



**Instituto Politécnico da Guarda**  
Escola Superior de Tecnologia e Gestão

**Caderno de exercícios  
SQL e MySQL**

**ESTE DOCUMENTO ESTÁ EM EDIÇÃO**  
**- FALTA TEXTO A EXPLICAR AS FIGURAS**  
**- CARECE DE REVISÕES NO TEXTO**  
**- NÃO IMPRIMIR!!**

11 de Dezembro de 2016

**Curso:** Comunicação Multimédia

**Unidade Curricular:** Base de Dados

**Ano Letivo:** 2016/2017

**Docente:** Paulo Jorge Costa Nunes

**Coordenador da área disciplinar:** José Fonseca



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>9</b>
<b>2</b>	<b>Registo de presenças de alunos nas aulas</b>	<b>11</b>
2.1	Registo de presenças . . . . .	12
2.2	Modelo ER . . . . .	13
2.3	Modelo físico em MySQL . . . . .	16
<b>3</b>	<b>Validação de dados</b>	<b>21</b>
3.1	SQL CHECK Constraint . . . . .	21
3.2	SQL DEFAULT Constraint . . . . .	23
3.3	Expressões regulares . . . . .	24
<b>4</b>	<b>Some of The Most Important SQL Commands</b>	<b>25</b>
<b>5</b>	<b>Exemplo normalização: Serviço Docente</b>	<b>27</b>
<b>6</b>	<b>Avaliação do pessoal docente do IPG</b>	<b>29</b>
6.1	Código SQL para criar tabelas . . . . .	29
<b>7</b>	<b>Lisboa Web Summit 2016</b>	<b>33</b>
<b>8</b>	<b>Formulário de Candidatura: Vigilante</b>	<b>35</b>
<b>9</b>	<b>Semana do Caloiro 2016</b>	<b>37</b>
9.1	Introdução . . . . .	37
9.2	Bilhete geral . . . . .	37
9.3	Bilhete geral situação 1 . . . . .	38
9.4	Bilhete geral situação 1 melhorada . . . . .	38
9.4.1	Script SQL para inserir dados nas tabelas . . . . .	40
9.5	Bilhete geral situação 2 . . . . .	42
9.6	Bilhete geral situação 2b . . . . .	44
9.7	Bilhete geral situação 3 . . . . .	45
9.8	Bilhete geral situação 4 . . . . .	46
9.9	Bilhete geral situação 5 . . . . .	46
9.10	Bilhetes diários . . . . .	46
9.11	Proposta de exercícios . . . . .	47
9.11.1	Bilhetes diários, situação 1 . . . . .	47
9.11.1.1	Normalização . . . . .	48
9.11.1.2	Limitações . . . . .	50
9.11.1.3	Refinamento 1 . . . . .	51
9.11.1.4	Refinamento 2 . . . . .	53
9.11.1.5	Conclusões . . . . .	58

9.11.2	Bilhetes diários, situação 2 . . . . .	59
<b>10</b>	<b>Modelação de dados</b>	<b>61</b>
10.1	Operações com strings . . . . .	62
10.2	Função MID() . . . . .	62
10.3	Função LENGTH(str) . . . . .	62
10.4	Função UCASE() . . . . .	62
10.5	Função UCASE() . . . . .	62
10.6	Função LOCATE() . . . . .	63
10.7	Função SUBSTRING_INDEX() . . . . .	63
10.8	Função CONCAT() . . . . .	63
<b>11</b>	<b>Linguagem SQL</b>	<b>65</b>
11.1	Base de dados SCUT . . . . .	65
11.1.1	Estrutura das tabelas . . . . .	66
11.1.2	Conversão da tabela passagens . . . . .	66
11.2	Cópia e restauração de base de dados . . . . .	66
11.3	Criação e cópia de tabelas . . . . .	67
11.4	Datas e horas . . . . .	67
11.5	Consultas com uma tabela . . . . .	68
11.6	Lista de todos os e todos os dados . . . . .	68
11.6.1	Lista de diversas componentes nome dos donos . . . . .	68
11.7	Projeção e DISTINCT . . . . .	69
11.8	Lista de questões para resolver em aula prática . . . . .	74
11.8.1	Carros . . . . .	74
11.8.2	Donos . . . . .	75
11.8.3	Carros e Donos . . . . .	75
11.8.4	Cidades e Distâncias . . . . .	75
11.8.5	Passagens . . . . .	75
11.9	Perguntas direcionadas a cada aluno . . . . .	76
11.9.1	Escolha da pessoa a investigar . . . . .	76
11.10	Alteração dos dados dos donos para nomes de alunos . . . . .	77
11.11	JUNÇÃO DE VÁRIAS TABELAS - Consultas com duas ou mais tabelas . . . .	78
11.11.1	Aula 2016-12-05/06 . . . . .	80
<b>12</b>	<b>Cópia e restauração de base de dados</b>	<b>83</b>
12.1	Criação e cópia de tabelas . . . . .	83

# Lista de Figuras

2.1	Folha de assiduidade em aulas, Modelo PED.045.01 . . . . .	12
2.2	Diagrama entidade relacionamento: aluno, aula. . . . .	13
2.3	Modelo entidade relacionamento antes de ligar as entidades aluno e aula. . . . .	14
2.4	Modelo entidade relacionamento depois de ligar as entidades aluno e aula. . . . .	15
2.5	Tipos de dados em MySQL. . . . .	16
2.6	Criação de uma base de dados MySQL com o phpmyadmin. . . . .	17
2.7	Modelo físico em MySQL/phpmyadmin. . . . .	17
2.8	Inserir um aluno MySQL/phpmyadmin. . . . .	18
2.9	Ver o aluno inserido na figura 2.8. . . . .	18
5.1	Serviço Docente Curso Comunicação Multimédia . . . . .	28
6.1	Avaliação do pessoal docente do IPG grelha de atividades a avaliar e respetivas ponderações. . . . .	30
7.1	Formulário Web Summit 2016 (General Attendee). . . . .	33
8.1	Formulário de Candidatura . . . . .	35
9.1	Dados do aluno e do bilhete na mesma entidade. . . . .	38
9.2	Modelo físico MySQL . . . . .	39
9.3	Dados nas tabelas . . . . .	41
9.4	Erro ao tentar inserir um aluno em duplicado (mesmo número). . . . .	41
9.5	Erro ao tentar inserir um aluno sem indicação do bilhete. . . . .	42
9.6	Modelo físico MySQL . . . . .	43
9.7	Dois alunos. Um com bilhete e outro sem. . . . .	44
9.8	Modelo físico MySQL . . . . .	44
9.9	Modelo físico MySQL . . . . .	45
9.10	Dois alunos. Um com bilhete e outro sem. . . . .	45
9.11	. . . . .	46
9.12	. . . . .	46
9.13	. . . . .	47
9.14	. . . . .	50
9.15	. . . . .	53
9.16	. . . . .	53
9.17	. . . . .	54
9.18	. . . . .	57
9.19	. . . . .	57



# Listings

2.1	Script SQLtabelas registro presenças. . . . .	19
3.1	. . . . .	21
3.2	. . . . .	22
3.3	. . . . .	22
3.4	. . . . .	22
3.5	. . . . .	22
3.6	. . . . .	23
3.7	. . . . .	23
3.8	. . . . .	23
3.9	. . . . .	24
3.10	. . . . .	24
3.11	. . . . .	24
3.12	. . . . .	24
9.1	Script SQLpara criar a tabela aluno1. . . . .	40
9.2	Script SQLpara inserir dados nas tabelas. . . . .	40
9.3	Script SQLpara inserir dados nas tabelas. . . . .	40
9.4	Script SQLpara inserir dados nas tabelas. . . . .	44
9.5	Script SQLpara inserir dados nas tabelas. . . . .	45





# Capítulo 1

## Introdução

Com este caderno de exercícios pretende-se contribuir para o ensino da linguagem SQL na unidade curricular de Base de Dados do curso de Licenciatura em Comunicação Multimédia. Os exercícios apresentados neste caderno foram resolvidos no gestor de base de dados MySQL.

Os exercícios estão organizados por capítulos e de acordo com as funcionalidade da linguagem SQL/MySQL. No início de cada capítulo são apresentados alguns conceitos e de seguida são apresentados diversos exercícios resolvidos e explicados com bastante detalhe. No final de cada capítulo é proposta a resolução de uma lista de exercícios com grau de dificuldade semelhante.

Um base de dados é um conjunto de dados organizado de forma a facilitar o seu uso pelos seus utilizadores. OS dados devem ser armazenados de forma segura para evitar o acesso por utilizadores não autorizados.

Para manipular base de dados foram desenvolvidos programas específicos denominados de Sistema de Gestão de Base de Dados (SGBD).

O MySQL é um sistema de gestão de base de dados(SGBD) relacional, que utiliza a linguagem Structured Query Language (SQL) como interface. É atualmente um dos SGBD mais populares, com mais de 10 milhões de instalações em todo o mundo [1]. Em agosto de 2016 O MySQL estava em segundo lugar numa classificação TOP3 de SGBDs [2]. A tabela 1.1 apresenta a lista dos 6 primeiros SGBD classificados. Podemos verificar que os três primeiros da lista têm pontuações muito próximas e que a soma da pontuação dos restantes três é menos de metade da pontuação de qualquer um dos três primeiros.

Entre as empresas que utilizam o SGBD MySQL estão: NASA, Friendster, Banco Bradesco, Dataprev, HP, Nokia, Sony, Lufthansa, U.S. Army, U.S. Federal Reserve Bank, Associated Press, Alcatel, Slashdot, Cisco Systems, Google, entre outros [1].

Tabela 1.1: Classificação de SGBDs

<b>Classificação</b>	<b>SGBD</b>	<b>Pontuação</b>
1	Oracle	1425.56
2	MySQL	1354.03
3	Microsoft SQL Server	1211.55
4	PostgreSQL	316.35
5	DB2 Relational	181.19
6	Microsoft Access	123.31

O SQL, ou linguagem de consulta estruturada, é uma linguagem padrão para gestão de base de dados relacionais. O SQL foi desenvolvido no início dos anos 70 pela IBM para demonstrar a viabilidade da implementação do modelo relacional proposto por Edgar Frank Codd (matemático britânico). Muitas das características originais do SQL foram inspiradas na álgebra relacional [3].

Existem muitos recursos na Internet para desenvolvimento em MySQL, de seguida são apresentados alguns:

1. Site oficial do MySQL — <http://www.mysql.com>.
2. W3Schools - SQL Tutorial — <http://www.w3schools.com/sql/>.

## Capítulo 2

# Registo de presenças de alunos nas aulas

O registo da assiduidade dos alunos nas aulas permite-nos obter um conjunto enorme de informação quer sobre cada um dos alunos quer sobre a turma. Por exemplo, para cada aluno podemos saber o número e horas de aulas presenciadas, o número e horas de faltas. Esta informação pode ser resumida por diversos itens: disciplina, tipo de aula (teórica, prática, etc.), hora do dia, dia da semana, semana, mês, etc. Por exemplo, queremos saber quantas aulas da disciplina de Base de Dados o aluno número 5001234 frequentou às segundas-feiras durante o mês de outubro.

Uma das atribuições da Comissão Nacional de Protecção de Dados (CNPd) é [4] [5]:

"A CNPD é a autoridade nacional que tem como atribuição controlar e fiscalizar o cumprimento das disposições legais e regulamentares em matéria de protecção de dados pessoais, em rigoroso respeito pelos direitos do homem e pelas liberdades e garantias consagradas na Constituição e na lei."

A CNPD para garantir os principais direitos de cada cidadão, a saber:

1. Direito de informação
2. Direito de acesso
3. Direito de retificação e eliminação
4. Direito de oposição
5. Outros Direitos

Os programadores de aplicações informáticas devem conhecê-los e garantir que as aplicações que desenvolvem os implementem corretamente.

Por exemplo, no direito 3, (Direito de retificação e eliminação) um cidadão pode exercer do direito de eliminação dos seus dados diretamente junto do responsável pelo tratamento dos dados. Assim a aplicações deve ter uma funcionalidade que permita ao responsável pelos dados eliminar imediatamente os dados do cidadão.

## 2.1 Registo de presenças

Neste exercício pretende-se desenvolver um modelo ER e respetiva implementação em MySQL para uma base de dados que permita registar as presenças dos alunos no IPG. Atualmente o registo é efetuado em papel. A figura 2.1 mostra o aspecto do modelo Modelo PED.045.01 para o registo de assiduidade dos alunos em aulas. Com base neste modelo podemos identificar as entidades apresentadas na tabela 2.1. No entanto, nesta primeira fase poderiam ter sido identificados outros conjuntos de entidades, dependendo da experiência e conhecimento do desenvolvedor do modelo ER.

IPG Politécnico da Guarda Polytechnic of Guarda		FOLHA DE ASSIDUIDADE EM AULAS						MODELO PED.045.01 Pág. ____ / ____	
<b>1. IDENTIFICAÇÃO DA UNIDADE CURRICULAR / CURSO / DOCENTE</b>									
Unidade curricular: _____				Curso: _____					
Ano letivo: _____ / _____		Tipo: <input type="checkbox"/> 1.º sem. <input type="checkbox"/> 2.º sem. <input type="checkbox"/> Anual		Ano curricular: _____		Turma: _____			
U.O.: <input type="checkbox"/> ESECD <input type="checkbox"/> ESS <input type="checkbox"/> ESTG <input type="checkbox"/> ESTH		Docente: _____				Assinatura: _____			
<b>2. REGISTO DE PRESENCAS</b>									
N.º Estudante	Nome	Datas							
		__/__/__	__/__/__	__/__/__	__/__/__	__/__/__	__/__/__	__/__/__	__/__/__
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
		Rubrica do Docente:	Rubrica do Docente:	Rubrica do Docente:	Rubrica do Docente:	Rubrica do Docente:	Rubrica do Docente:	Rubrica do Docente:	Rubrica do Docente:
		_____	_____	_____	_____	_____	_____	_____	_____

Figura 2.1: Folha de assiduidade em aulas, Modelo PED.045.01

Podemos facilmente concluir que o número de folhas de assiduidade necessárias para registar as presenças dos alunos numa **unidade curricular** de um **curso** pertencente a uma **escola** e leccionadas por um **docente**, dependem quer do número de alunos quer do número de aulas. Note-se que uma disciplina pode ser lecionada por vários docentes. Assim, o número de folhas

Tabela 2.1: Entidades identificadas no Modelo PED.045.01

Entidade	Atributos
Folha de assiduidade	
Unidade curricular	Nome
Curso	Nome
Docente	Nome
Registo de presença	Nº estudante, nome do estudante, data1, data2, ...
Estudante	Nº estudante, Nome

depende ainda do número de docentes. De facto, cada docentes tem de manter as presenças dos alunos nas suas aulas.

Considere-se o exemplo da unidade curricular de **Base de Dados** do curso de **Comunicação Multimédia**. Com 40 alunos inscritos à unidade curricular e um número de aulas previstas de 30, são necessárias 10 folhas calculadas pela seguinte fórmula:  $ArredondaParaCima(40/22) \times ArredondaParaCima(30/7) = 2 \times 5 = 10$ . Onde 22 é o número de alunos por folha e 7 é o número de aulas por folha.

Para simplificar o modelo ER vamos considerar que apenas pretendemos registar as presenças para uma dada unidade curricular. Desta forma pretendemos registar os **alunos** que estiveram **presentes** em cada **aula**. O relacionamento entre as entidades aluno e aula é de muitos para muitos sem obrigatoriedade de ambos os lados. O seja, **um aluno** pode estar presente em **zero ou muitas aulas**. Em **uma aula** podem estar **zero ou muitos alunos**.



Figura 2.2: Diagrama entidade relacionamento: aluno, aula.

## 2.2 Modelo ER

A figura 2.3 mostra uma captura de ecrã do programa **MySQL Workbench** com o modelo entidade relacionamentopara o registo de presenças. A figura 2.4 mostra outra captura de ecrã após o utilizador efetuar uma ligação de **n** para **n** entre as entidades aluno e aula. O programa adicionou uma terceira tabela (aluno\_has\_aula) automaticamente.

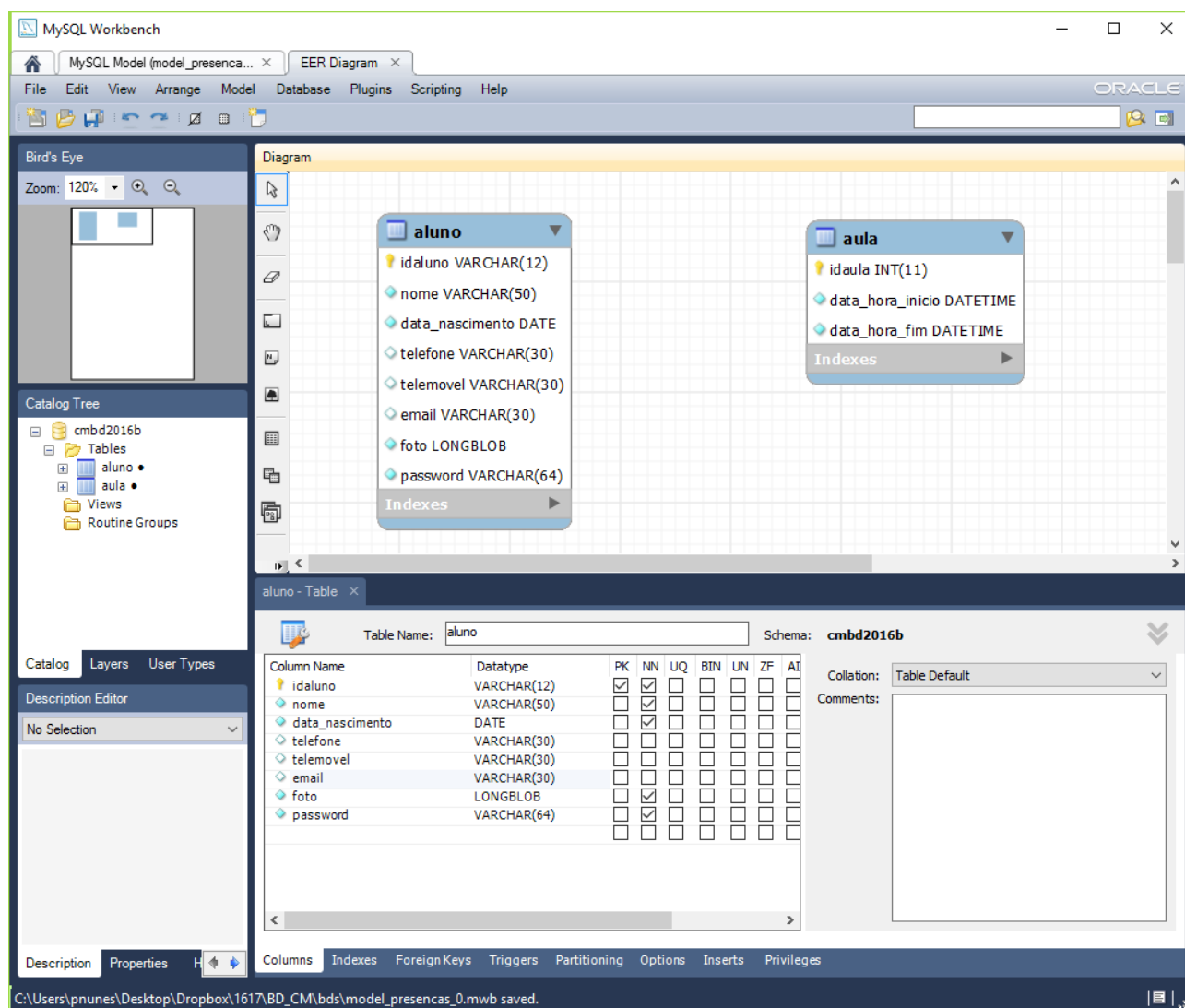


Figura 2.3: Modelo entidade relacionamento antes de ligar as entidades aluno e aula.

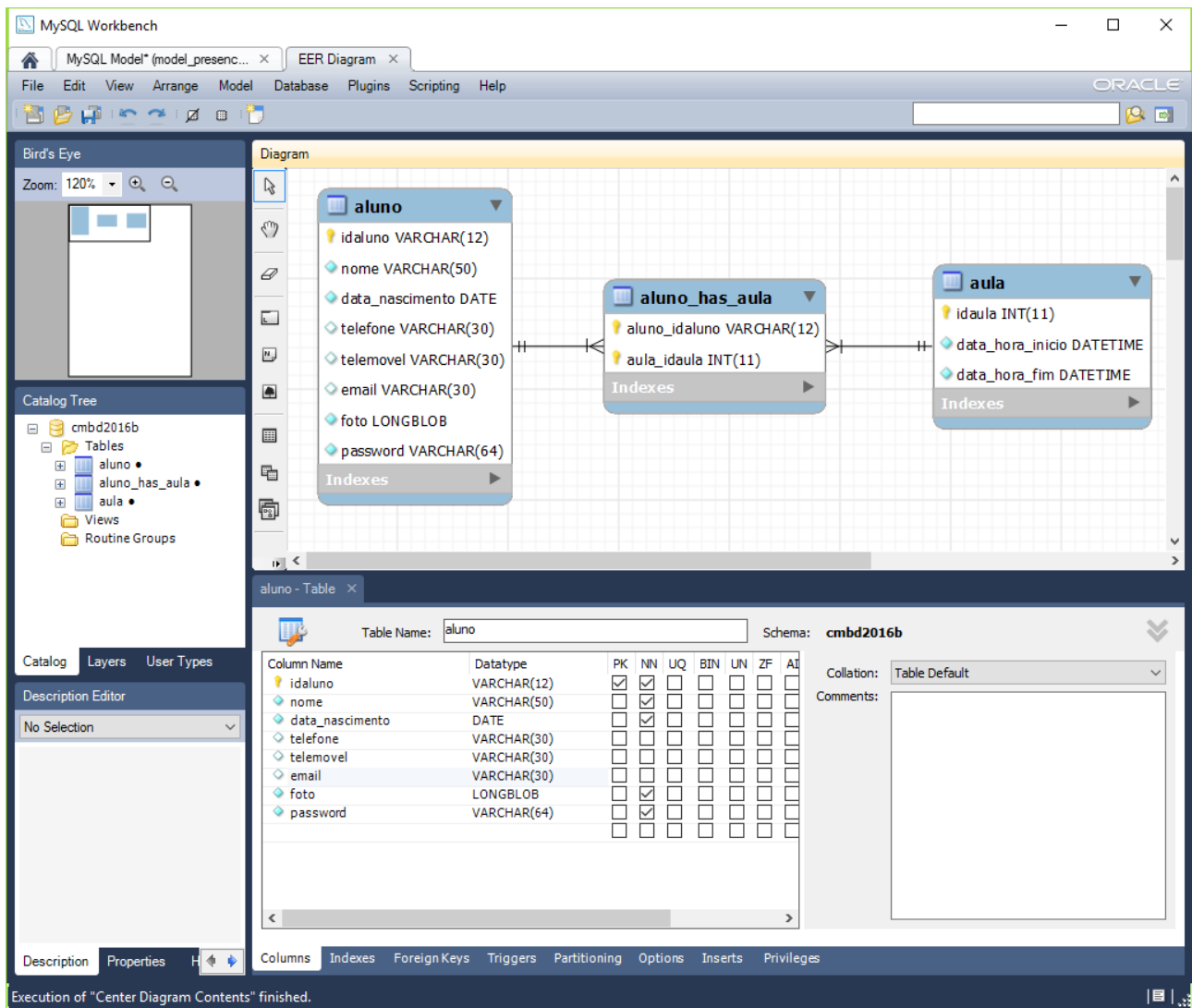
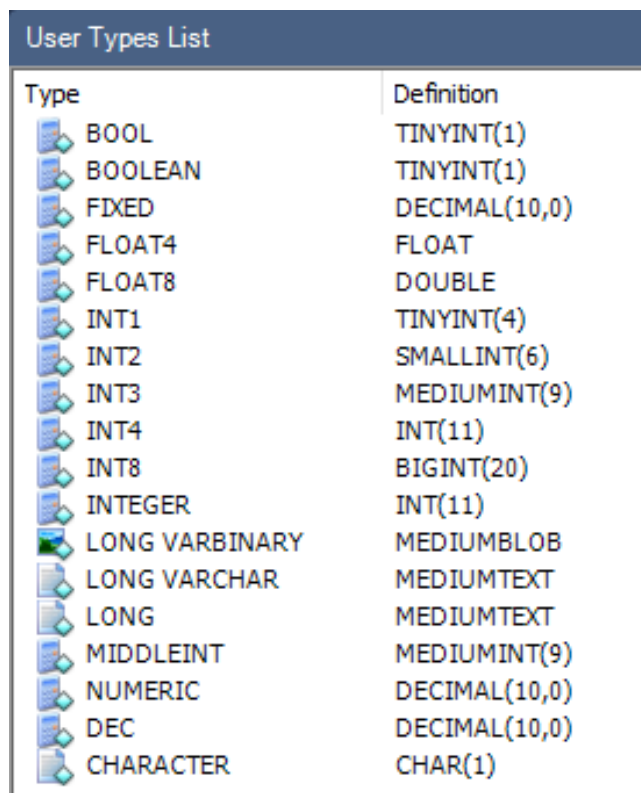


Figura 2.4: Modelo entidade relacionamentodepois de ligar as entidades aluno e aula.



Type	Definition
BOOL	TINYINT(1)
BOOLEAN	TINYINT(1)
FIXED	DECIMAL(10,0)
FLOAT4	FLOAT
FLOAT8	DOUBLE
INT1	TINYINT(4)
INT2	SMALLINT(6)
INT3	MEDIUMINT(9)
INT4	INT(11)
INT8	BIGINT(20)
INTEGER	INT(11)
LONG VARBINARY	MEDIUMBLOB
LONG VARCHAR	MEDIUMTEXT
LONG	MEDIUMTEXT
MIDDLEINT	MEDIUMINT(9)
NUMERIC	DECIMAL(10,0)
DEC	DECIMAL(10,0)
CHARACTER	CHAR(1)

Figura 2.5: Tipos de dados em MySQL.

## 2.3 Modelo físico em MySQL

A 2.6 ilustra a utilização da aplicação web **phpmyadmin** para criar uma base de dados MySQL com o nome **presencasA** e codificação de caracteres em **cp1250\_general\_ci** (permite armazenar caracteres de várias línguas da Europa Central).



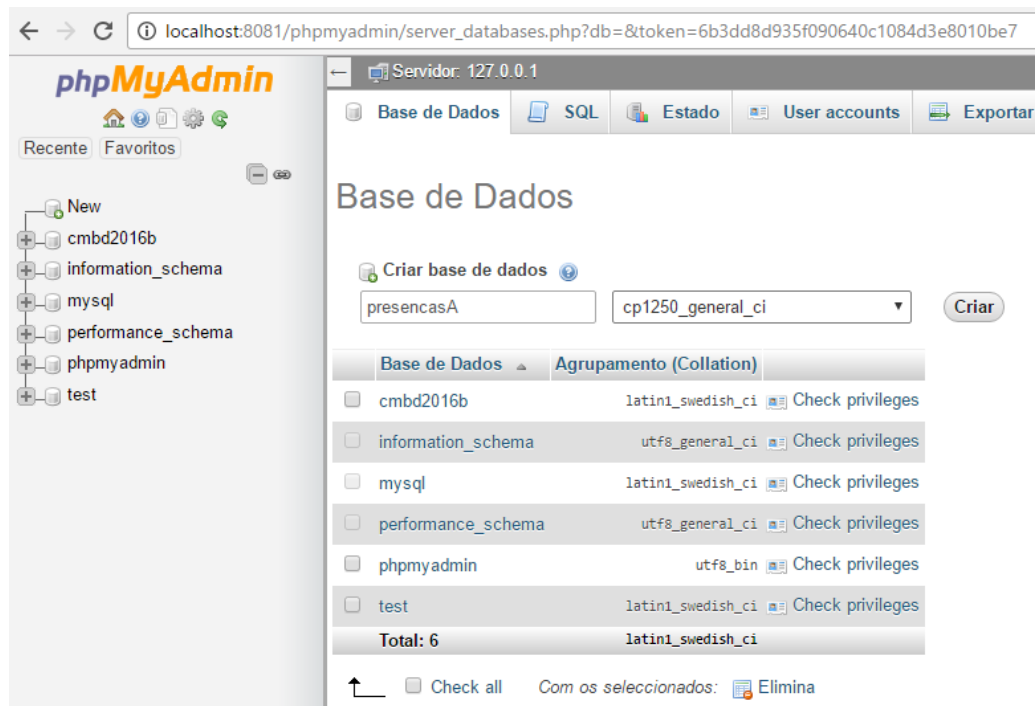


Figura 2.6: Criação de uma base de dados MySQL com o phpmyadmin.

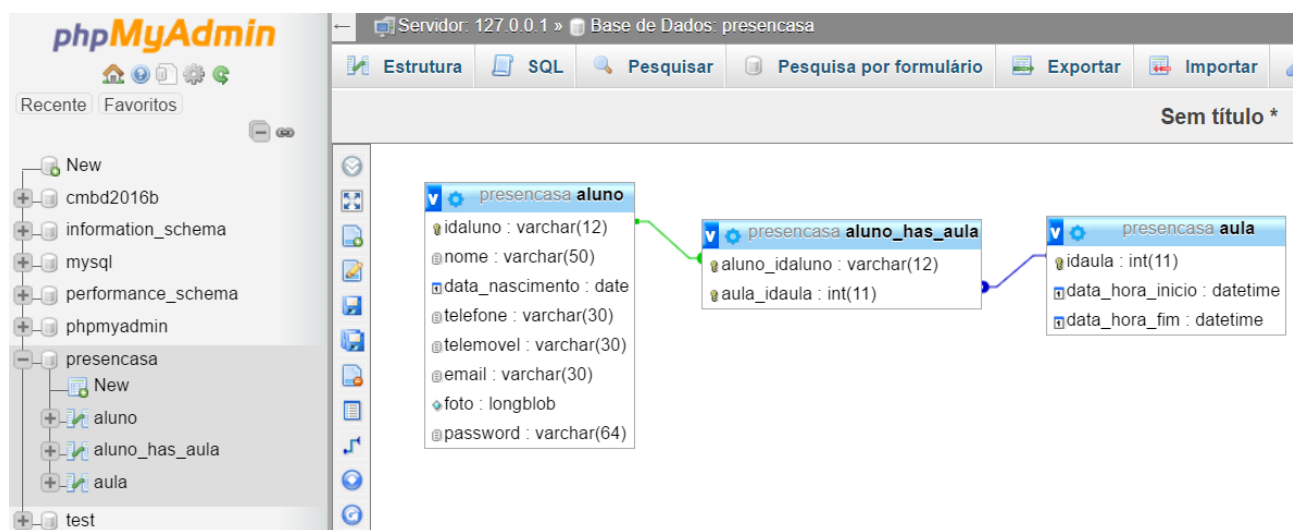


Figura 2.7: Modelo físico em MySQL/phpmyadmin.

phpMyAdmin

Recente Favoritos

Novo

cmbd2016b

Novo

aluno

aula

presenca

information\_schema

mysql

performance\_schema

phpmyadmin

presencasa

Novo

aluno

aluno\_has\_aula

aula

test

Servidor: 127.0.0.1 » Base de Dados: presencasa » Tabela: aluno

Procurar Estrutura SQL Pesquisar Insere Exportar Importar Privilégios Operações

Coluna	Tipo	Funções	Nulo	Valor
idaluno	varchar(12)			5007514
nome	varchar(50)			Sónia Gabriela dos Santos Soares
data_nascimento	date			1988-10-01
telefone	varchar(30)		<input checked="" type="checkbox"/>	
telemovel	varchar(30)		<input checked="" type="checkbox"/>	
email	varchar(30)		<input checked="" type="checkbox"/>	
foto	longblob			Binário - não editar (0 Bytes) Choose File No file chosen (Tamanho máximo: 2,048KB)
password	varchar(64)			

Executar

Figura 2.8: Inserir um aluno MySQL/phpmyadmin.

phpMyAdmin

Recente Favoritos

Novo

cmbd2016b

Novo

aluno

aula

presenca

information\_schema

mysql

performance\_schema

phpmyadmin

presencasa

Novo

aluno

aluno\_has\_aula

aula

test

Servidor: 127.0.0.1 » Base de Dados: presencasa » Tabela: aluno

Procurar Estrutura SQL Pesquisar Insere Exportar Importar Privilégios Operações

✓ A mostrar registos de 0 - 0 (1 total, A consulta demorou 0.0006 segundos.)

SELECT \* FROM `aluno`

☐ Mostrar tudo | Número de registos: 25 | Filtrar registos: Pesquisar esta tabela

+ Opções

	idaluno	nome	data_nascimento	telefone	telemovel	email	foto	password
<input type="checkbox"/> Editar <input checked="" type="checkbox"/> Copiar <input checked="" type="checkbox"/> Apagar	5007514	Sónia Gabriela dos Santos Soares	1988-10-01	NULL	NULL	NULL		

☐ Check all Com os seleccionados: ☒ Editar ☒ Copiar ☒ Apagar ☒ Exportar

☐ Mostrar tudo | Número de registos: 25 | Filtrar registos: Pesquisar esta tabela

Operações resultantes das consultas

☒ Vista de impressão ☒ Exportar ☒ Mostrar gráfico ☒ Criar visualização

Figura 2.9: Ver o aluno inserido na figura 2.8.

```

1 SET @@OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
2 SET @@OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
3 SET @@OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';
4
5 -- Table `aluno`
6
7 CREATE TABLE IF NOT EXISTS `aluno` (
8   `idaluno` VARCHAR(12) NOT NULL ,
9   `nome` VARCHAR(50) NOT NULL ,
10  `data_nascimento` DATE NOT NULL ,
11  `telefone` VARCHAR(30) CHARACTER SET 'latin1' COLLATE 'latin1_general_ci' NULL ,
12  `telemovel` VARCHAR(30) CHARACTER SET 'latin1' COLLATE 'latin1_general_ci' NULL ,
13  `email` VARCHAR(30) NULL ,
14  `foto` LONGBLOB NOT NULL ,
15  `password` VARCHAR(64) NOT NULL ,
16  PRIMARY KEY (`idaluno`) )
17 ENGINE = InnoDB
18 DEFAULT CHARACTER SET = latin1;
19
20 -- Table `aula`
21
22 CREATE TABLE IF NOT EXISTS `aula` (
23   `idaula` INT(11) NOT NULL AUTO_INCREMENT ,
24   `data_hora_inicio` DATETIME NOT NULL ,
25   `data_hora_fim` DATETIME NOT NULL ,
26   PRIMARY KEY (`idaula`) )
27 ENGINE = MyISAM
28 DEFAULT CHARACTER SET = latin1
29 COLLATE = latin1_general_ci;
30
31 -- Table `aluno_has_aula`
32
33 CREATE TABLE IF NOT EXISTS `aluno_has_aula` (
34   `aluno_idaluno` VARCHAR(12) NOT NULL ,
35   `aula_idaula` INT(11) NOT NULL ,
36   PRIMARY KEY (`aluno_idaluno`, `aula_idaula`) ,
37   CONSTRAINT `fk_aluno_has_aula_aluno`
38     FOREIGN KEY (`aluno_idaluno`)
39     REFERENCES `aluno` (`idaluno`)
40     ON DELETE NO ACTION
41     ON UPDATE NO ACTION,
42   CONSTRAINT `fk_aluno_has_aula_aula1`
43     FOREIGN KEY (`aula_idaula`)
44     REFERENCES `aula` (`idaula`)
45     ON DELETE NO ACTION
46     ON UPDATE NO ACTION)
47 ENGINE = InnoDB
48 DEFAULT CHARACTER SET = latin1;
49
50 CREATE INDEX `fk_aluno_has_aula_aula1_idx` ON `aluno_has_aula` (`aula_idaula` ASC) ;
51 CREATE INDEX `fk_aluno_has_aula_aluno_idx` ON `aluno_has_aula` (`aluno_idaluno` ASC) ;
52
53 SET SQL_MODE=@@OLD_SQL_MODE;
54 SET FOREIGN_KEY_CHECKS=@@OLD_FOREIGN_KEY_CHECKS;
55 SET UNIQUE_CHECKS=@@OLD_UNIQUE_CHECKS;

```

Listing 2.1: Script SQLtabelas registro presenças.



# Capítulo 3

## Validação de dados

### 3.1 SQL CHECK Constraint

The CHECK constraint is used to limit the value range that can be placed in a column.

If you define a CHECK constraint on a single column it allows only certain values for this column.

If you define a CHECK constraint on a table it can limit the values in certain columns based on values in other columns in the row.

SQL CHECK Constraint on CREATE TABLE The following SQL creates a CHECK constraint on the "P\_Id" column when the "Persons" table is created. The CHECK constraint specifies that the column "P\_Id" must only include integers greater than 0.

```
1 CREATE TABLE Persons
2 (
3     P_Id int NOT NULL,
4     LastName varchar(255) NOT NULL,
5     FirstName varchar(255),
6     Address varchar(255),
7     City varchar(255),
8     CHECK (P_Id>0)
9 )
10
```

Listing 3.1: .

To allow naming of a CHECK constraint, and for defining a CHECK constraint on multiple columns, use the following SQL syntax:

```
1 CREATE TABLE Persons
2 (
3   P_Id int NOT NULL,
4   LastName varchar(255) NOT NULL,
5   FirstName varchar(255),
6   Address varchar(255),
7   City varchar(255),
8   CONSTRAINT chk_Person CHECK (P_Id>0 AND City='
      Sandnes ')
9 )
10
```

Listing 3.2: .

SQL CHECK Constraint on ALTER TABLE To create a CHECK constraint on the "P\_Id" column when the table is already created, use the following SQL:

MySQL / SQL Server / Oracle / MS Access:

```
1 ALTER TABLE Persons
2   ADD CHECK (P_Id>0)
3
```

Listing 3.3: .

To allow naming of a CHECK constraint, and for defining a CHECK constraint on multiple columns, use the following SQL syntax:

MySQL / SQL Server / Oracle / MS Access:

```
1 ALTER TABLE Persons
2   ADD CONSTRAINT chk_Person CHECK (P_Id>0 AND
      City='Sandnes ')
3
```

Listing 3.4: .

To DROP a CHECK Constraint To drop a CHECK constraint, use the following SQL:

SQL Server / Oracle / MS Access:

```
1 ALTER TABLE Persons
2   DROP CONSTRAINT chk_Person
3
```

Listing 3.5: .

MySQL:

```
1 ALTER TABLE Persons
2 DROP CHECK chk_Person
3
```

Listing 3.6: .

## 3.2 SQL DEFAULT Constraint

The DEFAULT constraint is used to insert a default value into a column.

The default value will be added to all new records, if no other value is specified.

SQL DEFAULT Constraint on CREATE TABLE The following SQL creates a DEFAULT constraint on the "City"column when the "Persons"table is created:

My SQL / SQL Server / Oracle / MS Access:

```
1 CREATE TABLE Persons
2 (
3 P_Id int NOT NULL,
4 LastName varchar(255) NOT NULL,
5 FirstName varchar(255),
6 Address varchar(255),
7 City varchar(255) DEFAULT 'Sandnes '
8 )
9
```

Listing 3.7: .

The DEFAULT constraint can also be used to insert system values, by using functions like GETDATE():

```
1 CREATE TABLE Orders
2 (
3 O_Id int NOT NULL,
4 OrderNo int NOT NULL,
5 P_Id int,
6 OrderDate date DEFAULT GETDATE()
7 )
8
```

Listing 3.8: .

SQL DEFAULT Constraint on ALTER TABLE To create a DEFAULT constraint on the "City"column when the table is already created, use the following SQL:

MySQL:

```
1 ALTER TABLE Persons
2 ALTER City SET DEFAULT 'SANDNES'
3
```

Listing 3.9: .

SQL Server / MS Access:

```
1 ALTER TABLE Persons
2 ALTER COLUMN City SET DEFAULT 'SANDNES'
3
```

Listing 3.10: .

Oracle:

ALTER TABLE Persons MODIFY City DEFAULT 'SANDNES' To DROP a DEFAULT Constraint To drop a DEFAULT constraint, use the following SQL:

MySQL:

T

```
1 ALTER TABLE Persons
2 ALTER City DROP DEFAULT
3
```

Listing 3.11: .

SQL Server / Oracle / MS Access:

```
1 ALTER TABLE Persons
2 ALTER COLUMN City DROP DEFAULT
3
```

Listing 3.12: .

### 3.3 Expressões regulares



# Capítulo 4

## Some of The Most Important SQL Commands

Some of The Most Important SQL Commands

Tabela 4.1

Comando	Designação
SELECT	extracts data from a database
UPDATE	updates data in a database
DELETE	deletes data from a database
INSERT INTO	inserts new data into a database
CREATE DATABASE	creates a new database
ALTER DATABASE	modifies a database
CREATE TABLE	creates a new table
ALTER TABLE	modifies a table
DROP TABLE	deletes a table
CREATE INDEX	creates an index (search key)
DROP INDEX	deletes an index



## Capítulo 5

### Exemplo normalização: Serviço Docente

Aplique as regras de normalização 1FN, 2FN e 3FN à tabela da figura 5.1. Acrescente os atributos que forem necessários. Apresente o modelo ER obtido.

Figura 5.1: Serviço Docente Curso Comunicação Multimédia

# Capítulo 6

## Avaliação do pessoal docente do IPG

Desenhe o diagrama ER o mais completo possível de forma a permitir avaliar o desempenho dos docentes do IPG. Considere a informação da figura 6.1.

A avaliação está dividida em três áreas (tabela 6.1). As áreas estão divididas em 16 sub-áreas (tabela 6.2). Cada sub-área tem vários critérios de avaliação (tabela 6.3).

Para cada docente deve ser possível registar todos os elementos de avaliação (critérios). Por exemplo, um docente foi orientador de Projetos/Estágios de Licenciatura de 15 alunos, então deve ser possível registar essas 15 ocorrências numa tabela para o efeito. Esta tabela deve ter pelo menos os seguintes campos: data de início, data de fim, descrição.

Tabela 6.1: Áreas

ID Área	Designação	Subtotal
1	Técnico-Científica	147,95
2	Pedagógica	290,75
3	Organizacional	94,46

### 6.1 Código SQL para criar tabelas

22122

Diário da República, 2.ª série — N.º 153 — 7 de agosto de 2015

<b>Artigo 9.º</b>	
<b>Cooperação</b>	
1 — O Faltante, em caso de divida ou insuficiência das informações prestadas através do Relatório de Atividades, tem competência para solicitar, em qualquer momento, aos órgãos executivo, científico e pedagógico, ou ao docente avaliado, os elementos necessários para proceder à avaliação final devendo essa solicitação ser feita por escrito e com indicação de prazo, o qual não poderá ser inferior a 10 dias úteis.	
2 — No caso de não serem fornecidos esses elementos, o Faltante, para além de informar o docente em causa, decidirá com os elementos disponíveis, podendo, no entanto, se assim o entender, aos seus competentes para o obter.	
<b>Artigo 10.º</b>	
<b>Classificação da avaliação de desempenho</b>	
1 — A classificação final da avaliação de desempenho tem por base a pontuação global atribuída através da grelha de critérios aprovada, sendo expressa em quatro classes de acordo com a seguinte correspondência:	
a) Excelente, pontuação superior a 90 %;	
b) Muito Bom, pontuação igual ou superior a 65 % e inferior ou igual a 90 %;	
c) Bom, pontuação igual ou superior a 40 % e inferior a 65 %;	
d) Insuficiente, pontuação inferior a 40 %.	
2 — A avaliação de desempenho negativa, para efeitos do disposto no ECPDESP e a expressa pela classificação "insuficiente".	
3 — A obtenção da menção de "Excelente" e "Muito Bom" resulta da obtenção de um mínimo de 30 pontos percentuais na componente técnico-científica, dos quais, pelo menos 10 pontos percentuais, devem ser obtidos na subcomponente "Resultados da Atividade de Investigação — Publicações e Conferências Científicas".	
4 — Quando da aplicação das diversas pontuações resulta uma pontuação global superior a 90 % e não seja alcançado o disposto no número anterior, a atribuição a pontuação de 95 % e a menção qualificada de "Muito Bom".	
<b>Artigo 11.º</b>	
<b>Alteração do Posicionamento Remuneratório</b>	
1 — Nos termos de n.º 4 do artigo 35.º-C do ECPDESP e obrigatória a alteração do posicionamento remuneratório sempre que um docente,	

no processo de avaliação de desempenho, tenha obtido, durante um período de seis meses consecutivos, a seguinte menção:

2 — Para efeitos de posicionamento remuneratório considera-se que o docente muda de posição quando tenha reunido uma pontuação mínima de 10 pontos;

3 — Para os efeitos previstos no número anterior, as classificações mencionadas e atribuídas a seguinte pontuação:

a) Excelente, corresponde a uma atribuição de 9 pontos no final do triénio, valendo anualmente 3 pontos;

b) Muito Bom, corresponde a uma atribuição de 6 pontos no final do triénio, valendo anualmente 2 pontos;

c) Bom, corresponde a uma atribuição de 3 pontos no final do triénio, valendo anualmente 1 ponto;

d) Insuficiente, corresponde a uma atribuição de 1 ponto negativo no final do triénio.

4 — A alteração do posicionamento remuneratório tem efeitos retroativos ao 1.º dia do ano seguinte ao qual foi alcançada a pontuação mínima necessária, em termos dos números anteriores.

5 — Sempre que por aplicação do disposto no artigo 35.º do ECPDESP não for possível proceder à alteração do posicionamento remuneratório, os docentes serão classificados de acordo com os pontos obtidos desde a última alteração de posicionamento remuneratório, subindo de escala, no dia 1 de janeiro de cada ano, os primeiros dessa lista, até que se esgote a verba disponível para o efeito em cada ano.

6 — Em caso de empate, os docentes serão ordenados por ordem decrescente da sua antiguidade no IPG.

7 — No caso civil imputado ao da ocorrência de alteração de posicionamento remuneratório, e independentemente do valor total acumulado de pontos obtidos no ano em que ocorre essa mudança, e sem apresentação de pontos sobranceiros, inicia-se o novo cômputo de pontos.

8 — Sempre que se verifique uma mudança de categoria, e consequentemente uma modificação remuneratória, causada os efeitos das avaliações de desempenho da anterior categoria, iniciando-se uma nova contagem com as avaliações de desempenho referentes à nova categoria, para efeitos de reposicionamento remuneratório.

**Artigo 12.º****Entrada em Vigor e Disposições Finais**

1 — As alterações ao sistema de avaliação previsto no presente regulamento entram em vigor após o final do ciclo de avaliação do desempenho (2015-2015).

2 — Eventuais dúvidas de aplicação de presente regulamento, serão decididas por despacho do Presidente do IPG.

**ANEXO****Avaliação do pessoal docente do IPG grelha de atividades a avaliar e respetivas ponderações**

Área	Subárea	Critérios	Ponderação		Mínimo de elementos	Ponderação máxima
			Pts	Utilidade		
Técnico-Científica	1 Formação académica Profissional (grau e pós-graduação)	1.1 Licenciatura	4		Se se consideram 6 mais elementos	10
		1.2 Pós-Graduação Provas Públicas	5			
		1.3 Mestrado	9			
		1.4 Doutoramento/Título Especialista (DL 306/2009)	10			
		1.5 Provas de Agregação	4			
	2 Resultados da Atividade de Investigação — Publicações e Conferências Científicas	2.1 Autor de obra completa ou tradução de livros científicos, excluindo livros que não a compilação de artigos já publicados	3,5	Per artigo	4	14
		2.2 Autor de artigos em revistas incluídas na ISI, SCIMAGO ou Google Scholar	2,5	Per artigo	4	10
		2.3 Autor de artigos incluídos em outras revistas com arbitragem científica ou Autor de capítulos em livros científicos com arbitragem	2	Per obra	2	4
		2.4 Coordenador ou Editor de publicações científicas multiautor				

(a)

22124

Diário da República, 2.ª série — N.º 153 — 7 de agosto de 2015

Área	Subárea	Critérios	Ponderação		Mínimo de elementos	Ponderação máxima
			Pts	Utilidade		
Técnico-Científica	4.4 Membro de comissões organizadoras de conferências técnico-científicas ou eventos científicos e artísticos	4.4.1	0,5	Per participação	3	1,5
		4.4.2	0,1	Per participação	3	0,3
		4.4.3	0,1	Per participação	4	0,9
		4.4.4	0,1	Per participação	5	0,5
		4.4.5	0,5	Per participação	3	1,5
	5 Investigação e Projeto	5.1	2	Per projeto	2	4
		5.2	1	Per projeto	2	2
		5.3	3	Per projeto	2	6
		5.4	1,5	Per projeto	2	3
		5.5	1	Per projeto	3	3
Pedagógica	6 Prémios e Patentes	6.1	1,5	Per prémio	2	3
		6.2	1,5	Per prémio	1	3
		6.3	2,5	Per prémio	1	2,5
		6.4	1,5	Per prémio	1	1,5
		6.5	1,5	Per prémio	1	1,5
	7 Experiência e Dedicção Docente	7.1	0,75	Per mo	10	7,5
		7.2	1	Per horas/semana	12	12
		7.3	0,75	Per unidade curricular	6	4,5
		7.4	0,4	Per unidade curricular	6	2,4
		7.5	4	Per manual	2	8
Organizacional	8 Qualidade do Desempenho Docente	8.1	5	Per empresa criada	1	5
		8.2	1,5	Per prémio	2	3
		8.3	1,5	Per prémio	1	3
		8.4	1,5	Per prémio	1	2,5
		8.5	1,5	Per prémio	1	2,5
	9 Formação ministrada	9.1	0,75	Per mo	10	7,5
		9.2	1	Per horas/semana	12	12
		9.3	0,75	Per unidade curricular	6	4,5
		9.4	0,4	Per unidade curricular	6	2,4
		9.5	4	Per manual	2	8
Organizacional	10 Qualidade do Desempenho Docente	10.1	5	Per empresa criada	1	5
		10.2	1,5	Per prémio	2	3
		10.3	1,5	Per prémio	1	3
		10.4	1,5	Per prémio	1	2,5
		10.5	1,5	Per prémio	1	2,5
	11 Formação ministrada	11.1	0,75	Per mo	10	7,5
		11.2	1	Per horas/semana	12	12
		11.3	0,75	Per unidade curricular	6	4,5
		11.4	0,4	Per unidade curricular	6	2,4
		11.5	4	Per manual	2	8
Organizacional	12 Qualidade do Desempenho Docente	12.1	5	Per empresa criada	1	5
		12.2	1,5	Per prémio	2	3
		12.3	1,5	Per prémio	1	3
		12.4	1,5	Per prémio	1	2,5
		12.5	1,5	Per prémio	1	2,5
	13 Formação ministrada	13.1	0,75	Per mo	10	7,5
		13.2	1	Per horas/semana	12	12
		13.3	0,75	Per unidade curricular	6	4,5
		13.4	0,4	Per unidade curricular	6	2,4
		13.5	4	Per manual	2	8

(c)

Diário da República, 2.ª série — N.º 153 — 7 de agosto de 2015

22123

Área	Subárea	Critérios	Ponderação		Mínimo de elementos	Ponderação máxima
			Pts	Utilidade		
Técnico-Científica	2.5 Autor de artigos publicados em atas de congressos indexados ao ISI ou a outros indexadores similares	2.5.1	3	Per artigo	3	6
		2.5.2	1	Per artigo	3	3
		2.5.3	0,1	Per resumo	3	0,3
		2.5.4	0,5	Per apresentação	3	1,5
		2.5.5	0,5	Per apresentação	3	1,5
	3 Orientação e Arbitragem	3.1	3	Per orientação	3	9
		3.2	1,5	Per orientação	4	6
		3.3	0,75	Per orientação	6	4,5
		3.4	0,5	Per orientação	6	3
		3.5	1,75	Per participação	3	5,25
Pedagógica	4.4 Membro de comissões organizadoras de conferências técnico-científicas ou eventos científicos e artísticos	4.4.1	0,5	Per participação	3	1,5
		4.4.2	0,1	Per participação	3	0,3
		4.4.3	0,1	Per participação	4	0,9
		4.4.4	0,1	Per participação	5	0,5
		4.4.5	0,5	Per participação	3	1,5
	5 Investigação e Projeto	5.1	2	Per projeto	2	4
		5.2	1	Per projeto	2	2
		5.3	3	Per projeto	2	6
		5.4	1,5	Per projeto	2	3
		5.5	1	Per projeto	3	3
Organizacional	6 Prémios e Patentes	6.1	1,5	Per prémio	2	3
		6.2	1,5	Per prémio	1	3
		6.3	2,5	Per prémio	1	2,5
		6.4	1,5	Per prémio	1	3
		6.5	1,5	Per prémio	1	1,5
	7 Experiência e Dedicção Docente	7.1	0,75	Per mo	10	7,5
		7.2	1	Per horas/semana	12	12
		7.3	0,75	Per unidade curricular	6	4,5
		7.4	0,4	Per unidade curricular	6	2,4
		7.5	4	Per manual	2	8

(b)

Diário da República, 2.ª série — N.º 153 — 7 de agosto de 2015

22125

Área	Subárea	Critérios	Ponderação		Mínimo de elementos	Ponderação máxima
			Pts	Utilidade		
Técnico-Científica	3.1 Autor de artigos publicados em atas de congressos indexados ao ISI ou a outros indexadores similares	3.1.1	0,04	Per hora de duração	150	6
		3.1.2	0,5	Per ação	1	0,5
		3.1.3	1	Per ação	1	1
		3.1.4	1,5	Per ação	1	1,5
		3.1.5	0,6	Per participação	4	0,6
	4 Participação em outras atividades académicas	4.1	0,75	Per unidade curricular	4	3
		4.2	2,75	Per curso	3	8,25
		4.3	2	Per continuação grupo	5	10
		4.4	0,5	Per aluno	15	7,5
		4.5	0,4	Per aluno	6	2,4
Pedagógica	5 Qualidade do Desempenho Docente	5.1	0,3	Per visita	3	0,9
		5.2	0,4	Per visita	3	1,2
		5.3	0,6	Per ponto na escala de classificação (0-5)	3	3
		5.4	3	Per registo	3	3
		5.5	0	N.º de registo = 1 ou 2	0	0
	6 Formação ministrada	6.1	0,75	Per mo	10	7,5
		6.2	1	Per horas/semana	12	12
		6.3	0,75	Per unidade curricular	6	4,5
		6.4	0,4	Per unidade curricular	6	2,4
		6.5	4	Per manual	2	8
Organizacional	7 Qualidade do Desempenho Docente	7.1	5	Per empresa criada	1	5
		7.2	1,5	Per prémio	2	3
		7.3	1,5	Per prémio	1	3
		7.4	1,5	Per prémio	1	2,5
		7.5	1,5	Per prémio	1	2,5
	8 Formação ministrada	8.1	0,75	Per mo	10	7,5
		8.2	1	Per horas/semana	12	12
		8.3	0,75	Per unidade curricular	6	4,5
		8.4	0,4	Per unidade curricular	6	2,4
		8.5	4	Per manual	2	8

(d)

Figura 6.1: Avaliação do pessoal docente do IPG grelha de atividades a avaliar e respetivas ponderações.

Tabela 6.2: Sub-áreas

ID Sub-área	ID Área	Designação
1	1	Formação académica/Profissional (graus e provas)
2	1	Resultados da Atividade de Investigação ? Publicações e Conferências Científicas
3	1	Orientação e Arbitragem
4	1	Reconhecimento técnico-científico
5	1	Investigação e Projetos
6	1	Prémios e Patentes
7	2	Experiência e Dedicção à Docência
8	2	Elaboração de Material Didático
9	2	Atualização pedagógica e técnico-científica
10	2	Participação em outras atividades académicas
11	2	Qualidade do Desempenho Docente
12	2	Cumprimento de prazos e outras obrigações profissionais
13	3	Prestação de serviços ao exterior, estudos/projetos ou pareceres elaborados
14	3	Formação ministrada
15	3	Participação em Órgãos Colegiais e Responsabilidade académica
16	3	Outras atividades

Tabela 6.3: Critérios

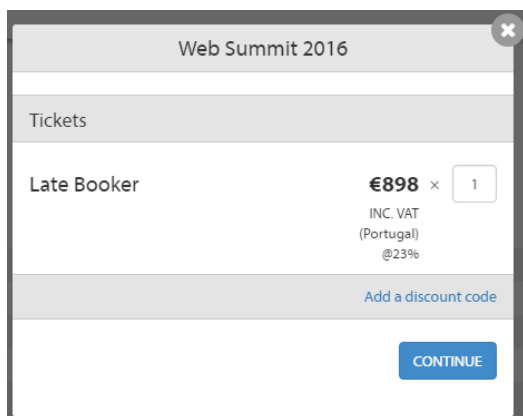
ID Cri- tério	ID Sub- área	Designação	Pontos	Unidade	Máximo elemen- tos	Pontuação
1	1	Licenciatura	4.00		0.00	10.00
2	1	Pós -Graduação/Provas Publicas	5.00		0.00	0.00
3	1	Mestrado	7.00		0.00	0.00
4	1	Doutoramento/Título Especialista(DL 206/2009)	9.00		0.00	0.00
5	1	Provas de Agregação	10.00		0.00	0.00
6	2	Autor de obra completa ou tradução de livros cient...	4.00	Por obra	2.00	8.00
7	2	Autor de artigos em revistas incluídas na ISI, SCI...	3.00	Por artigo	4.00	14.00
8	2	Autor de artigos incluídos em outras revistas com ...	2.00	Por artigo	4.00	10.00
9	2	Coordenador ou Editor de publicações científicas m...	2.00	Por obra	2.00	4.00
10	2	Autor de artigos publicados em atas de congressos ...	2.00	Por artigo	3.00	6.00
11	2	Autor de artigo publicado em atas de congressos co...	1.00	Por artigo	3.00	3.00
12	2	Publicação em livro de resumos	0.00	Por resumo	3.00	0.00
13	2	Comunicação oral	0.00	Por apresentação	3.00	1.00
14	2	Comunicação em poster	0.00	Por apresentação	3.00	1.00
15	2	Elaboração de relatório científico, técnico, cultu...	1.00	Por relatório	2.00	3.00
16	3	Orientação ou coorientação de Teses de Doutora- ment...	3.00	Por orientação	3.00	9.00
17	3	Orientação ou coorientação de Tese/Projeto de Mest...	1.00	Por orientação	4.00	6.00
18	3	Orientação de Projetos/Estágios de Licenciatura ou...	0.00	Por orientação	6.00	4.00
19	3	Orientação de Projetos/Estágios de outros cursos (...)	0.00	Por orientação	6.00	3.00
20	3	Júri de Tese de Doutoramento desde que não tenha s...	1.00	Por participação	3.00	5.00
21	3	Júri de Tese/Projetos Mestrado desde que não te- nha...	1.00	Por participação	4.00	4.00
22	3	Júri de avaliação em Relatório de Estágio de Bacha...	0.00	Por participação	6.00	2.00
23	3	Júri de avaliação em Relatório de Estágio de CET/C...	0.00	Por participação	6.00	1.00
24	3	Júri de Provas Académicas (ex <sup>o</sup> Especialista) públi...	1.00	Por participação	3.00	3.00
25	3	Júri de concursos (ou provas académicas) docu- menta...	1.00	Por participação	3.00	3.00
26	4	Avaliador/revisor de artigos científicos submetido...	1.00	Por artigo revisto	3.00	3.00
27	4	Avaliador/revisor de artigos científicos submetido...	1.00	Por artigo revisto	3.00	3.00
28	4	Membro de comissões científicas de conferências ou...	0.00	Por participação	3.00	2.00
29	4	Membro de comissões organizadoras de conferências ...	0.00	Por participação	3.00	1.00
30	4	Moderador de sessão em conferência	0.00	Por participação	3.00	0.00
31	4	Comunicações orais por convite	0.00	Por participação	4.00	2.00
32	4	Número de citações em revistas indexadas ao ISI, e...	0.00	Por cada citação	6.00	0.00
33	4	Número de citações em revistas não indexadas ao IS...	0.00	Por participação	5.00	0.00
34	4	Participação em redes de investigação internaciona...	0.00	Por participação	3.00	1.00
35	4	Autor ou coautor de exposições artísticas ou cient...	1.00	Por atividade	3.00	4.00
36	5	Responsável ou investigador principal de projeto c...	2.00	Por projeto	2.00	4.00
37	5	Membro colaborador da equipa de projeto com fi- nanc...	1.00	Por projeto	2.00	2.00
38	5	Responsável ou investigador principal de projeto c...	3.00	Por projeto	2.00	6.00
39	5	Membro colaborador da equipa de projeto com fi- nanc...	1.00	Por projeto	2.00	3.00
40	5	Membro de Centro de Investigação externo avaliado ...	1.00	Por ano completo	3.00	3.00
41	5	Júri ou avaliador de projetos de investigação fina...	1.00	Por participação	3.00	3.00
42	5	Criação de empresas oriundas do meio académico	5.00	Por empresa cri- ada	1.00	5.00
43	6	Prémio ou Distinção Nacional	1.00	Por prémio	2.00	3.00
44	6	Prémio ou Distinção Internacional	3.00	Por prémio	1.00	3.00
45	6	Patentes e protótipos registadas	2.00	Patente	1.00	2.00
46	7	Experiência profissional no ensino superior polité...	0.00	Por ano	10.00	7.00
47	7	Número médio anual de horas letivas efetivas de co...	1.00	Por horas/se- mana	12.00	12.00
48	7	Número de unidades curriculares com conteúdo di- fer...	0.00	Por unidade cur- ricular	6.00	4.00
49	7	N.º de unidades curriculares lecionadas com número...	0.00	Por unidade cur- ricular	6.00	2.00
50	8	Manuais e livros de texto de apoio à docência ou a...	4.00	Por manual	2.00	8.00
51	8	Elaboração de apontamentos	0.00	2,5	0.00	3.00
52	9	Cursos ou ações de atualização pedagógica, científ...	0.00	Por hora de dura- ção	150.00	6.00
53	9	Conclusão de parte escolar de Mestrado	0.00	Por ação	1.00	0.00
54	9	Conclusão de parte escolar de Doutoramento	1.00	Por ação	1.00	1.00
55	9	Conclusão de Pós-Doutoramento	1.00	Por ação	1.00	1.00
56	9	Participação em congressos, palestras, conferência...	0.00	Por participação	4.00	0.00
57	10	Participação na elaboração de programas de diferen...	0.00	Por unidade cur- ricular	4.00	3.00
58	10	Pessoa Responsável por pedido (PEP) de criação ou ...	2.00	Por curso	3.00	8.00
59	10	Participação em grupos ou comissões académicas, in...	2.00	Por comissão/- grupo	5.00	10.00
60	10	Orientação e acompanhamento de alunos em ensino cl...	0.00	Por aluno	15.00	7.00
61	10	Orientação e acompanhamento de estágios curricu- lar...	0.00	Por aluno	6.00	2.00
62	10	Organização de visitas de estudos	0.00	Por visita	3.00	0.00
63	10	Membro de comissão organizadora de outros con- gress...	0.00	Por evento	3.00	1.00
64	11	Resultados da avaliação pelos alunos	0.00	Por ponto, na es- cala de classifica- ção (0-5)	0.00	3.00
65	12	Cumprimento de prazos e outras obrigações profissi...	3.00	Sem registos	0.00	3.00
66	13	Responsável	1.00	Por atividade	6.00	6.00
67	13	Corresponsável ou participante	0.00	Por atividade	8.00	4.00
68	14	Lecionação de seminários, cursos de formação e dis...	0.00	Por hora de for- mação	150.00	9.00
69	15	Presidente de órgão de gestão estatutário (CTC, CP...	0.00	Por mês completo	36.00	7.00
70	15	Vice-presidente de órgão de gestão, quando estatut...	0.00	Por mês completo	36.00	2.00
71	15	Secretário de órgão de gestão, quando estatutariam...	0.00	Por mês completo	36.00	1.00
72	15	Membro de órgãos colegiais estatutários (CTC, CP, ...	0.00	Por mês completo	36.00	3.00
73	15	Coordenador de UTC	0.00	Por mês completo	36.00	5.00
74	15	Coordenador de curso	0.00	Por mês completo	36.00	4.00
75	15	Membro de Comissão Científica, Pedagógica ou Ci- ent...	0.00	Por mês completo	36.00	2.00
76	15	Responsável ou corresponsável por unidade/servi- ços...	0.00	Por mês completo	36.00	4.00
77	15	Responsável por Área disciplinar	0.00	Por mês completo	36.00	3.00
78	16	Presidente de júri de seleção/seriação (ex <sup>o</sup> mestrad...	2.00	Por participação	3.00	6.00



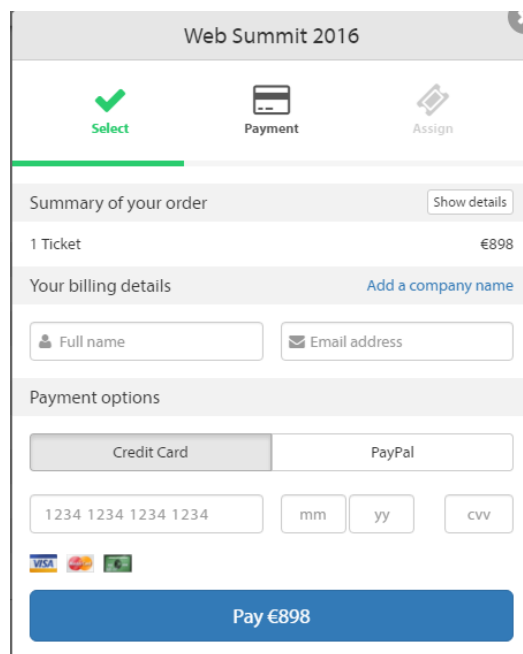
# Capítulo 7

## Lisboa Web Summit 2016

Desenhe o diagrama ER o mais completo possível de forma a permitir armazenar os dados contantes no formulário de compra de bilhete para o Web Summit<sup>1</sup>2016, ilustrado na figura 7.1.



(a) Quantidade



(b) Pagamento

Figura 7.1: Formulário Web Summit 2016 (General Attendee).

<sup>1</sup>[https://websummit.net/tickets?utm\\_source=pub-summit](https://websummit.net/tickets?utm_source=pub-summit)



## Capítulo 8

# Formulário de Candidatura: Vigilante

Desenhe o diagrama ER o mais completo possível de forma a permitir os dados contantes no Formulário de Candidatura<sup>1</sup> da figura 8.1.

**Formulário de Candidatura**  
**Vigilantes - Grande Lisboa e Setúbal**

**Identificação**

Nome \*

Telefone \*  Email \*

Nº. Identificação Fiscal \*  Género \*

**Dados Pessoais**

Morada \*

C. Postal \*

Estado Civil \*  Nacionalidade \*

Altura \*  Peso \*

Data Nascimento \*   
Formato AAAA-MM-DD

**Qualificações e Competências**

Habilitações Académicas

Tem Cartão Profissional?  Dt. Validade Cartão

Carta de Condução \*

Conhecimentos Inglês

Conhecimentos Informática

Outras Competências

Referências na Securitas?

**Candidatura**

Motivações e Apresentação \*

Figura 8.1: Formulário de Candidatura

<sup>1</sup><https://webtec.securitas.pt/jobs/apply/vigilantes-grande-lisboa-e-setubal-10>



# Capítulo 9

## Semana do Caloiro 2016

### 9.1 Introdução

A Semana do Caloiro do IPG organizado pela Associação Académica da Guarda realiza-se de 31 de outubro a 6 de novembro, no Pavilhão Municipal da Guarda. O estudantes têm direito a comprar bilhetes baratos para participar nas festas. O público em geral paga um pouco mais. Os alunos são livres de participar ou não nas festas. Assim, pode comprar um bilhete ou não. Nas secções seguintes são apresentados os modelos ER de acordo com diversas situações que se pretendem para o modelo. Na secção 9.2 é abordado o caso de existirem apenas bilhetes gerais e na secção 9.10 o caso em que existem apenas bilhetes diários.

### 9.2 Bilhete geral

Na tabela 9.1 são apresentados os preços dos bilhetes gerais.

Tabela 9.1: Precário de bilhetes gerais

Designação	Geral €
Geral alunos IPG	40,00
Geral	60,00

Desenhe o diagrama ER para cada uma das seguintes situações (restrições impostas pela Direção da AAG):

1. Um aluno **tem de comprar** um bilhete geral.
2. Um aluno **só pode comprar** um bilhete geral.

3. Um aluno **pode comprar vários** bilhetes gerais **de uma só vez**.
4. Um aluno **pode comprar vários** bilhetes gerais **um de cada vez**.
5. Um aluno **pode comprar vários** bilhetes gerais **várias vezes**.

### 9.3 Bilhete geral situação 1

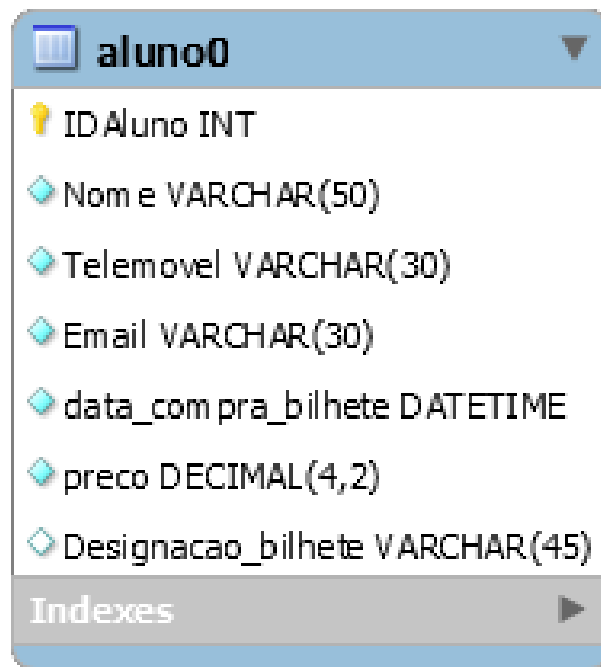


Figura 9.1: Dados do aluno e do bilhete na mesma entidade.

A entidade não está normalizada. Porquê?

### 9.4 Bilhete geral situação 1 melhorada

Um aluno **tem de comprar** um bilhete geral. O modelo ER da 9.2 deve ler-ser da seguinte maneira:

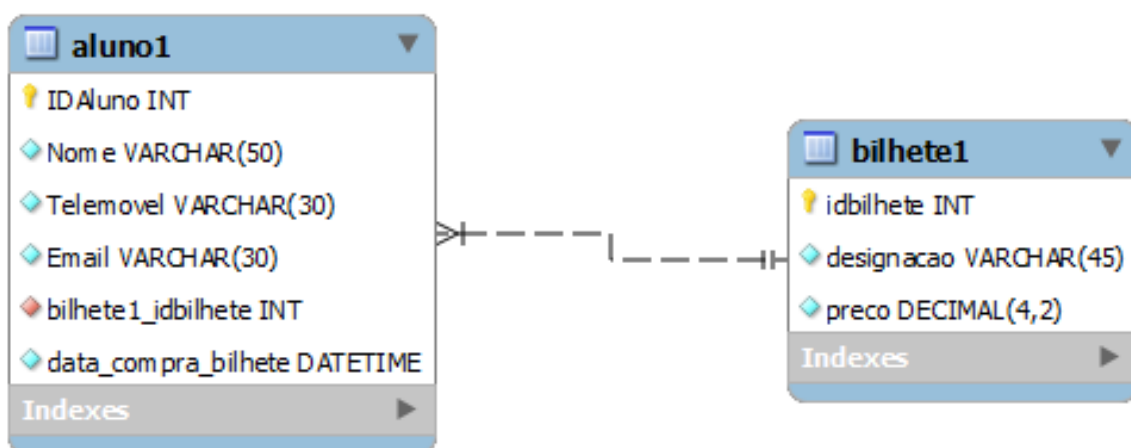
1. Cada *Aluno* **tem de Comprar um e só um** *Bilhete*.
2. e
3. Cada *Bilhete* **poder ser Comprado por um ou muitos** Alunos.

Os passos para a concepção física modelo de dados são os seguintes:

1. Converter entidades em tabelas.
2. Converter relações em chaves estrangeiras.
3. Converter atributos em colunas.
4. Modificar o modelo de dados físico com base em restrições físicas / necessidades.

Comparando o modelo de dados físico mostrado no diagrama da figura 9.2b com o modelo de dados lógico mostrado na figura 9.2a, vemos as principais diferenças entre os dois:

1. Os nomes de entidades estão agora nomes de tabela.
2. Os atributos são agora os nomes das colunas.
3. O tipo de dados para cada coluna é especificado. Os tipos de dados podem ser diferentes, dependendo da base de dados real que está sendo usado.



(a) Modelo ER. Relacionamento 1:n, obrigatório do lado n



(b) Pagamento

Figura 9.2: Modelo físico MySQL

Utilizando o modelo ER da figura 9.2b (b) não é possível inserir um aluno se não for indicado o bilhete. Isto significa que os dados do aluno só podem ser inseridos quando o aluno compra o bilhete. Na prática levaria a algumas dificuldades:

- Não seria possível ter os dados dos alunos previamente. Ou seja, antes da venda de bilhetes começar.
- O processo de compra de bilhetes requer a digitação dos dados dos alunos no ato da compra o que requer mais tempo quer por parte do aluno quer por parte do funcionário que vende os bilhetes.

### 9.4.1 Script SQL para inserir dados nas tabelas

```

1  -----
2  -- Table `SemanaCaloiroIPG`.`aluno1`
3  -----
4  DROP TABLE IF EXISTS `SemanaCaloiroIPG`.`aluno1` ;
5
6  CREATE TABLE IF NOT EXISTS `SemanaCaloiroIPG`.`aluno1` (
7    `IDAluno` INT NOT NULL ,
8    `Nome` VARCHAR(50) NOT NULL ,
9    `Telemovel` VARCHAR(30) CHARACTER SET 'latin1' COLLATE 'latin1_general_ci' NOT NULL ,
10   `Email` VARCHAR(30) NOT NULL ,
11   `bilhete1_idbilhete` INT NOT NULL ,
12   `data_compra_bilhete` DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP ,
13   PRIMARY KEY (`IDAluno`) ,
14   INDEX `fk_aluno1_bilhete1_idx` (`bilhete1_idbilhete` ASC) ,
15   UNIQUE INDEX `Email_UNIQUE` (`Email` ASC) ,
16   UNIQUE INDEX `Telemovel_UNIQUE` (`Telemovel` ASC) ,
17   CONSTRAINT `fk_aluno1_bilhete1`
18     FOREIGN KEY (`bilhete1_idbilhete`)
19     REFERENCES `SemanaCaloiroIPG`.`bilhete1` (`idbilhete`)
20     ON DELETE NO ACTION
21     ON UPDATE NO ACTION)
22 ENGINE = InnoDB;

```

Listing 9.1: Script SQL para criar a tabela aluno1.

```

1  INSERT INTO `bilhete1` (`idbilhete`, `designacao`, `preco`)
2    VALUES ('1', 'Geral alunos IPG', '40.00');
3  INSERT INTO `aluno1` (`IDAluno`, `Nome`, `Telemovel`, `Email`, `bilhete1_idbilhete`)
4    VALUES ('5008643', 'António Moura', '961234567', 'amoura@sal.ipg.pt', '1');

```

Listing 9.2: Script SQL para inserir dados nas tabelas.

```

1  INSERT INTO `bilhete1` (`idbilhete`, `designacao`, `preco`)
2    VALUES ('1', 'Geral alunos IPG', '40.00');
3  INSERT INTO `aluno1` (`IDAluno`, `Nome`, `Telemovel`, `Email`, `bilhete1_idbilhete`)
4    VALUES ('5008643', 'António Moura', '961234567', 'amoura@sal.ipg.pt', '1');

```

Listing 9.3: Script SQL para inserir dados nas tabelas.



+ Opções

	IDAluno	Nome	Telemovel	Email	bilhete1_idbilhete	data_compra_bilhete
<input type="checkbox"/> Editar  Copiar  Apagar	5008643	António Moura	961234567	amoura@sal.ipg.pt	1	2016-10-30 17:27:37

↑ ☐ Check all Com os seleccionados: Editar Copiar Apagar Exportar

(a) Tabela aluno1

+ Opções

	idbilhete	designacao	preco
<input type="checkbox"/> Editar  Copiar  Apagar	1	Geral alunos IPG	40.00

↑ ☐ Check all Com os seleccionados: Editar Copiar Apagar Exportar

(b) Tabela bilhete1

Figura 9.3: Dados nas tabelas

Servidor: 127.0.0.1 » Base de Dados: semanacaloiroipg » Tabela: aluno1

Procurar Estrutura SQL Pesquisar Inserir Mais

Mostrar Caixa do query

```
1 INSERT INTO `aluno1` (`IDAluno`, `Nome`, `Telemovel`, `Email`, `bilhete1_idbilhete`)
VALUES ('5008643', 'António Moura', '961234567', 'amoura@sal.ipg.pt', '1')
```

[ Edit inline ] [ Editar ] [ Create PHP code ]

**Erro**

**Comando SQL:**

```
INSERT INTO `aluno1` (`IDAluno`, `Nome`, `Telemovel`,
```

**Mensagens do MySQL :**

#1062 - Duplicate entry '5008643' for key 'PRIMARY'

Figura 9.4: Erro ao tentar inserir um aluno em duplicado (mesmo número).

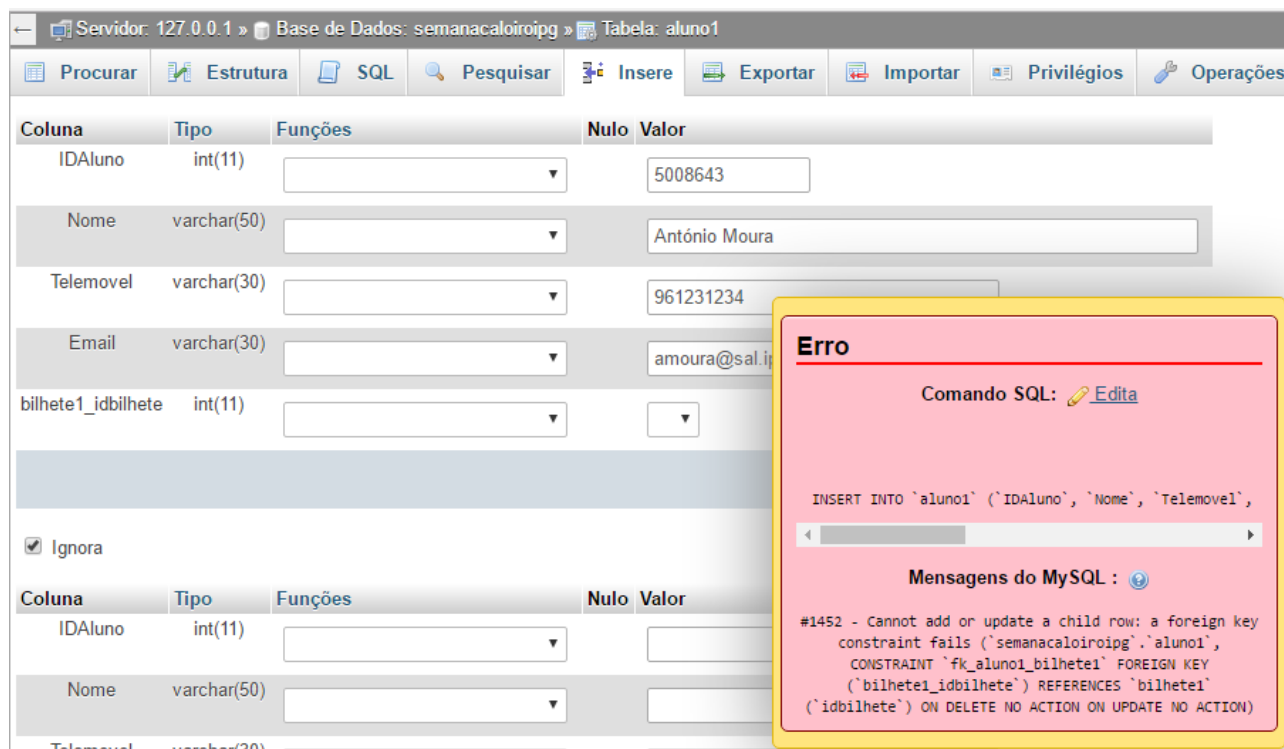


Figura 9.5: Erro ao tentar inserir um aluno sem indicação do bilhete.

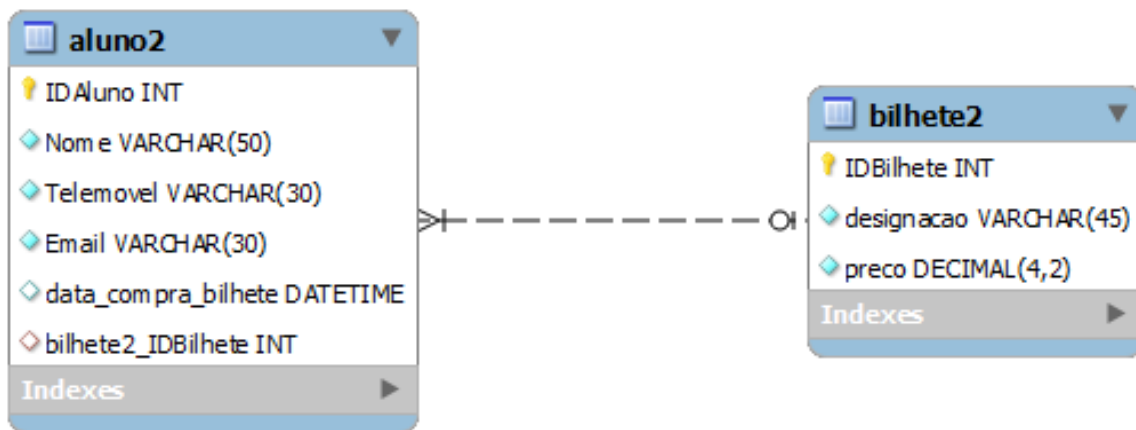
## 9.5 Bilhete geral situação 2

Um aluno **só pode comprar** um bilhete geral. O modelo ER da figura 9.6 deve ler-ser da seguinte maneira:

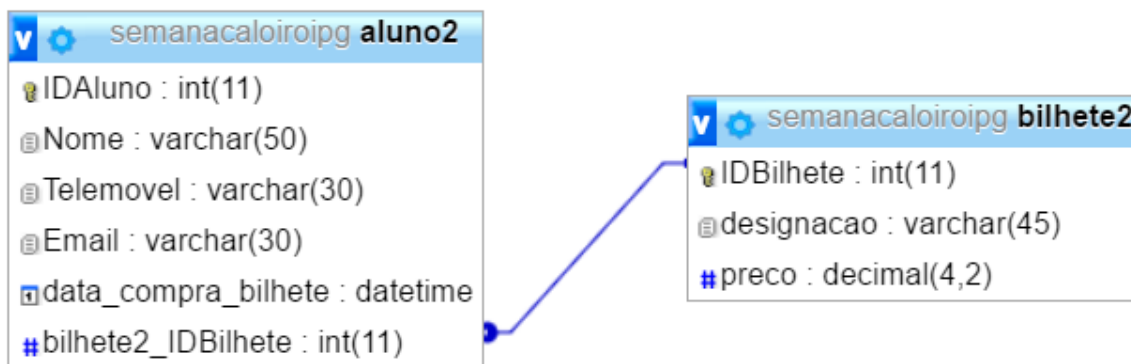
1. Cada *Aluno* **pode Comprar um e só um** *Bilhete*.
2. Cada *Bilhete* **poder ser Comprado por um ou muitos** Alunos.

Com o modelo ER da figura 9.6 já é possível ter os dados dos alunos mesmo antes de se iniciar o processo de venda. Assim, podemos fazer diversas consultas. Por exemplo, quantos alunos podem comprar bilhete; quantos alunos já compraram ou não bilhete.

Com este modelo temos um desperdício de memória quando houver muitos alunos que não comprem bilhete. Pois os atributos `data_compra_bilhete` e `IDBilheteFK` ficam por preencher.



(a) Modelo ER. Relacionamento 1:n, não obrigatório do lado n



(b) Pagamento

Figura 9.6: Modelo físico MySQL

```

1 delete from aluno2;
2 delete from bilhete2;
3 INSERT INTO `bilhete2` (`idbilhete`, `designacao`, `preco`)
4 VALUES ('1', 'Geral alunos IPG', '40.00');
5 INSERT INTO `aluno2` (`IDAluno`, `Nome`, `Telemovel`, `Email`, `data_compra_bilhete`, `
  IDBilheteFK`)
6 VALUES ('5008643', 'António Moura', '961234567', 'amoura@sal.ipg.pt', '2016-10-30
  12:41:33', '1');
7 INSERT INTO `aluno2` (`IDAluno`, `Nome`, `Telemovel`, `Email`)
8 VALUES ('5008631', 'Cerveira Marques', '935554567', 'cmarques@sal.ipg.pt');

```

Listing 9.4: Script SQL para inserir dados nas tabelas.

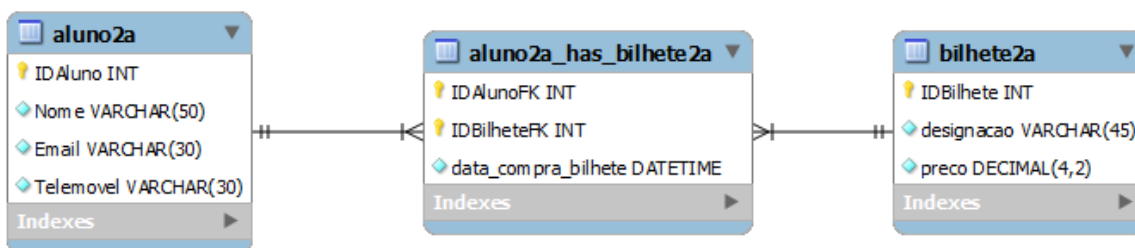
+ Opções

	IDAluno	Nome	Telemovel	Email	data_compra_bilhete	IDBilheteFK
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Apagar	5008631	Cerveira Marques	935554567	cmarques@sal.ipg.pt	NULL	NULL
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Apagar	5008643	António Moura	961234567	amoura@sal.ipg.pt	2016-10-30 12:41:33	1

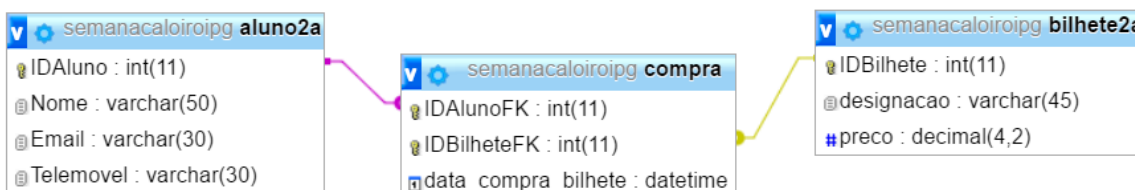
Figura 9.7: Dois aluno. Um com bilhete e outro sem.

## 9.6 Bilhete geral situação 2b

No modelo ER da figura 9.8a foi criada uma nova tabela para registar a compra dos bilhete gerais. Agora não temos desperdício de espaço. No entanto, temos mais uma tabela e para registar uma venda é necessário preencher a chave primária, agora composta por dois campos do tipo INT (IDAlunoFK e IDBilheteFK) e a data de compra do tipo DATETIME.



(a) Modelo ER. Relacionamento 1:n, não obrigatório do lado n



(b) Pagamento

Figura 9.8: Modelo físico MySQL

```

1 delete from aluno2a;
2 delete from bilhete2a;
3 delete from compra;
4 INSERT INTO `bilhete2a` (`idbilhete`, `designacao`, `preco`)
5   VALUES ('1', 'Geral alunos IPG', '40.00');
6 INSERT INTO `aluno2a` (`IDAluno`, `Nome`, `Telemovel`, `Email`)
7   VALUES ('5008643', 'António Moura', '961234567', 'amoura@sal.ipg.pt');
8 INSERT INTO `aluno2` (`IDAluno`, `Nome`, `Telemovel`, `Email`)
9   VALUES ('5008631', 'Cerveira Marques', '935554567', 'cmarques@sal.ipg.pt');
10 INSERT INTO `compra` (IDAlunoFK, IDBilheteFK) VALUES ('5008631', '1');
11 INSERT INTO `compra` (IDAlunoFK, IDBilheteFK) VALUES ('5008643', '1');

```

Listing 9.5: Script SQL para inserir dados nas tabelas.

IDBilhete	designacao	preco
1	Geral alunos IPG	40.00

(a) Bilhetes

IDAluno	Nome	Email	Telemovel
5008631	Cerveira Marques	cmarques@sal.ipg.pt	935554567
5008643	António Moura	amoura@sal.ipg.pt	961234567

(b) Alunos

IDAlunoFK	IDBilheteFK	data_compra_bilhete
5008631	1	2016-10-30 22:11:43
5008643	1	2016-10-30 22:34:20

(c) Compras

Figura 9.9: Modelo físico MySQL

## 9.7 Bilhete geral situação 3

Um aluno **pode comprar vários** bilhetes gerais **de uma só vez**. Com base no modelo ER da secção anterior (9.6), podemos ver que basta adicionar o atributo **Quantidade** na tabela compra.

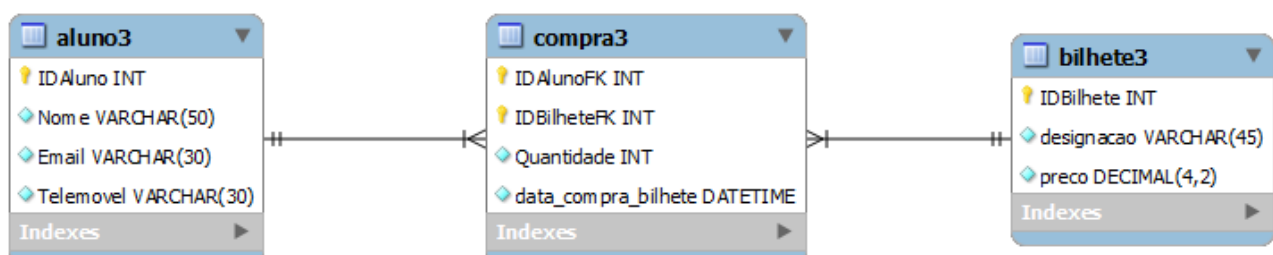


Figura 9.10: Dois aluno. Um com bilhete e outro sem.

## 9.8 Bilhete geral situação 4

Um aluno **pode comprar vários** bilhetes gerais **um de cada vez**. Com base no modelo ER da secção 9.6, podemos ver que basta adicionar mais um atributo IDCompra à chave primária da tabela compra. Assim o aluno pode comprar um bilhete quantas vezes quiser.

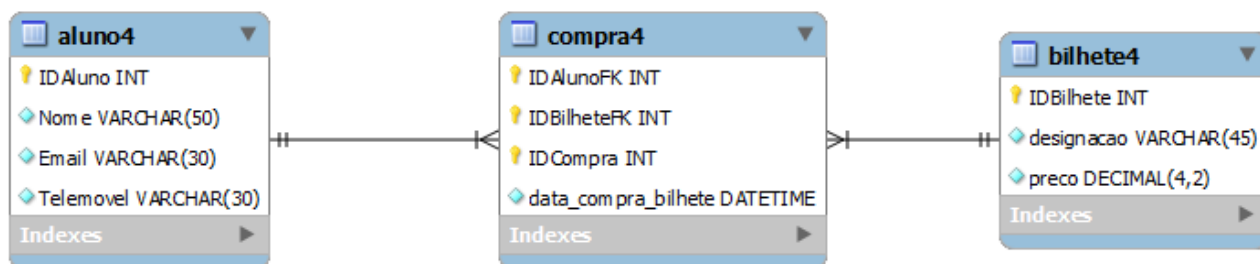


Figura 9.11

## 9.9 Bilhete geral situação 5

Um aluno **pode comprar vários** bilhetes gerais **várias vezes**. Com base no modelo ER da secção 9.6, podemos ver que basta adicionar mais um atributo IDCompra à chave primária da tabela compra. Assim o aluno pode comprar um bilhete quantas vezes quiser.

Semelhante à situação 3 da secção 9.7, podemos ver que basta adicionar o atributo Quantidade na tabela compra.

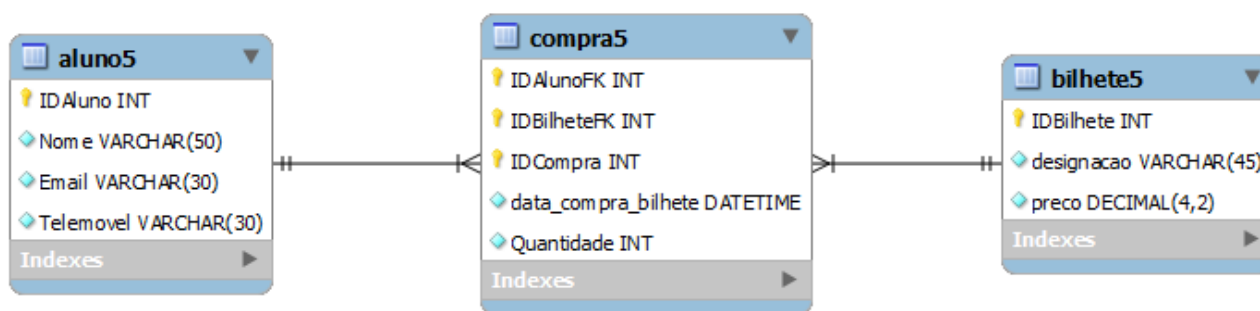


Figura 9.12

## 9.10 Bilhetes diários

Na tabela 9.2 são apresentados os preços dos bilhetes para cada dia da Semana do Caloiro.

Tabela 9.2: Precário dos bilhetes gerais

\$ IDBilhete	Designação	Preços €
1	31 Outubro Segunda-feira	9,00
2	01 Novembro Terça-feira	8,00
3	02 Novembro Quarta-feira	8,00
4	03 Novembro Quinta-feira	8,00
5	04 Novembro Sexta-feira	10,00
6	05 Novembro Sábado	10,00
7	06 Novembro Domingo	8,00

## 9.11 Proposta de exercícios

Desenhe o diagrama ER para cada uma das seguintes situações (restrições impostas pela Direção da AAG):

1. Um aluno **pode comprar** um exemplar de cada bilhete diário quando o desejar. Pode comprar hoje dois e depois de amanhã um
2. Um aluno **pode comprar** de uma só vez um exemplar de cada bilhete diário

### 9.11.1 Bilhetes diários, situação 1

Um aluno **pode comprar** um exemplar de cada bilhete diário quando o desejar. Pode comprar hoje dois e depois de amanhã um. O modelo ER da figura 9.13 permite a cada aluno comprar um bilhete para cada dia da Semana do Caloiro. A chave primária da tabela **compra** é composta pelos atributos **IDAlunoFK** e **IDBilheteFK**. Isto garante que cada aluno só pode comprar um bilhete para cada dia.

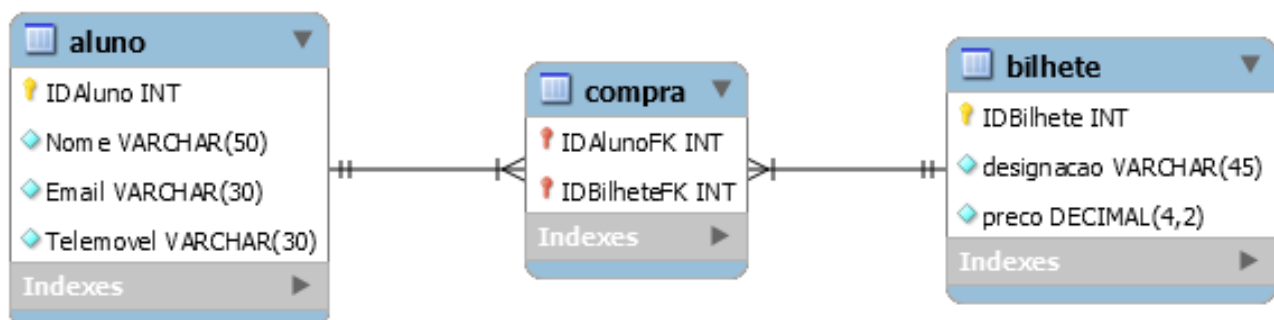


Figura 9.13

A tabela 9.3 mostra os dados (fictícios) de alguns alunos. Na tabela 9.4 são apresentados vários registos de compra de bilhetes de três alunos. A chave primária composta (**IDAlunoFK**, **IDBilheteFK**) não permite a um aluno comprar mais do que um bilhete para cada dia. As

linhas com o texto cortado (cor encarnada) indica que estes dados não podem fazer para da tabela **compra**.

Tabela 9.3: Tabela de alunos

\$ IDAluno	Nome	E-Mail	Telemóvel
5008643	António Moura	amarco@sal.ipg.pt	964107790
5008631	Cerveira Marques	cstéphane@sal.ipg.pt	934892380
5008364	Christel Dias	ctania@sal.ipg.pt	947329503
5008360	Dias Sousa	ddaniel@sal.ipg.pt	931132221
5008630	José Correia	jpedro@sal.ipg.pt	954781538
5007935	Manuel Rodrigues	mrenato@sal.ipg.pt	936183640
5008330	Manuel Cabral	mricardo@sal.ipg.pt	950760821
5008322	Raquel Gonçalves	rana@sal.ipg.pt	960325491
5008413	Ricardo Costa	rfilipe@sal.ipg.pt	918196002
5008613	Sofia Rodrigues	scatarina@sal.ipg.pt	915440266

Tabela 9.4: Tabela de compra de bilhetes

\$ IDAlunoFK	\$ IDBilheteFK
5008643	1
5008643	2
5008643	3
<del>5008643</del>	<del>3</del>
5008631	1
5008631	4
<del>5008631</del>	<del>4</del>
5008613	1

#### 9.11.1.1 Normalização

**As tabelas estão na 1ª Forma Normal (1FN) ?**

Sem grupos de dados relacionados ou elementos repetidos ?

1. Tem uma chave primária
2. Cada coluna é atómica
3. Não há grupos repetidos de colunas



Tabela 9.5: 1ª Forma Normal (1FN) das tabelas

Tabela	Tem uma chave primária?	Cada coluna é atômica?	Não há grupos repetidos de colunas?
aluno	✓	✓	✓
bilhete	✓	✓	✓
compra	✓	✓	✓

**As tabelas estão na 2ª Forma Normal (2FN) ?** Vamos ver apenas as tabelas com chave composta. As restantes que estejam na 1FN, também já estão na 2FN.

1. Está na 1FN
2. Cada coluna **não chave** depende da totalidade da chave primária (composta). Se a tabela tiver chave primário simples então já está na 2FN.

Tabela 9.6: 2ª Forma Normal (2FN) das tabelas

Tabela	Está na 1FN?	A chave primária é composta?	Cada coluna <i>não chave</i> depende da totalidade da chave primária (composta)?
aluno	✓	×	⇒ 2FN
bilhete	✓	×	⇒ 2FN
compra	✓	✓	Não tem colunas. ⇒ 2FN

**As tabelas estão na 3ª Forma Normal (3FN) ?** Apenas para tabelas com chave simples.

1. Está na 2FN
2. Cada coluna não chave só é dependente da chave primária (simples).

Tabela 9.7: 3ª Forma Normal (3FN) das tabelas

Tabela	Está na 2FN?	A chave primária é simples?	Cada coluna não chave só é dependente da chave primária (simples)?
aluno	✓	✓	✓ (Nome, EMail, Telemovel) ⇒ 3FN
bilhete	✓	✓	✓ (Designacao, Preco) ⇒ 3FN
compra	✓	×	Não tem. ⇒ 3FN

Os atributos **E-Mail** e **Telemovel** são ambas chaves candidatas na entidade **aluno**, logo não são consideradas dependentes uma da outra. Na realidade o **E-Mail** determina o **Telemovel** e vice-versa.

### 9.11.1.2 Limitações

Com o modelo ER apresentado na figura 9.13 não é possível saber quando é que um aluno comprou os bilhetes. Este pode ter comprado dois num dia, à mesma hora, e outro no dia seguinte. Para corrigir esta situação podemos adicionar o atributo **data** na tabela **compra** resultando no modelo da figura 9.14.

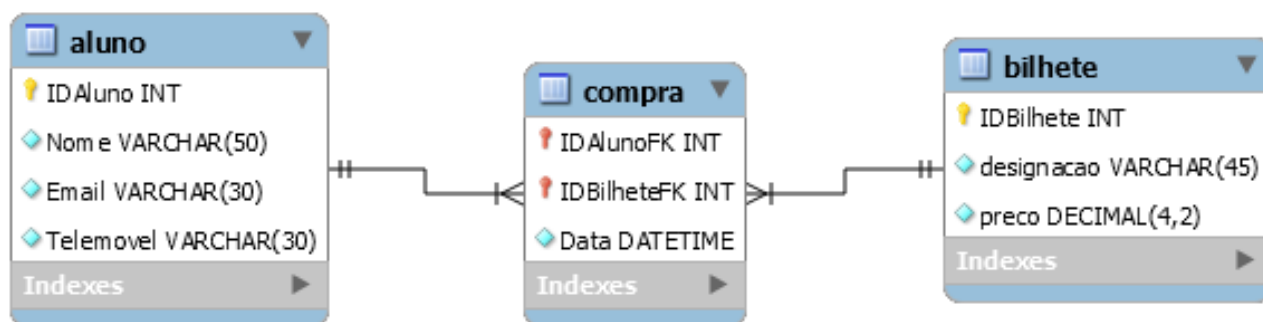


Figura 9.14

Tabela 9.8: Tabela de compra de bilhetes. Aluno 5008643, compra dois, mais um.

\$ IDAlunoFK	\$ IDBilheteFK	Data
5008643	1	2016-10-31 10:30:00
5008643	2	2016-10-31 10:30:00
5008643	3	2016-11-01 12:20:45
5008631	1	2016-11-03 19:13:00
5008613	1	2016-10-31 10:30:00

Na tabela 9.8 podemos ver que os dois primeiros bilhetes foram comprados em conjunto, constituindo uma compra. Logo devem ter e têm a mesma data. No entanto, nesta situação o atributo data na entidade **compra** é redundante. De facto, uma compra só pode ter uma data. Logo, permite inconsistências. Por exemplo, se for alterada a data do primeiro bilhete para **2016-10-31 15:00:00**.

O modelo ER da figura 9.14 apresenta as seguintes anomalias:

1. **Insert** - Quando se acrescenta um novo bilhete na mesma compra a um aluno tem de se ter o cuidado de colocar a mesma data dos outros bilhetes, senão passamos a ter uma nova compra.
2. **Delete** - Não apresenta nenhuma.

3. **Update** - A data aparece tantas vezes quantas o número de bilhetes que um aluno comprou de uma só vez, pelo que alterando-se a data de uma linha temos de ter o cuidado de alterar as outras.

Por outro lado, a data poderia não ser considerado redundante mas sim outra compra. Portanto, passaríamos a ter três compras em vez de duas. O que não é verdade. Para resolver esta anomalia podemos acrescentar um novo atributo (**IDCompra**) para registar o número da compra.

Nas duas secções seguintes vamos explorar duas forma de refinar o modelo ER.

### 9.11.1.3 Refinamento 1

Neste caso começamos adicionar o atributo **IDCompra** na entidade **compra**. Depois, vamos normalizar a entidade.

Tabela 9.9: Tabela de **compra** de bilhetes com **IDCompra**.

\$ IDAlunoFK	\$ IDBilheteFK	Data	IDCompra
5008643	1	2016-10-31 10:30:00	1
5008643	2	2016-10-31 10:30:00	1
5008643	3	2016-11-01 12:20:45	2
5008631	1	2016-11-03 19:13:00	3
5008613	1	2016-10-31 10:30:00	4

Devido à alteração na tabela **compra** devemos verificar se está normalizada (3FN). Não está na 1FN porque tem um grupo relacionado/repetido, **Data**, **IDCompra**, **IDAlunoFK** porque o **IDCompra** determina a **Data** e o **IDAlunoFK**. Ou seja, sabendo-se o **IDCompra** pode-se saber/consultar a quer a **Data** quer o aluno (**IDAlunoFK**). Vamos então aplicar a regra para a 1FN:

---

Cria tabelas separadas para cada grupo de dados relacionados e identifica cada linha com uma única coluna ou conjunto de colunas (chave primária)

---

Devemos criar uma nova entidade denominada **cabecalho\_compra** (ou outro nome adequado) com o grupo repetido. Devemos definir uma chave primária para a nova entidade a partir dos atributos existentes ou criar um novo atributo. O **IDCompra** é uma chave candidata não porque os seus valores não se repetem. Logo, esta é escolhida para ser a chave da entidade **cabecalho\_compra**. A estrutura é apresentada na tabela 9.10. As linhas repetidas da tabela 9.9 foram removidas.

Tabela 9.10: Nova entidade **cabecalho\_compra**

\$ IDCompra	IDAlunoFK	Data
1	5008643	2016-10-31 10:30:00
2	5008643	2016-11-01 12:20:45
3	5008631	2016-11-03 19:13:00
4	5008613	2016-10-31 10:30:00

A tabela 9.11 mostra a nova estrutura da entidade **compra**. Com já foi referido atrás o grupo relacionado a **Data**, **IDCompra**, **IDAlunoFK** foi removido da tabela. No entanto, deve ser adicionada a chave primária da nova entidade como chave estrangeira (IDCompraFK). Qual deve ser a chave primária da entidade **compra**?

Para garantir que um aluno possa comprar mais do que um bilhete a chave deve ser composta pelos dois atributos **IDCompraFK**, **IDBilheteFK**.

Tabela 9.11: Tabela de **compra** de bilhetes com **IDCompraFK**.

\$ IDCompraFK	\$ IDBilheteFK
1	1
1	2
2	3
3	1
4	1

A figura 9.15 mostra o modelo ER resultante. O preço pode mudar ao longo do tempo. Por isso, deve ficar armazenado em cada um dos detalhes da venda. Adicionando o atributo quantidade temos a possibilidade de comprar vários bilhetes diários para o mesmo dia. A figura 9.16 ilustra o respectivo modelo ER.

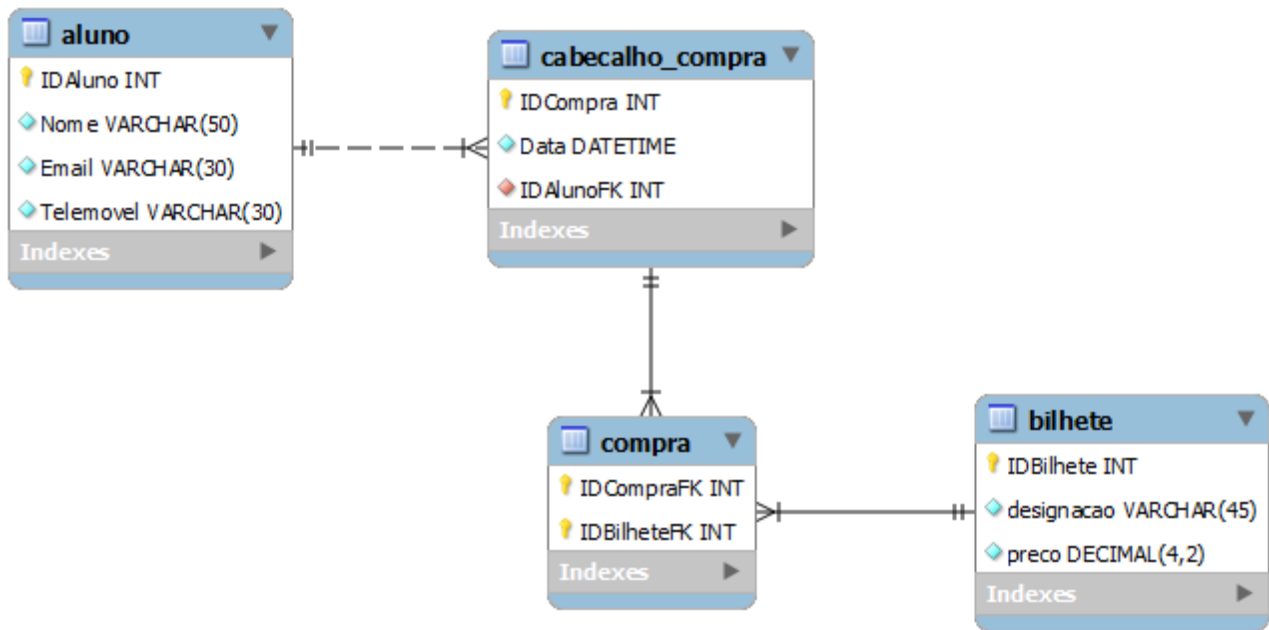


Figura 9.15

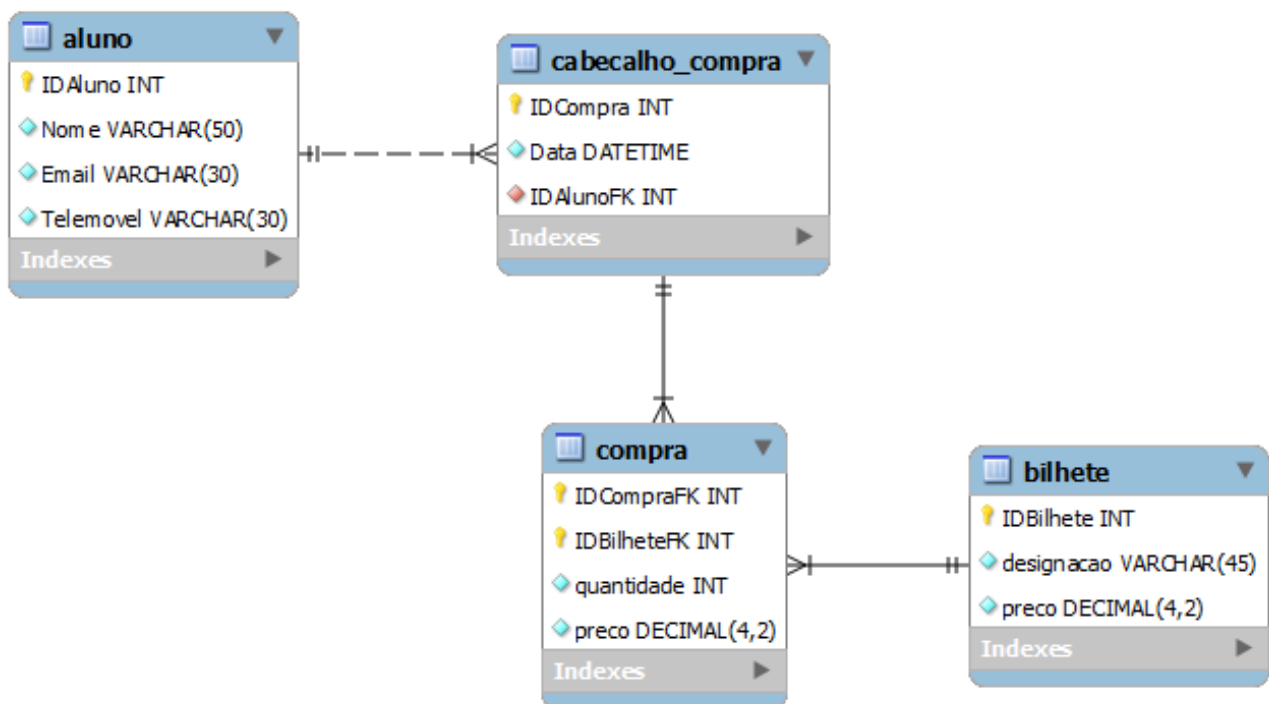


Figura 9.16

#### 9.11.1.4 Refinamento 2

Neste refinamento, em vez de começarmos por adicionar o atributo **IDCompra**, vamos remover o atributo **Data** que é redundante na entidade **compra**. Depois, vamos aplicar o processo de normalização à entidade **compra** e à nova entidade que resulta de eliminar a redundância.

Tabela 9.12: Tabela de compra de bilhetes

\$ IDAlunoFK	\$ IDBilheteFK	Data
5008643	1	2016-10-31 10:30:00
5008643	2	2016-10-31 10:30:00
5008643	3	2016-11-01 12:20:45
5008631	1	2016-11-03 19:13:00

A tabela **compra** (tabela 9.12) não está na 1FN porque a **data** é redundante, repete-se para cada conjunto dos bilhetes de um aluno, comprados de uma só vez. Logo, a data deve ser removida da entidade **compra** e criada uma nova entidade (data\_compra) com a data e uma chave primária, por exemplo **IDData**. Deve-se agora relacionar as duas tabelas colocando a chave primária na tabela **compra** como chave estrangeira (**IDDataFK**). A figura 9.17 apresenta o novo diagrama e as tabelas 9.13 e 9.14 os mesmos dados da tabela 9.8 sem redundância.

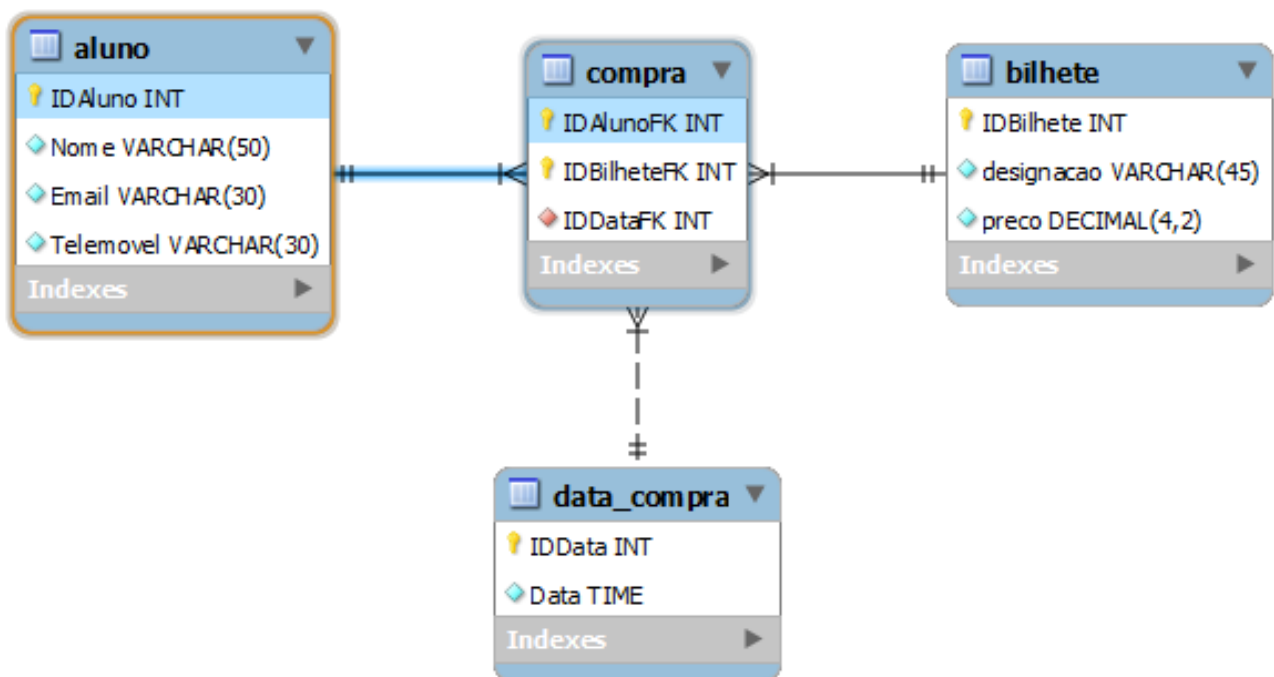


Figura 9.17

Tabela 9.13: Tabela de **compra** de bilhetes, com **IDData** e sem **data**.

\$ IDAlunoFK	\$ IDBilheteFK	IDDataFK
5008643	1	1
5008643	2	1
5008643	3	2
5008631	1	3

Tabela 9.14: Tabela de **data\_compra**.

\$ IDData	Data
1	2016-10-31 10:30:00
2	2016-11-01 12:20:45
3	2016-11-03 19:13:00

Tabela 9.15: Tabela de **compra** de bilhetes. Dois alunos com o mesmo **IDDataFK** (1).

\$ IDAlunoFK	\$ IDBilheteFK	\$ IDDataFK
5008643	1	1
5008643	2	1
5008643	3	2
5008631	1	3
5008613	1	1

Na tabela **compra** poderíamos colocar o mesmo **IDDataFK** para todas as compras efetuadas nessa data. No entanto, temos uma anomalia quando pretendemos alterar a data de uma compra. Ou seja, todas as outras também ficam com a data alterada. Isto acontece porque a data apesar de ser igual não é a mesma porque depende do **IDAlunoFK**. Portanto, o **IDAlunoFK** deve constar na entidade **data\_compra** como chave estrangeira. A tabela 9.16 ilustra a entidade com o novo atributo.

Tabela 9.16: Tabela de **data\_compra**.

IDAluno	\$ IDData	Data
5008643	1	2016-10-31 10:30:00
5008643	2	2016-11-01 12:20:45
5008631	3	2016-11-03 19:13:00
5008613	4	2016-10-31 10:30:00

Tabela 9.17: Tabela de **compra** de bilhetes. Dois alunos com a mesma data mas **IDDataFK** diferente.

\$ IDAlunoFK	\$ IDBilheteFK	\$ IDDataFK
5008643	1	1
5008643	2	1
5008643	3	2
5008631	1	3
5008613	1	4

Devido à alteração na tabela **data\_compra**, a tabela **compra** não está na **2FN** porque o **IDDataFK** depende apenas de parte da chave composta, **IDAlunoFK**. Ou seja sabendo o **IDData** sabe-se o cliente consultando a tabela **data\_compra**. Logo este atributo deve ser removido da tabela **compra**.

Tabela 9.18: Tabela de **compra** de bilhetes. Sem o atributo **IDAlunoFK**

\$ IDBilheteFK	\$ IDDataFK
1	1
2	1
3	2
1	3
1	4

A tabela **data\_compra** contém na realidade os dados de uma compra. Ou seja, um aluno que numa data compra bilhetes. O **IDData** poderia ser o número da compra (**IDCompra**). Assim, esta entidade passa a ter o nome de **compra** e a entidade **compra** passa a ter o nome de **detalhes\_compra** porque na realidade armazena os detalhes da compra. Ou seja os bilhetes comprados numa compra.

Tabela 9.19: Tabela **compra** anteriormente **data\_compra**.

IDAluno	\$ IDCompra	Data
5008643	1	2016-10-31 10:30:00
5008643	2	2016-11-01 12:20:45
5008631	3	2016-11-03 19:13:00
5008613	4	2016-10-31 10:30:00

Tabela 9.20: Tabela **detalhes\_compra** anteriormente **compra**.

\$ IDBilheteFK	\$ IDDataFK
1	1
2	1
3	2
1	3
1	4

A figura 9.18 mostra o modelo ER resultante. O preço pode mudar ao longo do tempo. Por isso deve ficar armazenado em cada um dos detalhes da venda. Adicionando o atributo quantidade temos a possibilidade de comprar vários bilhetes diários para o mesmo dia. A figura 9.19 ilustra o respectivo modelo ER.



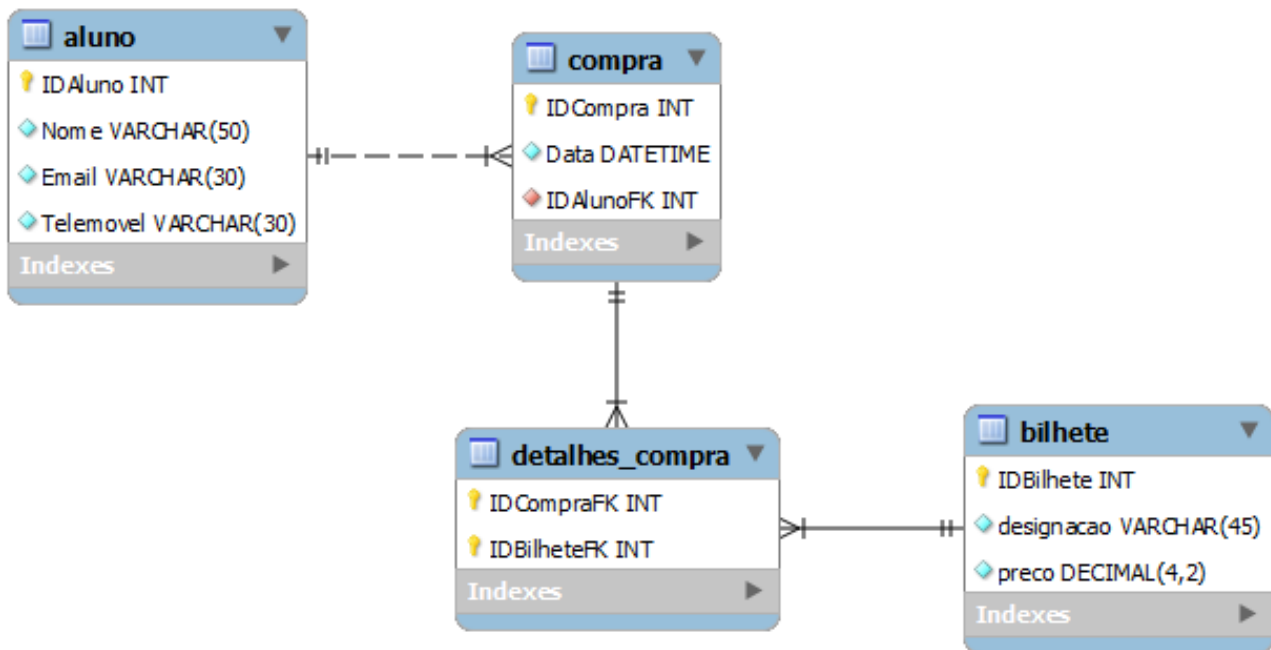


Figura 9.18

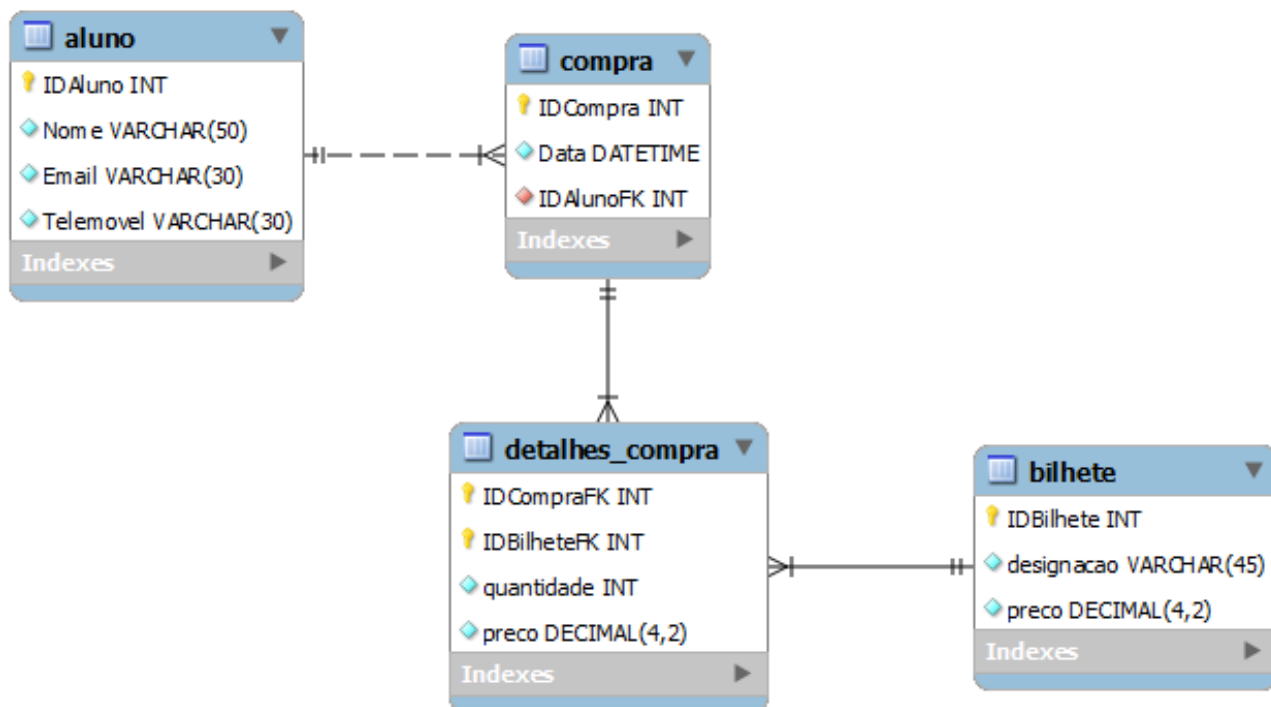
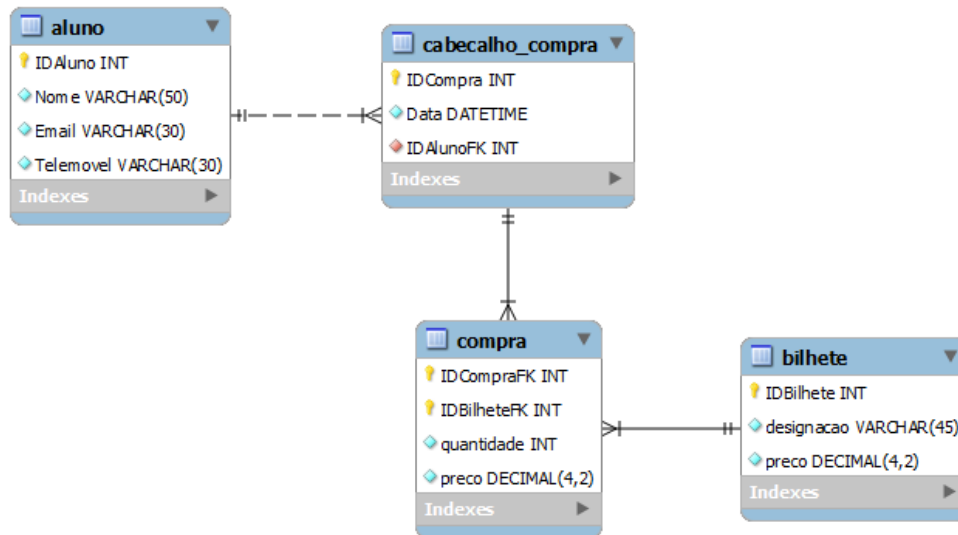


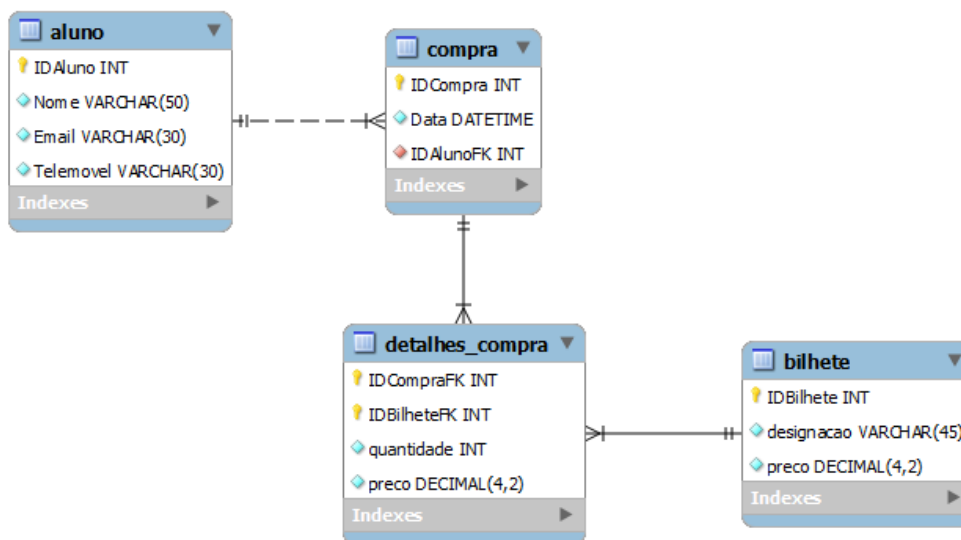
Figura 9.19

### 9.11.1.5 Conclusões

Nas secções anteriores seguimos dois caminhos diferentes para modelar a venda de bilhetes diários na Semana do Caloio do IPG. Se observarmos os dois modelos ER finais, podemos verificar que são iguais com excepção do nome de duas entidades, as que foram criadas no processo de modelação. A figura ?? mostra os dois modelos ER obtidos. Note-se que o atributo quantidade não é necessário porque cada aluno só pode comprar um bilhete. No entanto neste modelo ficará preenchido com o valor 1.



(a) Modelação 1



(b) Modelação 2

Ambos os casos de modelação foram morosos porque não foi inicialmente identificadas mais entidades. Nomeadamente a entidade **cabecalho\_compra** no primeiro caso e a entidade **compra** no segundo.

### 9.11.2 Bilhetes diários, situação 2

Um aluno **pode comprar** de uma só vez um exemplar de cada bilhete diário.



# Capítulo 10

## Modelação de dados

Vocabulário e relações entre modelo conceptual, lógico e físico.

<http://www.1keydata.com/datawarehousing/data-modeling-levels.html>

## 10.1 Operações com strings

No endereço <http://dev.mysql.com/doc/refman/5.7/en/string-functions.html> é apresentada uma lista de todas as funções para operações com strings.

## 10.2 Função MID()

The MID() Function The MID() function is used to extract characters from a text field.

SQL MID() Syntax `SELECT MID(column_name,start,length) AS some_name FROM table_name;`

Parameter Description `column_name` Required. The field to extract characters from `start` Required. Specifies the starting position (starts at 1) `length` Optional. The number of characters to return. If omitted, the MID() function returns the rest of the text

## 10.3 Função LENGTH(str())

Returns the length of the value in a text field.

SQL Syntax `SELECT LENGTH(column_name) FROM table_name;`

```
mysql> SELECT LENGTH('text'); -> 4
```

## 10.4 Função UCASE()

The UCASE() Function The UCASE() function converts the value of a field to uppercase.

SQL UCASE() Syntax `SELECT UCASE(column_name) FROM table_name;`

Syntax for SQL Server `SELECT UPPER(column_name) FROM table_name;`

## 10.5 Função LCASE()

The LCASE() Function The LCASE() function converts the value of a field to lowercase.

SQL LCASE() Syntax SELECT LCASE(column\_name) FROM table\_name; Syntax for SQL Server SELECT LOWER(column\_name) FROM table\_name;

## 10.6 Função LOCATE()

LOCATE(substr,str), LOCATE(substr,str,pos)

The first syntax returns the position of the first occurrence of substring substr in string str. The second syntax returns the position of the first occurrence of substring substr in string str, starting at position pos. Returns 0 if substr is not in str. Returns NULL if substr or str is NULL.

```
mysql> SELECT LOCATE('bar', 'foobarbar'); -> 4 mysql> SELECT LOCATE('xbar',  
'foobar'); -> 0 mysql> SELECT LOCATE('bar', 'foobarbar', 5); -> 7
```

POSITION(substr IN str) is a synonym for LOCATE(substr,str).

## 10.7 Função SUBSTRING\_INDEX()

SUBSTRING\_INDEX(str,delim,count)

Returns the substring from string str before count occurrences of the delimiter delim. If count is positive, everything to the left of the final delimiter (counting from the left) is returned. If count is negative, everything to the right of the final delimiter (counting from the right) is returned. SUBSTRING\_INDEX() performs a case-sensitive match when searching for delim.

```
mysql> SELECT SUBSTRING_INDEX('www.mysql.com', '.', 2); -> 'www.mysql' mysql>  
SELECT SUBSTRING_INDEX('www.mysql.com', '.', -2); -> 'mysql.com'
```

## 10.8 Função CONCAT()

CONCAT(str1,str2,...)

Returns the string that results from concatenating the arguments. May have one or more arguments. If all arguments are nonbinary strings, the result is a nonbinary string. If the arguments include any binary strings, the result is a binary string. A numeric argument is converted to its equivalent nonbinary string form.

CONCAT() returns NULL if any argument is NULL.

mysql> SELECT CONCAT('My', 'S', 'QL'); -> 'MySQL' mysql> SELECT CONCAT('My', NULL, 'QL'); -> NULL mysql> SELECT CONCAT(14.3); -> '14.3' For quoted strings, concatenation can be performed by placing the strings next to each other:

```
mysql> SELECT 'My' 'S' 'QL'; -> 'MySQL'
```



# Capítulo 11

## Linguagen SQL

### 11.1 Base de dados SCUT

Considere os scripts, disponíveis na página da disciplina:

- `scut.sql` — Tabelas
- `dbo.carros.Table.sql` — Dados carros (10,000).
- `dbo.cidades.Table.sql` — Dados cidades (entra e saída de autoestradas) (10)
- `dbo.distancias.Table.sql` — Dados distâncias entre cidades (45).
- `dbo.donos.Table.sql` — Dados utentes das SCUTs (10,000).
- `dbo.passagens.Table.sql` — Dados de registo de passagens de utentes nas SCUTs (3,000,000).

Comece por criar uma base de dados e de seguida corra os scripts para criar as tabelas e preencher os dados.

### 11.1.1 Estrutura das tabelas

```

1 CREATE TABLE passagem(
2     IDUtenteFK int NOT NULL,
3     IDCidade int,
4     Data datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
5     EntradaSaida varchar(50) NOT NULL
6 );
7 CREATE TABLE donos(
8     IDUtente int primary key,
9     Nome varchar(80) NOT NULL,
10    Codigo varchar(15) NOT NULL
11 );
12 CREATE TABLE distancias(
13     IDCidadeA int NOT NULL,
14     IDCidadeB int NOT NULL,
15     Distancia float NOT NULL,
16     Preco decimal(12,2),
17     primary key (IDCidadeA, IDCidadeB)
18 );
19 CREATE TABLE cidades(
20     IDCidade int primary key,
21     Cidade varchar(50) NOT NULL
22 );
23 CREATE TABLE carros(
24     IDCarro int primary key,
25     IDUtenteFK int NOT NULL,
26     Matricula varchar(20) NOT NULL,
27     Marca varchar(30) NOT NULL,
28     Modelo varchar(30) NOT NULL,
29     Ano smallint NOT NULL
30 );

```

### 11.1.2 Conversão da tabela passagens

Conversão da tabela passagens com **data** to tipo **varchar(20)** para a tabela **passagens2** com data do tipo **DATETIME**.

```

1 delete from passagens2;
2 INSERT INTO passagens2 (IDCarroFK, IDCidade, Data, EntradaSaida)
3     SELECT IDCarroFK,
4           IDCidade,
5           STR_TO_DATE(DATA, '%d-%m-%Y %H:%i:%s.%f'),
6           EntradaSaida
7 FROM passagens;

```

## 11.2 Cópia e restauração de base de dados

SHOW [FULL] TABLES [FROM | IN db\_name] [LIKE 'pattern' | WHERE expr] show tables in scut SHOW TABLES;

```
DESCRIBE carros;
```

```
Tables_in_scut carros cidades distancias donos passagens
```

```
1 Cópia
2 mysqldump -u root --password= --databases scut > /var/www/EUSOUENGENHEIRO/tmp2/passagens.sql
3
4 Restauro de cópia:
5 mysql -u root --password= scut < /var/www/EUSOUENGENHEIRO/tmp2/passagens.sql
6
7 mysqldump -u root --password= --databases scut > /var/www/EUSOUENGENHEIRO/tmp2/scutLinux.sql
8
9 mysql -u root --password= scut_CMBD < /var/www/EUSOUENGENHEIRO/tmp2/scutLinux.sql
```

## 11.3 Criação e cópia de tabelas

Criação e de uma tabela numa base de dados com base numa tabela de outra base de dados.

```
1 CREATE TABLE scut_CMBD.carros SELECT * FROM scut.carros
2 CREATE TABLE scut_CMBD.cidades SELECT * FROM scut.cidades
3 CREATE TABLE scut_CMBD.distancias SELECT * FROM scut.distancias
4 CREATE TABLE scut_CMBD.passagens SELECT * FROM scut.passagens
5
6 Elimina repetidos:
7 CREATE TABLE scut_CMBD.donos SELECT IDUtente, Nome, NumeroDispositivoEletronico FROM scut.donos
   group by IDUtente, Nome, NumeroDispositivoEletronico
8
9 CREATE TABLE scut_CMBD.carros SELECT Matricula, Marca, Modelo, Ano, IDUtenteFK, IDCarro FROM scut.
   carros GROUP BY Matricula, Marca, Modelo, Ano, IDUtenteFK, IDCarro
```

## 11.4 Datas e horas

The following table lists the most important built-in date functions in MySQL:

Tabela 11.1

Função	Designação
NOW()	Returns the current date and time
CURDATE()	Returns the current date
CURTIME()	Returns the current time
DATE()	Extracts the date part of a date or date/time expression
EXTRACT()	Returns a single part of a date/time
DATE_ADD()	Adds a specified time interval to a date
DATE_SUB()	Subtracts a specified time interval from a date
DATEDIFF()	Returns the number of days between two dates
DATE_FORMAT()	Displays date/time data in different formats

```

1 STR_TO_DATE(DATA, '%d-%m-%Y %H:%i:%s.%f')
2 % 25-06-2004 07:43:52.803
3 SELECT STR_TO_DATE(DATA, '%Y-%m-%d %H:%i:%s.%f') from passgens limit 5
4 ->
5 2004-06-25 07:43:52
6 2004-06-25 09:31:24
7 2004-09-30 05:13:04
8 2004-09-30 07:11:05
9 2004-04-01 20:15:24
10
11 SELECT STR_TO_DATE('May 1, 2013', '%M %d,%Y');
12 -> '2013-05-01'
13
14 select date_format(sysdate(), '%d/%m/%Y %H:%i:%s');
15 -> 11/12/2016 20:43:09

```

## 11.5 Consultas com uma tabela

## 11.6 Lista de todos os e todos os dados

```
1 SELECT * FROM donos
```

### 11.6.1 Lista de diversas componentes nome dos donos

```

1 SELECT
2     Nome ,
3     SUBSTRING_INDEX(Nome, ' ', 1) Primeiro_Nome ,
4     SUBSTRING_INDEX(Nome, ' ', -1) Ultimo_Nome ,
5     CONCAT(SUBSTRING_INDEX(Nome, ' ', 1), ' ', SUBSTRING_INDEX(Nome, ' ', -1)) Nome_Apelido ,
6     SUBSTRING_INDEX(Nome, ' ', 2) Primeiro_e_Segundo_Nome ,
7     SUBSTRING(Nome, LOCATE(' ', Nome)) Segundo_e_Restantes ,
8     SUBSTRING_INDEX(SUBSTRING_INDEX(Nome, ' ', 2), ' ', -1) Segundo_Nome
9 FROM donos
10 WHERE Nome LIKE '%Catarina Martins%'
11 ORDER BY Segundo_Nome

```

Tabela 11.2: Conteúdo da tabela donos

Catarina Martins Brito Barros Aguiar Pires Pires	Catarina	Pires	Catarina Pires	Catarina Martins	Martins Brito Barros Aguiar Pires Pires	Martins
Catarina Martins Barbosa Queiros	Catarina	Queiros	Catarina Queiros	Catarina Martins	Martins Barbosa Queiros	Martins
Catarina Martins Castro Proenca Carneiro Santos Ferraz	Catarina	Ferraz	Catarina Ferraz	Catarina Martins	Martins Castro Proenca Carneiro Santos Ferraz	Martins
Catarina Martins Car- mona Moreira Fonseca	Catarina	Fonseca	Catarina Fonseca	Catarina Martins	Martins Carmona Moreira Fonseca	Martins
Catarina Martins Mourão Pires Carneiro	Catarina	Carneiro	Catarina Carneiro	Catarina Martins	Martins Mourão Pires Carneiro	Martins

## 11.7 Projeção e DISTINCT

Selecionar a coluna **Marca** da tabela **carros** com e sem repetições.

```
1  -- marca com repetições
2  SELECT marca FROM carros;
3
4  -- marca sem repetições
5  SELECT DISTINCT marca FROM carros;
6
7  -- marca e modelo sem repetições
8  SELECT DISTINCT marca, modelo FROM carros;
9
10 -- marca e modelo sem repetições ordenada por marca e modelo
11 SELECT DISTINCT marca, modelo
12    FROM carros
13   ORDER BY marca, modelo;
```

Tabela 11.3: Marca dos carros na tabela **carros**.

Registo	Marca
1	Opel
2	Porche
3	Peugeot
4	Peugeot
5	BMW
6	Citroen
7	Audi
8	Peugeot
9	Peugeot
10	Citroen
11	Audi
12	Opel
13	Toyota
14	Seat
15	Citroen
16	Fiat
17	Citroen
18	BMW
19	Audi
20	Opel
21	Peugeot
22	Fiat
23	Bentley
24	BMW
25	Citroen
26	Ferrari
27	Citroen
28	Citroen
29	Bentley
30	Audi
..	..
10000	..

Tabela 11.4: Marca sem repetição da tabela **carros**.

<b>Registo</b>	<b>Marca</b>
1	Audi
2	Bentley
3	BMW
4	Citroen
5	Ferrari
6	Fiat
7	Mitsubishi
8	Opel
9	Peugeot
10	Porche
11	Renault
12	Seat
13	Toyota

Tabela 11.5: Marca e modelo sem repetição da tabela **carros**.

<b>Registro</b>	<b>Marca</b>	<b>Modelo</b>
1	Opel	Corsa
2	Porche	Cheyenne
3	Peugeot	205
4	BMW	X3
5	Citroen	C4
6	Audi	A2
7	Peugeot	305
8	Citroen	C2
9	Audi	TT
10	Toyota	Avensis
11	Seat	Marbella
12	Citroen	C3
13	Fiat	600
14	Audi	A6
15	Opel	Kadet
16	Fiat	Punto
17	Bentley	Continental
18	BMW	520
19	Ferrari	Testarrosa
20	Citroen	C1
21	Audi	A3
22	Audi	A1
23	Audi	A4
24	BMW	530
25	Seat	Ibiza
26	BMW	320
27	Seat	Cordoba
28	Audi	A5
29	Renault	Kangoo
30	Citroen	C5
31	Renault	Megane
32	Mitsubishi	Lancer
33	Fiat	Panda
34	Porche	Carrera
35	BMW	X6
36	Audi	A8
37	Renault	Espace
38	Mitsubishi	Colt



Tabela 11.6: Marcas e modelos de carros ordenadas por marca e modelo

Registo	Marca	Modelo
1	Audi	A1
2	Audi	A2
3	Audi	A3
4	Audi	A4
5	Audi	A5
6	Audi	A6
7	Audi	A8
8	Audi	TT
9	Bentley	Continental
10	BMW	320
11	BMW	520
12	BMW	530
13	BMW	X3
14	BMW	X6
15	Citroen	C1
16	Citroen	C2
17	Citroen	C3
18	Citroen	C4
19	Citroen	C5
20	Ferrari	Testarrosa
21	Fiat	600
22	Fiat	Panda
23	Fiat	Punto
24	Mitsubishi	Colt
25	Mitsubishi	Lancer
26	Opel	Corsa
27	Opel	Kadet
28	Peugeot	205
29	Peugeot	305
30	Porche	Carrera
31	Porche	Cheyenne
32	Renault	Espace
33	Renault	Kangoo
34	Renault	Megane
35	Seat	Cordoba
36	Seat	Ibiza
37	Seat	Marbella
38	Toyota	Avensis

## 11.8 Lista de questões para resolver em aula prática

### 11.8.1 Carros

Lista:

1. Lista de todos os carros.
2. Lista da matricula e marca de todos os carros.
3. Lista da matricula e marca de todos os carros ordenada por matrícula.
4. Que marcas e modelos de carros existem na tabela carros ?

Que:

1. Que carros existem na tabela de carros ?
2. Que marcas de carros existem na tabela carros ?
3. Que marcas e modelos de carros existem na tabela carros ?

Existem:

1. Existem carros da marca **Fiat** na tabela de carros ?
2. Existem carros do ano 1996 na tabela de carros ?

Quantos:

1. Quantos carros existem na tabela de carros ?
2. Quantas marcas de carros existem na tabela carros ?
3. Quantas marcas e modelos de carros existem na tabela carros ?
4. Quantos carros existem de cada marca ?
5. Quantos carros existem da marca **Audi** do ano **2000** ?

### 11.8.2 Donos

### 11.8.3 Carros e Donos

Lista:

1. Lista de todos os carros e respetivos donos.
- 2.
- 3.
- 4.

### 11.8.4 Cidades e Distâncias

Lista:

1. Lista de todas as cidades.
2. Lista de todas as distâncias.
3. Qual a distância entre as cidades **Faro** e **Braga** ?
4. Qual a o valor a pagar entre as cidades **3** e **10** ?

### 11.8.5 Passagens

Quantos:

1. Quantas passagens existem ?
2. Quantos carros passaram nas SCUTs ?
3. Quantos carros passaram nas SCUTs por dia?
4. Quantos carros passaram nas SCUTs por semana?
5. Quantos carros passaram nas SCUTs por mês?
6. Quantos carros passaram nas SCUTs por ano?

## 11.9 Perguntas direcionadas a cada aluno

Suponha que é um agente secreto e foi-lhe encarregada a seguinte missão:

1. Investigar o dono de um carro cujo nome seja igual ou parecido com o seu.
2. Elaborar um conjunto de perguntas pertinentes acerca dessa pessoa.
3. Escrever as consultas SQL necessárias para responder às perguntas do item anterior.

### 11.9.1 Escolha da pessoa a investigar

Exemplo para o nome: Rita Maria dos Santos Martins. Vamos começar por procurar nomes começados por **Rita** e terminados por **Martins**. Resultam três nomes. Escolhemos a 3ª pessoa porque é a única que também tem no nome **Santos**.

```
1 SELECT *
2 FROM donos
3 WHERE Nome like 'Rita%Martins'
```

Tabela 11.7: Pessoas com nome: 'Rita%Martins'

#	IDUtente	Nome	NumeroDispositivoEletronico
1	1519903	Rita Carmona Carreira Calheiros Martins	10974-981
2	9313877	Rita Barbosa Brito Carmona Martins	13621-183
3	14597531	Rita Nunes Santos Barbosa Mota Martins	2528-121

Vamos agora ver qual o carro que a pessoa com o nome **Rita Nunes Santos Barbosa Mota Martins** e IDUtente **14597531** tem. Para isso vamos fazer um consulta na tabelas de carros.

```
1 SELECT *
2 FROM carros
3 WHERE IDUtenteFK = 14597531
```

O resultado é:

Tabela 11.8: Carro da Rita ...

Matricula	Marca	Modelo	Ano	IDUtenteFK	IDCarro
07-JO-11	Ferrari Testarrosa	1990	14597531	5909	

Agora já sabemos os dados do carro da Rita. De seguida, vamos fazer perguntas acerca das suas viagens da **Rita** nas auto-estradas de Portugal.

```

1  SELECT *
2    FROM passagens
3   WHERE IDUtenteFK = 4008
4

```

Resultam 376 registos de entrada e saída. Ou seja 188 (376/2) passagens nas SCUTs.

## 11.10 Alteração dos dados dos donos para nomes de alunos

```

UPDATE donos set Nome = 'Sónia Gabriela dos Santos Soares' where IDUtente = 14998371;
UPDATE donos set Nome = 'Inês Dinis Henriques Ferreira' where IDUtente = 14997863;
UPDATE donos set Nome = 'Daniela Filipa Nunes Jacinto' where IDUtente = 14992793;
UPDATE donos set Nome = 'Renato Manuel da Silva Rodrigues' where IDUtente = 14990807;
UPDATE donos set Nome = 'Liliana Alexandra Monteiro Teles' where IDUtente = 14981672;
UPDATE donos set Nome = 'Margarida Pires de Carvalho' where IDUtente = 14978204;
UPDATE donos set Nome = 'Diogo Filipe Anacleto Fernandes' where IDUtente = 14975658;
UPDATE donos set Nome = 'Maria João Almeida Duarte' where IDUtente = 14972118;
UPDATE donos set Nome = 'Ana Raquel da Silva Gonçalves' where IDUtente = 14964296;
UPDATE donos set Nome = 'Ricardo Manuel Gonçalves Cabral' where IDUtente = 14959889;
UPDATE donos set Nome = 'Daniel Dias de Sousa' where IDUtente = 14958261;
UPDATE donos set Nome = 'Tania Christel Santos Dias' where IDUtente = 14954201;
UPDATE donos set Nome = 'Ana Cláudia Nunes Martins' where IDUtente = 14954117;
UPDATE donos set Nome = 'Diogo de Sousa Proença' where IDUtente = 14948804;
UPDATE donos set Nome = 'Filipe Ricardo dos Santos Melo Costa' where IDUtente = 14946446;
UPDATE donos set Nome = 'Pedro Dinis Maldonado Ângelo' where IDUtente = 14940167;
UPDATE donos set Nome = 'Artur Vasco de Carvalho Rodrigues' where IDUtente = 14936741;
UPDATE donos set Nome = 'Catarina Sofia Pereira Rodrigues' where IDUtente = 14934483;
UPDATE donos set Nome = 'Diana Rafaela Nunes Barbosa' where IDUtente = 14932719;
UPDATE donos set Nome = 'Diogo Carlos Mota Valverde' where IDUtente = 14932632;
UPDATE donos set Nome = 'Francisco José Esteves Marques' where IDUtente = 14932254;
UPDATE donos set Nome = 'Frederico António Rosa Samora de Melo Leote' where IDUtente =
14931939;
UPDATE donos set Nome = 'Frederico Rodrigues Ferreira Durão Venâncio' where IDUtente =
14921386;
UPDATE donos set Nome = 'Georgina Maria Vieira Gonçalves' where IDUtente = 14916268;
UPDATE donos set Nome = 'Gerson Filipe Pinheiro de Jesus Inácio' where IDUtente = 14915476;
UPDATE donos set Nome = 'Jorge Guilherme Nascimento Cabral Rodrigues' where IDUtente =
14914556;
UPDATE donos set Nome = 'Laura Carina de Sá Filipe' where IDUtente = 14910712;
UPDATE donos set Nome = 'Pedro José Pena Correia' where IDUtente = 14902877;
UPDATE donos set Nome = 'Stéphane Cerveira Marques' where IDUtente = 14887210;
UPDATE donos set Nome = 'Marco António Pingarilho de Moura' where IDUtente = 14854969;
UPDATE donos set Nome = 'Pedro da Fonseca Duarte' where IDUtente = 14831279;
UPDATE donos set Nome = 'Ana Catarina Pereira Carvalho' where IDUtente = 14815543;
UPDATE donos set Nome = 'Diana Cunha Teixeira' where IDUtente = 14797260;
UPDATE donos set Nome = 'Maria Manuel Vareiro Moreira' where IDUtente = 1037065;
UPDATE donos set Nome = 'Orlando José Sá da Silva' where IDUtente = 1174166;
UPDATE donos set Nome = 'Pedro Samuel Mendes Oliveira' where IDUtente = 1089885;
UPDATE donos set Nome = 'Sara Elisa Alves da Costa' where IDUtente = 1089334;
UPDATE donos set Nome = 'Sergio Miguel Gabriel Nunes Contreiras' where IDUtente = 1023346;
UPDATE donos set Nome = 'Carlos Jorge do Espirito Santo' where IDUtente = 1009208;
UPDATE donos set Nome = 'Licínio Tomé Sousa Pontes Lima' where IDUtente = 1006661;

```

## 11.11 JUNCTÃO DE VÁRIAS TABELAS - Consultas com duas ou mais tabelas

```

SELECT STR_TO_DATE(DATA, '%Y-%m-%d %H:%i:%s.%f') from passagens limit 5

SELECT Matricula, datediff(sysdate(), DATA) FROM passagens LIMIT 10;

select NOW();
select CURDATE();

SELECT * FROM passagens LIMIT 10;
SELECT IDCarroFK, datediff(sysdate(), DATA) Dias FROM passagens LIMIT 10;

select date_format(sysdate(), '%d/%m/%Y %H:%i:%s');

-- data
select * from passagens where date(data) = str_to_date('2004-06-25', '%Y-%m-%d') LIMIT 10;
select * from passagens
  where date(data) >= str_to_date('2004-06-01', '%Y-%m-%d') AND date(data) <= str_to_date
    ('2004-06-03', '%Y-%m-%d')
LIMIT 10;

-- data e hora 4, 1006, 2004-06-25 07:43:52, 0
-- 131
select * from passagens
  where data >= str_to_date('2004-06-01 08:00:00', '%Y-%m-%d %H:%i:%s') AND
    data <= str_to_date('2004-06-01 08:30:00', '%Y-%m-%d %H:%i:%s')
LIMIT 10;

-- 10,000 x 10,000 -> 100,000,000
-- croos join
SELECT * FROM carros, donos;
SELECT * FROM carros, donos WHERE carros.IDUtenteFK = donos.IDUtente
LIMIT 10;

-- com Alias
SELECT * FROM carros c, donos d WHERE c.IDUtenteFK = d.IDUtente
LIMIT 10;

-- com join
SELECT * FROM carros join donos ON carros.IDUtenteFK = donos.IDUtente
LIMIT 10;

00-FL-51  Porche  Cheyenne  1999  1032963 7319  1032963 Bruna Valbom Mora Martins 19124-571
00-UD-97  Bentley Continental 1994  1057677 4137  1057677 Maria Carrico Real Ferreira 4815-310
03-OI-06  Opel  Corsa 2001  1088210 175 1088210 Rita Coelho Ferraz Amorim Mour 18067-978
01-HU-80  Seat  Cordoba 2001  1142227 3948  1142227 Cristina Aguiar Proenca Rosado Dinis
          Ferreira 3868-746
03-OS-95  Peugeot 305 1991  1183062 7976  1183062 Dina Arouca Rego Dias 5655-319
02-BM-14  Audi  A2 2005  1187890 9438  1187890 Bruno Vivas Cabral Dias Ferreira Real 1320-536
02-VR-05  Audi  A5 2006  1205322 786 1205322 Rita Dinis Tinoco Carneiro Rego Le 16607-933
03-HK-55  Peugeot 305 1998  1294708 7651  1294708 Ana Rego Santos Belchior Pinheiro 12360-327
00-BM-09  Toyota  Avensis 2007  1297933 5641  1297933 Paulo Amorim Bogas Belchior Moura
          16810-818
04-MA-55  Citroen C2 2001  1310920 9434  1310920 Soraia Carrasco Silva Tinoco 4334-733

-- projeção

```

## 11.11. JUNCTÃO DE VÁRIAS TABELAS - CONSULTAS COM DUAS OU MAIS TABELAS 79

```
SELECT carros.Matricula, donos.Nome FROM carros join donos ON carros.IDUtenteFK = donos.
    IDUtente
LIMIT 10;
-- 05-MS-49, Carlos Jorge do Espirito Santo
-- 12-PR-74, Vitor Tinoco Le

SELECT * FROM passagens;
describe passagens;
-- IDCidadeFK, varchar(50), YES, , ,
-- IDCarroFK, varchar(50), YES, , ,
-- Data, datetime, YES, , ,
-- EntradaSaida, varchar(50), YES, , ,

-- 2 tabelas
SELECT c.Cidade, p.* FROM passagens p join cidades c ON p.IDCidadeFK = c.IDCidade
LIMIT 10;
Aveiro, 4, 1006, 2004-06-25 07:43:52, 0
Aveiro, 4, 1006, 2004-06-25 07:43:52, 0

-- 3 tabelas
SELECT ci.Cidade, p.*, c.*
FROM passagens p join cidades ci ON p.IDCidadeFK = ci.IDCidade
join carros c ON c.IDCarro = p.IDCarroFK
LIMIT 10;

-- 4 tabelas
SELECT ci.Cidade, p.*, c.* , d.*
FROM passagens p
    join cidades ci ON p.IDCidadeFK = ci.IDCidade
    join carros c ON p.IDCarroFK = c.IDCarro
    join donos d ON c.IDUtenteFK = d.IDUtente
LIMIT 10;

0 utilizador com nome Francisco no nome passou na autoestrada do dia 2004-07-09?
0 utilizador com nome Francisco no nome passou na autoestrada do dia 2004-07-09 entre as
    09:00:00 e as 10:00:00?

-- 7, 5475, 2004-07-09 09:58:43, 0, 00-AG-30, Peugeot, 205, 1990, 6420646, 5475, 6420646,
    Francisco Pinheiro Brito Almeida, 10192-100

-- datas e horas

Quem não tem carro ?
Quem tem mais do que um carro ?
Quem tem dois do que um carro ?
Quem tem três do que um carro ?

-- Lista de donos e número de carros que têm.
SELECT d.Nome, count(*) numero_carros
FROM carros c join donos d ON c.IDUtenteFK = d.IDUtente
group by d.Nome
    ORDER by d.Nome
LIMIT 10;

-- Lista de donos e número de carros que têm.
SELECT d.Nome, count(*) numero_carros
FROM carros c join donos d ON c.IDUtenteFK = d.IDUtente
group by d.Nome
```

```

ORDER by numero_carros DESC
LIMIT 10;

-- Saídas em Lisboa dia 2004-12-24.
select * from passagens p join cidades ci
where date(data) = str_to_date('2004-12-24','%Y-%m-%d') and cidade = 'Lisboa' and
EntradaSaida=1
LIMIT 10;

-- Quantidade de saídas em Lisboa dia 2004-12-24.
select count(*) 'Número saídas' from passagens p join cidades ci
where date(data) = str_to_date('2004-12-24','%Y-%m-%d') and cidade = 'Lisboa' and
EntradaSaida=1
LIMIT 10;
-- Número saídas
-- 11944

-- Quantidade de saídas em Lisboa dia 12-24 de todos os anos
select year(data) Ano, count(*) 'Número saídas' from passagens p join cidades ci
where day(data) = 24 AND month(data) = 12 and cidade = 'Lisboa' and EntradaSaida=1
group by year(data)
LIMIT 10;

-- Quantidade de saídas em Lisboa dia 2004-12-24, hora a hora
select hour(data) Hora, count(*) 'Número saídas' from passagens p join cidades ci
where date(data) = str_to_date('2004-12-24','%Y-%m-%d') and cidade = 'Lisboa' and
EntradaSaida=1 -- and IDCarroFK=3906
group by hour(data)
LIMIT 10;

select hour(data) Hora, count(*) 'Número saídas' from passagens p join cidades ci
where date(data) = str_to_date('2004-12-25','%Y-%m-%d') and cidade = 'Lisboa' and
EntradaSaida=1 -- and IDCarroFK=3906
group by hour(data)
LIMIT 10;

select
-- sub-consultas
-- Lista de donos e número de carros que têm.
-- MUITO TEMPO !!!!!!!
SELECT d.Nome
FROM donos d
where (select count(*) from carros c where c.IDUtenteFK = d.IDUtente) >0
ORDER by d.Nome
LIMIT 10;

```

### 11.11.1 Aula 2016-12-05/06

1. Donos com nome começado por 'Paulo '

```

SELECT Nome
FROM donos
WHERE Nome LIKE 'Paulo%'

```

```

- - - - -
P a u l o %

```

começado donos S



2. Donos com nome começado por 'Paulo' e terminados com 'Nunes'

```
SELECT Nome
FROM donos
WHERE Nome LIKE 'Paulo%Nunes'
```

3. Donos com nome começado por 'Paulo' e com 'Nunes' em qualquer posição do nome

```
SELECT Nome
FROM donos
WHERE Nome LIKE 'Paulo%Nunes%'

Paulo Nunes Azeredo Bogas Mota Carmona Mora
Edita Apagar Paulo Lopes Carreira Nunes Mora
Edita Apagar Paulo Rosado Mendes Valbom Nunes
Edita Apagar Paulo Domingos Soeiro Rocha Carneiro Nunes Seixas
Edita Apagar Paulo Ramalho Azenha Real Macedo Nunes Peres
Edita Apagar Paulo Nunes Matos Silva Ferraz Arantes Silva
```

```
SELECT Nome
FROM donos
WHERE Nome LIKE 'C%'
```

4. Donos em que o nome contém os três nomes em qualquer posição.

```
SELECT Nome
FROM donos
WHERE Nome LIKE '%Paulo%' AND
Nome LIKE '%Jorge%' AND
Nome LIKE '%Nunes%'
```

4.1

```
SELECT *
FROM donos
WHERE Nome LIKE 'Pedro%Dinis'

4890444 Pedro Nascimento Tinoco Ferro Dinis 11838-811
8301189 Pedro Amorim Eanes Nogueira Rego Dinis 5293-350
```

5. Donos com nome começado por Paulo, Jorge em qualquer posição e Nunes no fim.

```
SELECT Nome
FROM donos
WHERE Nome LIKE 'Pedro%' AND
Nome LIKE '%Jorge%' AND
Nome LIKE '%An'
```

Carros

```
select * from donos where IDUtente = 6420646
6420646 Francisco Pinheiro Brito Almeida 10192-100
```

1. Qual o carro do dono com IDUtente = 4890444

```
SELECT *
FROM carros
WHERE IDUtenteFK = 6420646

00-AG-30 Peugeot 205 1990 6420646 5475
38-OJ-08 Audi A4 2000 6420646 1045
51-CS-62 BMW 520 1999 6420646 3607
```

Que marcas de carros existem ?

```
select Marca FROM carros;
select distinct Marca FROM carros;
```

```
select distinct Marca, Modelo
FROM carros
```

```

-- WHERE
-- GROUP BY
ORDER by Marca
--HAVING
Passagens
describepassagens;

SELECT 'select marca, modelo', ' where marca ='', marca, '', ' and modelo ='', modelo, ''
limit 1' FROM carros group by marca, modelo

Listar os donos com nome começado por Dulce e terminado com Martins.
SELECT * FROM `donos` WHERE nome LIKE 'Dulce%Martins'
9696014 Dulce Cunha Paredes Seixas Amado Martins 15090-548

Carros do dono cujo código é 13596450
select * FROM carros WHERE IDUtenteFK = 13596450
00-AC-58 Opel Corsa 1999 13596450 9559
12-SP-15 Porsche Cheyenne 2001 13596450 9136

Quem é o dono do carro 9559?
SELECT * FROM donos WHERE IDUtente = 13596450
13596450 Carmo Brito Bogas Azeredo Sequeira Clementino 2466-62

Quais as passagens do carro com código 9559?
SELECT * FROM passagens WHERE IDCarroFK = 9559
Mostrando registos 0 - 29 (310 total, 0 Query demorou 0.0761 sec)

SELECT * FROM cidades WHERE cidade = 'Lisboa'
IDCidade (Lisboa) = 8

IDCidade      varchar(50) YES  NULL
IDCarroFK     varchar(50) YES  NULL
Data          varchar(50) YES  NULL
EntradaSaida  varchar(50) YES  NULL

Quais as passagens do carro com código 9559 na cidade de Lisboa?
SELECT * FROM passagens WHERE IDCidade = '8' AND IDCarro = 9559
Mostrando registos 0 - 11 (12 total, 0 Query demorou 1.2117 sec)

Agrupar e Contar Dados

Quantos carros existem de cada marca?
SELECT marca, COUNT(*) FROM carros GROUP BY marca

Quantos carros existem ordenados pela marca?
SELECT marca, COUNT(*) FROM carros GROUP BY marca ORDER BY MARCA

Quantos carros existem ordenados por quantidade de carros?
SELECT marca, COUNT(*) N FROM carros GROUP BY marca ORDER BY N

Quantos carros existem ordenados por quantidade de carros do maior para o menor?
SELECT marca, COUNT(*) N FROM carros GROUP BY marca ORDER BY N desc

Quantos carros existem por marca e modelo?
SELECT marca, modelo, COUNT(*) N FROM carros GROUP BY marca, modelo ORDER BY N

Quantas as passagens do carro com código 9559 na cidade de Lisboa por mês?
SELECT month(data), count(*) FROM `passagens2` where IDUtenteFK = 9559 group by month(data)

```

# Capítulo 12

## Cópia e restauração de base de dados

`SHOW [FULL] TABLES [FROM | IN db_name] [LIKE 'pattern' | WHERE expr]` show tables in scut `SHOW TABLES;`

`DESCRIBE carros;`

Tables\_in\_scut carros cidades distancias donos passagens

```
1 Cópia
2 mysqldump -u root --password= --databases scut > /var/www/EUSOUENGENHEIRO/tmp2/passagens.sql
3
4 Restauro de cópia:
5 mysql -u root --password= scut < /var/www/EUSOUENGENHEIRO/tmp2/passagens.sql
6
7 mysqldump -u root --password= --databases scut > /var/www/EUSOUENGENHEIRO/tmp2/scutLinux.sql
8
9 mysql -u root --password= scut_CMBD < /var/www/EUSOUENGENHEIRO/tmp2/scutLinux.sql
```

### 12.1 Criação e cópia de tabelas

Criação e de uma tabela numa base de dados com base numa tabela de outra base de dados

```
1 CREATE TABLE scut_CMBD.carros SELECT * FROM scut.carros
2 CREATE TABLE scut_CMBD.cidades SELECT * FROM scut.cidades
3 CREATE TABLE scut_CMBD.distancias SELECT * FROM scut.distancias
4 CREATE TABLE scut_CMBD.passagens SELECT * FROM scut.passagens
5
6 Elimina repetidos:
7 CREATE TABLE scut_CMBD.donos SELECT IDUtente, Nome, NumeroDispositivoEletronico FROM scut.donos
   group by IDUtente, Nome, NumeroDispositivoEletronico
8
9 CREATE TABLE scut_CMBD.carros SELECT Matricula, Marca, Modelo, Ano, IDUtenteFK, IDCarro FROM scut.
   carros GROUP BY Matricula, Marca, Modelo, Ano, IDUtenteFK, IDCarro
```



# Bibliografia

- [1] “Mysql,” <https://pt.wikipedia.org/wiki/MySQL>, accessed September 16, 2016.
- [2] “Db-engines ranking of relational dbms,” <http://db-engines.com/en/ranking/relational+dbms>, accessed September 15 2016.
- [3] “Sql,” <https://pt.wikipedia.org/wiki/SQL>, accessed September 16, 2016.
- [4] “Princípios sobre a utilização de dados biométricos no Âmbito do controlo de acessos e de assiduidade,” <https://www.cnpd.pt/bin/orientacoes/principiosbiometricos.htm>, accessed September 24, 2016.
- [5] A. D. REPÚBLICA, “Lei n.º 67/98 de 26 de outubro - lei da protecção de dados pessoais,” <https://www.cnpd.pt/bin/legis/nacional/LPD.pdf>, accessed September 24, 2016.