

Faculdade de Engenharia da Universidade do Porto



CouchSurfing



BDAD 2015/2016 – Mestrado Integrado em Engenharia Informática e Computação

(Carlos Manuel Milheiro de Oliveira Pinto Soares)

Autoras:

Andreia Rodrigues up201404691@fe.up.pt

Inês Gomes up201405778@fe.up.pt

Catarina Ramos up201406219@fe.up.pt

Resumo

Este relatório, elaborado no âmbito da unidade curricular “Base de Dados” do 2º ano e 2º semestre do Mestrado Integrado de Engenharia Informática e Computação, tem como objetivo o desenvolvimento da segunda parte do projeto proposto pelo professor da cadeira no princípio do semestre.

Nesta segunda parte, foi nos proposto a elaboração de um modelo relacional, LDD e LMD com base no tema escolhido na primeira fase.

Neste relatório temos também uma versão atualizada do diagrama de classes UML proposto na primeira parte do projeto, devido a algumas alterações que tivemos que implementar.

Índice

Resumo.....	2
Índice	3
1.Introdução	4
2.Diagrama UML.....	5
3.Eschema relacional.....	6
4. Instruções LDD (Linguagem de Definição de dados)	7
5.Instruções LMD (Linguagem Manipulação de Dados).....	13
7. Conclusões	15
8. Bibliografia	16

1. Introdução

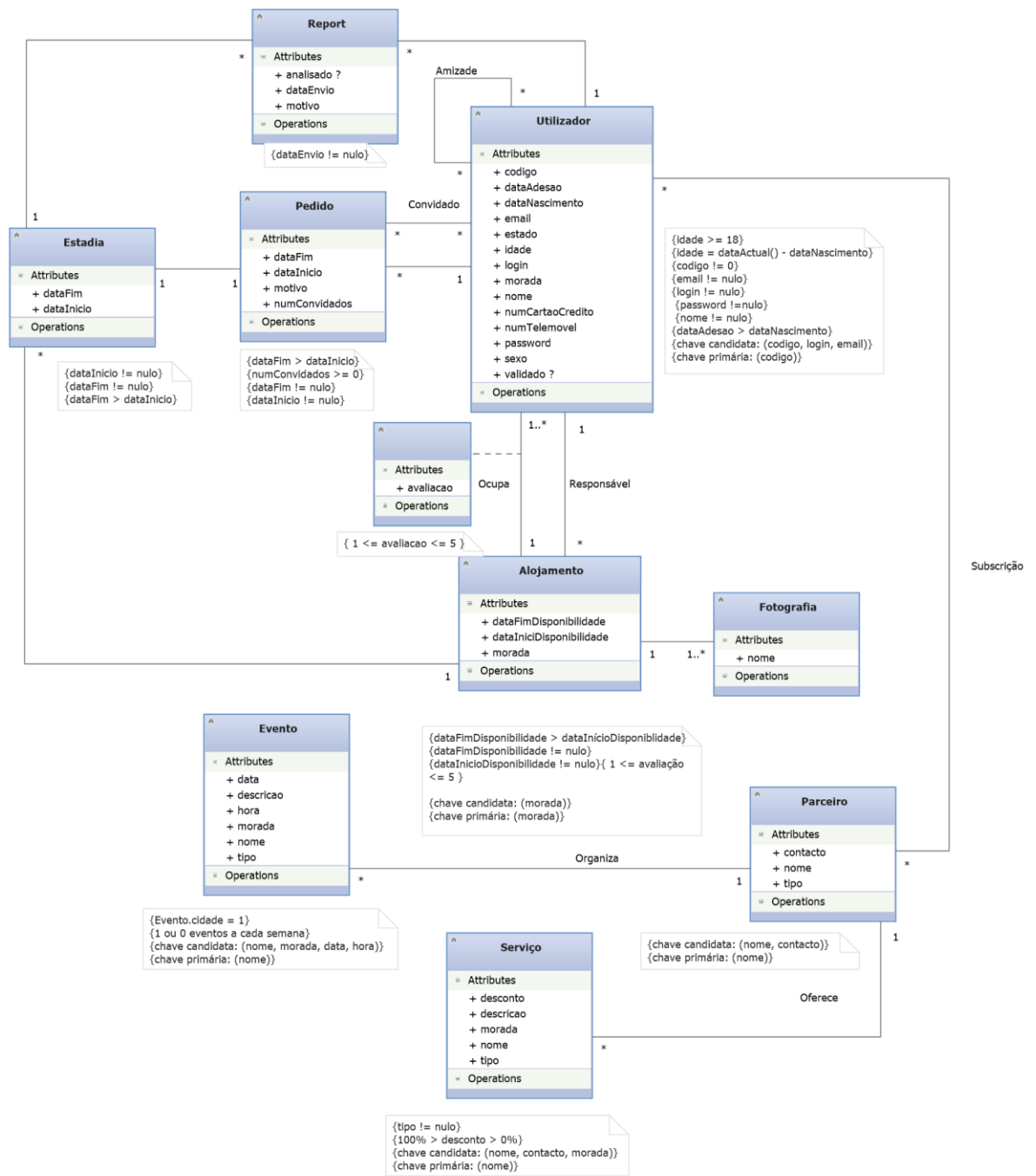
O *CouchSurfing* é uma rede de hospedagem gratuita para viajantes que procuram uma experiência diferente e inovadora. A ideia foi criada em 2003 e foi implementada por uma organização sem fins lucrativos.

Para usufruir deste tipo de alojamento, existe um site oficial destinado a este tipo de serviço. Dentro deste, os utilizadores após a sua inscrição, têm os meios necessários á pesquisa de um alojamento ou prestação do mesmo numa certa cidade de um dado país. A pesquisa inclui também as datas entre as quais o viajante deseja um alojamento e o número de convidados que pretendem participar nesta viagem.

Aquando a viagem, os utilizadores podem usufruir de diversos serviços e eventos semanais, divulgados pelos parceiros ligados à organização, que promovem novas experiências e o contacto com os locais e outros *couchSurfers*.

Um dos problemas que mais se opõe a este conceito, é garantir a segurança dos que estão envolvidos, tanto como a sua legitimidade e a das residências.

2. Diagrama UML



3. Esquema relacional

Utilizador (codigoUtilizador, dataAdesao, dataNascimento, mail, login, morada, estado, idade, nome, numeroCC, numeroTel, password, sexo, validado)

Amizade (codigoUtilizador1->Utilizador, codigoUtilizador2->Utilizador)

Alojamento (idAlojamento, morada, dataInicioDisponibilidade, dataFimDisponibilidade, codigoResponsável -> Utilizador, idFotografia -> Fotografia)

Fotografia (idFotografia, nome, idAlojamento->Alojamento)

Ocupa (idOcupacao, codigoUtilizador -> Utilizador, idAlojamento->Alojamento, avaliação)

Responsável (idResponsável, codigoUtilizador -> Utilizador, idAlojamento->Alojamento)

Pedido (idPedido, dataInicioEstadia, dataFimEstadia, motivo, numeroConvidados, codigoAlojamento->Alojamento, codigoUtilizador->Utilizador)

Convidado (codigoUtilizador->Utilizador, idPedido -> Pedido)

Estadia (idEstadia, codigoUtilizador->Utilizador, idPedido->Pedido)

Reclamação (idReclamação, motivo, dataEnvio, analisado, idEstadia->Estadia, codigoUtilizador -> Utilizador)

Parceiro (idParceiro, nome, tipo, contacto)

Subscrição (codigoUtilizador->Utilizador, idParceiro->Parceiro)

Evento (idEvento, morada, nome, descrição, data, tipo, hora, idParceiro->Parceiro)

Serviço (idServiço, desconto, morada, nome, descrição, tipo, idParceiro->Parceiro)

Oferece (idOferta, idParceiro->Parceiro, idServico->Servico)

Organiza (idOrganizacao, idParceiro->Parceiro, idEvento -> Evento)

Utilizador	Atributo	Nulos?	Domínio
	código	N	ID
	nome	N	Text
	idade	N	Integer
	email	N	Text
	morada	Y	Text
	dataAdesao	N	Date
	dataNascimento	N	Date
	sexo	Y	Text
	login	N	Text
	password	N	Text
	estado	Y	Numeric
	numCartaoCredito	Y	Integer
	numTelemovei	Y	Integer
	validado	Y	Numeric
Chave-Candidata	{código, email, login}		
Chave-Primária	{código}		
Mais acedidos	{código,nome,login,estado}		

Alojamento	Atributo	Nulos?	Domínio
	idAlojamento	N	ID
	dataInicioDisponibilidade	N	Date
	DataFimDisponibilidade	N	Date
	morada	N	Text
	codigoResponsavel	N	ID
	idFotografia	N	Integer
Chave-Candidata	{idAlojamento, morada, codigoResponsavel,idFotografia}		
Chave-Primária	{idAlojamento}		
Mais acedidos	{idAlojamento, idFotografia, codigoResponsavel, avaliacao}		

Ocupa	Atributo	Nulos?	Domínio
	idOcupacao	N	ID
	codigoUtilizador	N	ID
	idAlojamento	N	ID
	avaliacao	N	Integer
Chave-Candidata	{idOcupacao,codigoUtilizador,idAlojamento}		
Chave-Primária	{idOcupacao}		
Mais acedidos	{idOcupacao, codigoUtilizador, idAlojamento, avaliacao}		

Fotografia	Atributo	Nulos?	Domínio
	idFotografia	N	ID
	nome	Y	Text
	idAlojamento	N	ID
Chave-Candidata	{idFotografia,idAlojamento,nome}		
Chave-Primária	{idFotografia}		
Mais acedidos	{idFotografia,idAlojamento}		

Amizade	Atributo	Nulos?	Domínio
	codigoUtilizador1	N	ID
	codigoUtilizador2	N	ID
Chave-Candidata	{codigoUtilizador1,codigoUtilizador2}		
Chave-Primária	{codigoUtilizador1,codigoUtilizador2}		
Mais acedidos	{codigoUtilizador1,codigoUtilizador2}		

Pedido	Atributo	Nulos?	Domínio
	idPedido	N	ID
	dataInicioEstadia	N	Date
	dataFimEstadia	N	Date
	numeroConvidados	Y	Integer
	motivo	Y	Text
	codigoUtilizador	N	ID
	codigoAlojamento	N	ID
Chave-Candidata	{idPedido, codigoUtilizador, codigoAlojamento}		
Chave-Primária	{idPedido}		
Mais acedidos	{idPedido,codigoUtilizador,codigoAlojamento}		

Convidado	Atributo	Nulos?	Domínio
	codigoUtilizador	N	ID
	idPedido	N	ID
Chave-Candidata	{codigoUtilizador,idPedido}		
Chave-Primária	{codigoUtilizador,idPedido}		
Mais acedidos	{codigoUtilizador,idPedido}		

Estadia	Atributo	Nulos?	Domínio
	idEstadia	N	ID
	codigoUtilizador	N	ID
	idPedido	N	ID
Chave-Candidata	{idEstadia,codigoUtilizador,idPedido}		
Chave-Primária	{idEstadia}		
Mais acedidos	{idEstadia,codigoUtilizador,idPedido}		

Reclamacao	Atributo	Nulos?	Domínio
	idReclamacao	N	ID
	dataEnvio	N	Date
	motivo	N	Text
	analizado	N	Numeric
	idEstadia	N	ID
	codigoUtilizador	N	ID
Chave-Candidata	{idReclamacao,idEstadia,codigoUtilizador}		
Chave-Primária	{idReclamacao}		
Mais acedidos	{idReclamacao,idEstadia,codigoUtilizador}		

Evento	Atributo	Nulos?	Domínio
	idEvento	N	ID
	nome	N	Text
	data	N	Date
	horaInicio	N	Numeric
	morada	N	Text
	tipo	Y	Text
	descricao	N	Text
	idParceiro	N	ID
Chave-Candidata	{idEvento, nome, data, hora, morada, idParceiro}		
Chave-Primária	{idEvento}		
Mais acedidos	{idEvento, idParceiro}		

Servico	Atributo	Nulos?	Domínio
	idServico	N	ID
	nome	N	Text
	tipo	Y	Text
	morada	N	Text
	desconto	N	Real
	descricao	N	Text
	idParceiro		ID
Chave-Candidata	{idServico, nome, morada, idParceiro}		
Chave-Primária	{idServico}		
Mais acedidos	{idServico, nome, tipo, desconto, idParceiro}		

Subscricao	Atributo	Nulos?	Domínio
	codigoUtilizador	N	ID
	idParceiro	N	ID
Chave-Candidata	{codigoUtilizador, idParceiro}		
Chave-Primária	{codigoUtilizador, idParceiro}		
Mais acedidos	{codigoUtilizador, idParceiro}		

Oferece	Atributo	Nulos?	Domínio
	idParceiro	N	ID
	idServico	N	ID
Chave-Candidata	{idParceiro, idServico}		
Chave-Primária	{idParceiro, idServico}		
Mais acedidos	{idParceiro, idServico}		

Parceiro	Atributo	Nulos?	Domínio
	idParceiro	N	ID
	nome	N	Integer
	tipo	N	Text
	contacto	N	Integer
Chave-Candidata	{idParceiro, nome}		
Chave-Primária	{idParceiro}		
Mais acedidos	{idParceiro, nome, tipo, contacto}		

4. Instruções LDD (Linguagem de Definição de dados)

```
CREATE TABLE Utilizador (
```

```
    código                INTEGER NOT NULL UNIQUE,  
    nome                  TEXT NOT NULL,  
    idade                 INTEGER NOT NULL,  
    email                 TEXT NOT NULL UNIQUE,  
    morada                TEXT,  
    dataAdesao            NUMERIC NOT NULL,  
    dataNascimento        NUMERIC NOT NULL,  
    sexo                  TEXT,  
    login                 TEXT NOT NULL UNIQUE,  
    password              TEXT NOT NULL,  
    estado                NUMERIC,  
    numCartaoCredito      INTEGER,  
    numTelemovel          INTEGER,  
    validado              NUMERIC CHECK( numCartaoCredito != NULL),  
    PRIMARY KEY(código),  
    CHECK( dataAdesao >= dataNascimento + 18)
```

```
);
```

```
CREATE TABLE Alojamento (
```

```
    idAlojamento          INTEGER NOT NULL UNIQUE,  
    dataInicioDisponibilidade NUMERIC NOT NULL,  
    dataFimDisponibilidade NUMERIC NOT NULL,  
    morada                 TEXT NOT NULL UNIQUE,  
    codigoResponsavel      INTEGER NOT NULL UNIQUE,  
    idFotografia           INTEGER NOT NULL UNIQUE,  
    PRIMARY KEY(idAlojamento,morada),  
    FOREIGN KEY(codigoResponsavel) REFERENCES Utilizador(código),  
    FOREIGN KEY(idFotografia) REFERENCES Fotografia(idFotografia) ON DELETE SET NULL,  
    CHECK (dataFimDisponibilidade > dataInicioDisponibilidade)
```

```
);
```

```

CREATE TABLE Ocupa (
    idOcupacao          INTEGER NOT NULL UNIQUE,
    codigoUtilizador     INTEGER NOT NULL UNIQUE,
    idAlojamento        INTEGER NOT NULL UNIQUE,
    avaliacao            INTEGER NOT NULL,
    PRIMARY KEY(idOcupacao),
    FOREIGN KEY(codigoUtilizador) REFERENCES Utilizador(codigo),
    FOREIGN KEY(idAlojamento) REFERENCES Alojamento(idAlojamento)
);

```

```

CREATE TABLE Fotografia (
    idFotografia         INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,
    nome                 TEXT NOT NULL UNIQUE,
    idAlojamento         INTEGER NOT NULL UNIQUE,
    FOREIGN KEY(idAlojamento) REFERENCES Alojamento(idAlojamento)
);

```

```

CREATE TABLE Responsavel (
    idResponsavel        INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,
    codigoUtilizador     INTEGER NOT NULL UNIQUE,
    idAlojamento         INTEGER NOT NULL UNIQUE,
    FOREIGN KEY(codigoUtilizador) REFERENCES Utilizador(codigo),
    FOREIGN KEY(idAlojamento) REFERENCES Alojamento(idAlojamento)
);

```

```

CREATE TABLE Amizade (
    codigoUtilizador1     INTEGER NOT NULL UNIQUE,
    codigoUtilizador2     INTEGER NOT NULL UNIQUE,
    PRIMARY KEY(codigoUtilizador1,codigoUtilizador2),
    FOREIGN KEY(codigoUtilizador1) REFERENCES Utilizador(codigo),
    FOREIGN KEY(codigoUtilizador2) REFERENCES Utilizador(codigo)
);

```

);

CREATE TABLE Pedido (

idPedido INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,
dataInicioEstadia NUMERIC NOT NULL,
dataFimEstadia NUMERIC NOT NULL,
numeroConvidados INTEGER CHECK (numeroConvidados >= 0),
motivo TEXT,
codAlojamento INTEGER NOT NULL UNIQUE,
codUtilizador INTEGER NOT NULL UNIQUE,
FOREIGN KEY(codAlojamento) REFERENCES Alojamento(idAlojamento),
FOREIGN KEY(codUtilizador) REFERENCES Utilizador(codigo),
CHECK (dataFimEstadia > dataInicioEstadia)

);

CREATE TABLE Convidado (

codigoUtilizador INTEGER NOT NULL UNIQUE,
idPedido INTEGER NOT NULL UNIQUE,
FOREIGN KEY(codigoUtilizador) REFERENCES Utilizador(codigo),
FOREIGN KEY(idPedido) REFERENCES Pedido(idPedido)

);

CREATE TABLE Estadia (

idEstadia INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,
codigoUtilizador INTEGER NOT NULL UNIQUE,
idPedido INTEGER NOT NULL UNIQUE,
FOREIGN KEY(codigoUtilizador) REFERENCES Utilizador(codigo),
FOREIGN KEY(idPedido) REFERENCES Pedido(idPedido)

);

CREATE TABLE Reclamação (

idReclamacao INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,
dataEnvio NUMERIC NOT NULL,

```

motivo                TEXT NOT NULL,
analisado              NUMERIC NOT NULL,
idEstadia              INTEGER NOT NULL UNIQUE,
codigoUtilizador       INTEGER NOT NULL UNIQUE,
FOREIGN KEY(idEstadia) REFERENCES Estadia(idEstadia)
FOREIGN KEY(codigoUtilizador) REFERENCES Utilizador(codigo)

);

CREATE TABLE Parceiro (
    idParceiro          INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,
    nome                INTEGER NOT NULL UNIQUE,
    tipo                TEXT NOT NULL,
    contacto            INTEGER NOT NULL
);

CREATE TABLE Evento (
    idEvento            INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,
    nome                TEXT NOT NULL UNIQUE,
    data                NUMERIC NOT NULL UNIQUE,
    hora                NUMERIC NOT NULL UNIQUE,
    morada              TEXT NOT NULL UNIQUE,
    tipo                TEXT,
    descricao           TEXT NOT NULL,
    idParceiro          INTEGER NOT NULL UNIQUE,
    FOREIGN KEY(idParceiro) REFERENCES Parceiro(idParceiro)
);

CREATE TABLE Servico (
    idServico           INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,
    nome                TEXT NOT NULL UNIQUE,
    tipo                TEXT,
    morada              TEXT NOT NULL UNIQUE,

```

```

desconto          REAL NOT NULL CHECK( 0% < desconto < 100%),
descricao         TEXT NOT NULL,
idParceiro        INTEGER NOT NULL UNIQUE,
FOREIGN KEY(idParceiro) REFERENCES Parceiro(idParceiro)
);

```

```

CREATE TABLE Subscricao (
    codigoUtilizador    INTEGER NOT NULL UNIQUE,
    idParceiro          INTEGER NOT NULL UNIQUE,
    FOREIGN KEY(codigoUtilizador) REFERENCES Utilizador(codigo),
    FOREIGN KEY(idParceiro) REFERENCES Parceiro(idParceiro)
);

```

```

CREATE TABLE Oferece (
    idOferta           INTEGER NOT NULL UNIQUE,
    idParceiro         INTEGER NOT NULL UNIQUE,
    idServico          INTEGER NOT NULL UNIQUE,
    PRIMARY KEY(idOferta),
    FOREIGN KEY(idParceiro) REFERENCES Parceiro(idParceiro),
    FOREIGN KEY(idServico) REFERENCES Servico(idServico)
);

```

```

CREATE TABLE Organiza (
    idOrganizacao      INTEGER NOT NULL UNIQUE,
    idParceiro         INTEGER NOT NULL UNIQUE,
    idEvento           INTEGER NOT NULL UNIQUE,
    PRIMARY KEY(idOrganizacao),
    FOREIGN KEY(idParceiro) REFERENCES Parceiro(idParceiro),
    FOREIGN KEY(idEvento) REFERENCES Evento(idEvento)
);

```

```
CREATE TRIGGER UpdateConvidados
AFTER INSERT Convidado
BEGIN
    Pedido.numeroConvidados++
END
```

5. Instruções LMD (Linguagem Manipulação de Dados)

```
INSERT INTO Utilizador (código, nome, idade, email, morada, dataAdesao, dataNascimento,
sexo, login, password, estado, numCartaoCredito, numTelemovel, validado)
```

```
VALUES (1, 'Andreia', 20, 'andreia@mail.com', 'Rua da Andreia, Porto, Portugal', '22-04-2016',
'13-03-1996', 'Feminino', 'andreia', '123', 123456789, null, '123456789', 1);
```

```
INSERT INTO Utilizador (código, nome, idade, email, morada, dataAdesao, dataNascimento,
sexo, login, password, estado, numCartaoCredito, numTelemovel, validado)
```

```
VALUES (2, 'Inês', 20, 'ines@mail.com', 'Rua da Inês, Madrid, Espanha', '22-04-2016', '08-08-
1996', 'Feminino', 'inês', '123', 0, null, '123456780', 0);
```

```
INSERT INTO Utilizador (código, nome, idade, email, morada, dataAdesao, dataNascimento,
sexo, login, password, estado, numCartaoCredito, numTelemovel, validado)
```

```
VALUES (3, 'Catarina', 20, 'catarina@mail.com', 'Rua da Catarina, São Miguel, Açores, Portugal',
'22-04-2016', '24-02-1996', 'Feminino', 'catarina', '123', 0, null, '123456781', 0);
```

```
INSERT INTO Alojamento (idAlojamento, dataInicioDisponibilidade, dataFimDisponibilidade,
morada, codigoResponsavel, idFotografia)
```

```
VALUES (1, '01-05-2016', '01-06-2016', 'Rua da Andreia, Porto, Portugal', 1, 1);
```

```
INSERT INTO Responsavel (idResponsavel, codigoUtilizador, idAlojamento) VALUES (1, 1, 1);
```

```
INSERT INTO Ocupa (idOcupacao, codigoUtilizador, idAlojamento, avaliacao) VALUES (1, 2, 1,
4);
```

```
INSERT INTO Ocupa (idOcupacao, codigoUtilizador, idAlojamento, avaliacao) VALUES (1, 3, 1,
4);
```

```
INSERT INTO Fotografia (idFotografia, nome, idAlojamento) VALUES (1, 'Fotografia.png', 1);
```

```
INSERT INTO Amizade (codigoUtilizador1, codigoUtilizador2) VALUES (1,2);
```

```
INSERT INTO Amizade (codigoUtilizador1, codigoUtilizador2) VALUES (2,3);
```


INSERT INTO Pedido (idPedido, dataInicioEstadia, dataFimEstadia, numeroConvidados, motivo, codAlojamento, codUtilizador) VALUES (1, '02-05-2016', '08-05-2016', 1, 'Eu quero ir á queima', 1, 2);

INSERT INTO Convidado (codigoConvidado, codigoUtilizador, idPedido) VALUES (3,2,1);

INSERT INTO Estadia (idEstadia , codigoUtilizador , idPedido) VALUES (1, 2, 1);

INSERT INTO Estadia (idEstadia , codigoUtilizador , idPedido) VALUES (1, 3, 1);

INSERT INTO Reclamação (idReclamacao , dataEnvio , motivo , analisado , idEstadia) VALUES (1, '09-05-2016', 'A água era fria', 0, 1, 3);

INSERT INTO Parceiro (idParceiro , nome , tipo , contacto) VALUES (1, 'Federação Académica do Porto', 'Estudantil', '123456781');

INSERT INTO Evento (idEvento, nome, data, hora, morada, tipo, descrição, idParceiro) VALUES (1, 'Queima', '01-05-16', '23:00', 'Queimódromo, Porto', null, 'Festa Academica', 1);

INSERT INTO Organiza (idOrganizacao, idParceiro, idEvento) VALUES (1, 1, 1);

INSERT INTO Parceiro (idParceiro , nome , tipo , contacto) VALUES (2, 'Senhor Guia Turistico', 'Guia Turistico, '123456782');

INSERT INTO Serviço (idServico, nome, tipo, morada, desconto, descrição, idParceiro) VALUES (1, 'Roteiro pela cidade do Porto', 'Turistico', 'Aliados, Porto, Portugal', 20%, 'Rota para estrangeiros', 2);

INSERT INTO Oferece (idOferta, idParceiro, idServico) VALUES (1, 2, 1);

INSERT INTO Subscrição (idSubscricao , codigoUtilizador , idParceiro) VALUES (1, 2, 2);

7. Conclusões

Ao desenvolver este trabalho, deparamo-nos com alguns lapsos no diagrama de classes UML, aquando a estruturação do modelo relacional (MR), e portanto, fizemos algumas alterações ao mesmo.

Esta segunda parte do trabalho permitiu-nos também, compreender e por em prática alguns dos conceitos em relação à Linguagem de Definição de Dados (LDD) e Linguagem de Modulação de Dados (LMD), onde o adaptamos ao tema do nosso projeto.

8. Bibliografia

<https://www.couchsurfing.com/>

<https://en.wikipedia.org/wiki/CouchSurfing>

<http://www.flyertalk.com/forum/budget-travel/1560406-hostels-vs-couchsurfing.html>

<http://www.fodors.com/community/europe/travel-safety-sexual-assault-couchsurfing-gorg.cfm>