

3º Relatório da Disciplina

# Bases de Dados

---

Mestrado Integrado em Engenharia Informática e Computação

**Professor Sérgio Nunes**

**Grupo 704**

Diana Silva Viana

Ricardo Oliveira Neto Leite

Rui Manuel Santos Barros

1 de Junho de 2014

# 1 Introdução

O presente relatório da disciplina de Bases de Dados do Mestrado Integrado em Engenharia Informática e Computação é a terceira e última entrega do trabalho prático de Base de Dados de 2013/2014. Este relatório tem como data limite de entrega 1 de Junho de 2014 pelas 23h55 e reparte-se em cinco secções.

As quatro primeiras secções advêm de correções das versões entregues na primeira e segunda entrega e estas são: descrição do contexto em que se insere esta base de dados e qual a sua pertinência; na segunda secção, por sua vez, define-se os principais conceitos da mesma; na terceira secção, apresenta-se o diagrama de classes UML e, na quarta secção, o seu esquema relacional.

Neste terceiro e último relatório, para além do acréscimo das correções feitas ao já anteriormente realizado adiciona-se a quinta secção: Interrogações à base de Dados em que se encontram 8 queries e 2 triggers. A última secção é apresentada neste relatório e nos ficheiros .sql que seguem com esta entrega.

# 2 Descrição do contexto

A base de dados que vamos criar tem como objetivo servir um site de boleias entre doze principais cidades de Portugal.

Quando um utilizador se regista no site ele pode agir como condutor ou como passageiro.

Como condutor pode criar boleias, receber pedidos, pagamentos e gerir as regras de utilização do seu carro. Por sua vez, como passageiro pode procurar a boleia que lhe interessa, verificar o ranking dos condutores, adicionar-se a uma viatura e avaliar condutor.

Cria-se desta forma uma comunidade de passageiros e de condutores que se juntam para partilhar uma viagem de carro e que com isso todos beneficiam: diminuindo custos de transporte, partilhando experiências e melhorando o ambiente!

### 3 Definições dos principais conceitos

Os principais conceitos da nossa base de dados são: utilizador, carro, boleia, trajeto, cidade, feedback e pagamento que se encontram de seguida devidamente definidos.

Cada utilizador em cada boleia pode assumir ou o papel de condutor ou o papel de passageiro. Cada boleia terá um condutor (responsável pela criação e gestão da boleia) e no máximo quatro passageiros. Sobre o utilizador será preciso guardar o seu nome, data de nascimento, sexo, email, username, password, os seus carros (permitindo-se no entanto que ele não adicione nenhum e comportando-se assim apenas como passageiro), os seus pagamentos, os seus feedbacks dados ou recebidos e as suas boleias como condutor e/ou como passageiro.

Sobre os carros teremos que armazenar o seu consumo, o seu ano de primeiro registo, as suas regras de utilização (poder ou não comer, poder ou não fumar, poder ou não levar animal de estimação de pequeno porte,...), as opções de conforto (como ar condicionado, bons estofos...), tipo de combustível, modelo e marca.

Sobre as boleias teremos que armazenar a sua data de partida, preço total (que se dividirá pelo número de utilizadores atuais no carro), número de lugares vagos, preço de utilização da mala, os seus utilizadores, os seus pagamentos, feedbacks e o seu trajeto. A todos os passageiros será sempre perguntado se pretende ou não utilização da mala.

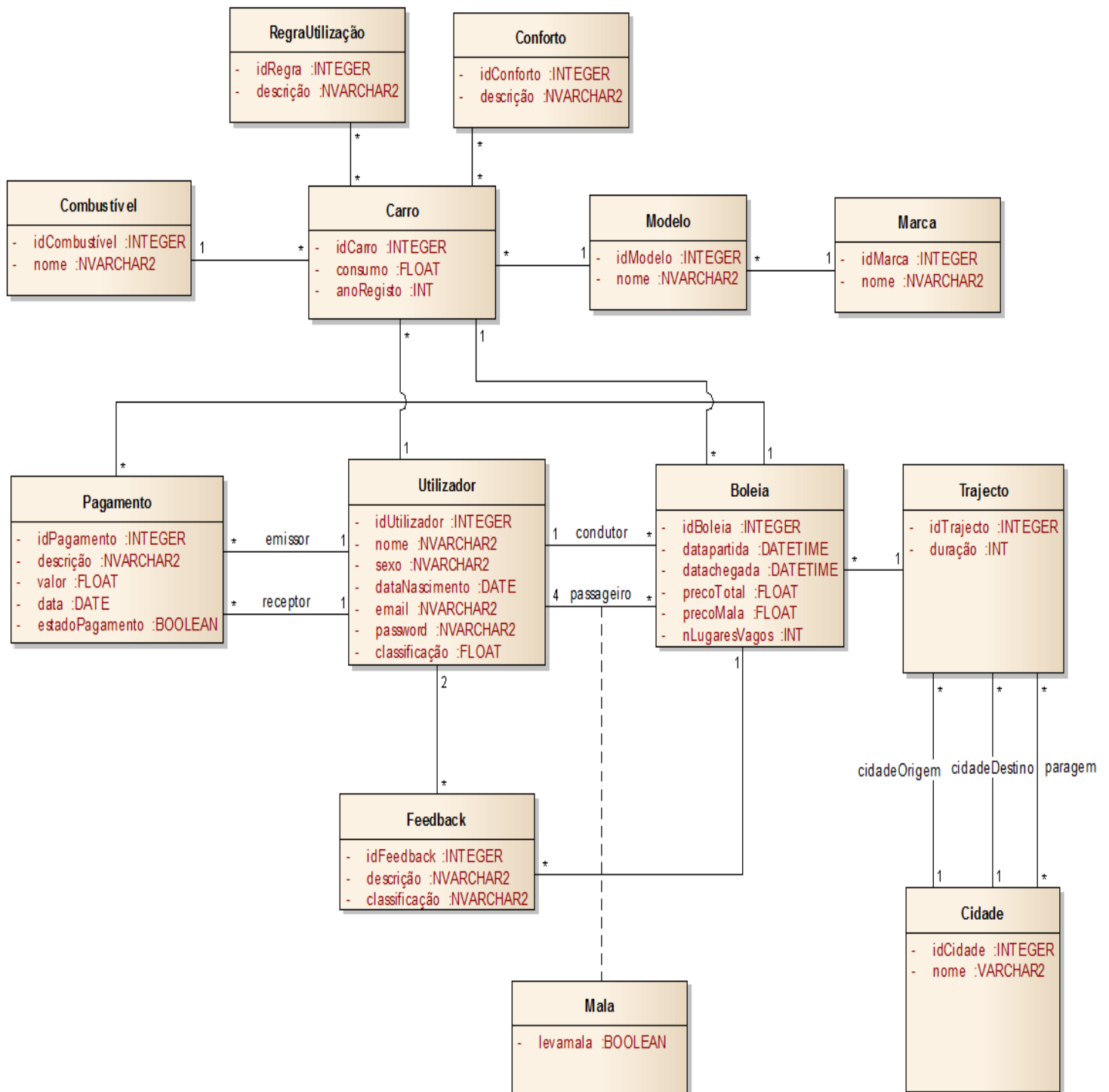
Cada trajeto tem uma determinada duração e pelo menos duas cidades: cidade origem e cidade destino. Permite-se ainda, se se entender vantajoso aos utilizadores, que se adicione de nenhuma a dez paragens ao trajeto. Paragem deve ser entendido como uma cidade entre a cidade destino e cidade origem.

Cada cidade pode tomar os seguintes valores: Porto, Braga, Vila Real, Viseu, Aveiro, Coimbra, Covilhã, Leiria, Caldas da Rainha, Lisboa, Évora, Faro.

Cada feedback tem de ter uma descrição associada, uma classificação e está associado a uma boleia.

Sobre os pagamentos teremos de armazenar uma descrição, o seu valor, data, estado e qual o utilizador pagante e recetor.

## 4 Diagrama de Classes UML



## 5 Esquema Relacional

**Utilizador**(idUtilizador, nome, sexo, dataNascimento, email, username, password)

**Marca**(idMarca, nome)

**Modelo**(idModelo, idMarca→Marca, nome)

**Combustivel**(idCombustivel, nome)

**Carro**(idCarro, consumo, anoRegisto, idUtilizador→Utilizador, idCombustivel→Combustivel, idModelo→Modelo)

**Condição**(idCondição, descrição)

**CarroCondição**(idCarro→Carro, idCondição→Condição)

**Conforto**(idConforto, descrição)

**CarroConforto**(idCarro→Carro, idConforto→Conforto)

**Cidade**(idCidade, nome)

**Trajecto**(idTrajecto, duração, idCidade→Cidade, idCidade→Cidade)

**Paragem**(idTrajecto→Trajecto, idCidade→Cidade)

**Boleia**(idBoleia, dataPartida, dataChegada, preçoTotal, preçoMala, nLugaresVagos, idTrajecto→Trajecto, idCarro→Carro, idCondutor→Condutor)

**Passageiro**(idUtilizador→Utilizador, idBoleia→Boleia, levamala)

**Feedback**(idFeedback, idBoleia→Boleia, emissor→Utilizador, receptor→Utilizador, descrição, classificação)

**Pagamento**(idPagamento, idBoleia→Boleia, emissor→Utilizador, receptor→Utilizador, descrição, valor, data, estadoPagamento)

## 5 Interrogações à base de dados

Dado que este projeto tem como objetivo servir um site de boleias entre doze principais cidades de Portugal, propõem-se as seguintes interrogações mais relevantes no contexto:

### 5.1 Queries

#### 5.1.1 TOP 10 Condutores com mais boleias criadas (ordenados pela sua classificação de forma decrescente)

```
.width 30 30 5 10 15
SELECT nome,email,CAST(strftime('%Y.%m%d','now') -
strftime('%Y.%m%d',datanascimento) as int) as idade, nboleias, classificacao FROM
Utilizador, (SELECT idUtilizador,COUNT(*) as nboleias FROM Utilizador, Boleia WHERE
Utilizador.idUtilizador = Boleia.idcondutor GROUP BY idUtilizador ORDER BY nboleias
DESC LIMIT 10) as v WHERE Utilizador.idUtilizador = v.idUtilizador ORDER BY
classificacao DESC;
```

```
sqlite> .width 30 30 5 10 15
sqlite> SELECT nome,email,CAST(strftime('%Y.%m%d','now') - strftime('%Y.%m%d',datanascimento) as int) as idade, nboleias, classificacao FROM Utilizador
, (SELECT idUtilizador,COUNT(*) as nboleias FROM Utilizador, Boleia WHERE Utilizador.idUtilizador = Boleia.idcondutor GROUP BY idUtilizador ORDER BY nb
oleias DESC LIMIT 10) as v WHERE Utilizador.idUtilizador = v.idUtilizador ORDER BY classificacao DESC;
nome email idade nboleias classificacao
-----
António Saraiva antonios@gmail.com 31 9 5.0
Tiago Castro ttcastro@gmail.com 26 8 5.0
Carlos Miranda ccmiranda@gmail.com 27 12 4.6
Afonso Santos apdsantos@gmail.com 37 8 4.3
Adriana Lima adrianalima@orkut.com 30 8 4.0
Pedro Letra pedro.letra12@gmail.com 27 8 4.0
Joana Pinheiro joanapinheiro@gmail.com 24 8 3.8
Rui Barros ruimsbarros08@gmail.com 24 9 3.6
Diana Viana dianaviana@gmail.com 24 12 3.5
Ricardo Leite ricardoleite@gmail.com 24 14 3.4
sqlite> |
```

*A relevância desta instrução prende-se com o interesse de o utilizador poder conferir os condutores que se destacam pelo maior número de boleias criadas, aumentando assim a confiança nesse mesmo condutor. De salientar ainda que simultaneamente é cedida informação relativamente à classificação deste conjunto de condutores.*

#### 5.1.2 TOP 10 Condutores com melhor classificação (ordenados por numero de feedbacks dados de forma decrescente)

```
.width 30 15 10
SELECT nome, classificacao,nfeedbacks FROM Utilizador, (SELECT
idUtilizador,COUNT(*) as nfeedbacks FROM Utilizador, Boleia, Feedback WHERE
Utilizador.idUtilizador = Boleia.idcondutor AND Feedback.boleia = Boleia.idBoleia AND
Feedback.receptor = Boleia.idCondutor GROUP BY idUtilizador ORDER BY
Utilizador.classificacao DESC LIMIT 10) as v WHERE Utilizador.idUtilizador =
v.idUtilizador ORDER BY nfeedbacks DESC;
```

```
sqlite> .width 30 15 10
sqlite> SELECT nome, classificacao, nfeedbacks FROM Utilizador, (SELECT idUtilizador, COUNT(*) as nfeedbacks FROM Utilizador, Boleia, Feedback WHERE Utilizador.idUtilizador = Boleia.idCondutor AND Feedback.boleia = Boleia.idBoleia AND Feedback.receptor = Boleia.idCondutor GROUP BY idUtilizador ORDER BY Utilizador.classificacao DESC LIMIT 10) as v WHERE Utilizador.idUtilizador = v.idUtilizador ORDER BY nfeedbacks DESC;
```

nome	classificacao	nfeedbacks
Carlos Miranda	4.6	11
Afonso Santos	4.3	7
Renato Velho	4.4	5
Angela Almeida	4.4	5
Adriana Lima	4.0	5
António Saraiva	5.0	1
André Graça	5.0	1
Bruna Pereira	5.0	1
Tiago Castro	5.0	1
Tomás Andrade	5.0	1

A confiança no condutor é um dos pilares para a aderência a este site. Por isso, quanto mais feedbacks forem dados a um condutor mais fiabilidade lhe confere. Foi neste contexto que surgiu esta instrução, informando ao utilizador não apenas as melhores classificações mas ordenando-as também por número de feedbacks dados. Como exemplificado na imagem acima, a classificação de 4.6 do condutor Carlos Miranda baseada em onze feedbacks confere-lhe um nível de confiança maior do que a classificação de 5.0 do António Saraiva que é baseada em apenas um feedback.

### 5.1.3 Quantos passageiros já transportou cada condutor (ordenados por número decrescente de passageiros)

```
.width 30 15
SELECT nome, COUNT(*) as npassageiros FROM Utilizador, Boleia, Passageiro WHERE Utilizador.idUtilizador = Boleia.idCondutor AND Boleia.idBoleia = Passageiro.boleia GROUP BY idCondutor ORDER BY npassageiros DESC;
```

```
sqlite> .width 30 15
sqlite> SELECT nome, COUNT(*) as npassageiros FROM Utilizador, Boleia, Passageiro WHERE Utilizador.idUtilizador = Boleia.idCondutor AND Boleia.idBoleia = Passageiro.boleia GROUP BY idCondutor ORDER BY npassageiros DESC;
```

nome	npassageiros
Carlos Miranda	11
Diana Viana	8
Ricardo Leite	8
Afonso Santos	7
Rui Barros	5
Renato Velho	5
Adriana Lima	5
Rita Pereira	5
Angela Almeida	5
Joana Pinheiro	4
António Saraiva	1
João Moutinho	1
José Soeiro	1
André Graça	1
Pedro Letra	1
Maria Conceição	1
Bruna Pereira	1
Tiago Castro	1
Leonor Antunes	1
Tomás Andrade	1

Quanto mais passageiros um condutor transportou, mais experiência o mesmo adquiriu neste processo de boleias bem como lhe confere um maior nível de confiança.

### 5.1.4 Para um determinado trajeto qual a média dos preços de cada boleia (Trajeto com id=67 que corresponde a Porto->Coimbra)

```
.width 10 5
SELECT idTrajeto, ROUND(AVG(precoTotal),1) as media FROM Trajeto, Boleia
WHERE Trajeto.idTrajeto = Boleia.trajeto AND idTrajeto=67 GROUP BY idTrajeto;
```

```
sqlite> .width 10 5
sqlite> SELECT idTrajeto, ROUND(AVG(precoTotal),1) as media FROM Trajeto, Boleia WHERE Trajeto.idTrajeto = Boleia.trajeto AND idTrajeto=67 GROUP
BY idTrajeto;
idTrajeto  media
-----  -
67         28.2
sqlite>
```

O preço total de cada boleia é definido pelo condutor, por isso, podemos ter nesta base de dados duas boleias com partidas, chegadas e paragens iguais, isto é, com o mesmo trajeto mas com um preço total diferente. Foi neste contexto, que percebemos a necessidade de dar ao utilizador a informação do valor médio desse trajeto pois acreditamos que se aproximará de um valor mais justo do mesmo. Colocamos a título de exemplo a consulta do trajeto com id 67.

### 5.1.5 Condutores que realizam cada trajeto (ordenados por número de boleias criadas)

```
.width 30 10 8
SELECT nome, idTrajeto, COUNT(*) as nboleias FROM Boleia, Trajeto, Utilizador
WHERE Boleia.trajeto = Trajeto.idTrajeto AND Boleia.idCondutor =
Utilizador.idUtilizador GROUP BY trajeto,idCondutor ORDER BY nboleias DESC;
```

```
sqlite> .width 30 10 8
sqlite> SELECT nome, idTrajeto, COUNT(*) as nboleias FROM Boleia, Trajeto, Utilizador WHERE Boleia.trajeto = Trajeto.idTrajeto AND Boleia.idCondut
or = Utilizador.idUtilizador GROUP BY trajeto,idCondutor ORDER BY nboleias DESC;
nome                idTrajeto  nboleias
-----
Ricardo Leite       20         7
Carlos Miranda     54         7
Afonso Santos      120        7
António Saraiva    16         6
Rui Barros         20         6
Diana Viana        20         5
Pedro Letra        67         5
Tiago Castro       67         5
Leonor Antunes     67         5
Rita Pereira       78         5
Angela Almeida     78         5
Diana Viana        11         4
Ricardo Leite      11         4
Carlos Miranda     67         4
Renato Velho       200        4
Joana Pinheiro     221        4
Adriana Lima       221        4
João Moutinho      120        3
Joana Pinheiro     200        3
Adriana Lima       200        3
Rui Barros         15         2
Diana Viana        15         2
Ricardo Leite      15         2
António Saraiva    15         2
Pedro Letra        54         2
Tiago Castro       54         2
José Soeiro        120        2
Rui Barros         78         1
Diana Viana        78         1
Ricardo Leite      78         1
António Saraiva    78         1
Renato Velho       78         1
Joana Pinheiro     78         1
Adriana Lima       78         1
João Moutinho      78         1
José Soeiro        78         1
```



*Esta instrução revela a frequência de todos os condutores num determinado trajeto. Esta informação revela-se útil na medida em que, se determinado condutor já realizou várias vezes um trajeto, pode ser provável que o realize novamente. Desta forma, o utilizador pode contactar o mesmo para perceber se em breve está a pensar fazer esse trajeto.*

### 5.1.6 Para um determinado trajeto, qual a lista de condutores que o realizou, ordenados por ordem decrescente de número de boleias dadas (Trajeto com id=78 que corresponde a Porto->Leiria->Caldas da Rainha->Lisboa)

```
.width 30 10 8
SELECT nome, idTrajeto, COUNT(*) as nboleias FROM Boleia, Trajeto, Utilizador
WHERE Boleia.trajeto = Trajeto.idTrajeto AND Boleia.idConductor =
Utilizador.idUtilizador AND idTrajeto = 78 GROUP BY nome ORDER BY nboleias
DESC;
```

```
sqlite> .width 30 10 8
sqlite> SELECT nome, idTrajeto, COUNT(*) as nboleias FROM Boleia, Trajeto, Utilizador WHERE Boleia.trajeto = Trajeto.idTrajeto AND Boleia.idConductor = Utilizador.idUtilizador AND idTrajeto = 78 GROUP BY nome ORDER BY nboleias DESC;
nome                                idTrajeto  nboleias
-----
Angela Almeida                      78         5
Rita Pereira                        78         5
Adriana Lima                        78         1
Afonso Santos                       78         1
André Graça                         78         1
António Saraiva                     78         1
Bruna Pereira                       78         1
Carlos Miranda                      78         1
Diana Viana                         78         1
Joana Pinheiro                      78         1
José Soeiro                         78         1
João Moutinho                       78         1
Leonor Antunes                      78         1
Maria Conceição                     78         1
Pedro Letra                         78         1
Renato Velho                        78         1
Ricardo Leite                       78         1
Rui Barros                          78         1
Tiago Castro                        78         1
Tomás Andrade                       78         1
sqlite>
```

*Esta instrução tem o mesmo interesse da instrução anterior, com a diferença de que agora o utilizador consulta a frequência de todos os condutores que já efetuaram o trajeto que efetivamente ele pretende. Foi usado como exemplo o trajeto com id 78 que corresponde a Porto->Leiria->Caldas da Rainha->Lisboa.*

### 5.1.7 Num determinado período e trajeto quais são as boleias disponíveis, ordenadas de forma crescente por preço total (Trajeto com id=78 que corresponde a Porto->Leiria->Caldas da Rainha->Lisboa e Data de partida entre “2014-08-01” e “2014-09-01”)

```
.width 8 10 19 19 10 10 13
SELECT idBoleia, idTrajeto, datapartida, datachegada, precoTotal, precoMala,
nLugaresVagos FROM Boleia, Trajeto WHERE Trajeto.idTrajeto = Boleia.trajeto
AND datapartida BETWEEN "2014-08-01" AND "2014-09-01" AND idTrajeto=78 AND
nLugaresVagos > 0 GROUP BY idBoleia ORDER BY precoTotal ASC;
```

```

sqlite> .width 8 10 19 19 10 10 13
sqlite> SELECT idBoleia,idTrajecto, datapartida, datachegada, precoTotal, precoMala, nLugaresVagos FROM Boleia, Trajecto WHERE Trajecto.idTrajecto = Bo
leia.trajecto AND datapartida BETWEEN "2014-08-01" AND "2014-09-01" AND idTrajecto=78 AND nLugaresVagos > 0 GROUP BY idBoleia ORDER BY precoTotal ASC;
idBoleia  idTrajecto  datapartida      datachegada      precoTotal  precoMala  nLugaresVagos
-----
109      78          2014-08-14 17:40:00 2014-08-14 19:55:00 12.0      14.0      1
121      78          2014-08-11 10:00:00 2014-08-11 13:08:00 14.5      9.0       1
114      78          2014-08-25 09:40:00 2014-08-25 12:08:00 15.0      6.0       1
123      78          2014-08-13 09:40:00 2014-08-13 12:08:00 15.0      6.0       1
103      78          2014-08-25 14:30:00 2014-08-25 18:38:00 17.0      8.0       1
120      78          2014-08-10 10:00:00 2014-08-10 13:08:00 17.5      8.0       1
110      78          2014-08-25 10:00:00 2014-08-25 13:08:00 18.0      8.0       1
115      78          2014-08-02 16:00:00 2014-08-02 18:08:00 18.0      5.0       1
104      78          2014-08-15 10:00:00 2014-08-15 13:08:00 19.0      8.0       1
105      78          2014-08-25 10:00:00 2014-08-25 13:08:00 20.0      8.0       1
106      78          2014-08-25 10:00:00 2014-08-25 13:08:00 20.0      8.0       1
111      78          2014-08-01 10:00:00 2014-08-01 13:08:00 20.0      8.0       1
116      78          2014-08-25 10:00:00 2014-08-25 13:08:00 20.0      8.0       1
124      78          2014-08-14 16:00:00 2014-08-14 18:08:00 21.0      5.0       1
125      78          2014-08-15 10:00:00 2014-08-15 13:08:00 22.0      8.0       1
102      78          2014-08-12 10:00:00 2014-08-12 13:08:00 25.0      9.0       1
108      78          2014-08-25 10:00:00 2014-08-25 13:08:00 25.0      10.0      1
113      78          2014-08-25 12:00:00 2014-08-25 17:08:00 25.0      8.0       1
118      78          2014-08-26 10:00:00 2014-08-26 13:08:00 25.0      8.0       1
122      78          2014-08-12 12:00:00 2014-08-12 17:08:00 25.0      8.0       1
127      78          2014-08-17 15:00:00 2014-08-17 18:08:00 25.0      8.0       1
128      78          2014-08-18 10:00:00 2014-08-18 13:08:00 25.0      8.0       1
101      78          2014-08-01 15:00:00 2014-08-01 18:08:00 27.0      6.0       1
126      78          2014-08-16 17:00:00 2014-08-16 20:08:00 27.0      8.0       1
107      78          2014-08-30 12:00:00 2014-08-30 15:08:00 30.0      8.0       1
112      78          2014-08-25 10:00:00 2014-08-25 13:08:00 30.0      9.0       1
117      78          2014-08-25 17:00:00 2014-08-25 20:08:00 30.0      8.0       1
119      78          2014-08-09 08:45:00 2014-08-09 11:08:00 31.0      3.0       1
sqlite>

```

*Esta consulta de informação é a mais frequente e fundamental neste tipo de sites. Um utilizador regista-se como passageiro e pesquisa para um trajeto que pretende efetuar num determinado período quais as boleias disponíveis, ordenados por preço total crescente.*

### 5.1.8 Número médio de lugares vagos por trajeto

```

.width 10 5
SELECT idTrajecto, ROUND(AVG(npassageiros),1) as media FROM Trajecto, (SELECT
trajeto, COUNT(*) as npassageiros FROM Passageiro, Boleia WHERE
Passageiro.boleia = Boleia.idBoleia GROUP BY idBoleia) as a WHERE a.trajeto =
Trajecto.idTrajecto GROUP BY idTrajecto;

```

```

sqlite> .width 10 5
sqlite> SELECT idTrajecto, ROUND(AVG(npassageiros),1) as media FROM Trajecto, (SELECT trajeto, COUNT(*) as npassageiros FROM Passageiro, Boleia WHERE
Passageiro.boleia = Boleia.idBoleia GROUP BY idBoleia) as a WHERE a.trajeto = Trajecto.idTrajecto GROUP BY idTrajecto;
idTrajecto  media
-----
11          3.0
20          4.0
54          2.0
78          1.0
120         2.0
200         3.7
sqlite>

```

*Esta instrução fornece informação relativa à adesão dos passageiros a um determinado trajeto. Por exemplo, a boleia com o id de trajeto 78, que corresponde a Porto->Leiria->Caldas da Rainha->Lisboa, tem como média de lugares vagos 1.0, o que revela que provavelmente quando se criar novamente uma boleia com este trajeto o condutor pode esperar a adesão a essa boleia de uma média de 3 passageiros. Num caso limite, podia mostrar ao utilizador que apesar de estar interessado em realizar o trajeto 77 Porto->Caldas da Rainha->Lisboa é provável que lhe compense passar por Leiria e realizar sim o trajeto 78 pois assim ganhará mais adesão à sua boleia.*

## 5.2 Triggers

### 5.2.1 Alerta antes de inserir um passageiro caso o número de lugares vagos para essa boleia seja zero

```
CREATE TRIGGER IF NOT EXISTS validaNpassageiros
BEFORE INSERT ON Passageiro
BEGIN
    SELECT RAISE(ABORT, 'Não pode inserir esse passageiro porque já não há lugares
vagos')
    WHERE EXISTS (SELECT * FROM Boleia
        WHERE idBoleia=NEW.boleia
        AND nLugaresVagos=0);
END;
```

*Este trigger pode ser testado com:*

```
INSERT INTO Passageiro (passageiro,boleia,levamala) VALUES (23,1,1);
```

### 5.2.2 Atualiza o número de passageiros de uma Boleia após a inserção de um novo passageiro nessa Boleia

```
CREATE TRIGGER IF NOT EXISTS atualizarPassageiros
AFTER INSERT ON Passageiro
BEGIN
    UPDATE Boleia set nLugaresVagos=(nLugaresVagos-1) WHERE
idBoleia=NEW.boleia;
END;
```

### 5.2.3 Atualiza o FeedBack de um Utilizador após a inserção de um feedback em que esse Utilizador é o recetor

```
CREATE TRIGGER IF NOT EXISTS atualizarFeedback
AFTER INSERT ON Feedback
BEGIN
    UPDATE Utilizador set classificacao=(SELECT ROUND(AVG(classificacao),1)
        FROM Feedback
        WHERE receptor=NEW.receptor)
    WHERE idUtilizador=NEW.receptor;
END;
```