavras> ::= <palavra> | <palavra> <palavras>
```

O símbolo não terminal <palavra> já foi definido anteriormente.

Número

Permite especificar um número inteiro.

Exemplos: 33, 520

```
<dígito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9  
<número> ::= <dígito> | <dígito><número>
```

Preço

Permite especificar um preço.

Exemplos: 1250, 12000

```
<preço> ::= <número>
```

Coordenada

Permite especificar uma coordenada topográfica (cartográfica) de qualquer ponto do país. Cada coordenada indica a distância, em metros, ao longo de um meridiano (coordenada P) ou ao longo de um paralelo (coordenada M), entre a origem de um referencial e o ponto em análise. Este referencial refere-se a um sistema de eixos ortonormados, com a origem num ponto junto à localidade de Melriça (Centro Cartográfico de Portugal).

Exemplo: -228,000.0

```
<coordenada> ::= <número>.<número> | <número>,<número>.<número> |
               -<número>.<número> | -<número>,<número>.<número>
```

Lista de números telefone_fax

Permite especificar vários números de telefone ou fax.

Exemplo: (089) 346787,(01) 569843

```
<número telefone_fax> ::= (<número>) <número>
<lista de números telefone_fax> ::=
  <número telefone_fax> |
  <número telefone_fax>,<lista de números telefone_fax>
```

Hora

Utiliza-se quando queremos especificar uma hora ou uma hora juntamente com um tempo de espera (hora).

Exemplo: 05:26
06:38+0:05

```
<hora simples> ::= 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 |
                  11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
                  22 | 23
<minuto> ::= 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 |
             13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
             25 | 26 | 27 |
             28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 |
             38 | 39 | 40 | 41 | 42 |
```

43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 |
58 | 59

<hora> ::= <hora simples>:<minuto>

<horas> ::= <hora> | <hora>+<hora>

Lista de horas

Utiliza-se quando queremos especificar várias horas.

Exemplo: 05:26+00:10,07:00,23:01

<lista de horas> ::= <horas> | <horas>,<lista de horas>

Horário

Permite especificar um horário de funcionamento.

Exemplo: 09:00-12:30,14:00-17:30

<horário turno> ::= <hora>-<hora>

<horário> ::= <horário turno> | <horário turno>,<horário>

Intervalo data

Permite especificar um período de tempo.

Exemplo: 01 Março-30 Setembro

<dia> ::= 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 |
14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27
| 28 | 29 | 30 | 31

<mês> ::= Janeiro | Fevereiro | Março | Abril | Maio | Junho |
Julho | Agosto | Setembro | Outubro | Novembro |
Dezembro

<data> ::= <dia> <mês>

<intervalo data> ::= <data>-<data>

Booleano

Permite especificar o valor lógico verdadeiro ou falso

Exemplos: V, F

```
<valor lógico verdadeiro> ::= V  
<valor lógico falso> ::= F  
<booleano> ::= <valor lógico verdadeiro> | <valor lógico falso>
```

Intervalo idade

Utiliza-se quando queremos especificar um intervalo entre duas idades.

Exemplo: [5,10]

```
<idade> ::= <dígito> | <dígito><dígito> | <dígito><dígito><dígito>  
<intervalo idade> ::= [<idade>,<idade>]
```

Dias da semana

Permite especificar os vários dias da semana.

Exemplo: Seg,Qui,Sáb

```
<dias> ::= Seg | Ter | Qua | Qui | Sex | Sáb | Dom  
<dias da semana> ::= <dias> | <dias>,<dias da semana>
```

Tabela

Permite representar uma tabela. Porque esta estrutura é complicada, decidiu-se não especificar nenhuma regra de produção (ver secção 2.5 para mais informação).

Instância_id

Este símbolo não terminal já foi especificado anteriormente.

Lista de instâncias_id

Permite especificar uma lista de identificadores de instâncias.

Exemplo: P_3,P_4,P_5,P_6,P_7,P_8

```
<lista de instâncias_id> ::=
```

<instância_id> |
<instância_id>,<lista de instâncias_id>

2.3 Significado dos valores dos atributos das classes (semântica)

Enquanto que em alguns atributos das classes é possível compreender imediatamente qual o tipo de dados que podem surgir nos atributos das instâncias dessas classes, existem outros, mais complexos, cujo significado se explica a seguir. Por uma questão de simplicidade decidiu-se utilizar a designação "valor dos atributos" para referir o valor dos atributos das instâncias das classes.

Quando no atributo de uma classe surgir `instância_id` (Operador), indica que o valor desse atributo será o identificador de uma instância da classe **Operador**.

Quando aparecer lista de `instâncias_id` (Paragem), o valor do atributo será constituído por vários identificadores de instâncias da classe **Paragem**.

No caso de surgir lista de `instâncias_id` (`Bilhetes_urbanos_passe` ou `Bilhetes_urbanos_outros`), o valor do atributo será uma lista de identificadores de instâncias tanto da classe **Bilhetes_urbanos_passe** como da classe **Bilhetes_urbanos_outros**. Repare-se que neste caso o atributo pode conter uma lista de identificadores de instâncias só da classe **Bilhetes_urbanos_passe**, ou uma lista de identificadores de instâncias apenas da classe **Bilhetes_urbanos_outros** ou ainda uma lista com instâncias das duas classes.

Também pode surgir, `instância_id` (Paragem) ou `instância_id` (`Paragem_e_local_aquisição_bilhetes_urbanos`) que indica que no atributo surgirá ou um identificador de uma instância da classe **Paragem** ou um identificador de uma instância da classe **Paragem_e_local_aquisição_bilhetes_urbanos**.

Portanto, sempre que surge a sequência de símbolos terminais ou, está-se perante uma alternativa de opções, com excepção de lista de instâncias_id (Bilhetes_urbanos_passe ou Bilhetes_urbanos_outros) que, como se viu, permite que surjam instâncias de ambas as classes.

Tendo em conta o significado atribuído a ou é fácil perceber que, lista de instâncias_id (Tipo_desconto) ou preço indica que no atributo surgirá ou uma lista de identificadores de instâncias da classe **Tipo_desconto** ou um preço.

As classes referidas foram dadas como exemplo, uma vez que nos atributos das classes podem ser especificadas instâncias ou listas de instâncias de quaisquer outras classes.

2.4 Classificação dos transportes e definição das heranças entre as classes

A primeira preocupação que surgiu para elaborar os planos de transporte de pessoas foi analisar os tipos de transportes públicos existentes no nosso país. Verificou-se então que um possível esquema de classificação poderia ser o apresentado na fig. 2.1..

Os transportes aparecem divididos em três grupos: os transportes terrestres, os aéreos e os marítimos. Cada um destes grupos divide-se por sua vez em outros grupos e assim sucessivamente. No entanto ao analisar este esquema, verificou-se que haviam meios de transportes com características comuns, pelo que se optou representar estas características numa só classe. Assim, classificaram-se os transportes em três classes: a classe **Urbano**, onde está representado o conhecimento sobre os transportes urbanos, a classe **Inter_urbano** onde está representado o conhecimento sobre os transportes inter-urbanos e a classe **Táxi**, onde é especificado o conhecimento sobre o táxi.

Considerou-se transporte urbano, aquele onde se utilizam os bilhetes pré-comprados, passes, etc. e que corresponde aos transportes públicos que realizam percursos de curta duração, ou seja, os percursos no interior das cidades ou vilas ou entre localidades próximas.

Nos transportes inter-urbanos, já não se utilizam os bilhetes atrás referidos, uma vez que nestes casos os bilhetes dependem da categoria de viagem (na linguagem comum utiliza-se o termo classe para especificar determinada categoria, por exemplo, 1ª classe, 2ª classe, classe turística, etc., no entanto para não confundir com o termo classe referente aos *enquadramentos*, decidiu-se utilizar o termo categoria de viagem). Estes transportes efectuam percursos de longa duração, ou seja, realizam as ligações entre as cidades, entre as vilas e entre os países.

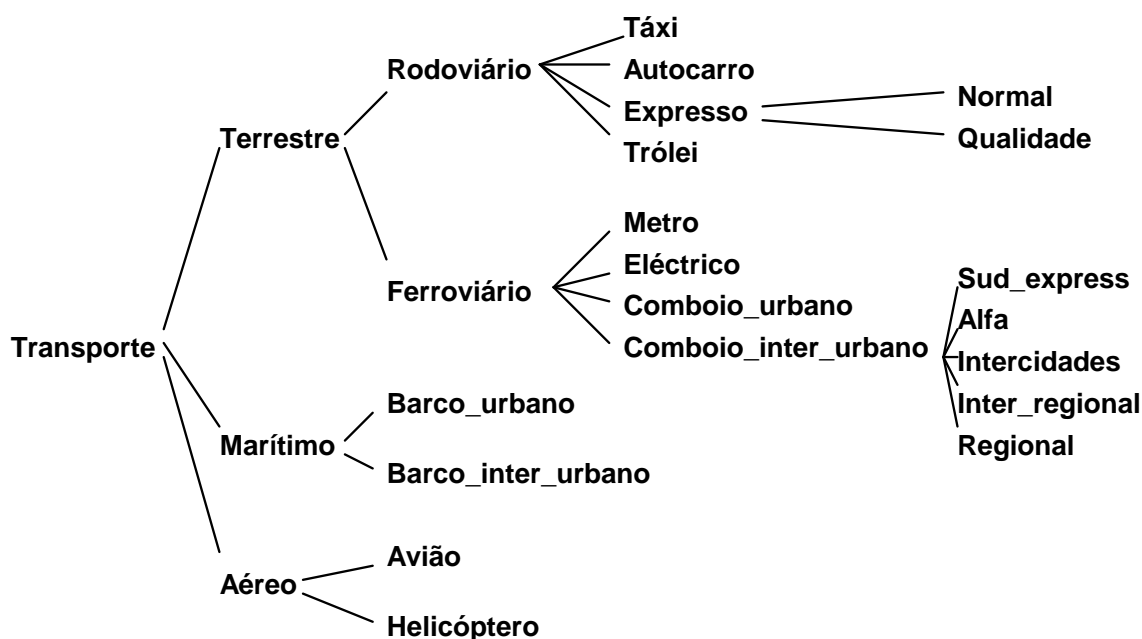


Fig. 2.1. Classificação dos transportes públicos em Portugal

No entanto existem características comuns entre as classes **Urbano** e **Inter_urbano**, que estão representadas numa classe abstracta chamada **Transporte**. Logo, estas classes herdam todos os atributos da classe **Transporte**. Nos atributos desta classe é possível definir o operador a que pertence determinado transporte, o período de tempo durante o qual determinado percurso se efectua (verificou-se que existem mais

transportes a funcionar durante os meses de verão), a hora de partida do transporte, o tempo de percurso e os dias da semana em que o percurso se efectua. O esquema de heranças para estas classes é o seguinte:

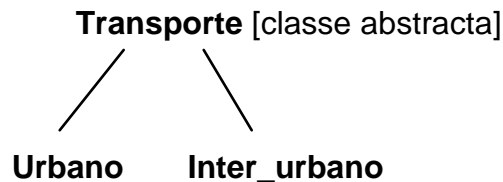


Fig. 2.2. Esquema de heranças para as classes **Urbano** e **Inter_urbano**

A classe **Urbano** permite especificar informação referente aos transportes urbanos, como por exemplo, o local de partida, de chegada, as paragens intermédias, o número de bilhetes necessários para ir de uma paragem a outra, etc.. A classe **Inter_urbano** permite especificar informação relativa aos transportes inter-urbanos, como por exemplo, o local de partida, de chegada, paragens intermédias, número do transporte (veículo), os preços, etc..

Embora à primeira vista se possa pensar que é possível definir em comum os atributos referentes ao local de partida, ao local de chegada e às paragens intermédias tanto dos transportes urbanos como dos inter-urbanos na classe **Transporte**, tal não é possível porque o valor desses atributos difere. Nos transportes urbanos, o valor dos atributos local de partida e local de chegada, só pode ser um identificador de uma instância ou da classe **Paragem** (quando na paragem não é possível comprar bilhetes) ou da classe **Paragem_e_local_aquisição_bilhetes_urbanos** (quando a compra do bilhete possa ser efectuada nas paragens), enquanto que nos transportes inter-urbanos, o valor destes atributos, corresponde ao identificador de uma instância ou da classe **Paragem** (quando a compra do bilhete não pode ser efectua nas paragens, mas em locais próprios) ou da classe **Paragem_e_local_aquisição_bilhetes_inter_urbanos** (quando a compra do bilhete possa ser efectuada nas paragens). A diferença reside no facto de que quando a compra do bilhete

é feita nas paragens, em relação aos transportes urbanos, é necessário indicar quais os tipos de bilhetes vendidos nessas paragens, enquanto que para os transportes inter-urbanos, quando nas paragens também se pode comprar os bilhetes, é necessário indicar quais os tipos de bilhetes vendidos, ou seja, se é possível comprar bilhetes para todos os tipos de transportes pertencentes a um operador. É que os tipos de bilhetes vendidos e utilizados para os transportes urbanos são diferentes dos vendidos e utilizados para os transportes inter-urbanos.

Em relação aos transportes urbanos, decidiu-se utilizar o conceito de bilhete, uma vez que neste tipo de transporte se usam vários tipos de bilhetes, como o bilhete simples, o pré_comprado, a série de vários bilhetes, o passe, etc.. Como se sabe, o preço destes bilhetes não depende do trajecto que o utente irá percorrer. Este aspecto não é similar nos transportes inter-urbanos, em que o preço do bilhete depende do trajecto que o utente percorrerá e da categoria de viagem em que este quer viajar. Assim, optou-se por não se considerar nos transportes inter-urbanos, classes referentes à entidade “Bilhete”.

Para definir as informações relativas às empresas que exploram determinado transporte, criou-se a classe **Operador**, classe abstracta que permite especificar informação relativa ao nome do operador.

Houve necessidade de definir a classe **Operador_urbano**, que como o nome indica é uma classe necessária para definir os transportes urbanos, porque nela está definida informação referente aos tipos de bilhetes utilizados por determinado operador (por exemplo, a empresa que gere o metropolitano de Lisboa, disponibiliza o bilhete simples, o bilhete simples obtido da máquina, a série de 10 bilhetes da máquina, os passes, etc.) e aos locais onde se podem comprar esses bilhetes, podendo estes ser ou não locais de paragem.

Definiu-se também a classe **Operador_inter_urbano**, para poder especificar por um lado, quais os tipos de transportes pertencentes a determinado operador (por exemplo, a CP (Caminhos de Ferro Portugueses) tem vários tipos de comboios, como o Alfa, o Intercidades, o

Inter_regional, o Regional, etc.) e por outro, os locais de aquisição dos bilhetes, podendo estes ser ou não locais de paragem.

Por último, para representar a informação referente aos operadores que gerem os táxis, definiu-se a classe **Operador_táxi** que também herda os atributos da classe **Operador** e que permite representar informação referente às moradas da Central de táxis, os seus números de telefone e de fax. O esquema a seguir, apresenta as classes existentes relativas aos operadores e as suas relações de herança:

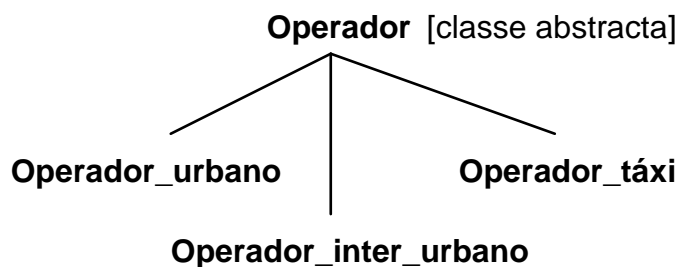


Fig. 2.3. Esquema de heranças para as classes **Operador_urbano**, **Operador_inter_urbano** e **Operador_táxi**

Na classe **Morada** estão definidas as características de localização de qualquer sítio, seja este paragem e local de aquisição de bilhetes, ou só paragem, ou só local de aquisição de bilhetes. Esta classe permite definir a rua (avenida, praça, praceta, etc.) e o número da porta, o código postal, o concelho, o distrito, a coordenada latitude e a coordenada longitude onde determinado local se situa.

Para guardar as informações relativas aos locais onde se adquirem os bilhetes, como por exemplo, a morada do local, os números de telefone, os números de fax, o horário de funcionamento, etc., definiu-se a classe **Local_aquisição_bilhetes**.

Houve necessidade de definir a classe **Local_aquisição_bilhetes_urbanos**, que herda todos os atributos da classe **Local_aquisição_bilhetes**, para poder especificar a informação relativa aos tipos de bilhetes vendidos para os transportes urbanos em determinado local. Repare-se que neste atributo (Tipos_bilhetes_vendidos) serão especificados os tipos de bilhetes vendidos dos bilhetes utilizados por determinado operador, (por exemplo,

há estações do metropolitano de Lisboa onde não existem máquinas para venda de bilhetes, e onde portanto não se podem comprar estes tipos de bilhetes).

Por outro lado, na classe **Local_aquisição_bilhetes_inter_urbanos**, que também herda todos os atributos da classe **Local_aquisição_bilhetes**, indica-se quais os bilhetes vendidos para os transportes inter-urbanos pertencentes a determinado operador em determinado local. Repare-se que nos transportes inter-urbanos o tipo de bilhete depende dos tipos de transportes disponibilizados pelo operador (tomando o exemplo da CP atrás referido, será que nas estações onde não passa o Sud_express ou o Alfa se podem adquirir bilhetes para estes comboios?).

A classe **Paragem** é utilizada na definição das características relativas aos sítios onde determinado transporte faz as suas paragens, mas onde não é possível adquirir bilhetes. Nos atributos desta classe é possível especificar a morada, os transportes que param nessa paragem e o seu nome.

Para evitar que houvesse duplicação dos nomes das paragens que são também locais de aquisição de bilhetes e um número excessivo de instâncias, definiu-se a classe **Paragem_e_local_aquisição_bilhetes_urbanos** para os transportes urbanos e a classe **Paragem_e_local_aquisição_bilhetes_inter_urbanos** para os transportes inter-urbanos. De notar que se estas classes não existissem, para um local de paragem, que é também local de aquisição de bilhetes, haveria necessidade de criar duas instâncias, uma da classe **Paragem** e outra da classe **Local_aquisição_bilhetes_urbanos** para os transportes urbanos ou da classe **Local_aquisição_bilhetes_inter_urbanos** para os transportes inter-urbanos, podendo-se juntar toda esta informação numa só instância da classe **Paragem_e_local_aquisição_bilhetes_urbanos** para os transportes urbanos ou da classe **Paragem_e_local_aquisição_bilhetes_inter_urbanos** para os transportes inter-urbanos.

O esquema das heranças é o seguinte:

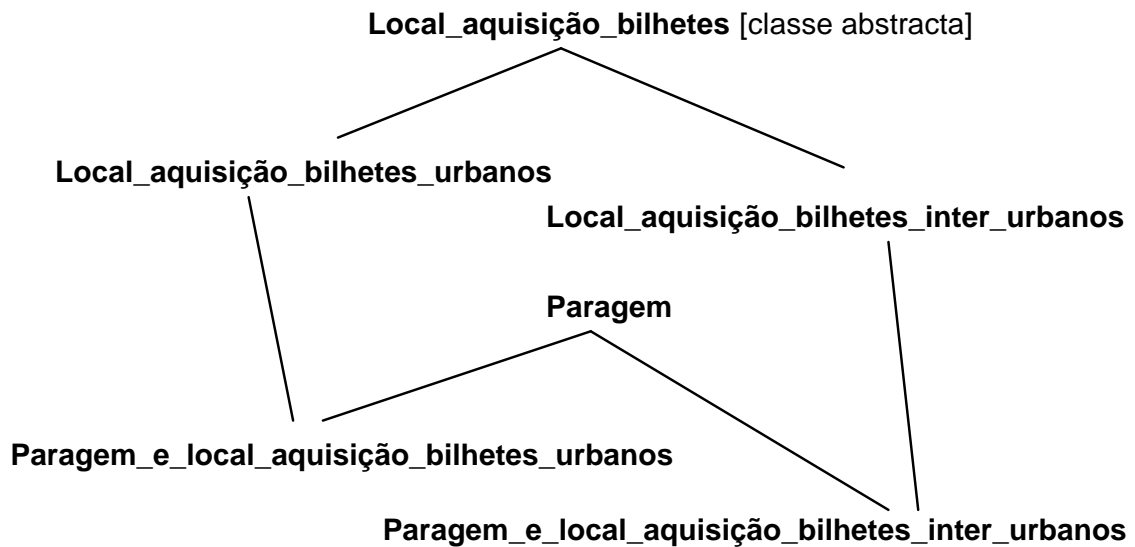


Fig. 2.4. Esquema de heranças para as classes **Local_aquisição_bilhetes_urbanos**, **Local_aquisição_bilhetes_inter_urbanos**, **Paragem_e_local_aquisição_bilhetes_urbanos** e **Paragem_e_local_aquisição_bilhetes_inter_urbanos**

Para representar a informação referente aos bilhetes utilizados nos transportes urbanos, criou-se a classe **Bilhetes_urbanos** que é uma classe abstracta e onde é possível especificar o nome do bilhete, o seu preço sem descontos, o preço com descontos, a validade do bilhete e a descrição do mesmo.

Por outro lado, verificou-se que há dois tipos de bilhetes completamente diferentes: os passes e os bilhetes que devem ser obliterados. Decidiu-se, por isso, criar duas novas classes, uma para cada tipo de bilhete. Obtiveram-se assim as classes **Bilhetes_urbanos_passe** e **Bilhetes_urbanos_outros**, que herdaram os atributos da classe **Bilhetes_urbanos**. Na classe **Bilhetes_urbanos_passe** é especificada e descrita a área onde é possível utilizar determinado tipo de passe. Na classe **Bilhetes_urbanos_outros** é especificado o número de bilhetes que são vendidos para determinado tipo de bilhete e o número de obliterações que permitem efectuar. Por exemplo, alguns operadores vendem conjuntos de 10 bilhetes, onde cada um pode ser obliterado 2 vezes e com os quais é possível fazer 20 viagens, desde que cada viagem

necessite apenas de uma obliteração. O esquema de heranças para estas classes é o seguinte:

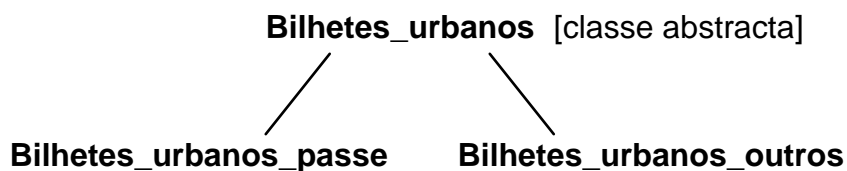


Fig. 2.5. Esquema de heranças para as classes **Bilhetes_urbanos_passe** e **Bilhetes_urbanos_outros**

Nas classes **Urbano** e **Tipo_categoria_preço** existe um atributo que permite especificar as características dos transportes urbanos e as categorias de viagem dos transportes inter-urbanos, respectivamente. Para tal, definiram-se as classes **Características_transporte** e **Características_transporte_inter_urbano**. A classe **Características_transporte** permite especificar o nome do transporte que tem essas características e as idades das crianças que o podem utilizar sem necessidade de bilhete ou com apenas meio bilhete. A classe **Característica_transporte_inter_urbano** para além de herdar todos os atributos da classe **Características_transporte**, possui mais um, onde estão definidas as outras características do transporte, que variam com as categorias de viagem nele existentes. Estas características estão definidas na classe **Outras_características**, onde se pode especificar se para determinada categoria de viagem é necessário fazer a marcação do lugar, se existe um bar, restaurante, etc.. Esta classe não é utilizada para os transportes urbanos, porque estes não possuem qualquer destas características. Enquanto que a classe **Características_transporte_inter_urbano** é utilizada apenas para os transportes inter-urbanos, a classe **Características_transporte** é utilizada para os transportes urbanos e em duas classes referentes aos transportes inter-urbanos: **Operador_inter_urbano** e **Local_aquisição_bilhetes_inter_urbanos**.

O esquema de herança para a classe **Características_transporte_inter_urbano** é o seguinte:

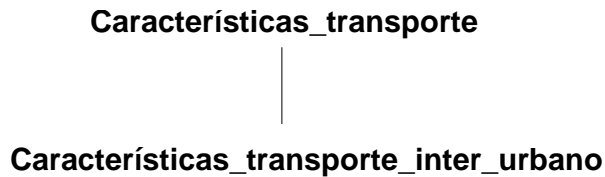


Fig. 2.6. Esquema de herança para a classe **Características_transporte_inter_urbano**

Como já foi referido anteriormente, as características dos transportes urbanos estão associadas à classe **Urbano**, uma vez que este tipo de transporte apenas possui uma categoria de viagem; para os transportes inter-urbanos, as características estão referidas na classe **Tipo_categoria_preço**, porque estas variam com as categorias de viagem existentes nestes transportes.

Na classe **Tipo_categoria_preço**, estão especificados o preço da viagem com descontos e sem descontos para determinada categoria de viagem. Quando o transporte inter-urbano não efectua paragens intermédias entre a paragem origem e a paragem destino, é necessário apenas especificar um preço (o valor do preço do bilhete para ir da paragem origem à paragem destino). Quando há paragens intermédias, utiliza-se uma tabela para especificar o preço do bilhete para a viagem entre quaisquer duas paragens. Quando existem descontos para determinada categoria de viagem, estes são também especificados na classe.

Os descontos que se podem fazer sobre o preço do bilhete num determinado transporte, dependem do operador que gere o transporte. Para tal, definiu-se a classe **Tipo_desconto** que pode ser utilizada pelos transportes urbanos e inter-urbanos. Nesta classe são especificados o nome do desconto, a percentagem do desconto e a condição necessária para o obter. Para especificar esta condição, utiliza-se a classe **Condição_desconto**, onde para além de esta ser descrita, indica-se o limite de idade do utilizador, quando é esta que define a condição.

Todas as classes atrás referidas, com excepção das classes **Operador**, **Operador_táxi** e **Morada** são utilizadas para definir apenas os transportes urbanos e inter-urbanos. Para definir o tipo de transporte táxi,

criaram-se as classes **Táxi**, **Preços_táxi** e **Preços**. O táxi é um meio de transporte especial, que não se enquadra nem nos transportes urbanos nem nos inter-urbanos, embora seja mais frequentemente utilizado como meio de transporte urbano.

A classe **Táxi** é utilizada para descrever informações referentes aos preços praticados por determinado operador e as moradas das várias praças de táxis (sítios próprios para obter um táxi) utilizadas pelos taxistas.

A classe **Preços_táxi** é utilizada para especificar em separado, quais os preços cobrados pelos taxistas durante o dia e durante a noite, visto estes preços serem diferentes.

A classe **Preços** é utilizada para especificar os vários preços a que um utilizador de táxi fica sujeito, como por exemplo, o preço do transporte da bagagem, o preço cobrado por o taxista ir buscar o utilizador a determinado sítio, o preço por quilómetro e o preço mínimo de utilização.

2.5 Definição das estruturas de informação

Classe Transporte

Nesta classe estão representadas as características comuns aos transportes urbanos e inter-urbanos. Esta classe é herdada pelas classes **Urbano** e **Inter_urbano**.

```
{Transporte
  {Superclasse Ø}
  {Classe abstracta}
  {Validade: intervalo data}
  {Horas_partida: lista de horas}
  {Tempo_percurso: lista de horas}
  {Dias_efectua_percurso: dias da semana}}
```

Os atributos desta classe são:

Validade - Indicação do período durante o qual se faz a ligação. Geralmente no Verão existe um maior número de transportes públicos.

Horas_partida - Indicação da hora (hora e minutos) ou das várias horas de partida do transporte do local de partida.

Tempo_percurso - Tempo em horas e minutos que o transporte demora a percorrer o trajecto entre qualquer paragem e a paragem seguinte. O primeiro tempo que aparece neste atributo, é o tempo gasto entre o local de partida e a paragem seguinte, sendo esta a primeira paragem que surge no atributo **Paragens_intermédias** definido quer na classe **Urbano**, quer na classe **Inter_urbano**. O segundo tempo corresponde ao tempo gasto entre a primeira paragem intermédia (paragem que segue ao local de partida e que é também a primeira paragem que surge no atributo **Paragens_intermédias**) e a paragem seguinte e assim sucessivamente. Para cada um dos tempos atrás descritos pode surgir mais um período, que corresponde ao tempo de espera para partir de uma determinada paragem. Estes tempos estão separados dos outros pelo sinal de adição "+". Geralmente esta situação acontece nos transportes inter-urbanos, uma vez que os transportes urbanos ficam pouco tempo à espera nas paragens, e portanto considera-se a hora de chegada e a hora de partida numa determinada paragem como sendo a mesma.

Dias_efectua_percurso - Dias da semana em que o transporte efectua o percurso.

Classe Operador

Esta classe é utilizada para especificar o nome da entidade (operador) que gere determinado transporte. O atributo desta classe é herdado pelas classes **Operador_urbano**, **Operador_inter_urbano** e **Operador_táxi**.

```
{Operador
  {Superclasse Ø}
  {Classe abstracta}
  {Nome_operador: palavras}}
```

O único atributo desta classe é:

Nome_operador - Nome da empresa que gere o meio de transporte.

Classe Operador_urbano

Esta classe é utilizada para definir os operadores dos transportes urbanos.

```
{Operador_urbano
  {Superclasse Operador}
  {Tipos_bilhetes_utilizados: lista de instâncias_id
    (Bilhetes_urbanos_outros ou Bilhetes_urbanos_passe)}
  {Local_aquisição_bilhetes: lista de instâncias_id
    (Local_aquisição_bilhetes_urbanos ou
    Paragem_e_local_aquisição_bilhetes_urbanos)}}}
```

Os atributos desta classe são:

Tipos_bilhetes_utilizados - Especificação dos bilhetes utilizados pelo operador.

Local_aquisição_bilhetes - Este atributo permite especificar as características dos locais onde é possível comprar os bilhetes, sendo estes, locais de paragem ou não. Algumas destas características são a morada, os números de telefone, de fax, o horário de funcionamento, os tipos de bilhetes vendidos, etc..

Classe Operador_inter_urbano

Esta classe é utilizada para definir os operadores dos transportes inter-urbanos.

```

{Operador_inter_urbano
  {Superclasse Operador}
  {Tipos_transportes_disponíveis: lista de instâncias_id
    (Características_transporte)}
  {Local_aquisição_bilhetes: lista de instâncias_id
    (Local_aquisição_bilhetes_inter_urbanos ou
    Paragem_e_local_aquisição_bilhetes_inter_urbanos)}}

```

Os atributos desta classe são:

Tipos_transportes_disponíveis - Permite especificar os transportes disponibilizados pelo operador. Como para este atributo apenas interessa indicar os tipos de transportes e não propriamente as suas características, a lista que surge neste atributo é uma lista de instâncias da classe **Características_transporte** e não da classe **Características_transporte_inter_urbano** como à primeira vista se poderia pensar.

Local_aquisição_bilhetes - Este atributo permite especificar as características dos locais onde é possível comprar os bilhetes, sendo estes, locais de paragem ou não. Algumas destas características são a morada, os números de telefone, de fax, o horário de funcionamento, os tipos de transportes que o operador possui, etc..

Classe Operador_táxi

Esta classe é utilizada para descrever as moradas da central de táxis, os seus números de telefone e números de fax.

```

{Operador_táxi
  {Superclasse Operador}
  {Moradas_central: lista de instâncias_id (Morada)}
  {Telefones_central: lista de números telefone_fax}
  {Faxes_central: lista de números telefone_fax}}

```

Os seus atributos são:

Moradas_central - Indicação das moradas da central de táxis pertencente ao operador.

Telefones_central - Especificação dos números de telefone da central.

Faxes_central - Indicação dos números de fax da central.

Classe Morada

Na classe **Morada** estão definidas as características de localização de determinada paragem e/ou do local onde é possível adquirir os bilhetes.

{Morada

{Superclasse Ø}

{Rua_avenida_nº: palavras}

{Código_Postal: palavras}

{Concelho: palavras}

{Distrito: palavras}

{Coordenada_M: coordenada}

{Coordenada_P: coordenada}}

Os atributos desta classe são:

Rua_avenida_nº - Rua, avenida (ou qualquer outra indicação, praça, largo, etc.), e número da porta.

Código_Postal - Código postal do local.

Concelho - Concelho do local.

Distrito - Distrito do local.

Coordenada_M - Coordenada M do local.

Coordenada_P - Coordenada P do local.

Classe Paragem

Esta classe é utilizada para descrever as características de um local de paragem.

```
{Paragem
  {Superclasse Ø}
  {Nome: palavras}
  {Morada: instância_id (Morada)}
  {Transportes_que_param_nessa_paragem: lista de
instâncias_id          (Inter_urbano ou Urbano)}}}
```

Os atributos desta classe são:

Nome - Nome da paragem.

Morada - Especificação da morada da paragem.

Transportes_que_param_nessa_paragem - Indicação dos transportes que param na paragem.

Classe Local_aquisição_bilhetes

Esta classe é utilizada para representar as informações sobre um local onde se podem adquirir os bilhetes, mas que não é local de paragem.

```
{Local_aquisição_bilhetes
  {Superclasse Ø}
  {Nome: palavras}
  {Morada: instância_id (Morada)}
  {Telefones: lista de números telefone_fax}
  {Faxes: lista de números telefone_fax}
  {Horário: horário}}
```

Os atributos desta classe são:

Nome - Nome do local onde se podem comprar os bilhetes.

Morada - Morada do local.

Telefones - Números de telefone do local.

Faxes - Números de fax do local.

Horário - Horário de abertura e de encerramento do local.

Classe Local_aquisição_bilhetes_urbanos

Esta classe é utilizada para especificar os tipos de bilhetes utilizados nos transportes urbanos, que são vendidos nos locais que não são sítios de paragem. Por exemplo, os passes não são vendidos em todos os sítios onde se adquirem os outros tipos de bilhetes.

```
{Local_aquisição_bilhetes_urbanos
  {Superclasse Local_aquisição_bilhetes}
  {Tipos_bilhetes_vendidos: lista de instâncias_id
    (Bilhetes_urbanos_passe ou Bilhetes_urbanos_outros)}}}
```

O atributo desta classe é:

Tipos_bilhetes_vendidos - Especificação dos tipos de bilhetes vendidos naquele local.

Classe Local_aquisição_bilhetes_inter_urbanos

Esta classe permite especificar os tipos de bilhetes que se podem comprar para os transportes inter-urbanos, nos vários locais onde se podem adquirir bilhetes, mas que não são sítios de paragem.

```
{Local_aquisição_bilhetes_inter_urbanos
  {Superclasse Local_aquisição_bilhetes}
  {Tipos_bilhetes_vendidos_para_tipos_transportes:
    lista de instâncias_id (Características_transporte)}}}
```

O único atributo desta classe é:

Tipos_bilhetes_vendidos_para_tipos_transportes - Permite especificar os transportes pertencentes a determinado operador, cujos bilhetes

são vendidos no local em questão. Como para este atributo apenas interessa indicar os tipos de transportes para os quais se vendem bilhetes e não propriamente as suas características, a lista que surge neste atributo é uma lista de instâncias da classe **Características transporte** e não da classe **Características transporte inter urbano**.

Classe Paragem_e_local_aquisição_bilhetes_urbanos

Esta classe é utilizada para descrever as características dos locais onde se podem comprar os bilhetes e que são também locais de paragem dos transportes urbanos.

```
{Paragem_e_local_aquisição_bilhetes_urbanos  
 {Superclasse Paragem,Local_aquisição_bilhetes_urbanos}}
```

A classe **Paragem** e a classe **Local_aquisição_bilhetes_urbanos** têm dois atributos comuns, **Nome** e **Morada**. É claro que na classe **Paragem_e_local_aquisição_bilhetes_urbanos** não haverá duplicação da informação e só irá surgir um atributo **Nome** e outro atributo **Morada**. Isto é possível porque os atributos **Nome** das duas classes têm o mesmo tipo de valor, ou seja, são ambos do tipo **Palavras**. O mesmo acontece para o atributo **Morada**.

Classe Paragem_e_local_aquisição_bilhetes_inter_urbanos

Esta classe é utilizada para descrever as características dos locais onde se podem comprar os bilhetes e que são também locais de paragem dos transportes inter-urbanos.

```
{Paragem_e_local_aquisição_bilhetes_inter_urbanos  
 {Superclasse  
 Paragem,Local_aquisição_bilhetes_inter_urbanos}}
```

Nesta classe surge a mesma situação explicada na classe anterior, sendo a solução adoptada a mesma.

Classe **Tipo_desconto**

Esta classe é utilizada para representar as informações sobre os descontos que determinado operador em determinado tipo de transporte pratica.

```
{Tipo_desconto
  {Superclasse Ø}
  {Nome_desconto: palavras}
  {Percentagem_desconto: número}
  {Condição_para_obter_desconto: instância_id
  (Condição_desconto)}}}
```

Os atributos são:

Nome_desconto - Nome do cartão que o cliente apresenta para adquirir o desconto, por exemplo, cartão jovem, cartão comboio azul, etc..

Percentagem_desconto - Percentagem de desconto que o operador faz em determinado tipo de transporte.

Condição_para_obter_desconto: Condição que o cliente deve reunir para obter o desconto.

Classe **Condição_desconto**

Esta classe é utilizada para representar as informações sobre a condição necessária para obter um tipo de desconto.

```
{Condição_desconto
  {Superclasse Ø}
  {Condição: palavras}
  {Idade_pertencer_condição: intervalo idade}}}
```

Os atributos são:

Condição - Especificação da condição para obter o desconto.

Idade_pertencer_condição - Indicação das idades mínimas e máximas que permitem satisfazer a condição.

Classe Características_transporte

Nesta classe são especificadas as características que descrevem determinado tipo de transporte.

<p>{Características_transporte {Superclasse ∅} {Nome_transporte: palavras} {Idade_sem_bilhete: intervalo idade} {Idade_meio_bilhete: intervalo idade}</p>
--

Os seus atributos são:

Nome_transporte - Nome do transporte, por exemplo, Intercidades, Alta Qualidade, etc..

Idade_sem_bilhete - Idade que a criança deve ter para não pagar bilhete.

Idade_meio_bilhete - Idade que a criança deve ter para pagar apenas meio bilhete.

Classe Características_transporte_inter_urbano

Nesta classe estão definidas as características que descrevem as várias categorias de viagem dos transportes inter-urbanos.

```
{Características_transporte_inter_urbano
  {Superclasse Características_transporte}
  {Outras_características: instância_id (Outras_características)}}
```

O seu atributo é:

Outras_características - Indicação de mais algumas características, como por exemplo, necessidade de marcação de lugar, existência de bar, etc..

Classe Outras_características

Esta classe permite especificar outras características dos transportes inter-urbanos para além das especificadas em **Características_transporte**.

```
{Outras_características
  {Superclasse ∅}
  {Marcação_lugar: booleano}
  {Bar: booleano}
  {Restaurante: booleano}
  {Lugares_cama: booleano}
  {WC: booleano}
  {Ar_condicionado: booleano}
  {Vídeo: booleano}
  {TV: booleano}
  {Telemóvel: booleano}}}
```

Os seus atributos são:

Marcação_lugar - Se o transporte precisar de marcação de lugar, o valor lógico deste atributo deverá ser V.

Bar - Verdadeiro (V) se o transporte possui bar.

Restaurante - Verdadeiro (V) se o transporte possui restaurante.

Lugares_cama - Verdadeiro (V) se o transporte possui lugares cama.

WC - Verdadeiro (V) se o transporte possui casa de banho.

Ar_condicionado - Verdadeiro (V) se o transporte tem ar condicionado.

Vídeo - Verdadeiro (V) se o transporte tem vídeo.

TV - Verdadeiro (V) se o transporte tem televisão.

Telemóvel - Verdadeiro (V) se o transporte tem telemóvel.

Classe **Tipo_categoria_preço**

A classe **Tipo_categoria_preço** permite especificar a tabela de preços, os tipos de descontos e as características do transporte, que estão associados a uma determinada categoria de viagem. Esta classe é apenas utilizada para os transportes inter-urbanos.

```
{Tipo_categoria_preço
  {Superclasse Ø}
  {Nome_categoria: palavras}
  {Preço_base: Tabela}
  {Características_transporte: instância_id
    (Características_transporte_inter_urbano)}
  {Descontos: lista de instâncias_id (Tipo_desconto)}}}
```

Os atributos desta classe são:

Nome_categoria - Permite especificar o nome da categoria de viagem, por exemplo, 1ª classe.

Preço_base - Indicação de uma tabela que contém os preços dos bilhetes, sem descontos, para ir de uma paragem a outra.

Características_transporte - Especificação das características do transporte, relativamente à categoria de viagem em questão.

Descontos - Indicação dos descontos que o operador faz neste tipo de categoria de viagem.

A tabela que poderá surgir no atributo **Preço_base** deverá obedecer à seguinte representação:

	P2	P3	P4	P5	P6	P7
P1	P12	P13	P14	P15	P16	P17
P2	-	P23	P24	P25	P26	P27
P3	-	-	P34	P35	P36	P37
P4	-	-	-	P45	P46	P47
P5	-	-	-	-	P56	P57
P6	-	-	-	-	-	P67

Na primeira linha da tabela estão representadas todas as paragens com excepção da primeira e na primeira coluna estão representadas todas as paragens com excepção da última. As outras células correspondem aos preços dos bilhetes para ir da paragem origem indicada na linha à paragem destino indicada na coluna. As células que não têm preço, correspondem a ligações que não se efectuam neste transporte inter-urbano específico.

Classe Bilhetes_urbanos

A classe **Bilhetes_urbanos** foi criada para descrever as características dos tipos de bilhetes utilizados nos transportes urbanos pertencentes a determinado operador.

```
{Bilhetes_urbanos
  {Superclasse Ø}
  {Classe abstracta}
  {Nome_bilhete: palavras}
  {Preço_base: preço}
  {Preço_com_descontos: lista de instâncias_id (Tipo_desconto)
ou      preço}
  {Validade_bilhete: palavras}
  {Descrição: palavras}}
```

Os seus atributos são:

Nome_bilhete - Nome do bilhete.

Preço_base - Preço, sem descontos, de determinado tipo de bilhete.

Preço_com_descontos - No caso de este tipo de bilhete não ter desconto, o valor deste atributo será o valor do atributo **Preço_base**; no caso de existirem descontos, o valor deste atributo corresponderá a um conjunto de identificadores de instâncias da classe **Tipo_desconto**, onde estão explicitadas as percentagens de descontos.

Validade_bilhete - Prazo em que é válido determinado tipo de bilhete.

Descrição - Descrição do bilhete.

Classe Bilhetes_urbanos_passe

Esta classe é utilizada para descrever as características do Passe.

```
{Bilhetes_urbanos_passe
  {Superclasse Bilhetes_urbanos}
  {Área_validade: palavras}
  {Descrição_área: palavras}}
```

Os seus atributos são:

Área_validade - Zona do percurso em que o tipo de Passe em questão pode ser utilizado.

Descrição_área - Descrição da área onde o tipo de Passe em questão pode ser utilizado.

Houve necessidade de criar esta classe, dado que o Passe tem características próprias que os outros bilhetes não têm.

Classe Bilhetes_urbanos_outros

Esta classe é utilizada para descrever as características de todos os outros bilhetes, que não o Passe, disponibilizados por determinado operador.

<pre>{Bilhetes_urbanos_outros {Superclasse Bilhetes_urbanos} {Número_bilhetes: número} {Número_oblitações: número}}</pre>

Os atributos são:

Número_bilhetes - Número de bilhetes vendidos, para o tipo de bilhete especificado.

Número_oblitações - Número de oblitações que é possível realizar com o número de bilhetes especificado no atributo anterior.

Classe Inter_urbano

A classe **Inter_urbano** é utilizada para especificar as características dos transportes inter-urbanos.

```

{Inter_urbano
  {Superclasse Transporte}
  {Número_transporte: número}
  {Nome_transporte: palavras}
  {Local_partida: instância_id
    (Paragem_e_local_aquisição_bilhetes_inter_urbanos) ou
    instância_id (Paragem)}
  {Local_chegada: instância_id
    (Paragem_e_local_aquisição_bilhetes_inter_urbanos) ou
    instância_id (Paragem)}
  {Paragens_intermédias: lista de instâncias_id
    (Paragem_e_local_aquisição_bilhetes_inter_urbanos ou
Paragem)}
  {Operador: instância_id (Operador_inter_urbano)}
  {Tipo_categoria_preço: lista de instâncias_id
(Tipo_categoria_preço)}}

```

Os seus atributos são:

Número_transporte - Número do veículo.

Nome_transporte - Nome do tipo de transporte, exemplo: Expresso, Avião, etc..

Local_partida - Especificação do local de partida. A instância que surge neste atributo pode ser instância da classe **Paragem** (se o local for apenas sítio de paragem) ou da classe **Paragem_e_local_aquisição_bilhetes_inter_urbanos** (se o local for sítio de paragem e de compra de bilhetes).

Local_chegada - Especificação do local de chegada. A instância referida neste atributo pode ser instância da classe **Paragem** (se o local for apenas sítio de paragem) ou da classe **Paragem_e_local_aquisição_bilhetes_inter_urbanos** (se o local for sítio de paragem e de compra de bilhetes).

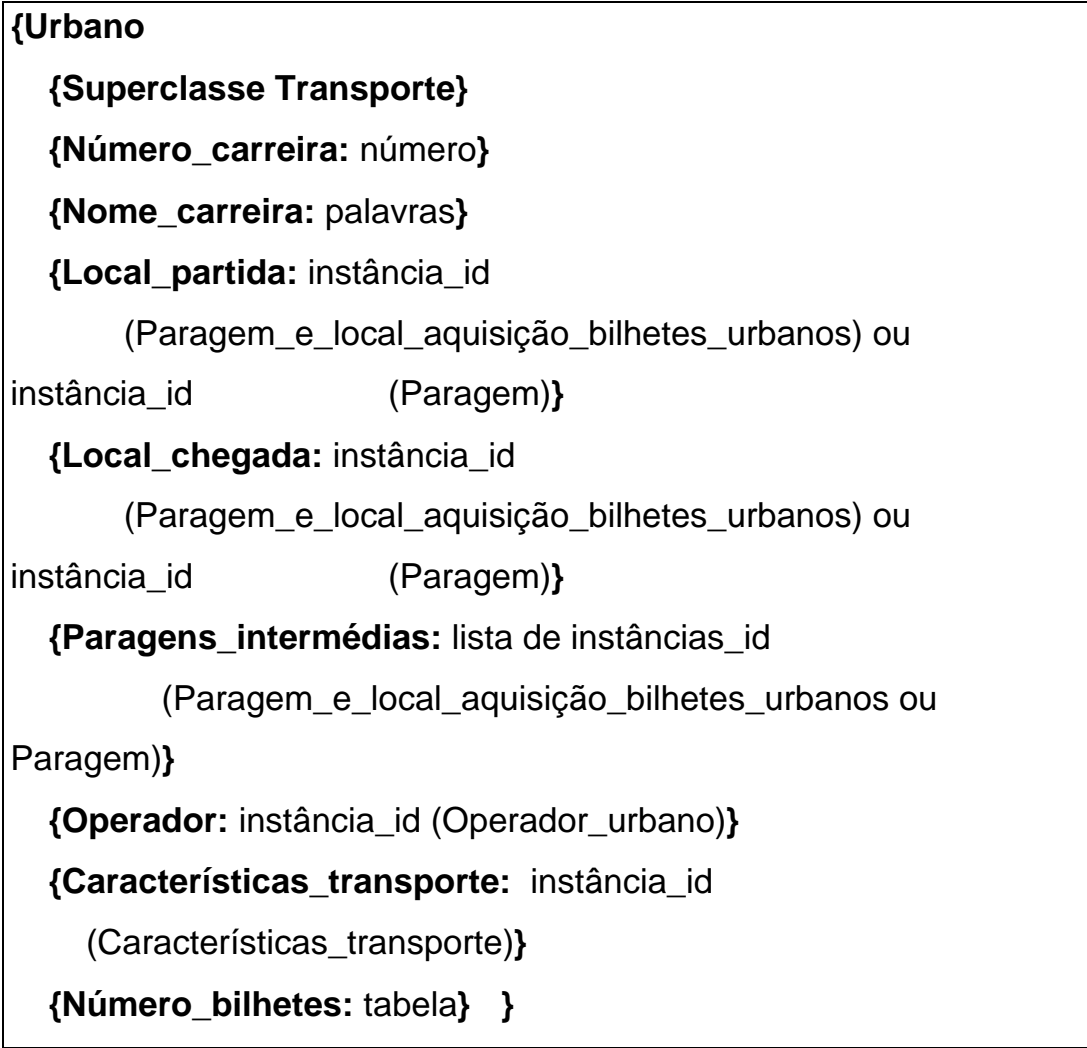
Paragens_intermédias - Indicação das paragens intermédias que surgem entre o local de partida e o local de chegada. Neste atributo não são especificadas as paragens origem e destino, uma vez que estas têm um atributo próprio para cada uma delas. Os identificadores das instâncias que surgem neste atributo, podem ser instâncias da classe **Paragem** (se o local for apenas sítio de paragem), da classe **Paragem_e_local_aquisição_bilhetes_inter_urbanos** (se o local for sítio de paragem e de compra de bilhetes) ou de ambas as classes.

Operador - Indicação das características do operador ao qual o transporte pertence.

Tipo_categoria_preço - Indicação das categorias de viagens, preços e descontos realizados no transporte em questão.

Classe Urbano

Esta classe é utilizada para especificar as características dos transportes urbanos.



Os seus atributos são:

Número_carreira - Número da carreira de um percurso (não confundir com número da viatura).

Nome_carreira - Nome da carreira de um percurso (geralmente o nome do local de chegada do percurso).

Local_partida - Indicação do local de partida. A instância que surge neste atributo pode ser instância da classe **Paragem** (se o local for apenas sítio de paragem) ou da classe **Paragem_e_local_aquisição_bilhetes_urbanos** (se o local for sítio de paragem e de compra de bilhetes).

Local_chegada - Indicação do local de chegada. O identificador da instância que surge neste atributo pode ser instância da classe **Paragem** (se o local for apenas sítio de paragem) ou da classe **Paragem_e_lo-**

cal_aquisição_bilhetes_urbanos (se o local for sítio de paragem e de compra de bilhetes).

Paragens_intermédias - Indicação das paragens intermédias que surgem entre o local de partida e o local de chegada. Neste atributo não são especificadas as paragens origem e destino, uma vez que estas têm um atributo próprio para cada uma delas. Os identificadores das instâncias que surgem neste atributo, podem ser instâncias da classe **Paragem** (se os locais forem apenas sítios de paragem), da classe **Paragem_e_lo-cal_aquisição_bilhetes_urbanos** (se os locais forem sítios de paragem e de compra de bilhetes) ou de ambas as classes.

Operador - Indicação das características do operador que gere o transporte.

Características_transporte - Especificação das características do transporte.

Número_bilhetes - Indicação de uma tabela que contém o número de bilhetes que são necessários para se efectuar os percurso entre quaisquer paragens.

Outra estrutura que se poderia utilizar tanto para representar o Número_bilhetes desta classe como o Preço_base da classe **Tipo_categoria_preço**, seria uma lista de listas, ou seja, uma lista que por sua vez é composta por outras listas.

Considere-se um exemplo aplicado à classe **Urbano**: um transporte urbano faz a ligação entre as seguintes paragens: P1, P2, P3, P4, P5. Para este caso a lista terá a seguinte forma:

(O1 (D2,D3,D4,D5),
O2 (D3,D4,D5),
O3 (D4,D5),
O4 (D5))

As designações O1,..., O4 indicadas antes de cada lista secundária denominam uma determinada paragem. A designação O1 corresponde à paragem origem, a O2 corresponde à primeira paragem intermédia e assim sucessivamente. A última designação corresponde à penúltima

paragem. Nos casos práticos, por uma questão de simplicidade, em vez de aparecerem O_1, O_2, \dots, O_4 , surgirão apenas números por forma a indicar a sequência das paragens. As designações D_2, D_3, \dots, D_5 dentro das listas secundárias correspondem ao número de bilhetes necessários para ir até uma determinada paragem, partindo da paragem origem indicada à frente da lista secundária em questão. Por exemplo, a notação D_4 que surge na primeira lista secundária, corresponde ao número de bilhetes necessários para ir da paragem O_1 para a paragem O_4 . Por outro lado, quando o número de bilhetes numa lista secundária é igual à lista secundária anterior, com excepção do primeiro número, então, em vez de se repetir a lista, utiliza-se uma lista vazia, que se representa por $()$.

Supondo que, para a situação anterior, os números de bilhetes são os a seguir indicados:

Da paragem O_1 :

- à paragem O_2 é necessário 1 bilhete
- à paragem O_3 é necessário 1 bilhete
- à paragem O_4 são necessários 2 bilhetes
- à paragem O_5 são necessários 2 bilhetes

Da paragem O_2 :

- à paragem O_3 é necessário 1 bilhete
- à paragem O_4 é necessário 1 bilhete
- à paragem O_5 é necessário 1 bilhete

Da paragem O_3 :

- à paragem O_4 é necessário 1 bilhete
- à paragem O_5 é necessário 1 bilhete

Da paragem O_4 :

- à paragem O_5 é necessário 1 bilhete

A lista de listas seria a seguinte:

- $(1 (1,1,2,2))$,
- $2 (1,1,1)$,

3 (),

4 ()

Classe Táxi

Esta classe é utilizada para descrever informações referentes às moradas das várias praças de táxis e os preços praticados por esse operador.

{Táxi

{Superclasse Ø}

{Moradas_praças_de_táxis: lista de instâncias_id (Morada)}

{Preços: instância_id (Preços_táxi)}

{Operador: instância_id (Operador_táxi)}}

Os atributos desta classe são:

Moradas_praças_de_táxis - Especificação das moradas das várias praças de táxis.

Preços - Indicação dos preços cobrados pelos taxistas abrangidos por determinado operador.

Operador - Indicação das características do operador.

Classe Preços_táxi

Esta classe é utilizada para especificar, em separado, quais os preços utilizados pelos taxistas durante o dia e durante a noite.

{Preços_táxi

{Superclasse Ø}

{Preços_dia: instância_id (Preços)}

{Preços_noite: instância_id (Preços)}}

Os atributos desta classe são:

Preços_dia - Indicação dos preços que estão em vigor durante o período que abrange a manhã e a tarde. Este período está indicado no atributo **Período_válido** da classe **Preços**.

Preços_noite - Indicação dos preços que estão em vigor durante o período que abrange a noite e a madrugada. Este período está indicado no atributo **Período_válido** da classe **Preços**.

Classe Preços

Esta classe é utilizada para especificar os vários preços a que um utilizador de táxi fica sujeito, como por exemplo, o preço do transporte da bagagem, o preço cobrado por o taxista ir buscar o utilizador a determinado sítio, o preço por quilómetro, etc..

```
{Preços
  {Superclasse Ø}
  {Período_válido: horário}
  {Preço_bandeirada: preço}
  {Preço_bagagem: preço}
  {Preço_chamado_pelo_telefone: preço}
  {Preço_por_kilómetro: preço}}
```

Os atributos desta classe são:

Período_válido - Indicação do horário durante o qual são utilizados os preços especificados nesta classe.

Preço_bandeirada - Preço mínimo de utilização.

Preço_bagagem - Preço cobrado se o utilizador tiver bagagem.

Preço_chamado_pelo_telefone - Preço cobrado se o utilizador chamou o táxi pelo telefone para a central. Neste caso podem surgir duas situações, ou o taxista põe o contador a funcionar quando recebe a mensagem para ir buscar o cliente, e neste caso este atributo não é utilizado para se obter o preço final, ou cobra um determinado preço

especificado neste atributo, e só põe o contador a funcionar quando o utilizador entrar no táxi.

Preço_por_kilómetro - Preço cobrado por cada quilómetro percorrido.

3. DEFINIÇÃO E REALIZAÇÃO DA BASE DE DADOS

3

3.1 Introdução

O objectivo deste capítulo é indicar a forma como a partir da representação do conhecimento sobre os transportes, se obteve uma base de dados desenvolvida num Sistema de Gestão de Bases de Dados Relacional (SGBDR).

A metodologia seguida começa por analisar o sistema num nível abstracto, que permite descrever os objectos considerados importantes no sistema, as interacções entre eles e as suas propriedades. A abordagem utilizada foi a abordagem Entidade-Associação, vulgarmente designada por E-R (Entity-Relationship em inglês).

O passo seguinte foi converter os diagramas obtidos numa base de dados que fosse suportada por um SGBDR, ou seja, tornou-se necessário converter os diagramas E-R em tipos de dados suportados pelos SGBDR, isto é, em relações.

Decidiu-se pelos sistemas relacionais em detrimento dos sistemas que suportam bases de dados desenhadas segundo outras abordagens, uma vez que são os sistemas mais utilizados actualmente e os mais simples e práticos de manipular.

A ferramenta utilizada como SGBDR foi o Microsoft Access versão 2.0, porque, na minha opinião, é a ferramenta mais completa e utilizada neste momento.

A organização deste capítulo é a seguinte:

Na secção 3.2 são explicados a abordagem Entidade-Associação, os conjuntos de entidades, de associações e os diagramas E-R obtidos para

os transportes inter-urbanos, urbanos e táxis. Na secção 3.3 é feita a conversão dos diagramas E-R obtidos em relações. Na secção 3.4 é feita a comparação entre as classes obtidas na representação do conhecimento sobre os transportes e as relações obtidas neste capítulo. Por último, na secção 3.5 é apresentada a informação que está guardada na base de dados e a forma como se introduz essa informação utilizando o Ms-Access 2.0.

3.2 Abordagem Entidade-Associação

A abordagem Entidade-Associação é a metodologia mais utilizada para a definição de bases de dados relacionais e é inspirada na chamada Teoria da Normalização. Esta abordagem foi proposta por P. Chen em 1976 ([5] e [7]).

Os principais conceitos desta abordagem são os de entidade, associação e atributo.

Entidades são objectos que são considerados importantes num sistema. A abordagem E-R geralmente começa por identificar todas as entidades importantes e agrupa-as em conjuntos com as mesmas propriedades.

A simbologia gráfica utilizada para representar um conjunto de entidades é um rectângulo. Considere-se por exemplo, o conjunto de entidades referentes aos transportes urbanos:



Fig. 3.1. Conjunto de entidades: Urbano

É importante ter presente a distinção entre entidades e conjunto de entidades. Uma entidade é uma instância (para o caso do conjunto de entidades **Urbano**, as entidades são os vários transportes urbanos

existentes no universo), enquanto que um conjunto de entidades é um conjunto de objectos com as mesmas propriedades (para o exemplo considerado, é o conjunto de todos os transportes urbanos existentes no universo). Como é vulgar utilizar a designação entidade quer se trate de uma entidade isolada ou de um conjunto de entidades, por forma a simplificar o texto, será esta também a opção utilizada.

O conjunto de entidades representam os objectos do sistema, mas não podem representar a forma como eles interactuam entre si, nem para que servem. Para tal, existe o conceito de associação. As associações permitem, portanto, representar as interacções entre entidades. A colecção de associações é denominada conjunto de associações e representa-se por um losango. Considere-se o exemplo da fig. 3.2., onde está representado o facto de os transportes urbanos (**Urbanos**) serem descritos (Tem_7) por várias características (**Características_transporte_urbano**).

A aridade de uma associação é o número de entidades que nela participam. Designam-se por associações binárias as associações entre duas entidades, por ternárias as associações entre três entidades e por n-árias as associações entre n entidades. Por outro lado, designam-se por argumentos de uma associação as entidades que nela participam.

As associações também se classificam quanto à sua multiplicidade, podendo ser uma associação de muitos-para-muitos, de muitos-para-um ou de um-para-um.

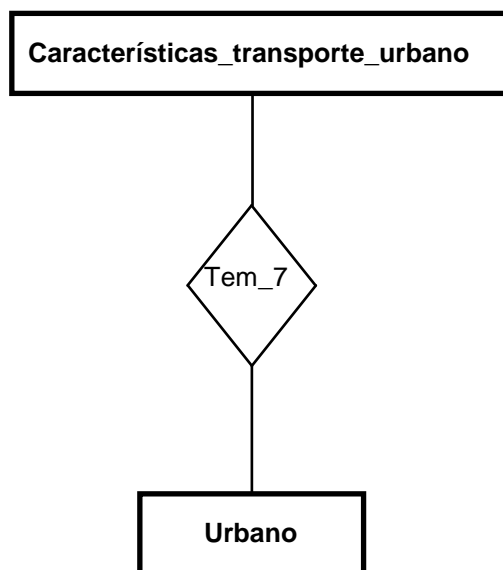


Fig. 3.2. . Conjunto de associações: Tem_7

Esta classificação reflecte a existência ou não de restrições relativamente ao número de entidades de um dado conjunto de entidades, argumento da associação, que podem estar relacionadas com entidades de outros conjuntos também argumentos da mesma associação. Assim e considerando apenas o caso das associações binárias, existe uma associação de:

a) **muitos-para-um**, quando cada elemento de um dos argumentos da associação apenas pode fazer parte de uma instância da associação, mas os elementos do outro argumento podem fazer parte dum qualquer número de instâncias da associação. Considere-se o seguinte facto, relativo aos transportes urbanos:

“... cada transporte urbano tem um determinado conjunto de características, mas este conjunto de características pode caracterizar vários transportes urbanos...”

Graficamente esta associação representa-se da seguinte forma¹:

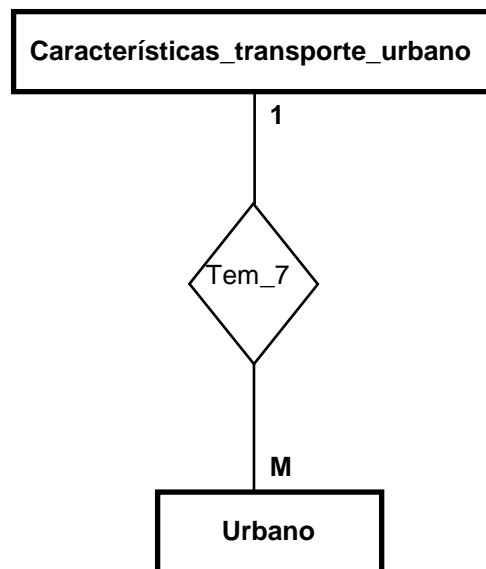


Fig. 3.3. Associação de muitos-para-um: Tem_7

b) **um-para-um**, quando cada elemento de um dado conjunto de entidades só pode estar associado a um elemento de outro conjunto de entidades e reciprocamente.

Representamos o seguinte facto:

“...cada paragem de transporte tem apenas uma morada e a cada morada apenas corresponde uma paragem...”

pele seguinte esquema:

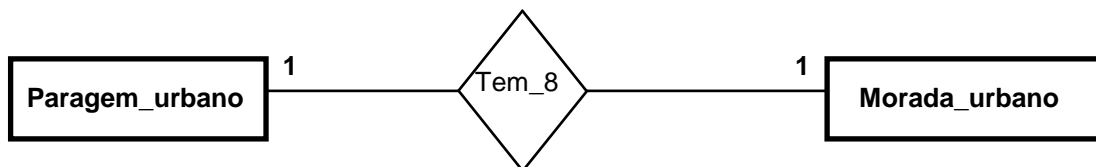


Fig. 3.4. Associação de um-para-um: Tem_8

¹ Decidiu-se colocar a letra que representa a multiplicidade perto do conjunto de entidades e não do conjunto de associações, para manter uma certa uniformidade com os restantes diagramas que irão aparecer no decorrer deste capítulo.

c) **muitos-para-muitos**, quando cada elemento de um dos argumentos de uma associação pode participar num qualquer número de instâncias das associações e cada elemento do outro argumento pode participar num qualquer número de instâncias da associação.

Considere-se o seguinte caso, relativo aos transportes inter-urbanos:

“...Para cada categoria de viagem podem ser efectuados vários descontos nos preços dos bilhetes e cada desconto pode ser efectuado em bilhetes de várias categorias...”

Graficamente tem-se:

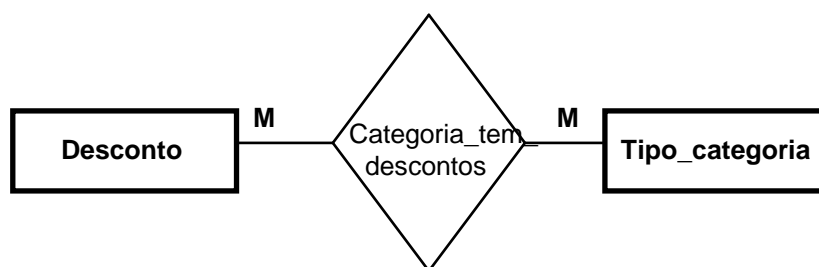


Fig. 3.5. Associação de muitos-para-muitos: Categoria_tem_descontos

A multiplicidade das associações não binárias segue os mesmos princípios estabelecidos para as associações binárias.

Outro conceito necessário nos diagramas E-R é o de atributo. Os atributos descrevem as propriedades de entidades e associações, sendo que, a cada atributo é atribuído um nome que é usado para referir determinada propriedade. Graficamente os atributos representam-se por ovais ligadas por segmentos de recta aos respectivos conjuntos de entidades e associações.

Considere-se o exemplo do conjunto de entidades **Morada_urbano**, na fig. 3.6.. **Morada_urbano_id**, **Rua_avenida_nº**, **Local**, **Código_postal**, **Concelho**, **Distrito**, **Coord_M** e **Coord_P** são os seus atributos.

Os diagramas E-R apenas incluem o conjunto de entidades e não as entidades em si. No entanto um requisito de todos os sistemas é de ser capaz de identificar as entidades sem ser necessário mostrá-las nos diagramas. Para tal define-se uma chave primária que não é mais do que

um conjunto de um ou mais atributos que permite identificar qualquer entidade no conjunto de entidades.

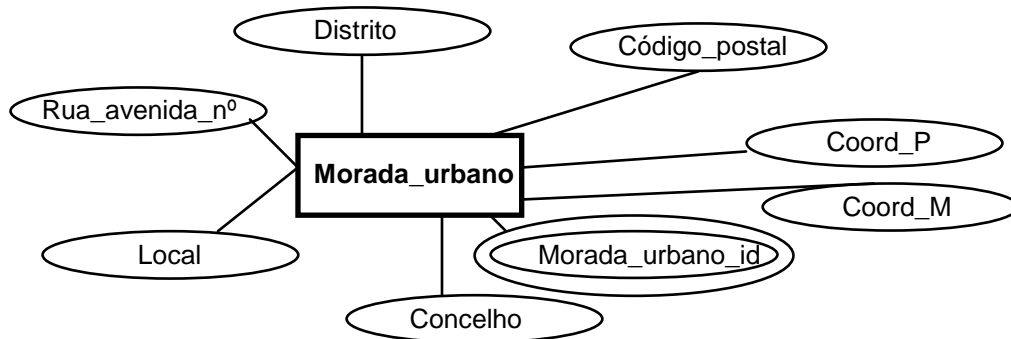


Fig. 3.6. Conjunto de entidades com atributos: Morada_urbano

Dado que as associações podem ser identificadas pelas entidades nelas envolvidas, o conjunto de associações não necessita de ter uma chave primária própria. A sua chave primária é o conjunto das chaves primárias das entidades nele envolvidas. As chaves primárias das associações não são representadas nos diagramas E-R.

Graficamente representa-se uma chave primária por uma oval dupla. Na figura anterior, a chave primária é o atributo **Morada_urbano_id**. Como nenhum dos atributos (ou conjunto de atributos) que caracterizam o conjunto de entidades **Morada_urbano** servem para caracterizar as entidades univocamente, teve que se adicionar um atributo que funcionasse como chave primária.

3.2.1 Transportes Inter-urbanos

Nas secções seguintes fala-se em percursos e ligações entre paragens. Convém salientar que se considera percurso, a trajetória completa que determinado transporte efectua a partir da paragem de partida até à última paragem. Ligações entre paragens referem-se às várias combinações de ligações que é possível fazer em determinado percurso (um percurso é também uma ligação).

3.2.1.1 Conjuntos de entidades

Os conjuntos de entidades definidos para este tipo de transporte são:

Inter_urbano

Operador_inter_urbano

Local_aquisição_bilhetes_inter_urbanos

Tipo_categoria

Desconto

Paragem

Morada

Parte_de

Chega_a

Outras_características

Localidade

Inter_urbano

No conjunto de entidades **Inter_urbano** é guardada informação referente ao número e nome do transporte, o local de partida e de chegada, o período de funcionamento de determinada ligação e os dias da semana em que se efectua a ligação. Uma entidade de **Inter_urbano** designa uma ligação e não o transporte em si. Se um transporte efectuar uma ligação partindo de um local X para chegar a um local Y e efectuar a ligação inversa, partindo de Y para chegar a X, estas duas ligações são entidades diferentes do conjunto de entidades **Inter_urbano**, embora efectuadas pelo mesmo transporte.

Operador_inter_urbano

Neste conjunto de entidades é guardado o nome da empresa que gere determinado transporte.

Local_aquisição_bilhetes_inter_urbanos

Permite especificar as características dos locais onde se podem comprar os bilhetes. Para além do nome do local, são especificados os números de telefone, de fax e o horário de atendimento ao público.

Tipo_categoria

Permite caracterizar determinada categoria de viagem. Esta entidade tem como atributos, o nome da categoria e as idades mínimas e máximas para viajar sem bilhete, para pagar apenas meio bilhete ou uma percentagem deste.

Desconto

Esta entidade permite especificar as características dos vários descontos que podem incidir nos preços dos bilhetes.

Para além do nome do desconto, são especificadas as suas percentagens, as condições para os obter e o período em que este é válido (geralmente em meses).

Paragem

Neste conjunto de entidades são especificados os nomes das paragens.

Morada

Permite especificar as moradas das paragens e dos locais onde se podem comprar os bilhetes. Neste conjunto de entidades são especificados o nome da rua-avenida-praceta, o número do local, o nome deste, o código postal, o concelho e distrito a que pertence e as suas coordenadas topográficas.

Parte_de

Nesta entidade são especificadas as horas de partida dos vários transportes de determinada paragem.

Chega_a

Esta entidade é semelhante à entidade anterior só que em vez das horas de partida são especificadas as horas de chegada dos vários transportes a determinada paragem.

Outras_características

Este conjunto de entidades permite especificar a existência de bar, restaurante, lugares cama, wc, ar condicionado, vídeo, tv e telemóvel e a necessidade ou não de marcação de lugar.

Localidade

Permite identificar as localidades onde se situam as moradas. Neste conjunto de entidades são especificados o nome e o tipo da localidade (cidade, vila, etc.).

3.2.1.2 Conjuntos de associações

Um-para-um

Para os transportes inter_urbanos apenas existem três conjuntos de associações deste tipo.

Tem_6

Representa o facto de uma paragem só ter uma morada e vice-versa. Uma morada apenas pode localizar uma determinada paragem.

Tem_5

A explicação é a mesma dada para o conjunto de associações anterior, ou seja, cada local, onde se podem adquirir os bilhetes, é caracterizado por apenas uma morada e vice-versa.

Tem_1

Uma categoria de viagem é representada por um conjunto de características, ou seja, por apenas uma entidade do conjunto de entidades **Outras_características**.

Muitos-para-um

Para os transportes inter-urbanos existem vários conjuntos de associações deste tipo.

Pertence_1

Representa o facto de um operador (empresa) ter vários transportes (veículos), mas cada veículo pertencer apenas a um operador.

Tem_4

Um operador (empresa) possui vários locais onde se podem adquirir os bilhetes (esses locais podem ser ou não sítios de paragem do transporte), no entanto, esses locais só pertencem a um operador.

Parte_1

De uma mesma paragem partem vários transportes a horas diferentes.

Chega

A uma mesma paragem chegam vários transportes a horas diferentes.

Situa-se1

Uma morada identifica um sítio de uma determinada localidade, no entanto, numa localidade existem várias moradas.

Muitos-para-muitos

Existem cinco conjuntos de associações deste tipo para os transportes inter-urbanos.

Categoria_tem_descontos

Este conjunto de associações representa o facto dos preços dos bilhetes para determinada categoria de viagem poderem sofrer vários tipos de descontos, enquanto que um mesmo tipo de desconto pode ser aplicado aos preços dos bilhetes de categorias diferentes.

Ligações_inter_urbano

Um determinado transporte (veículo) efectua várias ligações, ou seja, parte de várias paragens origem e chega a várias paragens destino. A partida de uma paragem a determinada hora, serve de ponto de partida para diferentes paragens destino, ou seja, existem várias ligações cuja paragem origem e hora de partida é a mesma, mas com paragem destino e hora de chegada diferentes. Por outro lado, existem várias ligações com a mesma paragem destino e hora de chegada mas com paragem origem e hora de partida diferentes.

Preço_inter_urbano

Para uma determinada categoria de viagem são praticados vários preços, dependendo também da paragem origem e da paragem destino. Para além do preço do bilhete correspondente a uma viagem (ida ou volta), é especificado também, quando disponível, o preço do bilhete para a ida e a volta.

Inter_urbanos_Categorias

Um transporte inter-urbano pode ter várias categorias e uma mesma categoria pode caracterizar vários transportes.

Cidade_próxima

Uma localidade pode ter várias cidades próximas e uma cidade pode estar próxima de várias localidades.

3.2.1.3 Diagrama Entidade-Associação

A fig. 3.7. corresponde ao diagrama E-R obtido para os transportes inter-urbanos.

3.2.2 Transportes Urbanos

3.2.2.1 Conjuntos de entidades

Os conjuntos de entidades definidos para este tipo de transporte foram:

- Urbano
- Operador_urbano
- Local_aquisição_bilhetes_urbanos
- Características_transporte_urbano
- Bilhete_urbano_passe
- Bilhete_urbano_outros
- Paragem_urbano
- Morada_urbano
- Partidas_urbano
- Localidade

Urbano

No conjunto de entidades **Urbano** é guardada informação referente ao número e nome da carreira e o local de partida e de chegada. Uma entidade de **Urbano** designa uma carreira e não o transporte em si. Geralmente um transporte efectua duas carreiras, a ida e a volta, que correspondem a duas entidades diferentes do conjunto de entidades **Urbano**.

Operador_urbano

Neste conjunto de entidades é guardado o nome da empresa que gere determinado transporte.

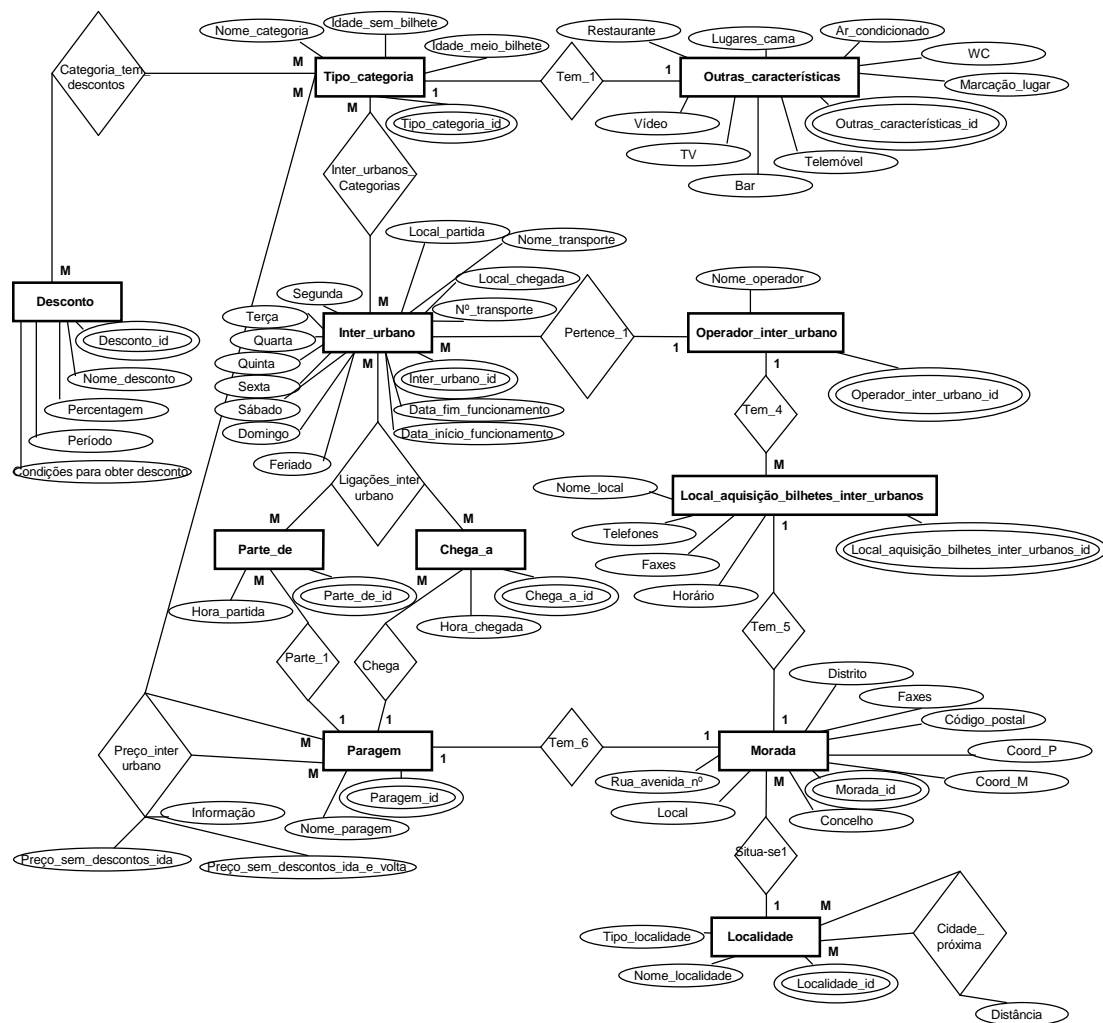


Fig. 3.7. Diagrama E-R para os transportes inter-urbanos

Características transporte urbano

Permite caracterizar os transportes urbanos. Neste conjunto de entidades para além do nome do transporte, são especificadas as idades mínimas e máximas para viajar sem bilhete, para pagar apenas meio bilhete ou determinada percentagem deste.

Local aquisição bilhetes urbanos

Permite especificar as características dos locais onde se podem comprar os bilhetes. Para além do nome do local, são especificados os números de telefone, de fax e o horário de atendimento ao público.

Bilhete urbano passe

Permite caracterizar os vários tipos de passes utilizados nos transportes urbanos. Para além do nome do passe, é necessário especificar o seu preço, o período de tempo e a área em que é válido, a descrição desta e uma breve descrição do passe.

Bilhete_urbano_outros

Permite caracterizar os vários tipos de bilhetes utilizados nos transportes urbanos. Para além do nome do bilhete, é necessário especificar o seu preço, o período de tempo em que é válido, o número de bilhetes que constituem determinado tipo de bilhete, o número de obliterações possíveis e uma breve descrição do bilhete.

Paragem_urbano

Neste conjunto de entidades são especificados os nomes das paragens.

Morada_urbano

Permite especificar as moradas das paragens e dos locais onde se podem comprar os bilhetes. Neste conjunto de entidades são especificados o nome da rua-avenida-praceta, o número do local, o nome deste, o código postal, o concelho e distrito a que pertence e as suas coordenadas topográficas.

Partidas_urbano

Permite especificar as partidas dos vários transportes urbanos, ou seja, as várias horas de partida que determinado transporte tem de cumprir saindo da paragem de partida, o seu período de funcionamento e os dias da semana em que se efectua a carreira a determinada hora.

Localidade

Permite identificar as localidades onde se situam as moradas. Neste conjunto de entidades são especificados o nome e o tipo da localidade (cidade, vila, etc.).

3.2.2.2 Conjuntos de associações

Um-para-um

Para os transportes urbanos existem dois conjuntos de associações deste tipo.

Tem_8

Representa o facto de uma paragem só ter uma morada e vice-versa. Uma morada apenas pode localizar uma determinada paragem.

Tem_10

A explicação é a mesma dada para o conjunto de associações anterior, ou seja, um determinado sítio, neste caso, cada local onde se podem adquirir os bilhetes é caracterizado por uma morada e vice-versa.

Muitos-para-um

Para os transportes urbanos existem vários conjuntos de associações deste tipo.

Pertence_2

Representa o facto de um operador (empresa) ter vários transportes (veículos), mas cada veículo pertencer apenas a um operador.

Tem_7

Este conjunto de associações representa o facto de existirem vários veículos com o mesmo conjunto de características, no entanto, um veículo só pode ser caracterizado por um conjunto de características.

Tem_9

Um operador (empresa) possui vários locais onde se podem adquirir os bilhetes (esses locais podem ser ou não sítios de paragem do transporte).

Parte_3

Para cada carreira que um transporte urbano efectue, existe apenas uma paragem considerada paragem de partida, no entanto uma paragem pode ser paragem de partida para vários transportes.

Parte_2

Um transporte urbano pode efectuar várias partidas a horas diferentes da paragem de partida, no entanto, a partida a uma determinada hora da paragem de partida está associada a apenas um transporte, mesmo que outro transporte parta da mesma paragem de partida à mesma hora.

Utiliza_1

Um operador urbano tem à disposição vários bilhetes tipo passe, mas os bilhetes deste tipo apenas podem ser utilizados nos transportes pertencentes a esse operador.

Utiliza_2

Um operador urbano tem à disposição vários tipos de bilhetes, mas estes bilhetes apenas podem ser utilizados nos transportes pertencentes a esse operador.

Situa-se2

Uma morada identifica um sítio de uma determinada localidade, no entanto, numa localidade existem várias moradas.

Muitos-para-muitos

Existem dois conjuntos de associações deste tipo para os transportes urbanos.

Número_bilhetes_urbano

Um transporte urbano geralmente faz a ligação entre várias paragens, ou seja, parte de muitas paragens e tem como destino várias paragens, sendo que, de uma paragem geralmente partem vários transportes e a uma mesma paragem podem chegar vários transportes (as paragens não são utilizadas apenas por um transporte). Neste conjunto de associações é especificado também o tempo de duração e o número de bilhetes necessários para se ir de uma paragem a outra.

Cidade_próxima

Uma localidade pode ter várias cidades próximas e uma cidade pode estar próxima de várias localidades.

3.2.2.3 Diagrama Entidade-Associação

O diagrama E-R obtido para os transportes urbanos corresponde à fig. 3.8..

3.2.3 Táxis

3.2.3.1 Conjuntos de entidades

Os conjuntos de entidades definidos para este tipo de transporte foram:

Operador_táxi

Preço_táxi

Morada_táxi

Localidade

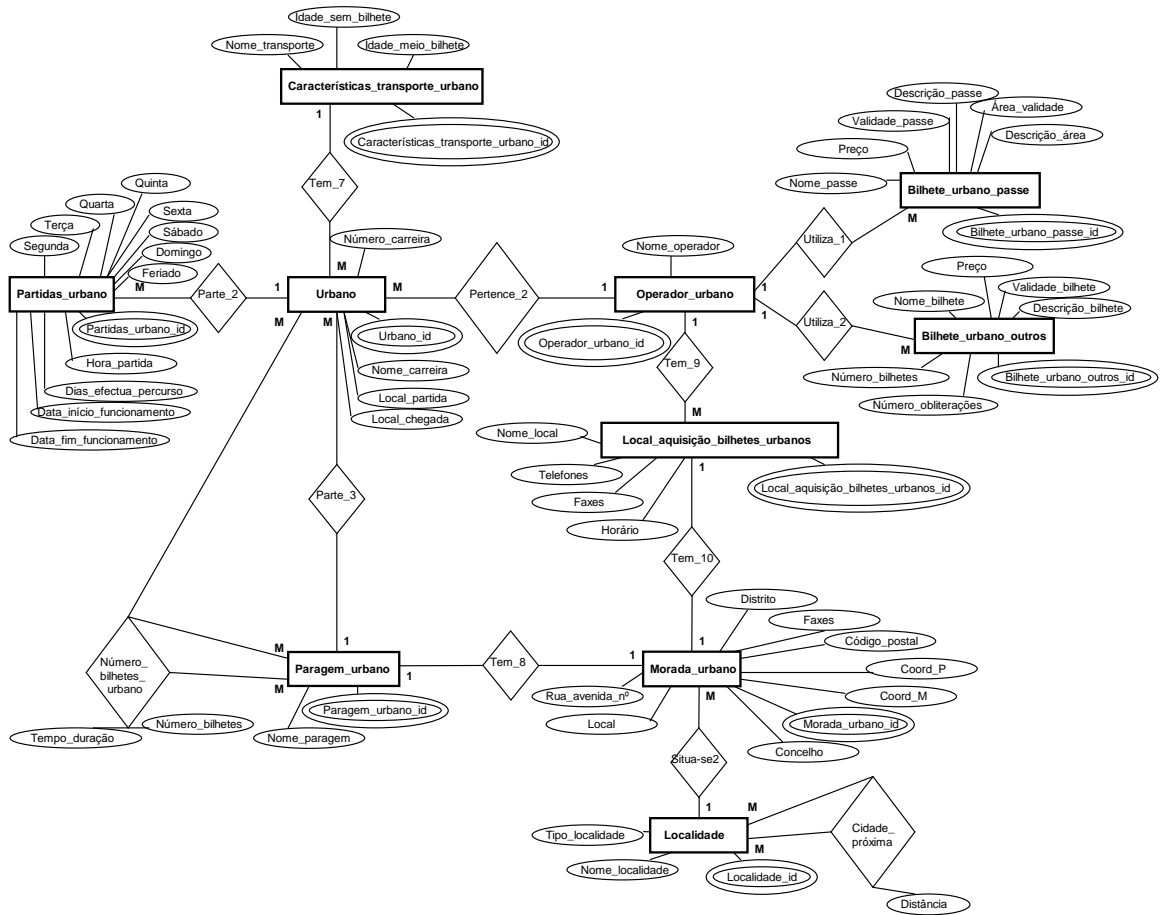


Fig. 3.8. Diagrama E-R para os transportes urbanos

Operador_táxi

Permite guardar informação sobre o nome do operador (empresa) e os números de telefone e de fax que servem para requisitar um táxi.

Preço_táxi

Este conjunto de entidades permite representar informação referente aos preços praticados pelos taxistas, o período em que se praticam tais preços (geralmente, durante a noite os preços são mais elevados), os dias da semana em que determinados preços estão em vigor, o preço acrescido por se utilizar a bagageira do táxi e o preço mínimo de utilização.

Morada_táxi

Permite especificar as moradas das praças de táxis onde se pode adquirir um táxi. Neste conjunto de entidades são especificados o nome da rua-avenida-praceta, o número do local, o nome deste, o código postal, o concelho e distrito a que pertence e as suas coordenadas topográficas.

Localidade

Permite identificar as localidades onde se situam as moradas. Neste conjunto de entidades são especificados o nome e o tipo da localidade (cidade, vila, etc.).

3.2.3.2 Conjuntos de associações

Muitos-para-um

Para os táxis existe apenas um conjunto de associações deste tipo.

Situa-se3

Uma morada identifica um sítio de uma determinada localidade, no entanto, numa localidade existem várias moradas.

Muitos-para-muitos

Para os táxis existem três conjuntos de associações deste tipo.

Preço_táxi_operador

Um operador pratica vários preços, que variam conforme a hora e o dia (fim de semana, feriado, dia útil). Por outro lado, os operadores praticam os mesmos preços, sob as mesmas condições.

Morada_praça_táxi_operador

Os taxistas pertencentes a determinado operador podem utilizar várias praças de táxis, enquanto que numa praça de táxi podem estar ao serviço táxis de diferentes operadores.

Cidade_próxima

Uma localidade pode ter várias cidades próximas e uma cidade pode estar próxima de várias localidades.

3.2.3.3 Diagrama Entidade-Associação

A figura seguinte corresponde ao diagrama E-R obtido para os táxis.

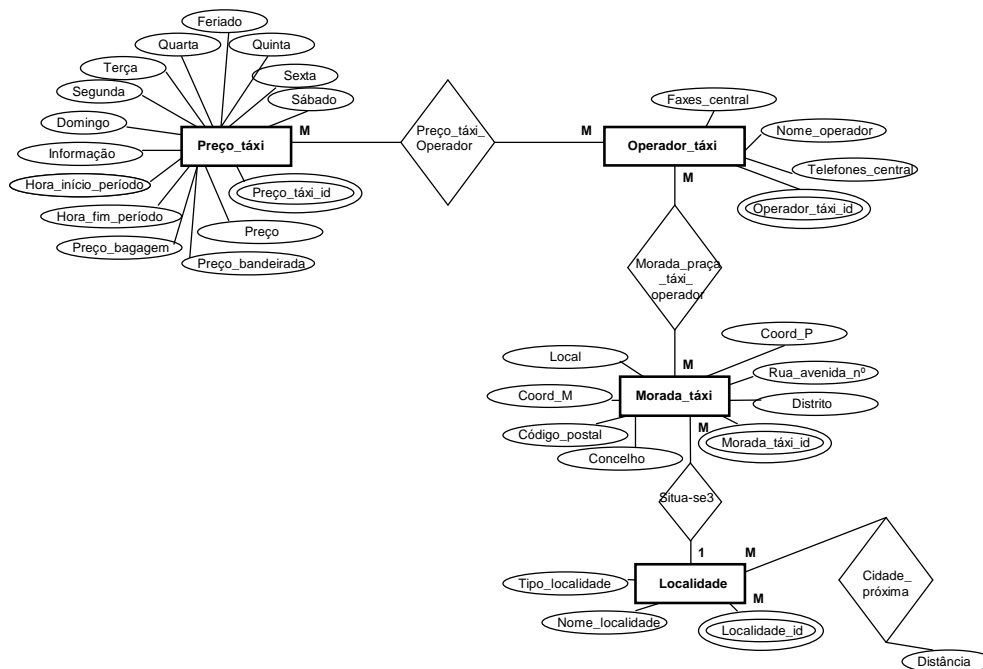


Fig. 3.9. Diagrama E-R para os táxis

Como se pode constatar, existem conjuntos de entidades comuns para os transportes inter-urbanos, para os transportes urbanos e para os táxis, no entanto, existem outros específicos a cada tipo de transporte.

O conjunto de entidades **Partidas_urbano** apenas foi criado para os transportes urbanos porque verificou-se que nos horários deste tipo de transporte não surge a hora de partida e a hora de chegada nas várias paragens tal como acontece para os transportes inter-urbanos, mas os minutos que o transporte gasta no percurso de uma paragem a outra. Para calcular a hora prevista de chegada a qualquer paragem é necessário saber a hora de partida da paragem de partida e somar-lhe todos os minutos que o transporte leva a percorrer as paragens intermédias entre a referida paragem de partida e a paragem para a qual se pretende calcular a hora de chegada. Contrariamente, para os transportes inter-urbanos foi

necessário criar os conjuntos de entidades **Parte_de** e **Chega_a** para poder especificar a hora de partida e a hora de chegada respectivamente, para qualquer paragem.

O conjunto de entidades **Outras_características** não foi criada para os transportes urbanos, uma vez que as características referidas nesse conjunto são próprias dos transportes inter-urbanos, como é o caso da existência ou não de restaurante, bar, etc. Os conjuntos de entidades **Tipo_categoria** e **Desconto** também não foram definidos para os transportes urbanos porque neste tipo de transporte apenas existe uma categoria de viagem e não se efectua nenhum tipo de desconto nos preços dos bilhetes.

Para os transportes urbanos houve necessidade de criar os conjuntos de entidades **Bilhete_urbano_passe** e **Bilhete_urbano_outros** porque para este tipo de transporte existem dois tipos de bilhetes, os passes e os outros bilhetes. Esta situação não se verifica para os transportes inter-urbanos, uma vez que não se utilizam passes e os outros bilhetes são todos do mesmo tipo (só se vendem em conjuntos de 1 bilhete).

De realçar, que o conjunto de entidades **Localidade** e o conjunto de associações **Cidade_próxima** são os únicos conjuntos que são iguais para os três tipos de transportes.

3.3 Conversão dos diagramas E-R em relações

O passo seguinte é converter os diagramas E-R numa base de dados. Ora, uma vez que o SGBD escolhido é um SGBDR (relacional) que suporta portanto, estruturas de dados relacionais, é necessário converter os diagramas E-R obtidos em relações ([6]).

Para tal, é necessário garantir que durante a conversão não se introduza redundância na base de dados, que se mantenham as restrições de integridade e se obtenha um número mínimo de relações.

Uma conversão ideal pressupõe que cada conjunto de entidades se transforme numa relação e que cada conjunto de associações se transforme igualmente numa relação.

Nesta fase da conversão é necessário garantir a preservação da integridade de entidades e da integridade referencial.

Enquanto que a integridade de entidades estabelece que nenhum componente da chave primária de uma relação pode apresentar valores nulos, a integridade referencial estabelece que nenhuma entidade pode fazer parte de uma relação a não ser que esta entidade esteja definida no conjunto de entidades. Outro critério a ter em conta nesta fase da conversão é tentar minimizar o número de relações.

Uma maneira de reduzir o número de relações é combinar qualquer conjunto de entidades, cujas entidades participem no máximo numa associação, com essa associação. Considere-se um exemplo retirado da base de dados que se está a desenvolver:

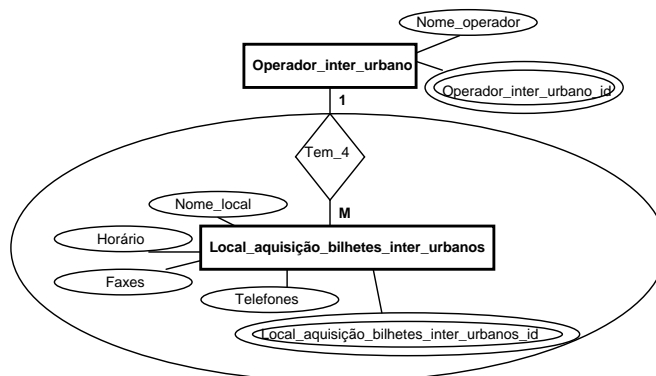


Fig. 3.10. Combinação do conjunto de associações **Tem_4** com o conjunto de entidades **Local_aquisição_bilhetes_inter_urbanos**

O conjunto de associações **Tem_4** estabelece que um operador possui vários locais onde se podem comprar os bilhetes utilizados pelos seus transportes e como é evidente, cada um desses locais apenas pertence a um operador (não é possível comprar bilhetes para transportes que não pertençam ao mesmo operador). Assim, em cada registo que representa um local virá indicado o seu operador, pelo que é possível combinar o conjunto de entidades **Local_aquisição_bilhetes_in-**

ter_urbanos com o conjunto de associações **Tem_4**, obtendo-se as relações da fig. 3.11..

Para simplificar os diagramas decidiu-se retirar as ovals que representam os atributos (incluindo a chave primária) e sublinhar a chave primária.

Uma vez que o conjunto de associações é combinado com o conjunto de entidades, é necessário que cada entidade deste conjunto de entidades guarde informação sobre a entidade do outro conjunto de entidades com a qual tem uma associação. Assim, em cada entidade é guardado o valor da chave primária da entidade com a qual tem a associação. Essa chave é chamada chave estrangeira e nos diagramas aparece a itálico.

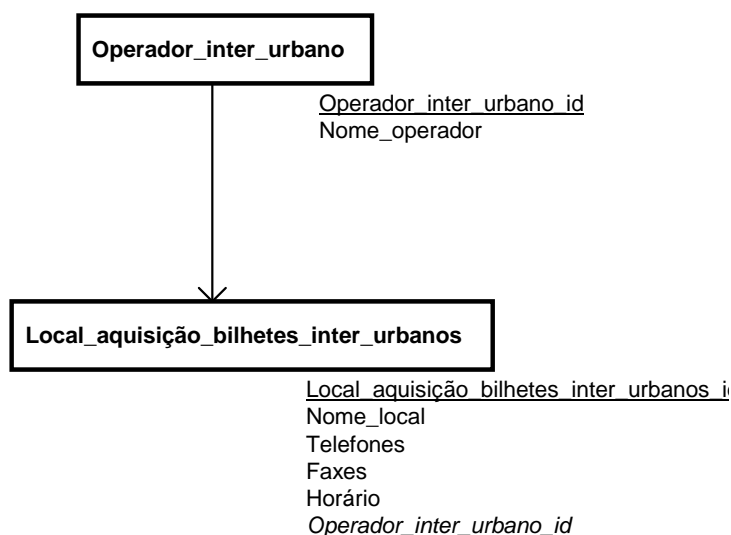


Fig. 3.11. Relações: Operador_inter_urbano e Local_aquisição_bilhetes_inter_urbanos

A seta que surge no segmento de recta que representa a ligação entre uma relação e outra indica a direcção da ligação, ou seja, que o valor da chave primária da relação onde começa a seta deve existir nessa relação antes de que seja usada como chave estrangeira na outra relação (onde acaba a seta). Tomando como exemplo as relações **Operador_inter_urbano** e **Local_aquisição_bilhetes_inter_urbanos** atrás apresentadas, antes de podermos utilizar um valor da chave primária como chave estrangeira em **Local_aquisição_bilhetes_inter_urbanos**, este valor já tem de estar definido na relação **Operador_inter_urbano**.

Aplicando a regra, que estabelece que qualquer associação de multiplicidade muitos-para-um pode ser combinada numa relação, aos diagramas E-R da base de dados em estudo, conseguiram-se combinar os conjuntos de entidades com os conjuntos de associações seguintes, originando as seguintes relações:

Conjunto de associações	Conjunto de entidades	Relação
Tem_4	Local_aquisição_bilhetes_inter_urbanos	Local_aquisição_bilhetes_inter_urbanos
Pertence_1	Inter_urbano	Inter_urbano
Parte_1	Parte_de	Parte_de
Chega	Chega_a	Chega_a
Tem_7	Urbano	Urbano
Tem_9	Local_aquisição_bilhetes_urbanos	Local_aquisição_bilhetes_urbanos
Pertence_2	Urbano	Urbano
Parte_2	Partidas_urbano	Partidas_urbano
Parte_3	Urbano	Urbano
Utiliza_1	Bilhete_urbano_passe	Bilhete_urbano_passe
Utiliza_2	Bilhete_urbano_outros	Bilhete_urbano_outros
Situa-se1	Morada	Morada
Situa-se2	Morada_urbano	Morada_urbano
Situa-se3	Morada_táxi	Morada_táxi

Tab. 3.1. Relações resultantes da combinação de conjuntos de entidades com conjuntos de associações de multiplicidade muitos-para-um

Considere-se um outro exemplo representado na fig. 3.12. Como cada entidade do conjunto de entidades **Outras_características** apenas diz respeito a uma entidade do conjunto de entidades **Tipo_categoria**, é possível juntar os dois conjuntos de entidades e o conjunto de associações numa só relação, tal como indicado na fig. 3.13..

Aplicando este raciocínio ao diagrama E-R, apenas se conseguiu combinar o conjunto de entidades **Outras_características**, o conjunto de entidades **Tipo_categoria** e o conjunto de associações **Tem_1** na relação **Tipo_categoria**.

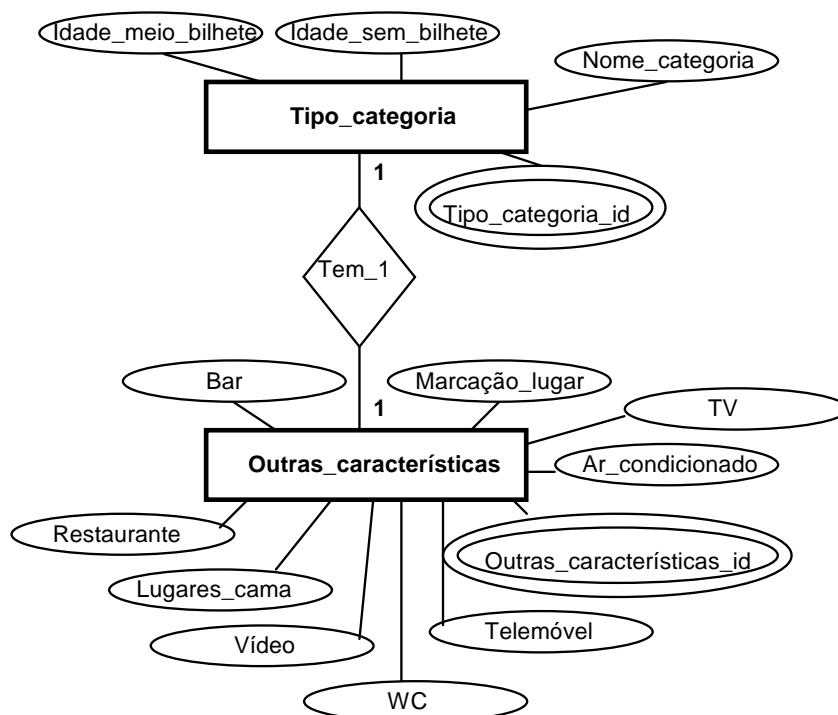


Fig. 3.12. Combinação do conjunto de associações Tem_1 com os conjuntos de entidades Tipo_categoria e Outras_características

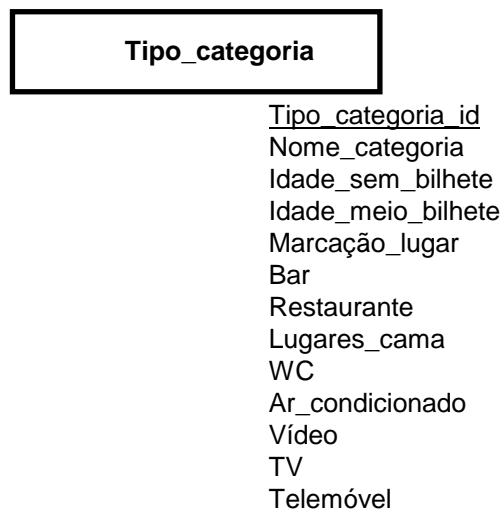


Fig. 3.13. Relação: Tipo_categoria

Não foi possível combinar numa só relação o conjunto de entidades **Local_aquisição_bilhetes_inter_urbanos** com o conjunto de entidades **Morada** e o conjunto de associações **Tem_5**, porque **Morada** é também argumento do conjunto de associações **Tem_6**. O mesmo acontece com o conjunto de entidades **Paragem**, o conjunto de entidades **Morada** e o

conjunto de associações **Tem_6**. O mesmo raciocínio é aplicado também aos transportes urbanos.

As relações obtidas foram:

Conjunto de associações	Conjunto de entidades	Relação
Tem_5	Local_aquisição_bilhetes_inter_urbanos	Local_aquisição_bilhetes_inter_urbanos
Tem_6	Paragem	Paragem
Tem_8	Paragem_urbano	Paragem_urbano
Tem_10	Local_aquisição_bilhetes_urbanos	Local_aquisição_bilhetes_urbanos

Tab. 3.2. Caso especial: Relações resultantes da combinação de conjuntos de entidades com conjuntos de associações

Os diagramas de relações obtidos para os transportes inter-urbanos, urbanos e táxis são os representados nas figuras 3.14., 3.15. e 3.16., respectivamente.

Utilizando outra notação, onde o nome da relação aparece seguido da lista de atributos (que estão separados por vírgulas e dentro de parêntesis curvos) e os atributos que pertencem à chave primária estão sublinhados e os que pertencem à chave estrangeira estão em itálico, obtiveram-se as relações para os transportes inter-urbanos, urbanos e táxis, especificadas no anexo A.

Elaboração de planos de transporte de pessoas

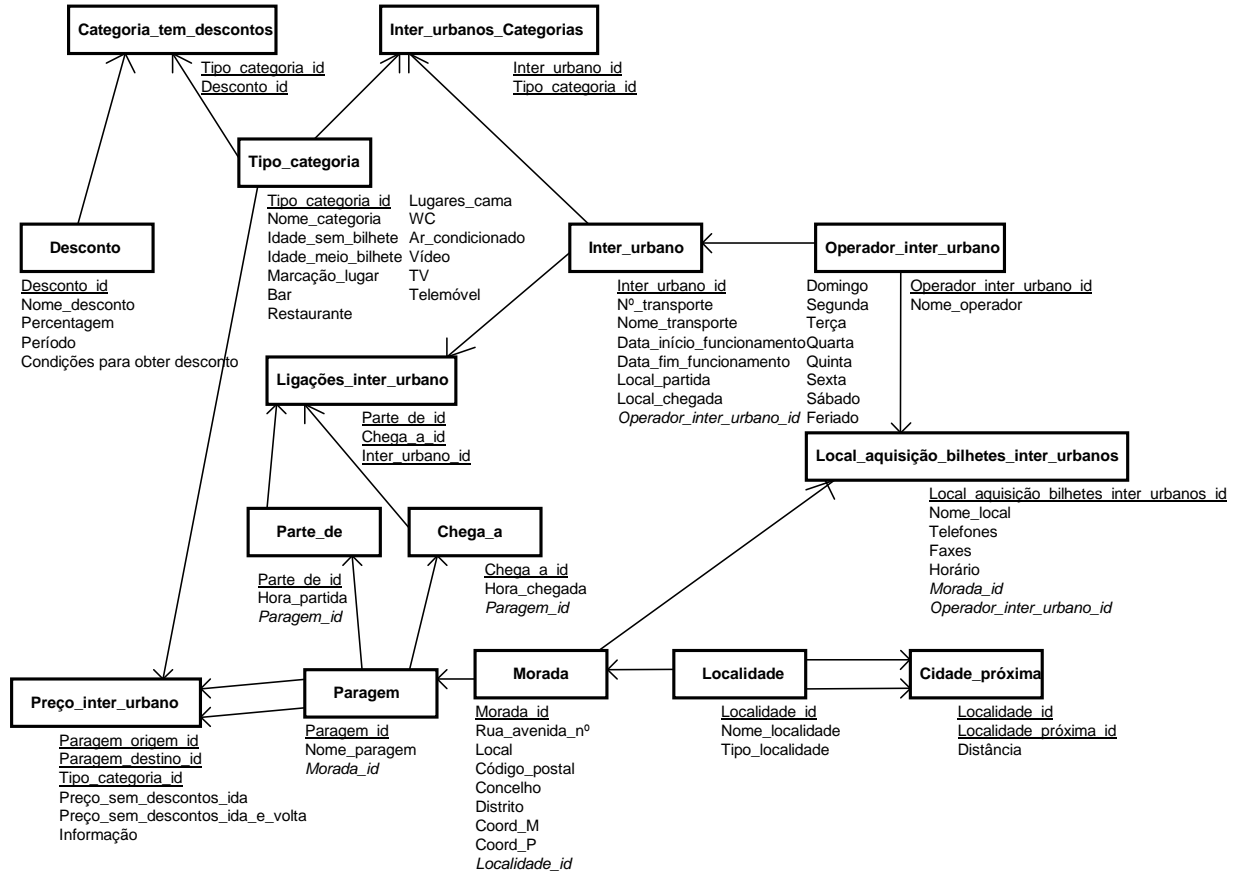


Fig. 3.14. Diagrama de relações para os transportes inter-urbanos

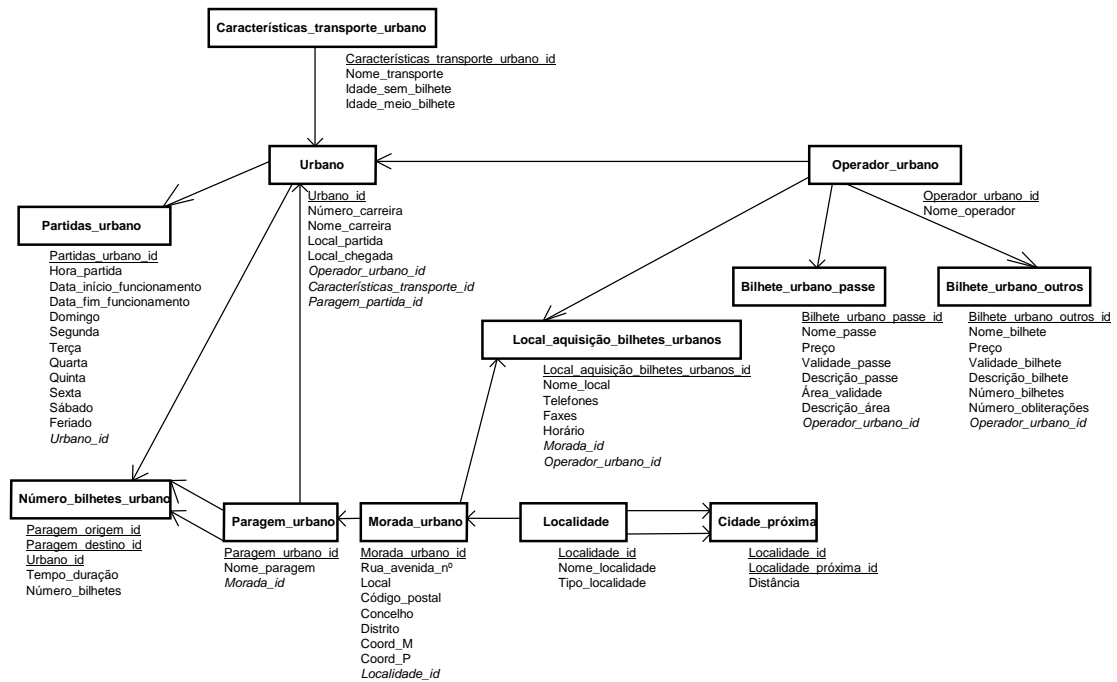


Fig. 3.15. Diagrama de relações para os transportes urbanos

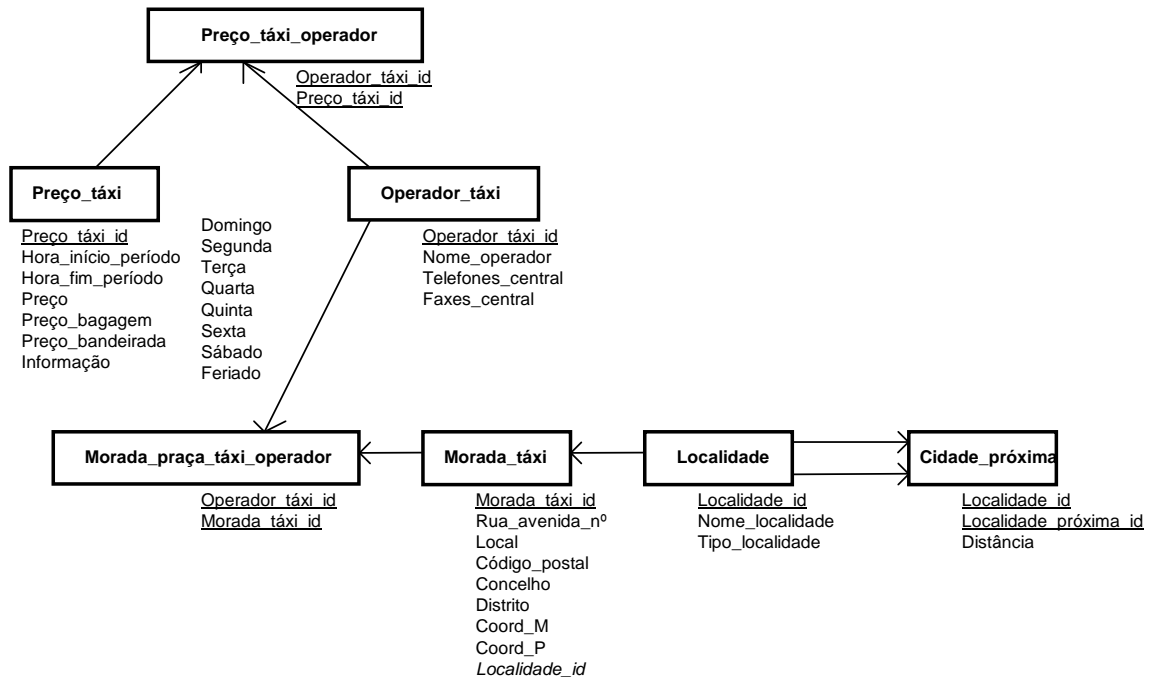


Fig. 3.16. Diagrama de relações para os táxis

3.4 Comparação das relações obtidas com as classes definidas na representação do conhecimento

Como se pode verificar, quase todas as classes definidas no primeiro capítulo referentes à representação do conhecimento originaram conjuntos de entidades ou de associações e posteriormente relações.

Algumas classes não originaram conjuntos de entidades ou de associações como é o caso das classes **Paragem_e_local_aquisição_bilhetes_urbanos** e **Paragem_e_local_aquisição_bilhetes_inter_urbanos**, uma vez que se veio verificar que não era necessário juntar num só conjunto de entidades, informação referente aos conjuntos de entidades **Paragem** e **Local_aquisição_bilhetes_inter_urbanos** para os transportes inter-urbanos e **Paragem_urbano** e **Local_aquisição_bilhetes_urbanos** para os transportes urbanos.

Por outro lado, a classe **Condição_Desconto** tornou-se não num conjunto de entidades mas num atributo do conjunto de entidades **Desconto**.

A classe **Tipo_categoria_preço** transformou-se no conjunto de entidades **Tipo_categoria** que não contém nenhum atributo referente aos preços dos bilhetes, uma vez que estes preços tornaram-se atributos do conjunto de associações criada para o efeito, **Preço_inter_urbano**.

As classes **Preços_táxi** e **Preços**, referentes aos táxis, foram agrupadas no conjunto de entidades **Preço_táxi**, uma vez que se verificou que não era necessário fazer explicitamente a distinção entre os preços praticados pelos taxistas durante o dia e durante a noite.

Quando se definiu e realizou a base de dados, tornou-se necessário criar outros conjuntos de entidades e de associações que não tinham sido definidos como classes. Exemplo disso são os conjuntos de entidades **Parte_de**, **Chega_a**, **Partidas_urbano** e os conjuntos de associações **Inter_urbanos_Categorias**, **Preço_inter_urbano**, **Ligações_inter_urbano**, **Categoria_tem_descontos**, **Inter_urbanos_Categorias**, **Número_bilhetes_urbano**, **Morada_praça_táxi_operador** e **Preço_táxi_operador**.

3.5 Realização da base de dados

3.5.1 Informação contida na base de dados

Na base de dados está guardada informação referente a alguns dos transportes inter-urbanos que fazem as ligações entre o Algarve e Lisboa, Barreiro, Setúbal, etc. e alguns dos transportes urbanos e táxis que circulam no Algarve e na zona de Lisboa.

Os transportes escolhidos para representar os transportes inter-urbanos foram o comboio intercidades, o autocarro-expresso e o avião.

Decidiu-se guardar informação de apenas estes três tipos de transportes inter-urbanos porque o que se pretende é criar uma base de dados mínima com a qual se consigam simular os planos de transportes. Por outro lado, estes três tipos de transportes representam os três grupos

de transportes inter-urbanos que fazem ligações entre o Algarve, Lisboa e arredores. Enquanto que o comboio intercity representa o transporte terrestre ferroviário, o autocarro-expresso representa o transporte terrestre rodoviário e o avião o transporte aéreo.

Em relação aos transportes urbanos consideraram-se alguns dos autocarros que circulam em Faro e arredores e alguns dos que circulam em Lisboa e arredores.

Para os táxis, consideraram-se apenas os táxis que circulam no Algarve e em Lisboa.

A informação contida na base de dados refere-se a transportes reais e não a casos fictícios.

3.5.2 Aspectos de realização

O SGBDR utilizado é o Microsoft Access 2.0 ([9]). Neste sistema as relações representam-se por tabelas (tables), enquanto que as ligações entre as relações representam-se por relações (relationships). Tal como se pode verificar pelos diagramas das figuras 3.14., 3.15. e 3.16., definiram-se treze tabelas para os transportes inter-urbanos, dez para os transportes urbanos, cinco para os táxis e duas tabelas partilhadas pelos três tipos de transportes.

Embora existam relações comuns para os vários tipos de transportes, como é o caso das relações **Morada**, **Morada_urbano** e **Morada_táxi** definidas para os transportes inter-urbanos, urbanos e táxis, respectivamente, que poderiam ser definidas na mesma tabela, decidiu-se defini-las em tabelas diferentes por uma questão de simplicidade na organização da informação.

Definiram-se consultas (queries) e formulários (forms) para cada um dos tipos de transportes, de modo a introduzir de forma esquematizada a informação sobre os transportes. O número destas consultas e formulários é de 34 e de 23, respectivamente. O modo de utilização destes formulários e a informação neles contida estão apresentados no anexo B.

4. ALGORITMOS E HEURÍSTICAS

4

4.1 Introdução

Este capítulo apresenta os algoritmos e as heurísticas utilizados na determinação dos planos de transporte de pessoas.

Os algoritmos estão divididos em três grupos:

1. Os algoritmos que determinam os planos de transportes possíveis para ir de Origem a Destino, utilizando todos os tipos de transportes, excepto os táxis.
2. Os algoritmos que determinam os melhores planos de entre os planos possíveis atrás referidos.
3. Os algoritmos que determinam os planos de transportes para os táxis.

Como já foi referido, o utilizador para além de indicar a localidade Origem e a localidade Destino, indica o dia e a hora a partir da qual pretende sair de Origem (esta opção será designada por opção 2) ou o dia e a hora antes da qual pretende chegar a Destino (e esta por opção 1). O utilizador especifica também os critérios que devem ser considerados na determinação dos melhores planos de transportes. Estes critérios reflectem a importância para o utilizador de factores como o preço dos bilhetes, o tempo de duração da viagem e a qualidade dos transportes utilizados, ou seja, se o utilizador pretende uma viagem de baixo custo, uma viagem que demore pouco tempo, etc.. As janelas onde o utilizador indica estes dados estão apresentadas no anexo C.

Para determinar os melhores planos de transportes, foram utilizadas técnicas de decisão multi-atributo, nomeadamente, o método da Pesagem Simples Aditiva, também designado por método das Ponderações.

Este capítulo divide-se em várias secções; na secção 4.2.1 são apresentados os algoritmos que determinam os planos de transportes para todos os tipos de transportes, excepto os táxis. Na secção 4.2.2 são apresentados os algoritmos que determinam os planos de transportes para os táxis. Na secção 4.3 estão indicadas as heurísticas que são utilizadas pelos algoritmos. Na secção 4.4 são explicadas as técnicas de decisão multi-atributo. Por último, na secção 4.5 são indicados os algoritmos que determinam os melhores de entre os planos de transportes determinados na secção 4.2.1.

4.2 Algoritmos que determinam os planos de transportes

Os algoritmos desenvolvidos nesta secção determinam os planos de transportes possíveis para ir de Origem a Destino utilizando todos os tipos de transportes, excepto os táxis e o plano de transporte para os táxis.

Para o primeiro tipo de planos, as informações fornecidas ao utilizador são a hora de partida e de chegada, a localização das paragens de partida e de chegada (rua ou avenida, número da porta, local, concelho e distrito), as categorias existentes, as condições disponíveis para cada categoria, os preços dos bilhetes, as idades limites que determinam a isenção total ou parcial de pagamento do bilhete e o nome do operador.

Em relação aos planos referentes aos táxis, são fornecidos ao utilizador a distância e o preço mínimos para ir de Origem a Destino, as moradas das praças de táxis em Origem e em Destino (conforme necessário) e os números de telefone e de fax das centrais de táxis.

O algoritmo que permite determinar os planos atrás referidos é o seguinte:

/* **Algoritmo Principal**

*/

Determinar os planos de transportes possíveis para todos os tipos de transportes, excepto o táxi (**Algoritmo DeterminaPlanos**)

Para cada um dos transportes principais determinados

Determinar os planos de transportes para os táxis (**Algoritmo DeterminaTáxis**)

Escolher os melhores planos em função das preferências do utilizador (**Algoritmo EscolheMelhores**)

/* Fim do algoritmo **Principal**

*/

Os planos referentes aos táxis não são considerados na determinação dos melhores planos de transportes devido à imprecisão dos dados que são determinados para este tipo de transporte.

As secções seguintes tratam dos algoritmos necessários para determinar os planos de transportes, utilizando todos os tipos de transportes, excepto os táxis (secção 4.2.1) e dos algoritmos necessários para determinar os planos de transportes referentes aos táxis (secção 4.2.2).

4.2.1 Algoritmos que determinam os planos de transportes para todos os tipos de transportes, excepto os táxis.

Para determinar os planos de transportes possíveis, consideraram-se quatro tipos de planos de transportes diferentes; planos de transportes directos, planos de transportes com origens alternativas à Origem, planos de transportes com destinos alternativos a Destino, planos de transportes com origens e destinos alternativos à Origem e a Destino, respectivamente.

Por exemplo, supondo que o utilizador pretende ir de Quarteira a Lisboa, os planos de transportes directos correspondem aos transportes directos que efectuem a ligação entre Origem e Destino, ou seja, entre Quarteira e Lisboa.

Um plano de transportes com origens alternativas à Origem, corresponde a um transporte que efectue a ligação entre a localidade Origem e uma localidade próxima a Origem (a este transporte chamaremos

transporte secundário) e ao transporte que efectue a ligação entre a localidade próxima a Origem, anteriormente referida, e a localidade Destino (a este transporte chamaremos transporte principal). A localidade próxima a Origem é determinada por um algoritmo, cujo objectivo é encontrar localidades próximas ao local fornecido, dando preferência às cidades.

Voltando ao caso prático e supondo que uma das localidades próximas encontradas foi Faro, um plano possível corresponde a um transporte que efectue a ligação entre Quarteira e Faro e um outro que efectue a ligação entre Faro e Lisboa.

Os planos de transportes com destinos alternativos a Destino correspondem à situação inversa da situação anterior, ou seja, a um transporte que efectue a ligação entre Origem e uma localidade próxima a Destino e a um outro transporte que efectue a ligação entre esta e Destino. A determinação das localidades próximas a Destino é feita pelo mesmo algoritmo que determina as localidades próximas a Origem.

Considere-se que uma das localidades próximas a Destino encontradas é Almada. Um plano de transportes possível corresponde a um transporte que efectue a ligação entre Quarteira e Almada e um outro que efectue a ligação entre Almada e Lisboa.

Por último, os planos de transportes com origens e destinos alternativos à Origem e a Destino, respectivamente, correspondem a um transporte que efectue a ligação entre Origem e uma localidade próxima a Origem, um outro transporte que efectue a ligação entre a localidade próxima a Origem e uma localidade próxima a Destino e por último, por um outro que efectue a ligação entre a localidade próxima a Destino e Destino.

Utilizando as localidades próximas a Origem e a Destino consideradas anteriormente, um plano de transportes possível corresponde a um transporte que efectue o percurso Quarteira-Faro, um outro transporte que efectue Faro-Almada e, por último, por um outro que ligue Almada a Lisboa.

Para os transportes principais consideraram-se apenas os transportes inter-urbanos, enquanto que para os secundários consideraram-se transportes inter-urbanos e urbanos. Esta opção tem duas razões; primeiro, devido ao agravamento, em termos de tempo que o programa sofreria para determinar transportes principais inter-urbanos e urbanos e os transportes

secundários para cada um deles; segundo, porque é mais provável que Origem e Destino sejam localidades distantes entre si e portanto que se utilize um transporte inter-urbano como transporte principal.

Como já foi referido anteriormente, um transporte é considerado urbano quando utiliza essencialmente bilhetes pré-comprados e passes, sendo os transportes inter-urbanos todos os outros com excepção do táxi. Por outro lado, os transportes urbanos geralmente efectuem percursos de curta duração, ao contrário dos inter-urbanos cujos percursos são de longa duração.

Por uma questão de simplicidade, decidiu-se chamar aos planos de transportes directos, de Situação 1; aos planos de transportes que incluam origens alternativas à Origem, de Situação 2; aos planos de transportes com destinos alternativos a Destino, de Situação 3; e por último, aos planos de transportes com origens e destinos alternativos à Origem e a Destino, respectivamente, de Situação 4. O algoritmo **DeterminaPlanos** que é utilizado pelo algoritmo **Principal**, verifica se as localidades pretendidas pelo utilizador como Origem e Destino existem na base de dados. Se tal acontecer, verifica qual o dia da semana a que corresponde o dia especificado pelo utilizador e se é ou não feriado. De seguida, determina os transportes inter-urbanos que correspondem à situação 1. Posteriormente, retira de entre estes transportes, aqueles que não funcionam devido ao seu período de funcionamento não abranger o dia especificado pelo utilizador, os que não funcionam no dia da semana correspondente ao dia especificado e por último, os que não funcionam aos feriados, se o dia especificado for um dia feriado. A seguir, retira também os que não podem ser utilizados devido à hora especificada pelo utilizador. Por último, verifica para as situações 2, 3 e 4, quais as ligações possíveis para cada situação e os transportes inter-urbanos e urbanos que efectuem essas ligações. Se Origem e/ou Destino não existirem, é enviada uma mensagem a informar que não é possível determinar os planos de transportes.

/* **Algoritmo DeterminaPlanos**

*/

Verificar se **Origem** e **Destino** existem na base de dados (**Algoritmo VerificaOrigemDestinoExistem**)

Em relação à data especificada pelo utilizador, verificar o dia da semana e se é um dia feriado (**Algoritmo DeterminaDataeHora**)

/* Corresponde à **Situação 1**

*/

Determinar os transportes inter-urbanos directos para ir de **Origem a Destino**

Retirar dos transportes anteriores, aqueles que não funcionam, tendo em conta o período de funcionamento do transporte, o dia da semana e do facto de o dia ser ou não feriado (**Algoritmo ApagaRegistosFunçãoPDF**)

Se não se retiraram todos os transportes

Verificar quais os que podem ser utilizados em função da hora especificada pelo utilizador (**Algoritmo ApagaRegistosHoraUtilizador**)

/* Corresponde às **Situações 2,3,4**

*/

Para cada uma das situações (2,3,4) (**Heurística H1**)

Determinar outras ligações tendo em conta situação (**Algoritmo ProcuraOutrasLigaçõesPrincipais**)

Determinar os transportes inter-urbanos e/ou urbanos que efectuem as ligações determinadas anteriormente (**Algoritmo ProcuraInterUrbanoseUrbanos234**)

/* Fim do algoritmo **DeterminaPlanos**

*/

O algoritmo **VerificaOrigemDestinoExistem**, verifica se os locais considerados como Origem e Destino existem na base de dados. Se pelo menos um dos locais não existir, o utilizador é informado que não é possível determinar os planos de transportes pedidos. Se os locais existirem, o algoritmo forma a ligação directa, com origem no local Origem e destino em Destino.

/* **Algoritmo VerificaOrigemDestinoExistem**

*/

Procurar o local na base de dados (**Algoritmo ProcuraLocalId**)

Se o local estiver na base de dados

Se o local for Origem

A origem da ligação directa é Origem

Se o local for Destino

O destino da ligação directa é Destino

Senão

Mostra mensagem a informar que o local não existe na base de dados e que não é possível construir planos de transportes

/* Fim do algoritmo **VerificaOrigemDestinoExistem**

*/

O algoritmo **ApagaRegistosFunçãoPDF** retira os transportes que não funcionam em função do respectivo período de funcionamento, do dia da

semana a que corresponde o dia especificado e do facto deste ser ou não um dia feriado.

Se o utilizador escolheu a opção 2 (pretende partir de Origem em determinado dia e a partir de determinada hora), é necessário rejeitar os transportes directos que não funcionam nesse dia. Este facto é verificado tendo em conta o período de funcionamento do transporte, ou seja, se o transporte funciona durante todos os meses do ano ou só durante o verão, etc.. Por outro lado, é necessário verificar o dia da semana correspondente ao dia especificado e verificar se esse dia é ou não um dia feriado, por forma a retirar os transportes que não funcionam nesses dias.

Se a opção escolhida for a opção 1 (o utilizador pretende chegar a Destino em determinado dia e até determinada hora), deve-se verificar para cada transporte directo, os que partem e chegam no mesmo dia. Para estes procede-se da mesma forma que para a opção 2. Para os transportes que chegam a Destino no dia especificado mas que partem no dia anterior, é necessário determinar o dia anterior e proceder para este, a todas as verificações já indicadas anteriormente.

Os transportes directos que verificam todas as condições indicadas anteriormente são seleccionados novamente em função da hora indicada pelo utilizador, utilizando o algoritmo **ApagaRegistosHoraUtilizador**. Se a opção for a 1, deve-se verificar se os transportes chegam antes ou após a hora especificada pelo utilizador. Os que chegam após são retirados. Por outro lado, se a opção for a 2, deve-se verificar quais os transportes que partem antes da hora especificada por forma a não serem considerados nos planos de transportes.

/* **Algoritmo ApagaRegistosFunçãoPDF**

*/

Se opção 2 (O utilizador pretende partir de Origem em determinado dia)

Retirar os transportes que não funcionam no dia especificado pelo utilizador

(Algoritmo ApagaFunçãoPeríodo)

Retirar os transportes que não funcionam no dia da semana do dia especificado pelo utilizador e no caso deste dia ser um dia feriado, retirar os transportes

que não funcionam aos feriados **(Algoritmo ApagaFunçãoDiadaSemanaeFeriado)**

Se opção 1 (O utilizador pretende chegar a Destino em determinado dia)

Para cada transporte

Verificar se parte e chega no mesmo dia **(Algoritmo ParteChegaMesmoDia)**

Se sim (o transporte parte no dia especificado pelo utilizador)
Igual a opção 2

Se não (o transporte parte no dia anterior ao especificado)
Determinar o dia anterior, determinar o dia da semana e se é um dia
feriado (**Algoritmo CalculaDataAnterior**)

Retirar o transporte se este não funciona no dia determinado (**Algoritmo
ApagaFunçãoPeríodo**)

Retirar o transporte se no dia da semana do dia determinado o transporte
não funciona. Se este dia for um dia feriado, retirar o transporte se este
não funciona aos feriados. (**Algoritmo**

ApagaFunçãoDiadaSemanaeFeriado)

/* Fim do algoritmo **ApagaRegistosFunçãoPDF**

*/

/* **Algoritmo ApagaRegistosHoraUtilizador**

*/

Se opção 2

Verificar se a hora de partida do transporte é anterior à hora especificada
pelo utilizador

Se sim

Retirar o transporte

Se opção 1

Verificar se a hora de chegada do transporte é posterior à hora especificada
pelo utilizador

Se sim

Retirar o transporte

/* Fim do algoritmo **ApagaRegistosHoraUtilizador**

*/

A procura das ligações que permitem determinar os transportes principais é feita pelo algoritmo **ProcuraOutrasLigaçõesPrincipais**. Este algoritmo procura para as situações 2 e 4, localidades próximas a Origem, o mesmo acontecendo para as situações 3 e 4, mas neste caso, próximas a Destino. O algoritmo dá preferência às cidades e só no caso de não as encontrar, é que procura outro tipo de localidades.

Por outro lado, só depois de fazer a procura dos locais é que estabelece as ligações, que variam conforme a situação. Considere-se que foram encontradas duas localidades próximas a Origem e outras duas próximas a Destino. Se a situação for a 2, para cada uma das localidades próximas a Origem estabelece uma ligação com Destino. Por outras palavras, se Origem1 e Origem2 designarem as duas localidades próximas a Origem, o algoritmo forma as ligações Origem1-Destino e Origem2-Destino. Se a situação for a 3, para cada uma das localidades próximas a Destino

estabelece uma ligação com Origem. Exemplificando, se Destino1 e Destino2 designarem as duas localidades em questão, as ligações estabelecidas são: Origem-Destino1 e Origem-Destino2. Por último, quando a situação é a 4, para cada localidade próxima a Origem, o algoritmo estabelece uma ligação com cada localidade próxima a Destino. Utilizando as designações atrás referidas, o algoritmo forma as seguintes ligações, Origem1-Destino1, Origem1-Destino2, Origem2-Destino1 e Origem2-Destino2.

/* **Algoritmo ProcuraOutrasLigaçõesPrincipais**

*/

Se Situação for

Situação 2 ou **Situação 4**

Determinar os locais próximos a **Origem**, de preferência cidades (**Algoritmo ProcuraLocaisPróximosId**) (**Heurística H2**)

Situação 3 ou **Situação 4.**

Determinar, os locais próximos a **Destino**, de preferência cidades (**Algoritmo ProcuraLocaisPróximosId**) (**Heurística H2**)

Se Situação for

Situação 2

Para cada local próximo a **Origem**

Estabelece uma ligação em que a origem é o local próximo a **Origem** e o destino é **Destino**

Situação 3

Para cada local próximo a **Destino**

Estabelece uma ligação em que a origem é **Origem** e o destino é o local próximo a **Destino**

Situação 4

Para cada local próximo a **Origem**

Para cada local próximo a **Destino**

Estabelece uma ligação em que a origem é o local próximo a **Origem** e o destino é o local próximo a **Destino**

/* Fim do algoritmo **ProcuraOutrasLigaçõesPrincipais**

*/

Depois de pesquisar as ligações possíveis para cada uma das situações, o algoritmo **ProcuralInterUrbanosEUrbanos234** procura os transportes inter-urbanos e urbanos que efectuam essas ligações. Neste mesmo algoritmo, é feita a procura das ligações secundárias e dos transportes que efectuam essas ligações. Se a situação for a 2 ou a 3, o algoritmo procura a ligação secundária correspondente e os transportes que

efectuam cada uma dessas ligações. Se o algoritmo encontrar algum transporte, verifica para cada transporte principal, quais os transportes secundários possíveis, em função das horas de partida e de chegada e da distância entre as respectivas paragens.

Quando a situação é a 4, o algoritmo procura as duas ligações secundárias e os transportes que as efectuam. No algoritmo, chama-se ligação secundária anterior à ligação secundária que se efectua antes da ligação principal, e ligação secundária posterior à que se efectua depois da ligação principal. Os transportes secundários de cada uma das ligações, chamam-se, respectivamente, transportes secundários anteriores e transportes secundários posteriores. De entre os transportes secundários, anteriores e posteriores, o algoritmo verifica quais são os possíveis, para cada transporte principal, em função das horas de partida e de chegada e da distância entre as respectivas paragens.

/* Algoritmo ProcuraInterUrbanosEUrbanos234

*/

Para cada ligação principal determinada

Determinar os transportes inter-urbanos que efectuam a ligação

Retirar os transportes principais que não devem ser considerados em função da situação. **(Algoritmo ApagaTransportesPrincipais) (Heurística H4)**

Se Situação for

Situação 2 ou Situação 3

Para a ligação principal considerada, determinar a ligação secundária

(Algoritmo ProcuraLigaçãoSecundária)

Determinar os transportes inter-urbanos e urbanos que efectuam a ligação secundária

Se se encontrou algum transporte

Se Situação 2

Estabelecer planos para esta situação tendo em conta a hora e a data especificadas pelo utilizador **(Algoritmo ApagaRegistosFunçãoData2)**

Se Situação 3

Estabelecer planos para esta situação tendo em conta a hora e a data especificadas pelo utilizador **(Algoritmo ApagaRegistosFunçãoData3)**

Situação 4

Para a ligação principal considerada, determinar as ligações secundárias. **(Algoritmo ProcuraLigaçãoSecundária)**. A estas ligações chamam-se ligação secundária anterior e ligação secundária posterior

```

ligação      Determinar os transportes inter-urbanos e urbanos que efectuam a
              secundária anterior
ligação      Determinar os transportes inter-urbanos e urbanos que efectuam a
              secundária posterior
              Se se encontrou pelo menos um transporte para cada uma das ligações
              secundárias
              Estabelecer planos para esta situação, tendo em conta a hora e a
data          especificadas pelo utilizador (Algoritmo
ApagaRegistosFunçãoData4)
/* Fim do algoritmo ProcuraInterUrbanosEUrbanos234
*/

```

O objectivo do algoritmo **ApagaTransportesPrincipais** é evitar que, para a situação 2, Origem seja uma paragem do transporte principal, que para a situação 3, Destino seja uma paragem do transporte principal e por último, que para a situação 4, se verifiquem ambos os casos. Imagine-se que o utilizador pretende ir de Quarteira a Lisboa e que existe um transporte que parte de Faro com destino a Lisboa passando por Quarteira. O algoritmo em questão, evita que surja um plano de transportes que efectue a ligação Quarteira-Faro num transporte e Faro-Lisboa noutro transporte que inclua a passagem por Quarteira.

```

/* Algoritmo ApagaTransportesPrincipais

```

```

*/
Se a Situação for
  Situação 2
    Retirar os transportes principais em que Origem seja uma das suas paragens
  Situação 3
    Retirar os transportes principais em que Destino seja uma das suas paragens
  Situação 4
    Retirar os transportes principais em que Origem seja uma das suas
paragens.
    Retirar os transportes principais em que Destino seja uma das suas paragens
/* Fim do algoritmo ApagaTransportesPrincipais
*/

```

O algoritmo **ProcuraLigaçãoSecundária** procura as ligações secundárias em função do tipo de situação. Assim, para a situação 2, a ligação secundária tem como origem a localidade Origem e como destino a localidade origem da ligação principal. Para a situação 3, a ligação secundária tem como origem a localidade destino da ligação principal e como destino a localidade Destino. Por último, para a situação 4, a ligação secundária anterior tem como origem a localidade Origem e como destino a

localidade origem da ligação principal, e a ligação secundária posterior tem como origem a localidade destino da ligação principal e como destino a localidade Destino.

/* **Algoritmo ProcuraLigaçãoSecundária**

*/

Se Situação for

Situação 2

A origem da ligação secundária é **Origem** e o destino é a origem da ligação principal.

Situação 3

A origem da ligação secundária é o destino da ligação principal e o destino é **Destino**.

Situação 4

Para uma ligação secundária, a origem é **Origem** e o destino é a origem da ligação principal.

Para a outra ligação secundária, a origem é o destino da ligação principal e o destino é **Destino**.

/* Fim do algoritmo **ProcuraLigaçãoSecundária**

*/

Os algoritmos **ApagaRegistosFunçãoData2**, **ApagaRegistosFunçãoData3** e **ApagaRegistosFunçãoData4** são utilizados no algoritmo **ProcuraInterUrbanoseUrbanos234**, respectivamente, para as situações 2, 3 e 4.

O algoritmo **ApagaRegistosFunçãoData2** retira, se a opção escolhida for a 1, os transportes principais que chegam depois da hora especificada (nesta situação, são os transportes principais os que chegam a Destino). Se não se retiram todos os transportes principais e se se encontram transportes secundários, então são determinados planos de transportes para esta situação. Contrariamente, se a opção escolhida for a 2 e uma vez que são os transportes secundários os que partem de Origem, retiram-se os que não funcionam no dia especificado pelo utilizador e os que partem antes da hora especificada. Se existirem transportes secundários que verificam todas estas condições e se existirem transportes principais, o algoritmo determina planos utilizando estes transportes.

/* **Algoritmo ApagaRegistosFunçãoData2**

*/

Se opção 1

Retirar os transportes principais cuja hora de chegada seja posterior à hora especificada pelo utilizador (**Algoritmo ApagaRegistosHoraUtilizador**)

Se não se retiraram todos os transportes principais e se existem transportes secundários

Estabelecer planos tendo em conta as ligações (**Algoritmo EstabeleceLigações23**)

Se opção 2

Retirar os transportes secundários que não funcionam no dia especificado pelo utilizador (**Algoritmo ApagaRegistosFunçãoPDF**)

Se não se retiraram todos os transportes secundários

Retirar os transportes secundários cuja hora de partida seja anterior à hora especificada pelo utilizador (**Algoritmo**

ApagaRegistosHoraUtilizador)

Se não se retiraram todos os transportes secundários e se existem transportes principais

Estabelecer planos tendo em conta as ligações (**Algoritmo EstabeleceLigações23**)

/* Fim do algoritmo **ApagaRegistosFunçãoData2**

*/

Utilizando o mesmo raciocínio descrito para a situação 2 e tendo em conta que para a situação 3, o transporte que parte de Origem é o transporte principal e o que chega a Destino é o transporte secundário, obtém-se o seguinte algoritmo:

/* **Algoritmo ApagaRegistosFunçãoData3**

*/

Se opção 1

Retirar os transportes secundários cuja hora de chegada seja posterior à hora especificada pelo utilizador (**Algoritmo ApagaRegistosHoraUtilizador**)

Se não se retiraram todos os transportes secundários e se existem transportes principais

Estabelecer planos tendo em conta as ligações (**Algoritmo EstabeleceLigações23**)

Se opção 2

Retirar os transportes principais que não funcionam no dia especificado pelo utilizador (**Algoritmo ApagaRegistosFunçãoPDF**)

Se não se retiraram todos os transportes principais

Retirar os transportes principais cuja hora de partida seja anterior à hora especificada pelo utilizador (**Algoritmo**

ApagaRegistosHoraUtilizador)

Se não se retiraram todos os transportes principais e se existirem transportes secundários

Estabelecer planos tendo em conta as ligações (**Algoritmo EstabeleceLigações23**)

/* Fim do algoritmo **ApagaRegistosFunçãoData3**

*/

O algoritmo **ApagaRegistosFunçãoData4** não é mais do que a adaptação dos algoritmos **ApagaRegistosFunçãoData2** e **ApagaRegistos-FunçãoData3** para a situação 4.

/* Algoritmo ApagaRegistosFunçãoData4

***/**

Se opção 1

Retirar os transportes secundários posteriores cuja hora de chegada seja posterior à hora especificada pelo utilizador (**Algoritmo ApagaRegistosHoraUtilizador**)

Se não se retiraram todos os transportes secundários posteriores e se existem transportes principais e transportes secundários anteriores

Estabelecer planos tendo em conta as ligações (**Algoritmo EstabeleceLigações4**)

Se opção 2

Retirar os transportes secundários anteriores que não funcionam no dia especificado pelo utilizador (**Algoritmo ApagaRegistosFunçãoPDF**)

Se não se retiraram todos os transportes secundários anteriores

Retirar os transportes secundários anteriores cuja hora de partida seja anterior à hora especificada pelo utilizador (**Algoritmo ApagaRegistosHoraUtilizador**)

Se não se retiraram todos os transportes secundários anteriores e se existem transportes principais e transportes secundários posteriores

Estabelecer planos tendo em conta as ligações (**Algoritmo EstabeleceLigações4**)

/* Fim do algoritmo ApagaRegistosFunçãoData4

***/**

O algoritmo **EstabeleceLigações23** constroi a matriz de decisão utilizada pela técnica de decisão multi-atributo (método da Pesagem Simples Aditiva) que permite determinar os melhores planos de transportes. Este algoritmo determina os planos de transportes para as situações 2 e 3 e utiliza o algoritmo **EstabelecePlanos23_1** e **EstabelecePlanos23_2**, respectivamente, para a opção 1 e opção 2.

/* Algoritmo EstabeleceLigações23

***/**

Se opção 1

Estabelece planos tendo em conta a situação e a opção (**Algoritmo EstabelecePlanos23_1**)

Se opção 2

Estabelece planos tendo em conta a situação e a opção (**Algoritmo EstabelecePlanos23_2**)

/* Fim do algoritmo EstabeleceLigações23

*/

O algoritmo **EstabeleceLigações4** também constroi a matriz de decisão utilizada pelo Método da Pesagem Simples Aditiva e utiliza os algoritmos **EstabelecePlanos4_1** e **EstabelecePlanos4_2**, para estabelecer planos de transportes referentes à situação 4, quando a opção indicada pelo utilizador for a 1 ou a 2, respectivamente.

/* **Algoritmo EstabeleceLigações4**

*/

Se opção 1

Estabelece planos tendo em conta a situação e a opção (**Algoritmo EstabelecePlanos4_1**)

Se opção 2

Estabelece planos tendo em conta a situação e a opção (**Algoritmo EstabelecePlanos4_2**)

/* Fim do algoritmo **EstabeleceLigações4**

*/

O algoritmo **EstabelecePlanos23_2** verifica primeiro se o transporte principal e o transporte secundário em análise são o mesmo transporte, rejeitando-os se tal acontecer. Como este algoritmo é utilizado, quer para a situação 2, quer para a situação 3, em vez de surgirem as designações de transporte principal e de transporte secundário, surgem as de 1º transporte e 2º transporte, que para a situação 2 são, respectivamente, o transporte secundário e o transporte principal e para a situação 3 o transporte principal e o transporte secundário. O algoritmo verifica também se a distância entre a paragem de chegada do 1º transporte e a paragem de partida do 2º transporte é superior a 500 metros e no caso de tal acontecer, o plano em questão é rejeitado. Considerou-se esta distância como a máxima admissível, para ser percorrida a pé, com ou sem bagagem. Neste algoritmo, estabelecem-se os planos partindo do 1º transporte e em função deste, determina-se o 2º.

O algoritmo **EstabelecePlanos23_2** verifica se o 1º transporte parte e chega no mesmo dia. Se tal acontecer, verifica se a hora de partida do 2º transporte é superior à hora de chegada do 1º transporte. Em caso afirmativo, o 2º transporte parte no dia especificado pelo utilizador e portanto é necessário verificar, utilizando o algoritmo **VerificaAdiciona23_2**, se

funciona nesse dia, se o utilizador tem tempo suficiente para efectuar o transbordo e se não tem de esperar mais do que 5 horas, por forma a poder ser considerado num plano de transportes. Se pelo contrário, a hora de partida do 2º transporte é inferior à hora de chegada do 1º transporte, considera-se que o 2º transporte será utilizado no dia seguinte, desde que o utilizador não tenha que esperar mais do que 5 horas. Por outro lado, se o 1º transporte chegar no dia seguinte, deve-se verificar também se a hora de partida do 2º transporte é superior à hora de chegada do 1º transporte, e no caso de tal acontecer, analisar se o 2º transporte funciona no dia posterior ao dia especificado, por forma a poder ser considerado num plano de transportes.

/* **Algoritmo EstabelecePlanos23_2**

*/

Verificar se o transporte principal e o transporte secundário são o mesmo transporte (**Heurística H5**)

Se forem diferentes

Determinar a distância entre a paragem de chegada do 1º transp. e a paragem de partida do 2º transp. (**Algoritmo VerificaDistância**)

Verificar se a distância é inferior a 500 metros

Se sim

Verificar se o 1º transp. parte e chega no mesmo dia (**Algoritmo VerificaParteChegaMesmoDia**)

Se sim (o 1º transp. chega dia especificado)

Verificar se a hora de partida do 2º transp. é superior à hora de chegada do 1º transp. (**Algoritmo HoraSuperiorTransportes**)

Se sim (o 2º transp. parte no dia especificado)

Verificar e adicionar, tendo em conta que Escolha1 é igual a 1 e Escolha2 é igual a 1 (**Algoritmo VerificaAdiciona23_2**)

Se não (o 2º transp. parte dia posterior)

Verificar e adicionar, tendo em conta que Escolha1 é igual a 2 e Escolha2 é igual a 2 (**Algoritmo VerificaAdiciona23_2**)

Se não (o 1º transp. chega dia posterior)

Verificar se a hora de partida do 2º transp. é superior à hora de chegada do 1º transp. (**Algoritmo HoraSuperiorTransportes**)

Se sim (o 2º transp. parte dia posterior)

Verificar e adicionar, tendo em conta que Escolha1 é igual a 2 e Escolha2 é igual a 2 (**Algoritmo VerificaAdiciona23_2**)

Se não

Não considerar

Se não

Não considerar

Se forem iguais

Não considerar

/* Fim do algoritmo **EstabelecePlanos23_2**

*/

Para o algoritmo **EstabelecePlanos23_1**, o raciocínio é idêntico, embora na ordem inversa ao anterior; a diferença reside no facto de se analisar primeiro o 2º transporte e só depois, e em função deste, o 1º. Este algoritmo estabelece planos considerando que, ou o 1º e o 2º transportes partem no mesmo dia, seja ele o dia especificado pelo utilizador, seja o dia anterior, ou o 1º transporte parte no dia anterior e o 2º no dia especificado.

/* **Algoritmo EstabelecePlanos23_1**

*/

/* Para a Situação 2, o 1º transporte é o transporte secundário e o 2º transporte é o transporte principal. Para a Situação 3 é ao contrário.

*/

Verificar se o transporte principal e o transporte secundário são o mesmo transporte (**Heurística H5**)

Se forem diferentes

Determinar a distância entre a paragem de chegada do 1º transp. e a paragem de partida do 2º transp. (**Algoritmo VerificaDistância**)

Verificar se a distância é inferior a 500 metros

Se sim

Verificar se o 2º transp. parte e chega no mesmo dia (**Algoritmo VerificaParteChegaMesmoDia**)

Se sim (o 2º transp. parte dia especificado)

Verificar se o 2º transp. funciona no dia especificado (**Algoritmo**

VerificaPDF)

Se sim

Verificar se a hora de chegada do 1º transp. é inferior à hora de partida do 2º transp. (**Algoritmo HoralInferiorTransportes**)

Se sim (o 1º transp. chega dia especificado)

Verificar se o 1º transp. parte num dia diferente ao que chega

(**Algoritmo**

VerificaParteChegaMesmoDia)

Se sim (o 1º transp. parte dia anterior)

Verificar e adicionar, tendo em conta que Escolha1 é igual a 2 e

Escolha2 é igual a 2 (**Algoritmo VerificaAdiciona23_1**)

Se não (o 1º transp. parte dia especificado)

Verificar e adicionar, tendo em conta que Escolha1 é igual a 1 e

Escolha2 é igual a 1 (**Algoritmo VerificaAdiciona23_1**)

Se não (o 1º transp. chega dia anterior)

Verificar se o 1º transp. parte num dia diferente ao que chega

(**Algoritmo**

VerificaParteChegaMesmoDia)

Se sim (o 1º transp. parte dia anterior ao anterior)

Não considerar

Se não (o 1º transp. parte dia anterior)

Verificar e adicionar, tendo em conta que Escolha1 é igual a 2 e

Escolha2 é igual a 2 (**Algoritmo VerificaAdiciona23_1**)

Se não

Não considerar

Se não (o 2º transp. parte dia anterior)

```

    Verificar se o 2º transp. funciona dia anterior (Algoritmo VerificaPDF)
    Se sim
        Verificar se a hora de chegada do 1º transp. é inferior à hora de partida
do      2º transp. (Algoritmo HoralInferiorTransportes)
        Se sim (o 1º transp. chega dia anterior)
            Verificar se o 1º transp. parte num dia diferente ao que chega
(Algoritmo      VerificaParteChegaMesmoDia)
            Se sim
                Não considerar
            Se não (o 1º transp. parte dia anterior)
                Verificar e adicionar, tendo em conta que Escolha1 é igual a 2 e
                    Escolha2 é igual a 3 (Algoritmo VerificaAdiciona23_1)
            Se não
                Não considerar
        Se não
            Não considerar
    Se não
        Não considerar
Se forem iguais
    Não considerar
/* Fim do algoritmo EstabelecePlanos23_1
*/
*/
```

O algoritmo **VerificaAdiciona23_2** recebe como parâmetros de entrada **Escolha1** e **Escolha2**, que estão definidos no algoritmo **EstabelecePlanos23_2**. Aquele algoritmo verifica se o tempo de espera entre o 1º e o 2º transporte não excede as 5 horas e se o utilizador tem tempo suficiente para efectuar o transbordo. Se tal acontecer, verifica em função do valor de **Escolha1**, se o 2º transporte funciona no dia especificado ou no dia posterior. De seguida e em função dos resultados das verificações atrás descritas e do valor de **Escolha2**, estabelece os planos de transportes.

Para se ter a certeza de que o utilizador terá tempo suficiente para apanhar o transporte seguinte do plano de transportes, decidiu-se considerar para os aviões um tempo de espera de 60 minutos e para os restantes transportes um tempo de 30 minutos. Estes tempos foram determinados empiricamente, no entanto, tal como enunciado na heurística H3 na secção 4.3, dependem do tipo de transporte que irá ser utilizado, do facto do utilizador ter ou não comprado antecipadamente todos os bilhetes necessários e do dia do ano em que se pretende realizar a viagem.

/* **Algoritmo VerificaAdiciona23_2**

*/

Determinar o tempo de espera entre o 1º e o 2º transp. (**Algoritmo DeterminaTempoEspera**)

Se for inferior a 5 horas e se for superior ou igual ao tempo necessário para efectuar o transbordo (**Heurística H3**)

Se Escolha1 for 1

Verificar se o 2º transp. funciona no dia especificado (**Algoritmo VerificaPDF**)

Se sim

Verifica

Se não

Não verifica

Se Escolha1 for 2

Verificar se o 2º transp. funciona no dia posterior (**Algoritmo VerificaPDF**)

Se sim

Verifica

Se não

Não verifica

Se verifica

Se Escolha2 for 1

Considerar o plano (o 1º transp. parte no dia especificado e o 2º transp. parte no dia especificado) (**Algoritmo AdicionaRegistosTPC23**)

Se Escolha2 for 2

Considerar o plano (o 1º transp. parte no dia especificado e o 2º transp. parte no dia posterior) (**Algoritmo AdicionaRegistosTPC23**)

Se não verifica

Não considerar

Se for superior a 5 horas ou se for inferior ao tempo necessário para efectuar o transbordo

Não considerar

/* Fim do algoritmo **VerificaAdiciona23_2**

*/

O algoritmo **VerificaAdiciona23_1** é utilizado pelo algoritmo **EstabelecePlanos23_1** e segue o mesmo raciocínio aplicado ao algoritmo **VerificaAdiciona23_2**.

/* **Algoritmo VerificaAdiciona23_1**

*/

Determinar o tempo de espera entre o 1º e o 2º transp. (**Algoritmo DeterminaTempoEspera**)

Se for inferior a 5 horas e se for superior ou igual ao tempo necessário para efectuar o transbordo (**Heurística H3**)

Se Escolha1 for 1

Verificar se o 1º transp. funciona no dia especificado (**Algoritmo VerificaPDF**)

Se sim

Verifica

Se não

```

    Não verifica
Se Escolha1 for 2
    Verificar se o 1º transp. funciona no dia anterior (Algoritmo VerificaPDF)
    Se sim
        Verifica
    Se não
        Não verifica

Se verifica
    Se Escolha2 for 1
        Considerar o plano (o 1º transp. parte no dia especificado e o 2º transp.
parte no        dia especificado) (Algoritmo AdicionaRegistosTPC23)
    Se Escolha2 for 2
        Considerar o plano (o 1º transp. parte no dia anterior e o 2º transp. parte
no            dia especificado) (Algoritmo AdicionaRegistosTPC23)
    Se Escolha2 for 3
        Considerar o plano (o 1º transp. parte no dia anterior e o 2º transp. parte
no            dia anterior) (Algoritmo AdicionaRegistosTPC23)
Se não verifica
    Não considerar
Se for superior a 5 horas ou se for inferior ao tempo necessário para efectuar o
transbordo
    Não considerar
/* Fim do algoritmo VerificaAdiciona23_1
*/
*/
```

Os algoritmos **EstabelecePlanos4_2** e **EstabelecePlanos4_1** são utilizados para estabelecer planos de transportes para a situação 4, quando a opção escolhida pelo utilizador é a 2 e a 1, respectivamente. A sequência da procura e as verificações realizadas seguem a mesma linha de orientação seguida nos algoritmos **EstabelecePlanos23_1** e **EstabelecePlanos23_2**. A diferença reside no facto de na situação 4 se fazer a procura de 3 transportes (principal+secundário anterior+secundário posterior), em vez de apenas 2 como acontece para as situações 2 e 3. Decidiu-se não explicar detalhadamente os algoritmos referentes à situação 4, uma vez que estes são explícitos e seguem o mesmo raciocínio dos algoritmos referentes às situações 2 e 3.

```
/* Algoritmo EstabelecePlanos4_2
```

```
*/
```

```
Verificar se o transp. princ. e o transp. sec. ant. são iguais ou se o transp. princ. e o
transp. sec. post. são iguais (Heurística H5)
```

```
Se ambos forem diferentes
```

```
    Determinar a distância entre a paragem de chegada do transp. sec. ant. e a
paragem de partida do transp. princ. e a distância entre a paragem de chegada
do transp. princ. e a paragem de partida do transp. sec. post. (Algoritmo
```

VerificaDistância)

Se forem ambas inferiores a 500 metros

Verificar se o transp. sec. ant. parte e chega no mesmo dia **(Algoritmo VerificaParteChegaMesmoDia)**

Se sim

Verificar se a hora de partida do transp. princ. é superior à hora de chegada do transp. sec. ant. **(Algoritmo HoraSuperiorTransportes)**

Se sim

Determinar o tempo de espera entre o transp. sec. ant. e o transp. princ. **(Algoritmo DeterminaTempoEspera)**

Se for inferior a 5 horas e se for superior ou igual ao tempo necessário para efectuar o transbordo **(Heurística H3)**

Verificar se o transp. princ. funciona no dia especificado **(Algoritmo VerificaPDF)**

Se sim

Verificar se o transp. princ. parte e chega no mesmo dia **(Algoritmo VerificaParteChegaMesmoDia)**

Se sim

Verificar se a hora de partida do transp. sec. post. é superior à hora de chegada do transp. princ. **(Algoritmo HoraSuperiorTransportes)**

Se sim

Verificar e adicionar, tendo em conta que Escolha1 é igual a Escolha2 é igual a 1 **(Algoritmo VerificaAdiciona4_2)**

Se não

Verificar e adicionar, tendo em conta que Escolha1 é igual a Escolha2 é igual a 2 **(Algoritmo VerificaAdiciona4_2)**

Se não

Verificar se a hora de partida do transp. sec. post. é superior à hora de chegada do transp. princ. **(Algoritmo HoraSuperiorTransportes)**

Se sim

Verificar e adicionar, tendo em conta que Escolha1 é igual a Escolha2 é igual a 2 **(Algoritmo VerificaAdiciona4_2)**

Se não

Não considerar

Se não

Não considerar

Se for superior a 5 horas ou se for inferior ao tempo necessário para efectuar o transbordo

Não considerar

Se não

Determinar o tempo de espera entre o transp. sec. ant. e o transp. princ. **(Algoritmo DeterminaTempoEspera)**

Se for inferior a 5 horas e se for superior ou igual ao tempo necessário para efectuar o transbordo **(Heurística H3)**

Verificar se o transp. princ. funciona no dia posterior **(Algoritmo VerificaPDF)**

Se sim

Verificar se o transp. princ. parte e chega no mesmo dia **(Algoritmo VerificaParteChegaMesmoDia)**

Se sim
Verificar se a hora de partida do transp. sec. post. é superior à hora de chegada do transp. princ. **(Algoritmo HoraSuperiorTransportes)**
Se sim
Verificar e adicionar, tendo em conta que Escolha1 é igual a 2 e Escolha2 é igual a 3 **(Algoritmo VerificaAdiciona4_2)**
Se não
Não considerar
Se não
Não considerar
Se não
Não considerar
Se for superior a 5 horas ou se for inferior ao tempo necessário para efectuar o transbordo
Não considerar
Se não
Verificar se a hora de partida do transp. princ. é superior à hora de chegada do transp. sec. ant. **(Algoritmo HoraSuperiorTransportes)**
Se sim
Determinar o tempo de espera entre o transp. sec. ant. e o transp. princ. **(Algoritmo DeterminaTempoEspera)**
Se for inferior a 5 horas e se for superior ou igual ao tempo necessário para efectuar o transbordo **(Heurística H3)**
Verificar se o transp. princ. funciona no dia posterior **(Algoritmo VerificaPDF)**
Se sim
Verificar se o transp. princ. parte e chega no mesmo dia **(Algoritmo VerificaParteChegaMesmoDia)**
Se sim
Verificar se a hora de partida do transp. sec. post. é superior à hora de chegada do transp. princ. **(Heurística H3)(Algoritmo HoraSuperiorTransportes)**
Se sim
Verificar e adicionar, tendo em conta que Escolha1 é igual a 2 e Escolha2 é igual a 3 **(Algoritmo VerificaAdiciona4_2)**
Se não
Não considerar
Se não
Não considerar
Se não
Não considerar
Se for superior a 5 horas ou se for inferior ao tempo necessário para efectuar o transbordo
Não considerar
Se não
Não considerar
Se pelo menos uma das distâncias for inferior a 500 metros
Não considerar
Se pelo menos um dos transp. sec. for igual ao transp. princ.
Não considerar
/* Fim do algoritmo **EstabelecePlanos4_2**

*/

/* **Algoritmo EstabelecePlanos4_1**

*/

Verificar se o transp. princ. e o transp. sec. ant. são iguais ou se o transp. princ. e o transp. sec. post. são iguais (**Heurística H5**)

Se ambos forem diferentes

Determinar a distância entre a paragem de chegada do transp. sec. ant. e a paragem de partida do transp. princ. e a distância entre a paragem de chegada do transp. princ. e a paragem de partida do transp. sec. post. (**Algoritmo VerificaDistância**)

Se forem ambas inferiores a 500 metros

Verificar se o transp. sec. post. parte e chega no mesmo dia (**Algoritmo VerificaParteChegaMesmoDia**)

Se sim

Verificar se o transp. sec. post. funciona no dia especificado (**Algoritmo VerificaPDF**)

Se sim

Verificar se a hora de chegada do transp. princ. é inferior à hora de partida do transp. sec. post. (**Algoritmo HoralInferiorTransportes**)

Se sim

Determinar o tempo de espera entre o transp. princ. e o transp. sec. post. (**Algoritmo DeterminaTempoEspera**)

Se for inferior a 5 horas e se for superior ou igual ao tempo necessário para efectuar o transbordo (**Heurística H3**)

Verificar se o transp. princ. parte num dia diferente ao que chega (**Algoritmo VerificaParteChegaMesmoDia**)

Se sim

Verificar se o transp. princ. funciona no dia anterior (**Algoritmo VerificaPDF**)

Se sim

Verificar se a hora de chegada do transp. sec. ant. é inferior à hora de partida do transp. princ. (**Algoritmo HoralInferiorTransportes**)

Se sim

Verificar e adicionar, tendo em conta que Escolha1 é igual a 2 (**Algoritmo VerificaeAdiciona4_1_2**)

Se não

Não considerar

Se não

Não considerar

Se não

Verificar se o transp. princ. funciona no dia especificado (**Algoritmo VerificaPDF**)

Se sim

Verificar se a hora de chegada do transp. sec. ant. é inferior à hora de partida do transp. princ. (**Heurística H3**)(**Algoritmo HoralInferiorTransportes**)

Se sim

Verificar e adicionar (**Algoritmo VerificaeAdiciona4_1_1**)

Se não


```

        Verificar se a hora de chegada do transp. sec. ant. é inferior
        à hora de partida do transp. princ. (Algoritmo
        HoraInferiorTransportes)
        Se sim
            Verificar e adicionar, tendo em conta que Escolha1 é
            (Algoritmo VerificaeAdiciona4_1_2)
        Se não
            Não considerar
        Se não
            Não considerar
            Se for superior a 5 horas ou se for inferior ao tempo necessário para
            efectuar o transbordo
        Se não
            Não considerar
        Se não
            Não considerar
        Se não
            Não considerar
        Se pelo menos uma das distâncias for inferior a 500 metros
            Não considerar
        Se pelo menos um dos transp. sec. for igual ao transp. princ.
            Não considerar
    /* Fim do algoritmo EstabelecePlanos4_1
    */

```

Os algoritmos **VerificaeAdiciona4_1_1** e **VerificaeAdiciona4_1_2** são utilizados no algoritmo **EstabelecePlanos4_1**. Estes algoritmos verificam se o tempo de espera entre os transportes é inferior a 5 horas, se o utilizador tem tempo suficiente para efectuar o transbordo e se os transportes funcionam em determinado dia.

```

/* Algoritmo VerificaeAdiciona4_1_1

```

```

*/
Determinar o tempo de espera entre o transp. sec. ant. e o transp. princ.
(Algoritmo DeterminaTempoEspera)
Se for inferior a 5 horas e se for superior ou igual ao tempo necessário para
efectuar o transbordo (Heurística H3)
    Verificar se o transp. sec. ant. parte num dia diferente ao que chega (Algoritmo
    VerificaParteChegaMesmoDia)
    Se sim
        Verificar se o transp. sec. ant. funciona no dia anterior (Algoritmo
        VerificaPDF)
        Se sim
            Considerar o plano (o transp. sec. ant. parte no dia anterior, o transp princ.
            parte no dia especificado e o transp. sec. post. parte no dia especificado)
            (Algoritmo AdicionaRegistosTPC4)
        Se não
            Não considerar
    Se não

```

Verificar se o transp. sec. ant. funciona no dia especificado (**Algoritmo VerificaPDF**)

Se sim

Considerar o plano (o transp. sec. ant. parte no dia especificado, o transp princ. parte no dia especificado e o transp. sec. post. parte no dia especificado) (**Algoritmo AdicionaRegistosTPC4**)

Se não

Não considerar

Se for superior a 5 horas ou se for inferior ao tempo necessário para efectuar o transbordo

Não considerar

/* Fim do algoritmo **VerificaAdiciona4_1_1**

*/

/* **Algoritmo VerificaAdiciona4_1_2**

*/

Determinar o tempo de espera entre o transp. sec. ant. e o transp. princ. (**Algoritmo DeterminaTempoEspera**)

Se for inferior a 5 horas e se for superior ou igual ao tempo necessário para efectuar o transbordo (**Heurística H3**)

Verificar se o transp. sec. ant. parte num dia diferente ao que chega (**Algoritmo VerificaParteChegaMesmoDia**)

Se sim

Não considerar

Se não

Verificar se o transp. sec. ant. funciona no dia anterior (**Algoritmo VerificaPDF**)

Se sim

Se Escolha1 for 1

Considerar o plano (o transp. sec. ant. parte no dia anterior, o transp princ. parte no dia especificado e o transp. sec. post. parte no dia especificado) (**Algoritmo AdicionaRegistosTPC4**)

Se Escolha1 for 2

Considerar o plano (o transp. sec. ant. parte no dia anterior, o transp princ. parte no dia anterior e o transp. sec. post. parte no dia especificado) (**Algoritmo AdicionaRegistosTPC4**)

Se Escolha1 for 3

Considerar o plano (o transp. sec. ant. parte no dia anterior, o transp princ. parte no dia anterior e o transp. sec. post. parte no dia anterior) (**Algoritmo AdicionaRegistosTPC4**)

Se não

Não considerar

Se for superior a 5 horas ou se for inferior ao tempo necessário para efectuar o transbordo

Não considerar

/* Fim do algoritmo **VerificaAdiciona4_1_2**

*/

O algoritmo **VerificaAdiciona4_2** é utilizado pelo algoritmo **EstabelecePlanos4_2**. Para além de verificar se o tempo de espera entre os

transportes é inferior a 5 horas e se o utilizador tem tempo suficiente para efectuar o transbordo, verifica, em função do parâmetro **Escolha1**, se o transporte secundário posterior funciona em determinado dia, estabelecendo os planos de transportes em função dos resultados destas verificações e do valor de **Escolha2**.

/* **Algoritmo VerificaeAdiciona4_2**

*/

Determinar o tempo de espera entre o transp. princ. e o transp. sec. post.
(Algoritmo DeterminaTempoEspera)

Se for inferior a 5 horas e se for superior ou igual ao tempo necessário para efectuar o transbordo **(Heurística H3)**

Se Escolha1 for 1

Verificar se o transp. sec. post. funciona no dia especificado **(Algoritmo VerificaPDF)**

Se sim

Verifica

Se não

Não verifica

Se Escolha1 for 2

Verificar se o transp. sec. post. funciona no dia posterior **(Algoritmo VerificaPDF)**

Se sim

Verifica

Se não

Não verifica

Se verifica

Se Escolha2 for 1

Considerar o plano (o transp. sec. ant. parte no dia especificado, o transp princ. parte no dia especificado e o transp. sec. post. parte no dia especificado) **(Algoritmo AdicionaRegistosTPC4)**

Se Escolha2 for 2

Considerar o plano (o transp. sec. ant. parte no dia especificado, o transp princ. parte no dia especificado e o transp. sec. post. parte no dia posterior) **(Algoritmo AdicionaRegistosTPC4)**

Se Escolha2 for 3

Considerar o plano (o transp. sec. ant. parte no dia especificado, o transp princ. parte no dia posterior e o transp. sec. post. parte no dia posterior) **(Algoritmo AdicionaRegistosTPC4)**

Se não verifica

Não considerar

Se for superior a 5 horas ou se for inferior ao tempo necessário para efectuar o transbordo

Não considerar

/* Fim do algoritmo **VerificaeAdiciona4_2**

*/

Os algoritmos **AdicionaRegistosTPC23** e **AdicionaRegistosTPC4** são utilizados, para as situações 2 e 3, e para a situação 4, respectivamente. Estes dois algoritmos determinam o tempo total da viagem, o preço total e as características (qualidade) de cada plano de transportes.

O tempo total da viagem para a situação 2, corresponde à diferença horária entre a hora de chegada do transporte principal e a hora de partida do transporte secundário. Para a situação 3 corresponde à diferença entre a hora de chegada do transporte secundário e a hora de partida do transporte principal. Por último, para a situação 4, é a diferença entre a hora de chegada do transporte secundário posterior e a hora de partida do transporte secundário anterior.

O preço total do plano de transportes corresponde à soma dos preços dos bilhetes dos transportes que compõem o plano de transportes.

É considerada como qualidade do plano de transportes, o número de características do transporte principal. Considerou-se a qualidade do plano de transportes desta forma, uma vez que, em princípio, o tempo gasto no transporte principal é superior ao gasto nos transportes secundários, pelo que o transporte principal é considerado mais importante e o que define a qualidade do plano de transportes.

A situação 1 não é referida neste algoritmo, porque o tempo, o preço e a qualidade do plano de transportes são calculados inicialmente aquando da criação da matriz de decisão.

/* **Algoritmo AdicionaRegistosTPC23**

*/

Se Situação for

Situação 2

O Tempo total de duração do plano de transportes é a diferença entre a hora de chegada do transporte principal e a hora de partida do transporte secundário

Situação 3

O Tempo total de duração do plano de transportes é a diferença entre a hora de chegada do transporte secundário e a hora de partida do transporte principal

O Preço do plano de transportes é o preço do bilhete do transporte principal mais o preço do bilhete do transporte secundário

A Qualidade do plano de transportes (número de características) é o número de características do transporte principal

/* Fim do algoritmo **AdicionaRegistosTPC23**

*/

/* **Algoritmo AdicionaRegistosTPC4**

*/

Designaremos o transporte secundário da ligação secundária anterior por transporte secundário anterior

Designaremos o transporte secundário da ligação secundária posterior por transporte secundário posterior

O Tempo total de duração do plano de transportes é a diferença entre a hora de chegada do transporte secundário posterior e a hora de partida do transporte secundário anterior

O Preço do plano de transportes é a soma dos preços dos seguintes bilhetes: bilhete do transporte secundário anterior, bilhete do transporte principal e bilhete do transporte secundário posterior

A Qualidade do plano de transportes (número de características) é o número de características do transporte principal

/* Fim do algoritmo **AdicionaRegistosTPC4**

*/

Alguns algoritmos poderiam ser optimizados, tornando-os recursivos, no entanto, por uma questão de simplicidade na explicação e de verificação de erros, são construídos numa forma, que apresenta de modo completo e fácil a procura sequencial de transportes.

4.2.2 Algoritmos que determinam o plano de transporte para os táxis

Decidiu-se indicar os táxis apenas para efectuar ligações secundárias, uma vez que às ligações principais correspondem, geralmente percursos longos, para os quais não é economicamente viável utilizar este tipo de transporte.

Para estabelecer a ligação secundária da situação 2 e a ligação secundária anterior da situação 4, são fornecidos ao utilizador algumas das moradas das praças de táxis da localidade Origem e alguns dos números de telefone das centrais de táxis. O destino desta ligação é a paragem de partida do transporte principal, cujo morada é também facultada ao utilizador.

Em relação à ligação secundária da situação 3 e à ligação secundária posterior da situação 4, são fornecidos alguns números de telefone das centrais de táxis da localidade destino da ligação principal e algumas das moradas das praças de táxis daquela localidade. Contudo, nas paragens dos transportes principais é usual existirem praças de táxis.

O preço fornecido ao utilizador é um preço mínimo, visto que a distância, a partir da qual este é calculado, é a mínima possível. Esta distância é calculada em função das coordenadas topográficas da origem e do destino da ligação, a partir das quais se obtém o comprimento do segmento de recta que os une. Além disso, o preço por quilómetro é o preço mínimo praticado pelos taxistas no local origem da ligação, ou seja, não é determinado em função do dia da semana nem da hora em que o utilizador pretende viajar, devido às incertezas que existem relativamente ao momento em que o transporte é utilizado. O preço calculado não inclui também o preço cobrado pelos taxistas quando estes são requisitados pelo utilizador via telefone. É indicado ainda, em separado, o preço mínimo de utilização (preço da bandeirada) e o preço cobrado pelo transporte de bagagem.

/* **Algoritmo DeterminaTáxis**

*/

Se Situação 2 ou Situação 4

Determinar as moradas de algumas praças de táxis de Origem e alguns dos números de telefone das centrais de táxis desta localidade

Determinar a morada da paragem de partida do transporte principal

Determinar a distância em quilómetros entre a localidade Origem e a paragem de partida do transporte principal (**Algoritmo DeterminaDistância**)

Determinar o preço mínimo por quilómetro, o preço pelo transporte de bagagem e o preço da bandeirada cobrados pelos taxistas da localidade Origem

Multiplicar a distância pelo preço mínimo por quilómetro e somar a esta quantia o preço da bandeirada

Se Situação 3 ou Situação 4

Determinar as moradas de algumas praças de táxis da localidade destino da ligação principal e alguns dos números de telefone das centrais de táxis desta localidade

Determinar a distância em quilómetros entre a localidade destino da ligação principal e Destino (**Algoritmo DeterminaDistância**)

Determinar o preço mínimo por quilómetro, o preço pelo transporte de bagagem e o preço da bandeirada cobrados pelos taxistas da localidade destino da ligação principal

Multiplicar a distância pelo preço mínimo por quilómetro e somar a esta quantia o preço da bandeirada

/* Fim do algoritmo **DeterminaTáxis**

*/

4.3 Heurísticas

As heurísticas que a seguir se apresentam, não são mais do que regras que foram consideradas na determinação dos planos de transportes possíveis e dos melhores planos de transportes. Algumas destas heurísticas foram utilizadas na definição dos algoritmos (heurísticas H1, H2, H3, H4 e H5), outras não o foram porque dizem respeito a uma caracterização diferente do problema (heurísticas H6, H7, H8 e H9).

Como já foi referido, os melhores planos de transportes para ir de Origem a Destino, podem não ser planos constituídos por apenas transportes que partam de Origem e cheguem a Destino, mas por transportes que partam de locais próximos a Origem e/ou cheguem a locais próximos a Destino (planos que correspondem às situações 2, 3 e 4).

Heurística H1:

Os melhores planos de transportes para ir de Origem a Destino, podem não ser constituídos por apenas transportes que partam de Origem e cheguem a Destino, mas por transportes que partam de locais próximos a Origem e/ou que cheguem a locais próximos a Destino.

Na procura dos locais próximos a Origem e a Destino, deve-se dar preferência às cidades em detrimento das vilas, uma vez que é mais provável que aí se encontrem mais e melhores transportes.

Heurística H2:

Deve-se dar preferência às cidades e só depois às vilas, uma vez que é mais provável que aí existam mais e melhores transportes.

Nos planos em que se verificam transbordos (mudança de transporte) é necessário verificar se o utilizador tem tempo suficiente para apanhar o transporte seguinte. Este tempo varia com o tipo de transporte (geralmente o “check-in” nos aviões demora mais tempo do que para qualquer outro tipo de transporte), com o período do ano em que se realiza a viagem (a

afluência de passageiros é maior no verão, na quadra natalícia, etc.) e com o facto do utilizador ter ou não comprado todos os bilhetes necessários.

Heurística H3:

Nos planos com transbordos, deve-se verificar se o utilizador tem tempo suficiente para apanhar o transporte seguinte. Este tempo varia com o tipo de transporte, o período do ano em que se pretende realizar a viagem e do facto do utilizador ter comprado, ou não, todos os bilhetes necessários, antecipadamente.

As duas heurísticas seguintes, reflectem duas situações a evitar na determinação dos planos de transportes.

Heurística H4:

Evitar planos de transportes em que, na situação 2, Origem seja uma paragem do transporte principal, na situação 3, Destino seja uma paragem do transporte principal e por último, na situação 4, os dois casos anteriores.

Heurística H5:

Não se devem estabelecer planos de transportes em que o transporte principal e o(s) secundários(s) sejam o mesmo transporte.

Outra forma de caracterizar o problema dos transportes, seria, em vez do utilizador indicar apenas o local Origem e o local Destino, que correspondem a cidades, vilas ou aldeias, indicar também as moradas ou as zonas do local Origem e do local Destino de onde pretende partir e chegar, respectivamente. Isto implicaria, que para além de se procurar os planos de transportes para ir de Origem a Destino, ter-se-ia que procurar o(s) transporte(s) para ir da morada ou zona indicada em Origem à paragem de partida em Origem e procurar o(s) transporte(s) para ir da paragem de chegada em Destino à morada ou zona indicada em Destino. Geralmente nestas situações, utilizam-se os transportes urbanos que circulam dentro das cidades e das vilas e os táxis. Por esta razão, designaremos estes transportes por transportes urbanos-locais e ao conjunto dos melhores planos e dos transportes urbanos-locais correspondentes, por planos de transporte total.

Uma das heurísticas mais evidentes, é que se devem procurar primeiro os melhores planos de transportes e só depois e em função destes é que se

devem procurar os melhores transportes urbanos-locais, uma vez que, os primeiros, são os transportes que consomem a maior parte do tempo e do dinheiro do total da viagem.

Heurística H6:

Devem-se procurar primeiro os melhores planos de transportes e só depois os transportes urbanos-locais correspondentes, uma vez que, os primeiros, são os transportes que geralmente consomem a maior parte do tempo e do dinheiro do total da viagem.

Nada garante no entanto, que o melhor plano de transportes e os melhores transportes urbanos-locais correspondentes constituam o melhor plano total. Deve-se considerar portanto, não só o melhor plano, mas os outros que também ficaram bem pontuados, por forma a verificar se algum deles, juntamente com os transportes urbanos-locais correspondentes, não constituem um melhor plano total.

Heurística H7:

O melhor plano de transportes e os melhores transportes urbanos-locais para este plano de transportes, podem não constituir o melhor plano total de transportes. Deve-se escolher, pelo menos, um outro plano de transportes (o segundo melhor), e os seus melhores transportes urbanos-locais, para verificar se no conjunto, não constituirão um plano total melhor.

Como nesta caracterização procuram-se os transportes urbanos-locais, é conveniente saber a quantidade de bagagem que o utilizador pretende transportar. Este facto influenciará a possibilidade de o utilizador poder fazer algum percurso a pé ou de utilizar como transporte urbano-local, o metro, o autocarro, etc. Por outro lado, quando a bagagem é considerada excessiva, deve-se indicar o táxi como transporte urbano-local.

Heurística H8:

Verificar a quantidade de bagagem que o utilizador pretende transportar, porque este facto influenciará a possibilidade de fazer algum percurso a pé, ou de utilizar transportes urbanos-locais. Quando a bagagem é considerada excessiva (em volume ou em peso), deve-se indicar o táxi como transporte urbano-local.

A quantificação de excessivo, também depende da idade do utilizador e de este ter ou não alguma deficiência motora.

Heurística H9:

Se a condição física do utilizador não for boa, deve-se indicar o táxi como transporte urbano-local.

4.4 Técnica de decisão multi-atributo

4.4.1 Introdução

Para determinar os melhores planos de transportes, decidiu-se utilizar as técnicas de decisão multi-atributo. Estas técnicas permitem resolver problemas de selecção de alternativas em número finito, em que os critérios são definidos pelos atributos ([8]).

De entre os vários métodos disponíveis, adoptou-se o Método da **Pesagem Simples Aditiva**, também chamado método das **Ponderações**.

A base de desenvolvimento deste método é uma matriz de avaliação, em que estão explicitadas as alternativas e os critérios. Cada elemento da matriz representa a classificação parcial de cada alternativa em função de cada critério de avaliação.

A matriz de decisão de m atributos e n alternativas representa-se por:

$$\begin{array}{c}
 \textit{A} \\
 \textit{L} \\
 \textit{T} \\
 \textit{E} \\
 \textit{R} \\
 \textit{N} \\
 \textit{A} \\
 \textit{T} \\
 \textit{I} \\
 \textit{V} \\
 \textit{A} \\
 \textit{S}
 \end{array}
 \begin{array}{c}
 \textit{A}1 \\
 \textit{A}2 \\
 \cdot \\
 \cdot \\
 \cdot \\
 \cdot \\
 \textit{A}n
 \end{array}
 \begin{array}{c}
 \textit{C}11 \dots \textit{C}1m \\
 \textit{C}21 \dots \textit{C}2m \\
 \cdot \\
 \cdot \\
 \cdot \\
 \cdot \\
 \textit{C}n1 \dots \textit{C}nm
 \end{array}$$

em que C_{ij} representa a classificação parcial da alternativa i no critério j.

As classificações parciais podem ser qualitativas ou quantitativas. As qualitativas são transformadas em unidades quantitativas.

A classificação parcial dos atributos é ponderada com pesos que representam o seu valor ou prioridade em função dos outros critérios.

Estes pesos formam o seguinte vector:

$$W = (w_1, w_2, \dots, w_j) \quad \text{em que} \quad \sum_j w_j = 1$$

ou seja o somatório de todos os pesos é igual à unidade.

Para obter a classificação final das alternativas é utilizado o método **da Pesagem Simples Aditiva** ou das **Ponderações**, que consiste na multiplicação da matriz de decisão pelo vector W , por forma a obtermos o vector S que representa o resultado da avaliação de cada alternativa.

Matricialmente fica:

$$\begin{bmatrix} S_1 \\ S_2 \\ \cdot \\ \cdot \\ \cdot \\ S_i \end{bmatrix} = \begin{bmatrix} C_{11} \dots C_{1j} \\ C_{21} \dots C_{2j} \\ \cdot \\ \cdot \\ \cdot \\ C_{i1} \dots C_{ij} \end{bmatrix} \begin{bmatrix} W_1 \\ W_2 \\ \cdot \\ \cdot \\ \cdot \\ W_j \end{bmatrix}$$

Finalmente, a melhor alternativa corresponde ao maior valor do vector S .

Seguidamente, são enunciadas as várias fases do processo de avaliação:

1. Definição dos objectivos gerais.
2. Identificação das alternativas.
3. Especificação do conjunto de critérios de avaliação e respectiva ordenação de preferências.
4. Determinação para cada alternativa, das classificações parciais correspondentes a cada critério.
5. Escolha com base num processo de avaliação.

Para compreender melhor esta última fase, é resolvido um caso prático ao mesmo tempo que se descrevem as suas várias etapas.

Em cada etapa, são referidos os nomes dos algoritmos que executam as suas operações. Alguns destes algoritmos estão listados na secção 4.5.

4.4.2 Caso prático

O caso prático em questão tem como objectivo determinar os dois melhores planos de transportes.

Considere-se que, finalizada a procura dos planos de transportes possíveis para ir de X a Y, obtêm-se os seguintes planos:

	Tempo (minutos)	Preço (escudos)	NºCaracterísticas
Plano 1	300	2000	5
Plano 2	270	2300	7
Plano 3	240	2500	5
Plano 4	320	3000	2

O quadro anterior mostra os tempos de duração, em minutos, dos vários planos, o(s) preço(s) do(s) bilhete(s) em escudos e o número de características dos transportes (existência de bar, WC, TV, vídeo, etc., que reflectem a qualidade dos transportes).

Com base no quadro anterior, a matriz de decisão de base fica:

	T	P	NºC
A1	300	2000	5
A2	270	2300	7
A3	240	2500	5
A4	320	3000	2

A notação utilizada é a seguinte:

A1 - Alternativa 1, que corresponde ao Plano 1.

A2 - Alternativa 2, que corresponde ao Plano 2.

A3 - Alternativa 3, que corresponde ao Plano 3.

A4 - Alternativa 4, que corresponde ao Plano 4.

T - Critério 1, que corresponde ao Tempo.

P - Critério 2, que corresponde ao Preço.

N^oC - Critério 3, que corresponde ao N^oCaracterísticas.

Na secção seguinte, são apresentadas juntamente com o caso prático, as várias fases referentes ao processo de avaliação.

4.4.2.1 Processo de avaliação

1) Filtragem

Nesta fase, são eliminadas as alternativas dominadas e as inadmissíveis, utilizando o método da Dominância e o método Conjuntivo, respectivamente.

a) Método da Dominância

Utilizando este método, eliminam-se as alternativas dominadas, ou seja, eliminam-se as alternativas que são superadas em um ou mais atributos por uma outra alternativa, desde que esta as iguale nos restantes atributos. Como resultado, poder-se-á eventualmente reduzir o número de alternativas.

O processo a seguir é o de comparar as duas primeiras alternativas, e caso uma domine a outra, elimina-se a dominada. A(s) não dominada(s) é (são) a seguir comparada(s) com a terceira alternativa e assim sucessivamente.

Ao analisarmos a matriz de decisão, verifica-se que a alternativa A4 é dominada por todas as outras alternativas, logo, é eliminada.

Verifica-se igualmente, que nenhuma alternativa está melhor classificada que todas as outras, em relação a todos os critérios, uma vez que se assim fosse, o problema estaria resolvido, dado que a escolha seria singular e teríamos determinado já a melhor solução.

A matriz de decisão fica reduzida a:

	T	P	NºC
A1	300	2000	5
A2	270	2300	7
A3	240	2500	5

b) Método Conjuntivo

Este método permite eliminar as alternativas inaceitáveis, ou seja, aquelas que possuem atributos com valores inadmissíveis, em relação à determinadas restrições.

No caso prático em estudo, não há nenhuma restrição aos valores dos atributos, o que leva a concluir que todas as alternativas são aceitáveis.

2) Ordenação de preferência

Esta fase é desenvolvida utilizando o método da Pesagem Simples Aditiva ou das Ponderações. Este método requer que os valores dos diversos atributos sejam numéricos e expressos em unidades normalizadas (adimensionais) de modo a poderem ser comparadas.

a) Quantificação dos atributos qualitativos

Nesta fase, utiliza-se uma escala (por exemplo, de 0 a 10 ou de 0 a 20) por forma a que determinada classificação qualitativa corresponda a um determinado intervalo de valores na escala escolhida.

Uma vez que, no caso prático em questão, não existe nenhum atributo qualitativo, não é necessário aplicar este método.

b) Normalização dos valores dos atributos

Nesta fase, faz-se a normalização das diferentes unidades, para torná-las adimensionais, por forma a facilitar a comparação dos atributos.

Esta normalização consiste na divisão de cada vector-linha pela respectiva norma euclidiana (módulo). (algoritmos **NormaEuclidiana** e **Normalização**)

Sendo C_{ij} , o elemento da linha i e da coluna j da matriz de decisão, o seu valor normalizado X_{ij} é obtido utilizando a seguinte equação:

$$X_{ij} = \frac{C_{ij}}{\sqrt{\sum_i C_{ij}^2}} \quad \begin{matrix} i = 1, \dots, n \\ j = 1, \dots, m \end{matrix}$$

ou seja, cada elemento da matriz decisão de base é dividido pela raiz quadrada do somatório dos quadrados de todos os elementos da mesma coluna (que correspondem ao mesmo atributo).

As normas euclidianas obtidas para o caso em estudo são:

$$|C1| = \sqrt{300^2 + 270^2 + 240^2} = 469.57$$

$$|C2| = 3942.08$$

$$|C3| = 9.95$$

A matriz de decisão normalizada fica:

	T	P	N°C
A1	0.639	0.507	0.503
A2	0.575	0.583	0.704
A3	0.511	0.634	0.503

c) Transformação linear

Para facilitar a comparação directa entre os atributos, procede-se a uma transformação linear. Assim, os atributos que são melhores quanto maior o seu valor, são divididos pelo maior valor que ele assume (na coluna), ou seja, o maior elemento ficará com o valor unitário. (algoritmo **TransfLinear**)

Matematicamente fica:

$$X_{ij} = \frac{X_{ij}}{X^{*j}} \quad \text{em que } X^{*j} = \max_i X_{ij}$$

Se ao atributo interessa o menor valor, cada elemento será o quociente do menor valor da coluna pelo seu valor, ou seja, o menor valor ficará igual à unidade. (algoritmo **TransfLinear**)

Neste caso a expressão a usar é a seguinte:

$$X_{ij} = \frac{X^{*j}}{X_{ij}} \quad \text{em que } X^{*j} = \min_i X_{ij}$$

Retornando ao caso prático, enquanto que os atributos Tempo e Preço são melhores quanto menor for o seu valor, o atributo N^oCaracterísticas é tanto melhor quanto maior o seu valor.

Para o atributo Tempo o menor valor é 0.511, enquanto que para o atributo Preço é 0.507. Contrariamente, para o atributo N^oCaracterísticas o melhor valor corresponde ao maior valor que é 0.704.

A nova matriz de decisão normalizada e com todos os elementos na mesma escala comparável é:

	T	P	N ^o C
A1	0.80	1.00	0.71
A2	0.89	0.87	1.00
A3	1.00	0.80	0.71

d) Método da Pesagem Simples Aditiva ou das Ponderações.

Neste método é atribuído um peso a cada atributo, de modo a que, a soma de todos os pesos seja igual à unidade. (algoritmo **CalculaPesos**)

Por outras palavras, se w_j é o peso do atributo j , então:

$$0 \leq w_j \leq 1 \quad \text{por forma a que } \sum_j w_j = 1 \quad j = 1, \dots, m$$

Em seguida, faz-se para cada alternativa o somatório do produto de cada um dos valores dos atributos pelos respectivos pesos, obtendo-se um

valor S_i para cada alternativa (um vector S). (algoritmos **MultiplicaPeso** e **Ponderações**)

Recapitulando, a equação que calcula S_i é:

$$S_i = \sum_j (X_{ij} * W_j)$$

Finalmente, a alternativa escolhida corresponde ao elemento de maior valor do vector S. (algoritmo **DeterminaMelhor**)

Em relação ao caso prático, foi atribuído uma importância de 50% para o Tempo, de 35% para o Preço e de 15% para o N°Características.

Numa escala unitária, os valores das ponderações serão:

Tempo - 0.50

Preço - 0.35

N°características - 0.15

Os valores “pesados” das três alternativas são:

$$S_1 = 0.80 * 0.50 + 1.00 * 0.35 + 0.71 * 0.15 = 0.86$$

$$S_2 = 0.90$$

$$S_3 = 0.89$$

A melhor alternativa é a que corresponde ao maior valor de S_i , que é a alternativa A2. A segunda melhor alternativa é A3 e por último a alternativa A1 que é a pior alternativa.

4.5 Algoritmos que determinam os melhores planos de transportes

Os algoritmos a seguir apresentados, permitem determinar os melhores planos de transportes, de entre os planos de transportes determinados na secção 4.2.1, em função dos critérios indicados pelo utilizador.

O algoritmo **EscolheMelhores** determina os melhores planos de transportes, utilizando as técnicas de decisão multi-atributo (método da Pesagem Simples Aditiva ou das Ponderações), já referidas.

O algoritmo verifica se existe alguma alternativa que esteja melhor classificada que todas as outras, relativamente a todos os critérios. Se tal alternativa existe, a procura acaba e essa alternativa é considerada a melhor alternativa e portanto, o melhor plano de transportes. Se essa alternativa não existe, o algoritmo retira da matriz de decisão todas as alternativas dominadas e as inadmissíveis.

De seguida, o algoritmo atribui para cada critério, o seu peso correspondente, em função das escolhas feitas pelo utilizador.

Como um dos critérios é a Qualidade do transporte, é necessário determinar o número de características que cada plano de transportes possui e introduzir esse número como o critério Qualidade na matriz de decisão.

Posteriormente, o algoritmo executa, para cada um dos critérios, as operações de normalização, transformação linear e multiplicação pelo respectivo peso. Para cada alternativa da matriz de decisão, somam-se os valores de todos os seus critérios.

Finalmente, o algoritmo determina as duas melhores somas na matriz de decisão e conseqüentemente, os planos de transportes que correspondem a essas somas. Decidiu-se fornecer um leque maior de planos, uma vez que não é tomado como critério de escolha, a hora indicada pelo utilizador (hora a partir da qual pretende sair de Origem ou a hora antes da qual pretende chegar a Destino). Assim, o melhor plano determinado pelo programa pode não ser o melhor para o utilizador, devido ao tempo que este pode ter que esperar antes de partir de Origem ou depois de chegar a Destino. Este critério não foi considerado na determinação dos melhores planos, devido à sua menor importância relativamente aos restantes critérios.

/* **Algoritmo EscolheMelhores**

*/

Verificar se existe alguma alternativa que esteja melhor classificada que todas as outras relativamente a todos os critérios

Se sim

Encontrou-se a solução ótima (a melhor alternativa)

O plano de transportes correspondente a essa alternativa é o melhor plano de transportes

Se não

Eliminar da matriz de decisão as alternativas dominadas e as inadmissíveis

Atribuir o peso a cada critério, em função das escolhas do utilizador relativamente ao Tempo, Preço e Qualidade (NºCaracterísticas) (**Algoritmo CalculaPesos**)

Para cada um dos planos de transportes existentes na Matriz de Decisão

Determinar o número de características que cada plano de transportes possui

Para cada um dos critérios considerados

Realizar as operações de normalização, transformação linear e multiplicação pelo peso **respectivo (Algoritmo CálculosCritérios)**

Para cada alternativa

Somar os valores de todos os critérios na alternativa em questão (**Algoritmo Ponderações**)

Determinar as duas melhores somas e os planos de transportes correspondentes (**Algoritmo DeterminaMelhores**)

/* Fim do algoritmo **EscolheMelhores**

*/

/* **Algoritmo CalculaPesos**

*/

Verificar qual a escolha do utilizador relativamente ao Preço, Tempo e Qualidade (NºCaracterísticas)

/* Estes pesos são determinados tendo em conta que a soma dos três pesos deve ser igual a 1.

*/

Se a escolha for

“O Tempo tem prioridade máxima”

Peso Tempo = 0.8; Peso Preço = 0.1; Peso NºCaracterísticas = 0.1

“O Preço tem prioridade máxima”

Peso Tempo = 0.1; Peso Preço = 0.8; Peso NºCaracterísticas = 0.1

“A Qualidade do transporte tem prioridade máxima”

Peso Tempo = 0.1; Peso Preço = 0.1; Peso NºCaracterísticas = 0.8

“Todos os factores têm a mesma prioridade”

Peso Tempo = 0.33; Peso Preço = 0.33; Peso NºCaracterísticas = 0.33

“Todos os factores têm prioridades diferentes” e “O Tempo é o factor mais importante seguido do Preço”

Peso Tempo = 0.50; Peso Preço = 0.35; Peso NºCaracterísticas = 0.15

“Todos os factores têm prioridades diferentes” e “O Tempo é o factor mais importante seguido da Qualidade”

Peso Tempo = 0.50; Peso Preço = 0.15; Peso NºCaracterísticas = 0.35

“Todos os factores têm prioridades diferentes” e “O Preço é o factor mais importante seguido do Tempo”

Peso Tempo = 0.35; Peso Preço = 0.50; Peso NºCaracterísticas = 0.15

“Todos os factores têm prioridades diferentes” e “O Preço é o factor mais importante seguido da Qualidade”

Peso Tempo = 0.15; Peso Preço = 0.50; Peso NºCaracterísticas = 0.35

“Todos os factores têm prioridades diferentes” e “A Qualidade é o factor mais importante seguido do Tempo”

Peso Tempo = 0.35; Peso Preço = 0.15; Peso NºCaracterísticas = 0.50

“Todos os factores têm prioridades diferentes” e “A Qualidade é o factor mais importante seguido do Preço”

Peso Tempo = 0.15; Peso Preço = 0.35; Peso NºCaracterísticas = 0.50

/* Fim do algoritmo **CalculaPesos**

*/

/* **Algoritmo CálculosCritérios**

*/

Determinar a norma euclidiana do critério em questão (**Algoritmo NormaEuclidiana**)

Normalizar os valores referentes ao critério (**Algoritmo Normalização**)

Executar a transformação linear aos valores referentes ao critério (**Algoritmo TransfLinear**)

Multiplica o peso atribuído ao critério com os valores referentes ao critério (**Algoritmo MultiplicaPeso**)

/* Fim do algoritmo **CálculosCritérios**

*/

4.6 Realização do programa desenvolvido

Os algoritmos foram desenvolvidos no Microsoft Access 2.0 utilizando a linguagem Microsoft Access Basic 2.0.

Foram desenvolvidas 75 rotinas, 97 consultas e 72 formulários, que permitem determinar e apresentar a informação sobre os planos de transportes possíveis e os melhores planos. Estes formulários são

apresentados no anexo C, em conjunto com uma simulação do programa desenvolvido.

O ficheiro que contém todas as tabelas, consultas, formulários e rotinas ocupa 6560 KB.

4.7 Exemplo de uma simulação do programa

Nesta secção são apresentados dois exemplos de planos de transportes possíveis e dos melhores planos que o programa determina, supondo que para a ligação Lisboa-Faro, se pretende partir de Lisboa no dia 3 de Janeiro de 1996 a partir das 7:30 e considerando num dos exemplos que o critério com prioridade máxima é o tempo e que no outro é o preço.

Para se ter uma ideia do tempo despendido pelo programa para determinar todos os planos de transportes possíveis e os melhores planos, convém indicar que na base de dados estão guardados 44 transportes inter-urbanos, 11 urbanos e 5 centrais de táxis. Os planos de transportes possíveis obtidos estão indicados na tabela Tab. 4.1..

Como no primeiro exemplo se indica o tempo de duração da viagem como o critério com prioridade máxima, dois dos planos escolhidos como melhores correspondem a planos constituídos por aviões, como se pode verificar na Tab. 4.2.. Os melhores planos de transportes obtidos no segundo exemplo, correspondem aos planos com os menores preços de bilhete, tal como indicados na tabela Tab. 4.3..

O tempo que o programa demorou² para determinar os planos de transportes possíveis foi de 107 segundos, enquanto que para determinar os melhores planos, tanto no primeiro exemplo como no segundo, demorou 5 segundos.

² O programa foi testado num PC 80486 DX2 a 66 MHz, com 16 MB de RAM.

Situação	Transporte principal								Transporte secundário anterior								Transporte secundário posterior									
	Nome	Origem	Destino	H. part.	H. cheg.	Categoria	Preço	Qual ³	Nome	Tipo ⁴	Origem	Destino	H. part.	H. cheg.	Categoria ⁵	Preço	Qual.	Nome	Tipo	Origem	Destino	H. part.	H. cheg.	Categoria	Preço	Qual.
1	Intercidades	Lisboa	Faro	17:30	22:00	1ª	3005	7																		
						2ª	1905	5																		
1	Avião	Lisboa	Faro	15:20	16:00	Única	13171	4																		
1	Avião	Lisboa	Faro	19:50	20:30	Única	13171	4																		
1	Expresso	Lisboa	Faro	7:45	12:25	Única	1900	3																		
1	Expresso	Lisboa	Faro	13:00	17:50	Única	1900	3																		
1	Expresso	Lisboa	Faro	14:15	18:55	Única	1900	3																		
1	Expresso	Lisboa	Faro	19:30	0:10	Única	1900	3																		
3	Expresso	Lisboa	Quarteira	12:30	18:00	Única	1100	3										Expresso	IU	Quarteira	Faro	18:30	19:00	Única	400	3
3	Expresso	Lisboa	Quarteira	11:00	18:00	Única	1100	3										Expresso	IU	Quarteira	Faro	18:30	19:00	Única	400	3
2	Expresso	Almada	Faro	11:00	16:00	Única	1500	5	Expresso	IU	Lisboa	Almada	7:45	8:05	Única	450	3									
4	Expresso	Almada	Quarteira	11:00	14:00	Única	1200	5	Expresso	IU	Lisboa	Almada	7:45	8:05	Única	450	3	Expresso	IU	Quarteira	Faro	18:30	19:00	Única	400	3
4	Expresso	Almada	Quarteira	11:00	14:00	Única	1200	5	Expresso	IU	Lisboa	Almada	7:45	8:05	Única	450	3	Expresso	U	Quarteira	Faro	16:19	16:51	Normal	400	
4	Expresso	Almada	Quarteira	11:00	14:00	Única	1200	5	Expresso	IU	Lisboa	Almada	7:45	8:05	Única	450	3	Expresso	U	Quarteira	Faro	18:28	19:00	Normal	400	

Tab. 4.1. Planos de transportes possíveis

³ A Qualidade do transporte corresponde ao número de características existente nele.

⁴ Transporte inter-urbano (IU) ou transporte urbano (U)

⁵ Quando o transporte é urbano, a categoria corresponde ao nome do bilhete.

Situação	Transporte principal								Transporte secundário anterior							Transporte secundário posterior										
	Nome	Origem	Destino	H. part.	H. cheg.	Categoria	Preço	Qual	Nome	Tipo	Origem	Destino	H. part.	H. cheg.	Categoria	Preço	Qual.	Nome	Tipo	Origem	Destino	H. part.	H. cheg.	Categoria	Preço	Qual.
1	Avião	Lisboa	Faro	15:20	16:00	Única	13171	4																		
1	Avião	Lisboa	Faro	19:50	20:30	Única	13171	4																		
1	Intercidades	Lisboa	Faro	17:30	22:00	1ª	3005	7																		
						2ª	1905	5																		

Tab. 4.2. Melhores planos de transportes quando o critério seleccionado corresponde ao Tempo com prioridade máxima

Situação	Transporte principal								Transporte secundário anterior							Transporte secundário posterior									
	Nome	Origem	Destino	H. part.	H. cheg.	Categoria	Preço	Qual	Nome	Tipo	Origem	Destino	H. part.	H. cheg.	Categoria	Preço	Qual.	Nome	Tipo	Origem	Destino	H. part.	H. cheg.	Categoria	Preço
3	Expresso	Lisboa	Quarteira	12:30	18:00	Única	1100	3									Expresso	IU	Quarteira	Faro	18:30	19:00	Única	400	3
3	Expresso	Lisboa	Quarteira	11:00	18:00	Única	1100	3									Expresso	IU	Quarteira	Faro	18:30	19:00	Única	400	3

Tab. 4.3. Melhores planos de transportes quando o critério seleccionado corresponde ao Preço com prioridade máxima

5. CONCLUSÕES

5

Embora se possam fazer alguns melhoramentos, tanto ao nível da caracterização do problema, como ao nível dos algoritmos desenvolvidos, julga-se ter sido cumprido o objectivo proposto.

Devido à especificidade de cada tipo de transporte, quer no que respeita ao tipo de pagamento da viagem, quer em relação aos percursos executados, classificaram-se os transportes em três tipos: os inter-urbanos, os urbanos e os táxis.

A primeira abordagem ao problema foi representar a informação adquirida sobre os transportes utilizando os *enquadramentos* como método de representação. Utilizou-se o formalismo BNF para representar as classes-enquadramento, instâncias enquadramento e valores dos atributos.

Para se estabelecerem planos de transportes, foi necessário desenvolver uma base de dados num SGBD, por forma a ser possível guardar e gerir a informação adquirida. Optou-se por uma base de dados relacional, uma vez que é de fácil manipulação e a mais utilizada actualmente.

Para definir a base de dados, começou-se por utilizar a abordagem Entidade-Associação, estabelecendo-se os conjuntos de entidades, os conjuntos de associações que existem para estes e os atributos que descrevem as propriedades de cada um destes conjuntos. Os conjuntos de associações classificaram-se quanto à sua aridade e multiplicidade.

Uma vez que a base de dados é relacional, foi necessário converter os diagramas E-R em relações, de modo a não se introduzir redundância na base de dados, a se manterem a integridade de entidades e a integridade referencial e a obter-se um número mínimo de relações.

O SGBDR utilizado foi o Microsoft Access 2.0. Para introduzir informação referente aos transportes na base de dados, definiram-se formulários e consultas, tais como os apresentados no anexo B.

Por último desenvolveram-se os algoritmos e heurísticas necessários para estabelecer planos de transportes e para determinar os melhores de entre eles, em função dos critérios indicados pelo utilizador.

Considerou-se que os planos não deveriam ser constituídos apenas por transportes directos, mas por vários transportes (planos com transbordos). Assim, estabeleceram-se planos com origens alternativas a Origem, planos com destinos alternativos a Destino e planos com origens e destinos alternativos a Origem e a Destino, respectivamente.

Para construir planos de transportes, os algoritmos seleccionam todos os transportes que satisfazem as exigências indicadas pelo utilizador e algumas hipóteses pré-estabelecidas que se consideraram válidas para todos os utilizadores (tais como, as durações máxima e mínima para se efectuar um transbordo e a distância máxima entre as paragens onde este ocorre).

Os táxis apenas são utilizados como transportes de ligação, ou seja, para efectuar percursos de curta duração. Devido ao tipo de informação contida na base de dados, só é possível determinar a distância mínima dos percursos em função das coordenadas topográficas dos locais.

Para determinar os melhores planos em função de critérios indicados pelo utilizador, foram utilizadas técnicas de decisão multi-atributo, especificamente o método da Pesagem Simples Aditiva. Estes critérios reflectem a importância que o utilizador atribui a factores como o tempo de duração da viagem, o preço dos bilhetes e a qualidade dos transportes utilizados.

As heurísticas desenvolvidas e utilizadas na definição dos algoritmos, reflectem algumas considerações e melhorias tidas em conta, quer para determinar os planos de transportes possíveis, quer os melhores planos.

Os algoritmos foram implementados num programa desenvolvido no próprio SGBDR, utilizando a linguagem de programação Microsoft Access Basic 2.0. Uma simulação do programa é apresentada no anexo C.

Embora este estudo permita estabelecer planos de transportes e determinar os melhores de entre estes, é possível introduzir algumas melhorias que se sugerem a seguir:

1. Introduzir mecanismos de aprendizagem que confirmam ao programa a capacidade de utilizar o conhecimento adquirido com os planos anteriormente determinados, na elaboração de novos planos.
2. Utilização das técnicas tradicionais de calendarização (scheduling) e de satisfação de constrangimentos (constraints-satisfaction) para a escolha dos percursos.
3. Introduzir a possibilidade do utilizador indicar, para além das localidades Origem e Destino, as moradas em Origem e em Destino, por forma a ter conhecimento também dos melhores transportes, em função dos critérios escolhidos por si, para ir da morada indicada em Origem à paragem de partida em Origem e da paragem de chegada em Destino à morada indicada em Destino. Estes transportes serão, essencialmente, transportes urbanos e táxis.
4. No caso dos táxis, o cálculo da distância entre dois locais deverá ser mais exacta, uma vez que na realidade esta não corresponde à medida do segmento de recta que os une. Uma solução possível, embora não a mais correcta, seria guardar numa tabela, as distâncias exactas (dos menores percursos rodoviários).
5. O utilizador poderá ser inquirido também quanto aos cartões de descontos que eventualmente possua, por forma a que o programa proceda aos descontos possíveis nos preços dos bilhetes.
6. Desenvolver uma interface mais agradável, onde se poderia mostrar ao utilizador as posições das localidades num mapa, o caminho a percorrer e as paragens necessárias em cada plano. Também se poderiam

introduzir mecanismos que permitissem a introdução de dados pelo utilizador, utilizando a voz.

6. BIBLIOGRAFIA

- [1] Minsky, M., 1975. "A Framework for Representing Knowledge" em The Psychology of Computer Vision - Winston (ed.), New York: McGraw Hill Book Co..

 - [2] Reichgelt, Han, 1991. "Knowledge Representation: An A.I. Perspective" - Ablex Publishing Corporation - Norwood, New Jersey

 - [3] Martins, João P., 1989. "Introduction to Computer Science" - Wadsworth Publishing Company, 1º Edition

 - [4] Coelho, Helder, 1988. "Computador e Informação" - Laboratório Nacional de Engenharia Civil, 2ª Edição

 - [5] Chen, P. P., 1976. "The Entity-Relationship Model - Toward a Unified View of Data", ACM Transactions on Database Systems, vol. 6, no. 1.

 - [6] Hawryszkiewicz, Igor T., 1990. "Relational Database Design - An Introduction" - Prentice Hall.

 - [7] Carapuça, Rogério, 1993. "Introdução aos Sistemas de Informação e Bases de Dados: Texto de Apoio" - IST/DEEC.

 - [8] Ackoff, R., 1962. "Scientific Method: Optimizing Applied Research Decisions" - John Wiley, New York.

 - [9] Schuy, Don, 1995. "Programming Microsoft Office" - SAMS Publishing.
- Alagic, Suad, 1986. "Relational Database Technology" - Springer-Verlag.
- Atre, S., 1980. "Database: Structured Techniques for Design, Performance and Management" - Wiley-Interscience Publication.

Coelho, Helder, 1995. "Inteligência Artificial em 25 Licções" - Fundação Calouste Gulbenkian.

Date, C.J., 1995. "An Introduction to Database Systems" - Addison-Wesley Publishing Company, Sixth Edition.

Date, C.J., 1995. "Relational Database - Writings 1991-1994" - Addison-Wesley Publishing Company.

Dean, Thomas; Allen, James; Aloimonos, Yiannis, 1995. "Artificial Intelligence, Theory and Practice" - The Benjamin/Cummings Publishing Company, inc.

"Guia Prático do Windows 3.1, Word 6.0, Excel 5.0, Access 2.0" - McGraw-Hill.

Kim, Won, 1995. "Modern Database Systems: The Object Model, Interoperability and Beyond" - ACM Press.

Korth, Henry F.; Silberschatz, Abraham, 1986. "Database System Concepts" - McGraw-Hill International Editions.

Loomis, Mary E. S., 1995. "Object Databases: The Essentials" - Addison-Wesley Publishing Company.

Martins, M. A. Costa, 1984. "Concepção duma Base de Dados" - Rés-Editora.

McGee, Margaret D.; Boyce, Judy, 1993. "Microsoft Access for Windows, Step by Step" - Microsoft Corporation.

Nelson, Stephen L., 1994. "Field Guide to Microsoft Access 2 for Windows" - Microsoft Press.

Rich, Elaine; Knight, Kevin, 1991. "Artificial Intelligence" - McGraw-Hill, Second Edition.

Ullman, Jeffrey D., 1982. "Principles of Database Systems" - Computer Science Press, Second Edition.

6.1 RELAÇÕES-NOTAÇÃO LINEAR

A

Neste anexo são apresentadas as relações obtidas no terceiro capítulo, utilizando outra notação, onde o nome da relação aparece seguido da lista de atributos (que estão separados por vírgulas e dentro de parênteses curvos) e os atributos que pertencem à chave primária estão sublinhados e os que pertencem à chave estrangeira estão em itálico.

Transportes Inter urbanos

Inter_urbano (Inter_urbano_id, N^o_transporte, Nome_transporte, Data_início_funcionamento, Data_fim_funcionamento, Local_partida, Local_chegada, Domingo, Segunda, Terça, Quarta, Quinta, Sexta, Sábado, Feriado, *Operador_inter_urbano_id*)

Operador_inter_urbano (Operador_inter_urbano_id, Nome_operador)

Local_aquisição_bilhetes_inter_urbanos

(Local_aquisição_bilhetes_inter_urbanos_id, Nome_local, Telefones, Faxes, Horário, *Morada_id*, *Operador_inter_urbano_id*)

Tipo_categoria (Tipo_categoria_id, Nome_categoria, Idade_sem_bilhete,

Idade_meio_bilhete, Marcação_lugar, Bar, Restaurante, Lugares_cama, Wc, Ar_condicionado, Vídeo, Tv, Telemóvel)

Inter_urbanos_Categorias (Inter_urbano_id, Tipo_categoria_id),

Desconto (Desconto_id, Nome_desconto, Percentagem, Período, Condições_para_obter_desconto)

Categoria_tem_descontos (Tipo_categoria_id, Desconto_id)

Paragem (Paragem_id, Nome_paragem, *Morada_id*)

Morada (Morada_id, Rua_avenida_nº, Local, Código_postal, Concelho,
Distrito, Coord_M, Coord_P, *Localidade_id*)

Parte_de (Parte_de_id, Hora_partida, *Paragem_id*)

Chega_a (Chega_a_id, Hora_chegada, *Paragem_id*)

Ligações_inter_urbano (Parte_de_id, Chega_a_id, Inter_urbano_id)

Preço_inter_urbano (Paragem_origem_id, Paragem_destino_id,
Tipo_categoria_id, Preço_sem_descontos_ida,
Preço_sem_descontos_ida_e_volta, Informação)

Localidade (Localidade_id, Nome_localidade, Tipo_localidade)

Cidade_próxima (Localidade_id, Localidade_próxima_id, Distância)

Transportes Urbanos

Urbano (Urbano_id, Número_carreira, Nome_carreira, Local_partida,
Local_chegada, *Operador_urbano_id*,
Características_transporte_id, *Paragem_partida_id*)

Características_transporte_urbano (Características_transporte_urbano_id,
Nome_transporte, Idade_sem_bilhete,
Idade_meio_bilhete)

Operador_urbano (Operador_urbano_id, Nome_operador)

Bilhete_urbano_passe (Bilhete_urbano_passe_id, Nome_passe, Preço,
Validade_passe, Descrição_passe, Área_validade,
Descrição_área, *Operador_urbano_id*)

Bilhete_urbano_outros (Bilhete_urbano_outros_id, Nome_bilhete, Preço,
Validade_bilhete, Descrição_bilhete,
Número_bilhetes, Número_obliterações,
Operador_urbano_id)

Local_aquisição_bilhetes_urbanos (Local_aquisição_bilhetes_urbanos_id,
Nome_local, Telefones, Faxes, Horário,
Morada_id, Operador_urbano_id)

Morada_urbano (Morada_urbano_id, Rua_avenida_nº, Local,
Código_postal, Concelho, Distrito, Coord_M, Coord_P, *Localidade_id*)

Paragem_urbano (Paragem_urbano_id, Nome_paragem, *Morada_id*)

Partidas_urbano (Partidas_urbano_id, Hora_partida,
Data_início_funcionamento, Data_fim_funcionamento, Domingo, Segunda,
Terça, Quarta, Quinta, Sexta, Sábado, Feriado, *Urbano_id*)

Número_bilhetes_urbano (Paragem_origem_id, Paragem_destino_id,
Urbano_id, Tempo_duração, Número_bilhetes)

Localidade (Localidade_id, Nome_localidade, Tipo_localidade)

Cidade_próxima (Localidade_id, Localidade_próxima_id, Distância)

Táxis

Operador_táxi (Operador_táxi_id, Nome_operador, Telefones_central,
Faxes_central)

Morada_táxi (Morada_táxi_id, Rua_avenida_nº, Local, Código_postal,
Concelho, Distrito, Coord_M, Coord_P, *Localidade_id*)

Morada_praça_táxi_operador (Operador_táxi_id, Morada_táxi_id)

Preço_táxi (Preço_táxi_id, Hora_início_período, Hora_fim_período, Preço,
Preço_bagagem, Preço_bandeirada, Informação, Domingo,
Segunda, Terça, Quarta, Quinta, Sexta, Sábado, Feriado)

Preço_táxi_operador (Operador_táxi_id, Preço_táxi_id)

Localidade (Localidade_id, Nome_localidade, Tipo_localidade)

Cidade_próxima (Localidade_id, Localidade_próxima_id, Distância)

6.2 INTRODUÇÃO DA INFORMAÇÃO NA BASE DE DADOS

B

Neste anexo é descrita a forma como se deve introduzir a informação na base de dados. A introdução dos dados referentes aos transportes inter-urbanos, aos urbanos e aos táxis é feita em janelas diferentes.

À pessoa que introduz a informação, chama-se operador da base de dados, para que não se confunda com utilizador, que é o termo que tem vindo a ser usado e que corresponde à pessoa que utiliza o programa para obter planos de transportes.

TRANSPORTES INTER- URBANOS

Na janela que a seguir se apresenta, é possível introduzir os dados referentes aos transportes inter-urbanos, tais como, o número do transporte, a data de início e a data final do período de funcionamento, os dias da semana em que se efectua a ligação e o local de partida e de chegada. O transporte tem um código interno gerado automaticamente pelo programa. Se o transporte pertence a um operador que já existe na base de dados, basta introduzir o código deste e posteriormente é o próprio programa que actualiza automaticamente o outro campo pertencente ao operador, ou seja, o seu nome.

Acontece o mesmo em relação às paragens; se as informações relativas a estas já existem na base de dados, não se devem introduzir novamente, porque embora as informações a introduzir coincidam com as já existentes, os códigos das paragens seriam diferentes. O operador da

base de dados deve verificar qual o código atribuído a cada paragem e indicar esse código quando necessário.

Se o operador não existir na base de dados, é necessário que o operador da base de dados, introduza esses dados, deixando que o programa proceda à geração automática do código, que poderá ser posteriormente utilizado para a introdução de outros transportes pertencentes ao mesmo operador.

A restante informação relativa aos locais de paragem, às partidas, às chegadas, aos preços, etc., não é feita aleatoriamente; existe um número atribuído a cada botão que indica a sua ordem. Por exemplo, é necessário introduzir primeiro as paragens, porque o código atribuído pelo programa a cada uma das paragens, tem de ser depois introduzido, pelo operador da base de dados, nas informações relativas às partidas e às chegadas.

Nas janelas, apenas aparecem os códigos gerados pelo programa que são necessários, ou seja, os códigos que irão ser utilizados pelo operador da base de dados e que serão necessários em outras janelas.

A seguir é descrita a forma como o operador da base de dados deve proceder para realizar certas operações:

- Para ir para um determinado registo
Utilizar os botões de navegação que estão incorporados na *scroll bar* horizontal.
- Para acrescentar um novo registo
Ir para o último registo e carregar no botão da *scroll bar* horizontal que permite ir para o registo seguinte.
- Para apagar um registo
Seleccionar o registo a apagar e carregar no seguinte botão:



- Para desfazer a introdução da informação num registo
Carregar no seguinte botão:



- Para fechar a janela

Fecha-se como uma janela do Windows 95 (com o botão mais à direita dos três botões que estão no canto superior direito).

Transportes Inter-urbanos

Nº do transporte: 580

Data início de funcionamento: 01/01

Data fim de funcionamento: 31/12

Local de partida: Faro

Local de chegada: Lisboa

Nome do transporte: Intercidades

Nome do operador: Caminhos de Ferro Portuguese

Código do transporte:

Código do operador: 1

Domingo:

Segunda:

Terça:

Quarta:

Quinta:

Sexta:

Sábado:

Feriado:

1-Paragens

2-Partidas

3-Chegadas

4-Ligações

5-Categorias

6-Descontos

7-Categorias-Descontos

8-Preços

9-Categorias - Transportes

10-Locais onde adquirir os bilhetes

Record:1 of 44

Para as seguintes janelas, são indicadas as informações que são introduzidas pelo operador da base de dados e as geradas automaticamente pelo programa.

Paragens

Operador: O nome da paragem, a sua morada e as suas coordenadas.

Programa: O código da paragem.

Transportes

Inter-urbanos - Paragens

Código da paragem:
Nome da paragem:
Rua-avenida e nº:
Local:
Código postal:
Concelho:
Distrito:
Coordenada M:
Coordenada P:

Record: 1 of 6

Partidas

Operador: A hora de partida e o código da paragem onde esta se efectua.

Programa: O código da partida.

Transportes

Inter-urbanos - Partidas

Código da partida:
Código da paragem:
Hora de partida:

Record: 1 of 5

Chegadas

Operador: A hora de chegada e o código da paragem onde esta se efectua.

Programa: O código da chegada.

Transportes

Inter-urbanos - Chegadas

Código da chegada:

Código da paragem:

Hora de chegada:

Record: 1 of 5

Ligações

Operador: Os códigos referentes à partida, à chegada e ao transporte.

Transportes

Inter-urbanos - Ligações

Código da partida:

Código da chegada:

Código do transporte:

Record: 1 of 15

Categorias

Operador: O nome da categoria, as idades para não pagar bilhete, as idades para pagar apenas meio bilhete, a existência ou não de bar, restaurante, tv, etc..

Programa: O código da categoria.

Transportes

Inter-urbanos - Categorias

Código da categoria:

Nome da categoria:

Idades para não pagar bilhete:

Idades para pagar meio bilhete:

Marcação de lugar:

Bar:

Restaurante:

Lugares cama:



WC:

Ar condicionado:

Vídeo:

TV:

Telemóvel:

Record: 1 of 2

Descontos

Operador: O nome do desconto, a percentagem, o período em que é válido e as condições para o obter.

Programa: O código do desconto.

Transportes

Inter-urbanos - Descontos



Código do desconto:

Nome do desconto:

Percentagem:

Período em que o desconto é válido:

Condições para obter o desconto:

Record: 1 of 1

Categorias-Descontos



Operador: Os códigos da categoria e do desconto.

Transportes

Inter-urbanos - Categorias-Descontos

Código da categoria:

Código do desconto:

Record:1 of 2

Preços

Operador: O código da paragem de partida, o código da paragem de chegada, o preço sem descontos de uma ida para ir da paragem de partida à paragem de chegada, o preço sem descontos de uma ida e volta para ir da paragem de partida à paragem de chegada e vice-versa, o código da categoria a que correspondem esses preços e qualquer informação suplementar.

Transportes

Inter-urbanos - Preços

Código da paragem de partida:



Código da paragem de chegada:

Preço sem descontos ida:

Preço sem descontos ida e volta:

Código da categoria:

Informação suplementar:

Record:1 of 30

Categorias-Transportes



Operador: Os códigos da categoria de viagem e do transporte.

Inter_urbanos - Tipo_clases - Inter_urbanos

Inter_urbanos - Tipo_categorias - Inter_urbanos

Código transporte:

Código categoria:

Record: 1 of 2

Locais onde adquirir os bilhetes

Operador: O nome do local, os números de telefone, os números de fax, o horário de atendimento ao público, a morada, as suas coordenadas e o código do respectivo operador.

Programa: O código do local.

Transportes

Inter-urbanos - Locais

Código do local:

Nome do local:

Telefones:

Faxes:

Horário:

Rua-avenida e n°:

Local:

Código postal:



Concelho:

Distrito:

Coordenada M:

Coordenada P:

Código do operador:

Record: 1 of 7

TRANSPORTES URBANOS

A introdução da informação referente às características do transporte e ao operador seguem os mesmos princípios utilizados na introdução da informação referente ao operador para os transportes inter-urbanos. Se não existir na base de dados nenhum transporte com as características a introduzir, é necessário preencher os campos referentes ao nome do

transporte, às idades para não pagar bilhete e às idades para pagar apenas meio bilhete.

Tal como acontecia para os transportes inter-urbanos, o código do transporte também é gerado automaticamente pelo programa, mas é o operador da base de dados que introduz a restante informação, como o número da carreira, o seu nome, o local de partida e o local de chegada.

O código da paragem de partida só poderá ser introduzido depois das informações relativas a esta terem sido introduzidas, ou seja, depois de o programa ter gerado o referido código.

Transportes

Transportes Urbanos

Número da carreira:

Nome da carreira:

Local de partida:

Local de chegada:

Nome do transporte:

Código da paragem de partida:

Idades para não pagar bilhete:

Idades para pagar meio bilhete:

Nome do operador:

Código do operador:

Código do transporte:

Código das características:

1-Paragens 2-Partidas 3-Bilhetes 4-Passes 5-Preços 6-Locais onde adquirir os bilhetes

Record: 11 of 11

Paragens

Operador: O nome da paragem, a sua morada e as suas coordenadas.

Programa: O código da paragem.

Transportes

Urbanos - Paragens

Código da paragem:

Nome da paragem:

Rua-avenida e nº:

Local:

Código postal:

Concelho:

Distrito:

Coordenada M:

Coordenada P:

Record: 1 of 4

Partidas

Operador: A hora de partida, a data de início e a data final do período de funcionamento, os dias da semana em que se efectua o percurso e o código do transporte.

Transportes

Urbanos - Partidas

Hora de partida:

Data início de funcionamento:

Data fim de funcionamento:

Código do transporte:

Domingo:

Segunda:

Terça:

Quarta:

Quinta:

Sexta:

Sábado:

Feriado:

Record: 1 of 5

Bilhetes

Operador: O nome do tipo de bilhete, o seu preço, o número de bilhetes que o constituem, o número de obliterações que é possível fazer com esse bilhete, a sua validade, a descrição do bilhete e o código do operador.

Transportes

Urbanos - Bilhetes

Nome bilhete:

Preço:



Número de bilhetes:

Número de obliterações:

Validade do bilhete:

Descrição do bilhete:

Código do operador:

Record: 1 of 2

Passes

Operador: O nome do passe, o seu preço, validade, descrição, área de validade, designação da área e o código do operador.

Transportes

Urbanos - Passes

Nome do passe:

Preço:



Validade do passe:

Descrição do passe:

Área de validade:

Descrição da área:

Código do operador:

Record: 1 of 1

Preços

Operador: O tempo de duração e o número de obliterações necessárias para ir da paragem de partida à paragem de chegada, os códigos do transporte, da paragem de partida e da paragem de chegada.

Transportes

Urbanos - Preços

Código da paragem de partida:

Código da paragem de chegada:

Tempo de duração:

Número de obliterações necessárias:

Código do transporte:

Record:1 of 9

Locais onde adquirir os bilhetes

Operador: O nome do local, os números de telefone, os números de fax, o horário de atendimento ao público, a morada, as suas coordenadas e o código do respectivo operador.

Programa: O código do local.

Transportes

Urbanos - Locais

Código do local:

Nome do local:

Telefones:

Faxes:

Horário:

Rua-avenida e nº:

Local:

Código postal:

Concelho:

Distrito:

Coordenada M:

Coordenada P:

Código do operador:

Record:1 of 2

TÁXIS

No caso de o operador já existir na base de dados, é necessário apenas indicar o seu código, sendo depois os restantes campos actualizados automaticamente pelo programa. Se o operador não existir na base de dados, é o operador da base de dados que deve introduzir as

informações. Neste caso, o programa gera automaticamente o código do operador.

Transportes

Táxis

Nome do operador: Táxis Fartáxis

Telefones da central: (089) 801689, 801747

Faxes da central:

Código do operador:

1-Moradas das praças 2-Preços 3-Moradas-Operador 4-Preços-Operador

Record:1 of 5

Moradas das praças

Operador: A morada das praças de táxis e as suas coordenadas.

Programa: O código da morada.

Transportes

Táxis - Moradas

Código da morada:

Rua-avenida e nº: Praça do Hospital

Local: Faro

Código postal: 8000 Faro

Concelho: Faro

Distrito: Faro

Coordenada M: -293750

Coordenada P: 18400

Record:1 of 6

Preços

Operador: Um dos preços praticados pelo operador, os dias da semana em que esse preço é utilizado, o turno do dia em que esse preço

é válido, o preço pelo transporte de bagagem, o preço mínimo de utilização (preço da bandeirada) e qualquer informação suplementar.

Programa: O código do preço.

Transportes

Táxis - Preços

Código do preço:

Hora inicial do período:

Hora final do período:

Preço:

Preço da bagagem:

Preço bandeirada:

Informação:

Domingo:

Segunda:

Terça:

Quarta:

Quinta:

Sexta:

Sábado:

Feriado:

Record:1 of 3

Moradas Operador

Operador: Os códigos do operador e da morada de uma das praças de táxis.

Transportes

Táxis - Moradas-Operadores

Código do operador:

Código da morada:

Record:1 of 6

Preços Operador



Operador: Os códigos do operador e de um dos preços.

Transportes

Táxis - Preços-Operadores

Código do operador:

Código do preço:

Record: 1 of 3

6.3 SIMULAÇÃO DO PROGRAMA

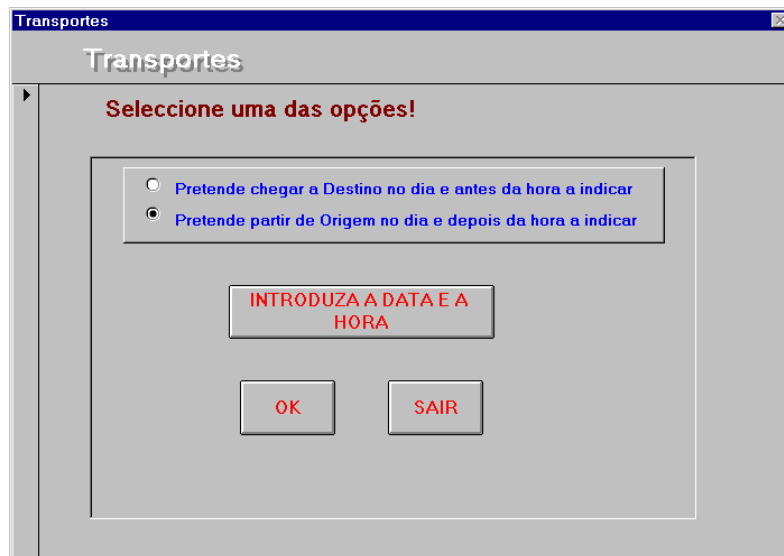
C

A interface que permite ao utilizador introduzir os dados relativos aos planos de transportes é composta por várias janelas e foi desenhada tendo em conta aspectos tais como a facilidade e a rapidez de introdução dos dados. Por estas razões, decidiu-se utilizar, sempre que possível, o rato como instrumento de selecção.

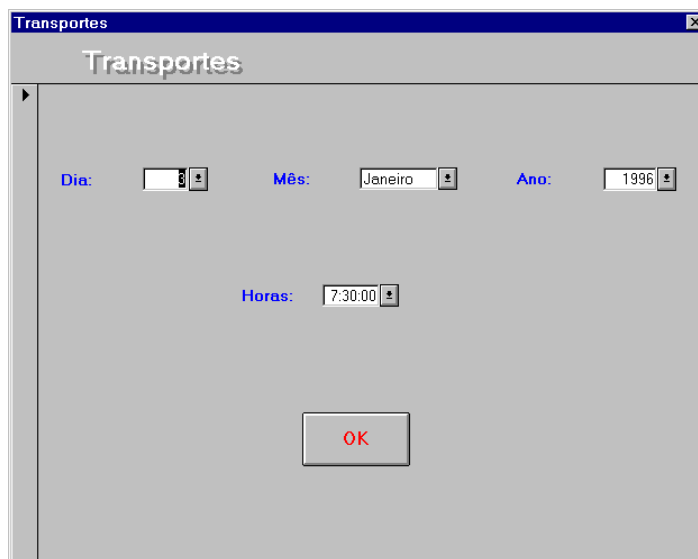
A primeira janela permite indicar o local de Origem e o local de Destino. Quando algum destes locais não está presente na lista de locais, digita-se o seu nome na lista utilizando o teclado.



A janela seguinte permite ao utilizador informar se pretende chegar a Destino até determinado dia e hora, ou partir de Origem depois de determinado dia e hora.



O botão “Introduzir a data e a hora” permite aceder à janela onde se indicam a data e a hora atrás referidas.

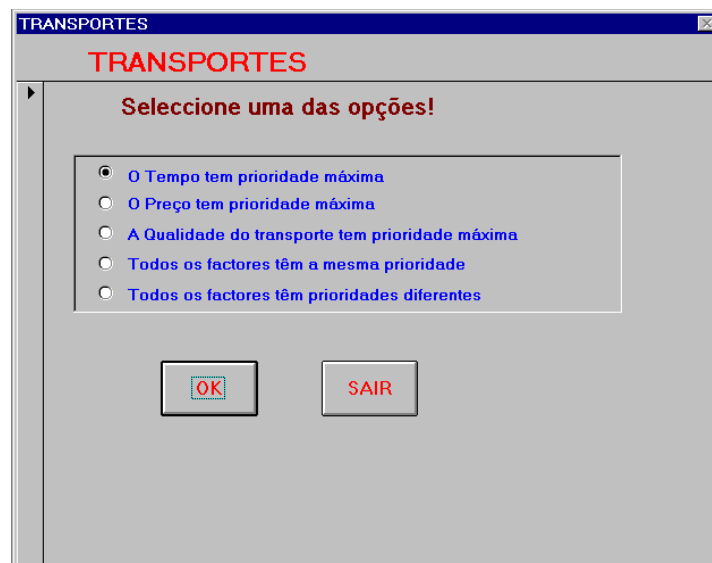


Por último, o utilizador deve escolher os critérios a considerar na determinação dos melhores planos. Estes critérios referem-se ao tempo de duração da viagem, ao seu preço e à qualidade do transporte utilizado.

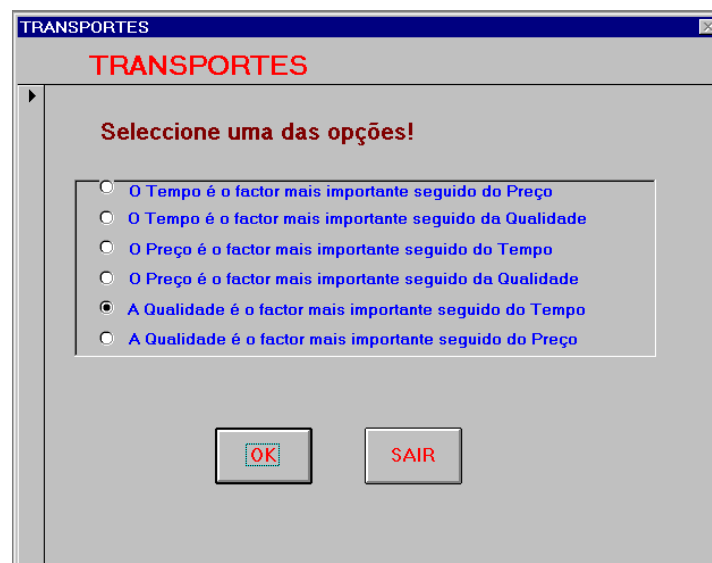
Se o utilizador pretende:

- A viagem com a menor duração possível, deve escolher “O Tempo tem prioridade máxima”.
- A viagem mais barata, deve escolher “O Preço tem prioridade máxima”.

- A viagem mais confortável, em termos da qualidade do transporte, deve escolher “A Qualidade do transporte tem prioridade máxima”.
- A viagem onde todos os factores têm igual importância, deve escolher “Todos os factores têm a mesma prioridade”.
- A viagem onde todos os factores têm prioridades diferentes, mas nenhum é prioritário relativamente aos outros, deve escolher “Todos os factores têm prioridades diferentes”.



Quando o utilizador escolhe a última opção, deve indicar numa outra janela, a importância que atribui a cada um dos factores.



Depois de se fornecerem ao programa todos estes dados, este determina os planos de transportes possíveis para ir de Origem a Destino, que são apresentados na seguinte janela:

The screenshot shows a window titled "TRANSPORTES POSSÍVEIS" with the following fields and options:

PARTIDA:		CHEGADA:	
Local de partida:	Almada	Local de chegada:	Faro
Distrito:	Setúbal	Distrito:	Faro
Rua_avenida e nº:	Pc. S. João Baptista	Rua_avenida e nº:	Terminal Rodoviário, Av. da Rep
Hora de partida:	11:00:00	Hora de chegada:	16:00:00
Nº do transporte:	66	Este plano de transportes inclui:	
Nome do transporte:	Expresso	Táxi(s):	<input checked="" type="checkbox"/>
Data:	3 de Janeiro	Transporte(s) secundário(s):	<input checked="" type="checkbox"/>

Buttons at the bottom: "Locais onde adquirir bilhetes", "Categorias e Preços", "Táxi", "Transportes secundários", "ESCOLHE OS MELHORES PLANOS DE TRANSPORTES", and "SAIR".

Footer: Record: 14 of 20

Analisando o conteúdo da janela, verifica-se que esta apresenta apenas um transporte, que corresponde a um transporte principal. Quando os planos não são constituídos por transportes directos, ou seja, quando correspondem às situações 2, 3 e 4, é necessário utilizar o botão "Transportes secundários", que, tal como o nome indica, apresenta o(s) transporte(s) secundário(s) que corresponde(m) ao transporte principal indicado na janela. Um exemplo de uma janela que fornece informação sobre o(s) transporte(s) secundário(s) que corresponde(m) a um determinado transporte principal é a seguir apresentada:

TRANSPORTES POSSÍVEIS

TRANSPORTES INTER-URBANOS- SITUAÇÃO 2

<p>PARTIDA:</p> <p>Local de partida: <input type="text" value="Lisboa"/></p> <p>Distrito: <input type="text" value="Lisboa"/></p> <p>Rua_avenida e nº: <input type="text" value="Terminal Rodoviário, Av. Casal F"/></p> <p>Hora de partida: <input type="text" value="7:45:00"/></p> <p>Nº do transporte: <input type="text"/></p> <p>Nome do transporte: <input type="text" value="Expresso"/></p> <p>Data: <input type="text" value="3 de Janeiro"/></p>	<p>CHEGADA:</p> <p>Local de chegada: <input type="text" value="Almada"/></p> <p>Distrito: <input type="text" value="Setúbal"/></p> <p>Rua_avenida e nº: <input type="text" value="Pc. S. João Baptista"/></p> <p>Hora de chegada: <input type="text" value="8:05:00"/></p>
---	---

Record: 1 of 1

Tal como já foi referido no quarto capítulo, os táxis são apenas utilizados como transportes secundários. Quando os planos correspondem à ligação secundária da situação 2 ou à ligação secundária anterior da situação 4, são fornecidas algumas moradas das praças de táxis de Origem e alguns dos números de telefone das centrais de táxis dessa localidade. Para aceder à janela com a informação sobre os táxis, utiliza-se o botão “Táxi” existente na janela dos transportes possíveis.

Táxis

Táxis

Praças de táxis

Rua_avenida e nº:

Local:

Distrito:

Record: 1 of 3

Telefones e Faxes

Nº de telefones: (01) 345622, 345672

Nº de faxes: (01) 467589

Record: 1 of 2

São também fornecidos ao utilizador, a morada da paragem de partida do transporte principal, a distância mínima do percurso, o preço total do táxi (preço da bandeirada mais o preço dos quilómetros percorridos) e o preço relativo ao transporte de bagagem.

Destino

PARAGEM:

Nome da paragem: Almada

Rua_avenida e nº: Pc. S. João Baptista

Local: Almada

Distrito:

Distância (em Km): 9

Preço mínimo: 467 Esc.

Preço bandeirada: 250 Esc.

Preço total mínimo: 716,65 Esc.

Preço bagagem: 300 Esc.

Record: 1 of 1

Quando os planos correspondem à ligação secundária da situação 3, ou à ligação secundária posterior da situação 4, são fornecidos alguns dos números de telefone das centrais de táxis da localidade destino da ligação principal e algumas das moradas das praças de táxis dessa localidade. A informação fornecida é a mesma indicada anteriormente para as outras situações.

Para tomar conhecimento sobre as categorias de viagem de cada transporte inter-urbano, os preços dos bilhetes e as características de cada categoria, existe o botão “Categorias e Preços”.

Classes e Preços

Categorias e Preços

Nome da categoria:	<input type="text" value="Unica"/>	Marcação de lugar:	<input checked="" type="checkbox"/>
Idades que dispensam bilhete:	<input type="text" value="Até aos 5 anos"/>	Bar:	<input type="checkbox"/>
Idade para adquirir apenas meio bilhete:	<input type="text" value="Dos 5 aos 10 anos"/>	Restaurante:	<input type="checkbox"/>
Preço de ida:	<input type="text" value="1.500,00 Esc."/>	Lugares cama:	<input type="checkbox"/>
Preço de ida e volta:	<input type="text" value="3.750,00 Esc."/>	WC:	<input checked="" type="checkbox"/>
Informações:	<input type="text"/>	Ar condicionado:	<input checked="" type="checkbox"/>
		Video:	<input checked="" type="checkbox"/>
		TV:	<input type="checkbox"/>
		Telemóvel:	<input checked="" type="checkbox"/>

Record: 1 of 1

Para os transportes urbanos existe o botão “Nº bilhetes e Preços”, que fornece o número de bilhetes necessários, os vários tipos de bilhetes que podem ser utilizados no transporte em questão e os seus preços.

Bilhetes e preços

Bilhetes e Preços

Número de obliterações necessárias:	<input type="text" value="1"/>
Nome do transporte:	<input type="text" value="Autocarro"/>
Idades que dispensam bilhete:	<input type="text" value="Até aos 4 anos"/>
Idades para adquirir apenas meio bilhete:	<input type="text" value="Até aos 12 anos"/>

Bilhetes:		Passes:	
Nome do bilhete:	<input type="text" value="Normal"/>	Nome do passe:	<input type="text" value="Quarteira-Faro e vice-versa"/>
Número mínimo de bilhetes:	<input type="text" value="1"/>	Preço:	<input type="text" value="8.020,00 Esc."/>
Número de obliterações:	<input type="text" value="1"/>		
Preço:	<input type="text" value="400,00 Esc."/>		

Record: 1 of 2

São também fornecidas a morada e outras informações relativas aos locais onde se podem adquirir os bilhetes. Os locais apresentados correspondem aos sítios onde é possível comprar os bilhetes nas localidades de partida de cada plano.

Locais onde adquirir bilhetes

Nome do local: Terminal Rodoviário, 45

Telefones:

Faxes:

Horário:

Rua_avenida e n°: Av. da Liberdade

Local: Almada

Código postal: 2800 Almada

Concelho: Almada

Distrito: Almada

Record: 1 of 2

O botão “Escolhe os melhores planos de transportes” permite determinar os melhores planos de transportes, apresentando-os por ordem decrescente de importância. O número de planos apresentados corresponde ao número de planos que obtiveram as duas melhores somas na matriz de decisão construída pelo método da Pesagem Simples Aditiva.

TRANSPORTES POSSÍVEIS

MELHORES TRANSPORTES

PARTIDA:

Local de partida: Lisboa

Distrito: Lisboa

Rua_avenida e n°: Rotunda do Aeroporto

Hora de partida: 15:20:00

N° do transporte: NI 205

Nome do transporte: Avião

Data: 3 de Janeiro

CHEGADA:

Local de chegada: Faro

Distrito: Faro

Rua_avenida e n°: Estrada do Aeroporto

Hora de chegada: 16:00:00

Este plano de transportes inclui:

Táxi(s):

Transporte(s) secundário(s):

Locais onde adquirir bilhetes Categorias e Preços SAIR Táxi Transportes secundários

Record: 1 of 2

A informação fornecida por esta janela é a mesma apresentada nas janelas correspondentes aos planos de transportes possíveis.

