

Trabalho Experimental – Etapa Final – Protocolo A

Engenharia Informática Base de Dados

Paulo Nogueira Martins
Daniel Moreira Lopes Alexandre
António Jorge Gonçalves de Gouveia

Autores

Guilherme Andrade - 73475
Bernardo Meneses - 74116
Rui Barbosa – 73144
Guilherme Cruz - 73752

Vila Real, 2022

Índice

Introdução	3
Enquadramento Teórico	3
Modelo Entidade-Relacionamento e Modelo Relacional	3
Procedimentos	3
Triggers	4
Desenvolvimento – etapa 1	5
1. Mapeamento do diagrama E-R para o modelo relacional	5
2. Normalização do modelo relacional até à 3ª Forma Normal	7
3. Implementação do modelo físico da base de dados com as respetivas restrições de integridade, recorrendo à linguagem SQL	14
4. Criação do diagrama da base de dados (apresentação gráfica das tabelas e suas relações)	17
Desenvolvimento – etapa 2	18
1. Registos em SQL	18
2. Questões de SQL	25
3. Procedimento em SQL	27
4. Trigger em SQL	28
Conclusão	28
Webgrafia	28

Introdução

Na unidade curricular de Base de Dados, foi nos proposto a elaboração de um projeto consistindo na criação de uma base de dados.

O objetivo deste trabalho seria de melhor adquirir conhecimentos sobre a arquitetura de uma base de dados em SQL (Structured Query Language).

Este relatório está dividido entre duas parte, “etapa 1” e “etapa 2”, cada uma é esta referente à primeira etapa e à segunda etapa do projeto, respetivamente.

Enquadramento Teórico

Modelo Entidade-Relacionamento e Modelo Relacional

Um modelo E-R é uma maneira de descrever o processo de armazenamento de dados. Desta forma definimos as entidades que representam as tabelas da base de dados, que estão ligadas por relacionamentos, a outras tabelas. As entidades são caracterizadas pelos atributos que as constituem, neste caso, as colunas de cada tabela.

Procedimentos

Procedimentos são o equivalente em SQL a subrotinas existentes em outras linguagens de programação.

Contêm:

- Variáveis
- Tipos de dados
- Parâmetros de entrada/saída
- Valores de retorno
- Execuções condicionais
- Ciclos
- Comentários

Triggers

Um Trigger é um tipo especial de Stored Procedure que é executado automaticamente como parte de uma modificação de dados.

Os Triggers estão associados com uma tabela numa base de dados e são executados pelo SGBD quando uma alteração específica ocorre numa tabela. As alterações podem ser o resultado de uma das seguintes declarações de modificação de dados:

- INSERT
- UPDATE
- DELETE

Existem 2 tipos de Triggers:

- After Triggers – são executados depois da alteração na tabela
- Instead-Of Triggers – são executados em vez da alteração na tabela

Desenvolvimento – etapa 1

1. Mapeamento do diagrama E-R para o modelo relacional

Jardins (ID, Nome, Nome_Alterativo)

Cidades (ID, Nome, Área, Localização (Continente, País))

Edifícios (Código, Área_Construída, Tipologia)

Funcionários (ID, Nome, Contacto)

Quadro (Anos_Serviço, Escalão)

Temporario (Seguro)

Jardineiros (ID, Nome, Apelido, Total_Horas, Endereço (Morada, CódigoPostal, Localidade))

Conter (ID_Jardins, ID_Cidades, ID_Jardineiros, Data_Inicio, Data_Contratar, Código_Edificios, ID_Funcionários, Data_Possuir)

ID_Jardins referencia **Jardins**

ID_Cidades referencia **Cidades**

ID_Jardineiros referencia **Jardineiros**

Data_Inicio referencia **Manutenção**

Data_Contratar referencia **Contratar**

Código_Edificios referencia **Edifícios**

ID_Funcionários referencia **Funcionários**

Data_Possuir referencia **Possuir**

ContratarJard (ID_Jardins, ID_Cidades, ID_Jardineiros, Data_Inicio, Código_Edificios, ID_Funcionários, Data_Possuir, Data_ContratarFunc, Data_ContratarJard, Salario_ContratarJard)

ID_Jardins referencia **Jardins**

ID_Cidades referencia **Cidades**

ID_Jardineiros referencia **Jardineiros**

Data_Inicio referencia **Manutenção**

Código_Edificios referencia **Edifícios**

ID_Funcionários referencia **Funcionários**

Data_Possuir referencia **Possuir**

Data_Contratar2 referencia **Contratar2**

ContratarFunc (ID_Jardins, ID_Cidades, ID_Jardineiros, Data_Inicio, Código_Edificios, ID_Funcionários, Data_Possuir, Data_ContratarJard, Data_ContratarFunc, Preço_ContratarFunc)

ID_Jardins referencia **Jardins**

ID_Cidades referencia **Cidades**

ID_Jardineiros referencia **Jardineiros**

Data_Inicio referencia **Manutenção**

Código_Edificios referencia **Edifícios**

ID_Funcionários referencia **Funcionários**

Data_Possuir referencia **Possuir**

Data_Contratar1 referencia **Contratar1**

Manutenção (~~ID_Jardins~~, ~~ID_Jardineiros~~, ~~Data_Contratar~~, Data_Inicio, Data_Fim)

ID_Jardins referencia **Jardins**

~~Data_Contratar~~ referencia **Contratar**

ID_Jardineiros referencia **Jardineiros**

Possuir (~~ID_Jardins~~, ~~ID_Cidades~~, ~~ID_Jardineiros~~, ~~Data_Inicio~~, ~~Data_Contratar~~, ~~Código_Edificios~~, ~~ID_Funcionários~~, Data_Possuir, Data_Fim_Possuir, Valor_Possuir)

ID_Jardins referencia **Jardins**

ID_Cidades referencia **Cidades**

ID_Jardineiros referencia **Jardineiros**

~~Data_Inicio~~ referencia **Manutenção**

~~Data_Contratar~~ referencia **Contratar**

~~Código_Edificios~~ referencia **Edificios**

~~ID_Funcionários~~ referencia **Funcionários**

Responsaveis (~~ID_Jardins~~, ~~ID_Cidades~~, ~~ID_Jardineiros~~, ~~Data_Inicio~~, ~~Data_Contratar~~, ~~Código_Edificios~~, ~~ID_Funcionários~~, ~~Data_Possuir~~, Horas_Extra)

ID_Jardins referencia **Jardins**

ID_Cidades referencia **Cidades**

ID_Jardineiros referencia **Jardineiros**

~~Data_Inicio~~ referencia **Manutenção**

~~Data_Contratar~~ referencia **Contratar**

~~Código_Edificios~~ referencia **Edificios**

~~ID_Funcionários~~ referencia **Funcionários**

~~Data_Possuir~~ referencia **Possuir**

2. Normalização do modelo relacional até à 3ª Forma Normal

Jardins (ID, Nome, Nome_Alternativo)

Jardins

1ª Forma

Já se encontra na 1ª forma normal.

2ª Forma

Já se encontra na 2ª forma normal.

3ª Forma

Já se encontra na 3ª forma normal.

Cidades (ID, Nome, Área, Localização (Continente, País))

Cidades

1ª Forma

Já se encontra na 1ª forma normal.

2ª Forma

Já se encontra na 2ª forma normal.

3ª Forma

Já se encontra na 3ª forma normal.

Edifícios (Código, Área_Construída, Tipologia)

Edifícios

1ª Forma

Já se encontra na 1ª forma normal.

2ª Forma

Já se encontra na 2ª forma normal.

3ª Forma

Já se encontra na 3ª forma normal.

Funcionários (ID, Nome, Contacto)

Funcionários

1ª Forma

Já se encontra na 1ª forma normal.

2ª Forma

Já se encontra na 2ª forma normal.

3ª Forma

Já se encontra na 3ª forma normal.

Quadro (Anos_Serviço, Escalão)

Quadro

1ª Forma

Já se encontra na 1ª forma normal.

2ª Forma

Já se encontra na 2ª forma normal.

3ª Forma

Já se encontra na 3ª forma normal.

Temporario (Seguro)

Temporario

1ª Forma

Já se encontra na 1ª forma normal.

2ª Forma

Já se encontra na 2ª forma normal.

3ª Forma

Já se encontra na 3ª forma normal.

Jardineiros (ID, Nome, Apelido, Total_Horas, Endereço (Morada, CódigoPostal, Localidade))

Jardineiros

1ª Forma

Relação: Jardineiros -> Contactos

Nova Entidade: Contactos (ID_Jardineiros, Contacto)

2ª Forma

Já se encontra na 2ª forma normal.

3ª Forma

Já se encontra na 3ª forma normal.

Contactos (ID_Jardineiros (FK), Contacto)

Contactos

1ª Forma

Já se encontra na 1ª forma normal.

2ª Forma

Não se encontra na 2ª forma porque o atributo **Contacto** depende funcionalmente da chave estrangeira ID_Jardineiros.

Contactos (ID_Contactos, ID_Jardineiros (FK), Contacto)

3ª Forma

Já se encontra na 3ª forma normal.

Conter (ID_Jardins, ID_Cidades, ID_Jardineiros, Data_Inicio, Data_Contratar, Código_Edificios, ID_Funcionários, Data_Possuir)

Conter

1ª Forma

Já se encontra na 1ª forma normal.

2ª Forma

Já se encontra na 2ª forma normal.

3ª Forma

Já se encontra na 3ª forma normal.

ContratarJard (ID_Jardins, ID_Cidades, ID_Jardineiros, Data_Inicio, Código_Edificios, ID_Funcionários, Data_Possuir, Data_ContratarFunc, Data_ContratarJard, Salario_ContratarJard)

Contratar

1ª Forma

Já se encontra na 1ª forma normal.

2ª Forma

Já se encontra na 2ª forma normal.

3ª Forma

Já se encontra na 3ª forma normal.

ContratarFunc (ID_Jardins, ID_Cidades, ID_Jardineiros, Data_Inicio, Código_Edificios,
ID_Funcionários, Data_Possuir, Data_ContratarJard, Data_ContratarFunc, Preço_ContratarFunc)

Contratar

1ª Forma

Já se encontra na 1ª forma normal.

2ª Forma

Já se encontra na 2ª forma normal.

3ª Forma

Já se encontra na 3ª forma normal.

Manutenção (ID_Jardins, ID_Jardineiros, Data_Contratar, Data_Inicio, Data_Fim)

Manutenção

1ª Forma

Já se encontra na 1ª forma normal.

2ª Forma

Já se encontra na 2ª forma normal.

3ª Forma

Já se encontra na 3ª forma normal.

Possuir (ID_Jardins, ID_Cidades, ID_Jardineiros, Data_Inicio, Data_Contratar, Código_Edificios, ID_Funcionários, Data_Possuir, Data_Fim_Possuir, Valor_Possuir)

Possuir

1ª Forma

Já se encontra na 1ª forma normal.

2ª Forma

Já se encontra na 2ª forma normal.

3ª Forma

Já se encontra na 3ª forma normal.

Responsaveis (ID_Jardins, ID_Cidades, ID_Jardineiros, Data_Inicio, Data_Contratar, Código_Edificios, ID_Funcionários, Data_Possuir, Horas_Extra)

Responsaveis

1ª Forma

Já se encontra na 1ª forma normal.

2ª Forma

Já se encontra na 2ª forma normal.

3ª Forma

Já se encontra na 3ª forma normal.

3. Implementação do modelo físico da base de dados com as respetivas restrições de integridade, recorrendo à linguagem SQL

```
CREATE DATABASE BaseDados;
```

```
USE BaseDados  
GO
```

```
CREATE TABLE Jardins (  
  ID_Jardins          int IDENTITY NOT NULL,  
  Nome                VARCHAR(MAX) NOT NULL,  
  Nome_Alternativo    VARCHAR(MAX) NOT NULL,  
  PRIMARY KEY (ID_Jardins));
```

```
CREATE TABLE Cidades (  
  ID_Cidades          int IDENTITY NOT NULL,  
  Nome                VARCHAR(MAX) NOT NULL,  
  Area                FLOAT(20) NOT NULL,  
  Localizacao_Continente  VARCHAR(MAX) NOT NULL,  
  Localizacao_Pais      VARCHAR(MAX) NOT NULL,  
  PRIMARY KEY (ID_Cidades));
```

```
CREATE TABLE Edificios (  
  Codigo_Edificio     int IDENTITY NOT NULL,  
  Area_Construida     VARCHAR(MAX) NOT NULL,  
  Tipologia           VARCHAR(MAX) NOT NULL,  
  PRIMARY KEY (Codigo_Edificio));
```

```
CREATE TABLE Funcionarios (  
  ID_Funcionarios     int IDENTITY NOT NULL,  
  Nome                VARCHAR(MAX) NOT NULL,  
  Contacto            integer NOT NULL,  
  PRIMARY KEY (ID_Funcionarios));
```

```
CREATE TABLE Jardineiros (  
  ID_Jardineiros      int IDENTITY NOT NULL,  
  Nome                VARCHAR(MAX) NOT NULL,  
  Apelido             VARCHAR(MAX) NOT NULL,  
  Contactos           INTEGER NOT NULL,  
  Total_Horas         INTEGER NOT NULL,  
  Endereço_Morada     VARCHAR(MAX) NOT NULL,  
  Endereço_Código_Postal  VARCHAR(MAX) NOT NULL,  
  Endereço_Localidade  VARCHAR(MAX) NOT NULL,  
  PRIMARY KEY (ID_Jardineiros));
```

```

CREATE TABLE Quadro (
IDQ                INTEGER NOT NULL,
Anos_Serviço       VARCHAR(MAX) NOT NULL,
Escalao            VARCHAR(MAX) NOT NULL,
Contacto           INTEGER NOT NULL,
PRIMARY KEY(IDQ),
FOREIGN KEY(IDQ) REFERENCES Funcionarios(ID_Funcionarios)
);

```

```

CREATE TABLE Temporario (
IDT                INTEGER NOT NULL,
Seguro             VARCHAR(MAX) NOT NULL,
Nome               VARCHAR(MAX) NOT NULL,
Contacto           INTEGER NOT NULL,
);

```

```

CREATE TABLE Conter (
ID_Jardins         INTEGER NOT NULL,
ID_Cidades         INTEGER NOT NULL,
PRIMARY KEY(ID_Jardins,ID_Cidades),
FOREIGN KEY(ID_Jardins) REFERENCES Jardins (ID_Jardins),
FOREIGN KEY(ID_Cidades) REFERENCES Cidades (ID_Cidades))

```

```

CREATE TABLE Contratar_Func(
ID_Funcionarios    INTEGER NOT NULL,
ID_Cidades         INTEGER NOT NULL,
Data_Contratar     DATE NOT NULL,
Preco              float(20) NOT NULL,
PRIMARY KEY(ID_Funcionarios,ID_Cidades,Data_Contratar),
FOREIGN KEY(ID_Funcionarios) REFERENCES Funcionarios (ID_Funcionarios),
FOREIGN KEY(ID_Cidades) REFERENCES Cidades (ID_Cidades))

```

```

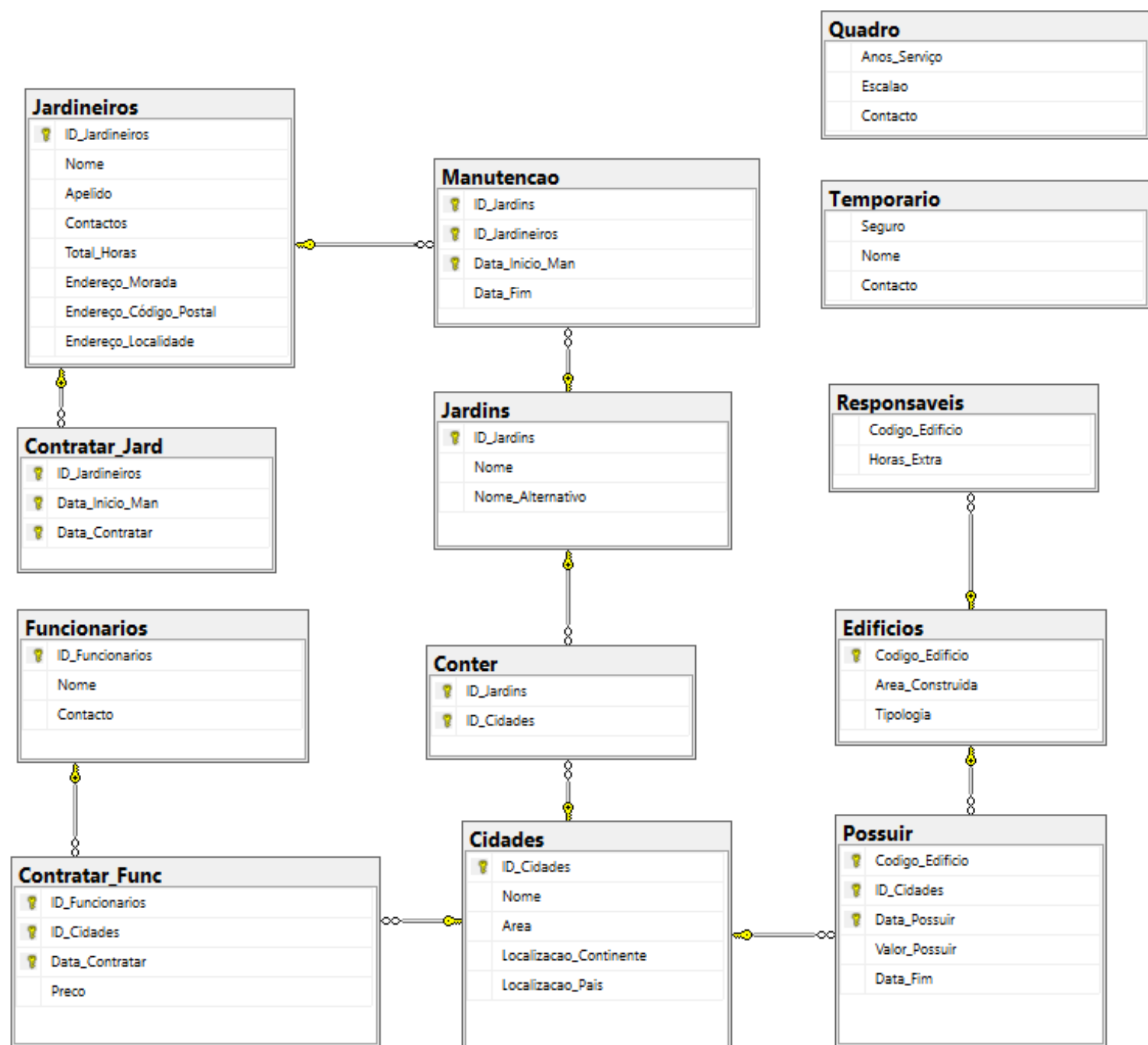
CREATE TABLE Possuir(
Codigo_Edificio    INTEGER NOT NULL,
ID_Cidades         INTEGER NOT NULL,
Data_Possuir       DATE NOT NULL,
Valor_Possuir      float NOT NULL,
Data_Fim           DATE NOT NULL,
PRIMARY KEY(Codigo_Edificio,ID_Cidades, Data_Possuir),
FOREIGN KEY(Codigo_Edificio ) REFERENCES Edificios (Codigo_Edificio ),
FOREIGN KEY(ID_Cidades) REFERENCES Cidades (ID_Cidades))

```

```
CREATE TABLE Manutencao(  
ID_Jardins          INTEGER NOT NULL,  
ID_Jardineiros      INTEGER NOT NULL,  
Data_Inicio_Man     DATE NOT NULL,  
Data_Fim            DATE NOT NULL,  
PRIMARY KEY(ID_Jardins,ID_Jardineiros,Data_Inicio_Man),  
FOREIGN KEY(ID_Jardins) REFERENCES Jardins (ID_Jardins),  
FOREIGN KEY(ID_Jardineiros) REFERENCES Jardineiros (ID_Jardineiros))
```

```
CREATE TABLE Responsaveis(  
Codigo_Edificio      int IDENTITY NOT NULL,  
Horas_Extra          INTEGER NOT NULL,  
FOREIGN KEY(Codigo_Edificio) REFERENCES Edificios (Codigo_Edificio))
```

```
CREATE TABLE Contratar_Jard(  
ID_Jardineiros      int IDENTITY NOT NULL,  
Data_Inicio_Man     DATE NOT NULL,  
Data_Contratar      DATE NOT NULL,  
PRIMARY KEY(ID_Jardineiros, Data_Inicio_Man, Data_Contratar),  
FOREIGN KEY(ID_Jardineiros) REFERENCES Jardineiros (ID_Jardineiros))
```

Desenvolvimento – etapa 2

1. Registos em SQL

“Insira, pelo menos, 3 registos em cada tabela”

```
--CREATE DATABASE BD2
```

```
USE BD2
```

```
GO
```

```
CREATE TABLE Jardins (
```

```
    ID_Jardins          int IDENTITY NOT NULL,
```

```
    Nome                VARCHAR(MAX) NOT NULL,
```

```
    Nome_Alternativo    VARCHAR(MAX) NOT NULL,
```

```
    PRIMARY KEY (ID_Jardins));
```

```
CREATE TABLE Cidades (
```

```
    ID_Cidades          int IDENTITY NOT NULL,
```

```
    Nome                VARCHAR(MAX) NOT NULL,
```

```
    Area                FLOAT(20) NOT NULL,
```

```
    Localizacao_Continente VARCHAR(MAX) NOT NULL,
```

```
    Localizacao_Pais    VARCHAR(MAX) NOT NULL,
```

```
    PRIMARY KEY (ID_Cidades));
```

```
CREATE TABLE Edificios (
```

```
    Codigo_Edificio     int IDENTITY NOT NULL,
```

```
    Area_Construida     VARCHAR(MAX) NOT NULL,
```

```
    Tipologia           VARCHAR(MAX) NOT NULL,
```

PRIMARY KEY (Codigo_Edificio));

```
CREATE TABLE Funcionarios (  
  ID_Funcionarios      int IDENTITY NOT NULL,  
  Nome                 VARCHAR(MAX) NOT NULL,  
  Contacto             integer NOT NULL,  
  PRIMARY KEY (ID_Funcionarios));
```

```
CREATE TABLE Jardineiros (  
  ID_Jardineiros       int IDENTITY NOT NULL,  
  Nome                 VARCHAR(MAX) NOT NULL,  
  Apelido              VARCHAR(MAX) NOT NULL,  
  Contactos            INTEGER NOT NULL,  
  Total_Horas          INTEGER NOT NULL,  
  Endereço_Morada      VARCHAR(MAX) NOT NULL,  
  Endereço_Código_Postal VARCHAR(MAX) NOT NULL,  
  Endereço_Localidade  VARCHAR(MAX) NOT NULL,  
  PRIMARY KEY (ID_Jardineiros));
```

```
CREATE TABLE Quadro (  
  IDQ                                INTEGER NOT NULL,  
  Anos_Serviço                      VARCHAR(MAX) NOT NULL,  
  Escalao                           VARCHAR(MAX) NOT NULL,  
  Contacto                          INTEGER NOT NULL,  
  PRIMARY KEY(IDQ),  
  FOREIGN KEY(IDQ) REFERENCES Funcionarios(ID_Funcionarios)  
);
```

```
CREATE TABLE Temporario (  
  IDT                INTEGER NOT NULL,  
  Seguro             VARCHAR(MAX) NOT NULL,  
  Nome               VARCHAR(MAX) NOT NULL,
```

```
    Contacto          INTEGER NOT NULL,  
);
```

```
CREATE TABLE Conter (  
    ID_Jardins          INTEGER NOT NULL,  
    ID_Cidades          INTEGER NOT NULL,  
    PRIMARY KEY(ID_Jardins,ID_Cidades),  
    FOREIGN KEY(ID_Jardins) REFERENCES Jardins (ID_Jardins),  
    FOREIGN KEY(ID_Cidades) REFERENCES Cidades (ID_Cidades))
```

```
CREATE TABLE Contratar_Func(  
    ID_Funcionarios      INTEGER NOT NULL,  
    ID_Cidades           INTEGER NOT NULL,  
    Data_Contratar       DATE NOT NULL,  
    Preco                float(20) NOT NULL,  
    PRIMARY KEY(ID_Funcionarios,ID_Cidades,Data_Contratar),  
    FOREIGN KEY(ID_Funcionarios) REFERENCES Funcionarios (ID_Funcionarios),  
    FOREIGN KEY(ID_Cidades) REFERENCES Cidades (ID_Cidades))
```

```
CREATE TABLE Possuir(  
    Codigo_Edificio      INTEGER NOT NULL,  
    ID_Cidades           INTEGER NOT NULL,  
    Data_Possuir         DATE NOT NULL,  
    Valor_Possuir        float NOT NULL,  
    Data_Fim             DATE NOT NULL,  
    PRIMARY KEY(Codigo_Edificio,ID_Cidades, Data_Possuir),  
    FOREIGN KEY(Codigo_Edificio ) REFERENCES Edificios (Codigo_Edificio ),  
    FOREIGN KEY(ID_Cidades) REFERENCES Cidades (ID_Cidades))
```

```
CREATE TABLE Manutencao(  
    ID_Jardins           INTEGER NOT NULL,  
    ID_Jardineiros       INTEGER NOT NULL,
```

```

Data_Inicio_Man    DATE NOT NULL,
Data_Fim           DATE NOT NULL,
PRIMARY KEY(ID_Jardins,ID_Jardineiros,Data_Inicio_Man),
FOREIGN KEY(ID_Jardins) REFERENCES Jardins (ID_Jardins),
FOREIGN KEY(ID_Jardineiros) REFERENCES Jardineiros (ID_Jardineiros))
CREATE TABLE Responsaveis(
Codigo_Edificio    int IDENTITY NOT NULL,
Horas_Extra        INTEGER NOT NULL,
FOREIGN KEY(Codigo_Edificio) REFERENCES Edificios (Codigo_Edificio))

```

```

CREATE TABLE Contratar_Jard(
IDQ                                     INTEGER NOT NULL,
ID_Jardineiros    int IDENTITY NOT NULL,
Data_              DATE NOT NULL,
Salario            INTEGER NOT NULL,
PRIMARY KEY(ID_Jardineiros, IDQ),
FOREIGN KEY(ID_Jardineiros) REFERENCES Jardineiros (ID_Jardineiros),
FOREIGN KEY(IDQ) REFERENCES Quadro(IDQ)
)

```

```

INSERT INTO Jardins(ID_Jardins, Nome, Nome_Alternativo)
VALUES (12345, 'Jardim do Morro', 'Jardim Morro')
INSERT INTO Jardins(ID_Jardins, Nome, Nome_Alternativo)
VALUES (12346, 'Jardim da Carreira', 'Jardim Bar')
INSERT INTO Jardins(ID_Jardins, Nome, Nome_Alternativo)
VALUES (54321, 'Jardins do Palácio de Cristal', 'Jardins do Palacio')

```

```

INSERT INTO Cidades(ID_Cidades, Nome, Area, Localizacao_Continente, Localizacao_Pais)
VALUES (62371, 'Paris', 105.4, 'Europa', 'França')
INSERT INTO Cidades(ID_Cidades, Nome, Area, Localizacao_Continente, Localizacao_Pais)
VALUES (23513, 'Toquio', 2194, 'Asia', 'Japão')
INSERT INTO Cidades(ID_Cidades, Nome, Area, Localizacao_Continente, Localizacao_Pais)
VALUES (12371, 'Lisboa', 100, 'Europa', 'Portugal')

```

```

INSERT INTO Edificios(Codigo_Edificio, Area_Construida, Tipologia)

```

```
VALUES (1252, 20, 'Habitacional')
INSERT INTO Edificios(Codigo_Edificio, Area_Construida, Tipologia)
VALUES (1324, 56, 'Hospitalar')
INSERT INTO Edificios(Codigo_Edificio, Area_Construida, Tipologia)
VALUES (6910, 104, 'Comercial')
INSERT INTO Funcionarios(ID_Funcionarios, Nome, Contacto)
VALUES (103, 'Guilherme Andrade', '937801245')
INSERT INTO Funcionarios(ID_Funcionarios, Nome, Contacto)
VALUES (703, 'Cristiano Ronaldo', '961718291')
INSERT INTO Funcionarios(ID_Funcionarios, Nome, Contacto)
VALUES (076, 'Andre Almeida', '911117891')
```

```
INSERT INTO Jardineiros(ID_Jardineiros, Nome, Apelido, Contactos, Total_Horas, Endereço_Morada,
Endereço_Código_Postal, Endereço_Localidade)
VALUES (1231, 'Jose', 'Fonte', 923137129, 27, 'Rua Ramiro Ferrao', '2809-001', 'Almada')
INSERT INTO Jardineiros(ID_Jardineiros, Nome, Apelido, Contactos, Total_Horas, Endereço_Morada,
Endereço_Código_Postal, Endereço_Localidade)
VALUES (9201, 'João', 'Alberto', 914768910, 54, 'Rua Infante Dom Henrique', '6301-861', 'Guarda')
INSERT INTO Jardineiros(ID_Jardineiros, Nome, Apelido, Contactos, Total_Horas, Endereço_Morada,
Endereço_Código_Postal, Endereço_Localidade)
VALUES (7261, 'Alberto', 'Caeiro', 962078918, 47, 'Rua da Mesura', '3044-519', 'Cernache')
```

```
INSERT INTO Quadro(Anos_Serviço, Escalao, Contacto)
VALUES (27, 'A', 910000002)
INSERT INTO Quadro(Anos_Serviço, Escalao, Contacto)
VALUES (3, 'B', 934777777)
INSERT INTO Quadro(Anos_Serviço, Escalao, Contacto)
VALUES (45, 'A', 911111111)
```

```
INSERT INTO Temporario(Seguro, Nome, Contacto)
VALUES ('Fidelidade', 'Joao Barbosa', 922222222)
INSERT INTO Temporario(Seguro, Nome, Contacto)
```

```
VALUES ('Mapfre', 'Bernardo Meneses', 912323434)
INSERT INTO Temporario(Seguro, Nome, Contacto)
VALUES ('Tranquilidade', 'Jose Fonte', 912121212)
```

```
INSERT INTO Conter(ID_Jardins, ID_Cidades)
VALUES (12252, 73212)
INSERT INTO Conter(ID_Jardins, ID_Cidades)
VALUES (71252, 91292)
INSERT INTO Conter(ID_Jardins, ID_Cidades)
VALUES (98516, 15021)
```

```
INSERT INTO Contratar_Func(ID_Funcionarios, ID_Cidades, Data_Contratar, Preco)
VALUES (809, 89812, '17-02-2009', 150000)
INSERT INTO Contratar_Func(ID_Funcionarios, ID_Cidades, Data_Contratar, Preco)
VALUES (762, 17283, '09-11-2021', 20000)
INSERT INTO Contratar_Func(ID_Funcionarios, ID_Cidades, Data_Contratar, Preco)
VALUES (105, 90901, '17-09-2016', 300000)
```

```
INSERT INTO Possuir(Codigo_Edificio, ID_Cidades, Data_Possuir, Valor_Possuir, Data_Fim)
VALUES (5455, 77777, '17-09-2022', 1500, '20-06-2025')
INSERT INTO Possuir(Codigo_Edificio, ID_Cidades, Data_Possuir, Valor_Possuir, Data_Fim)
VALUES (1234, 66666, '11-03-2011', 120000, '11-03-2021')
INSERT INTO Possuir(Codigo_Edificio, ID_Cidades, Data_Possuir, Valor_Possuir, Data_Fim)
VALUES (4321, 55555, '15-05-2019', 135000, '15-08-2019')
```

```
INSERT INTO Manutencao(ID_Jardins, ID_Jardineiros, Data_Inicio_Man, Data_Fim)
VALUES (52734, 3214, '7-11-2005', '11-7-2015')
INSERT INTO Manutencao(ID_Jardins, ID_Jardineiros, Data_Inicio_Man, Data_Fim)
VALUES (98252, 1071, '11-8-2014', '5-7-2012')
INSERT INTO Manutencao(ID_Jardins, ID_Jardineiros, Data_Inicio_Man, Data_Fim)
VALUES (11114, 1111, '1-11-2002', '11-1-2021')
```

```
INSERT INTO Responsaveis(Codigo_Edificio, Horas_Extra)
VALUES (7777, 123)
```

```
INSERT INTO Responsaveis(Codigo_Edificio, Horas_Extra)
VALUES (1224, 211)
```

```
INSERT INTO Responsaveis(Codigo_Edificio, Horas_Extra)
VALUES (9123, 713)
```

```
INSERT INTO Contratar_Jard(ID_Jardineiros, Data_, Salario)
VALUES (1234, '31-01-2007', 20000)
```

```
INSERT INTO Contratar_Jard(ID_Jardineiros, Data_, Salario)
VALUES (7123, '31-03-2017', 18000)
```

```
INSERT INTO Contratar_Jard(ID_Jardineiros, Data_, Salario)
VALUES (9013, '13-11-2014', 25000)
```


2. Questões de SQL

“Escreva a linguagem SQL que responda a cada uma das seguintes perguntas”

- 2.1. Qual o 1.º jardineiro contratado? [Jardineiro (Nome), Funcionário (Nome), Data, Salário]

```
Select Top 1 data_, Jardineiros.Nome, Funcionarios.Nome, Contratar_Jard.Salario
from Jardineiros, Funcionarios, Contratar_Jard
where Jardineiros.Id_jardineiros = contratar_jard.Id_jardineiros
Order By data_ ASC
```

- 2.2. Quantos jardineiros foram contratados por cada funcionário? [Funcionário (Nome), N_Jardineiros]

```
SELECT F.Nome, COUNT(*) AS N_Jardineiros
FROM Funcionarios F, Contratar_Jard CJ, Quadro Q, Jardineiros J
WHERE F.Nome = Q.IDQ
AND Q.IDQ = CJ.IDQ
AND J.ID_Jardineiros = CJ.ID_Jardineiros
GROUP BY J.Nome
```

- 2.3. Qual a cidade com mais jardins? [Cidade (Nome), (Quantidade)]

```
SELECT TOP 1 C.Nome, COUNT(DISTINCT J.ID_Jardins) AS Quantidade
FROM Cidades C, Jardins J, Conter CO
WHERE C.ID_Cidades = Co.ID_Cidades
AND Co.ID_Jardins = J.ID_Jardins
ORDER BY Quantidade ASC
```

- 2.4. Qual a cidade que mais funcionários contratou nos últimos 30 dias? [Cidade (Nome), Número de Funcionários]

```
SELECT TOP 1 C.Nome, COUNT(DISTINCT F.ID_Funcionarios) AS Num_Func
FROM Cidades C, Contratar_Func CF, Funcionarios F
WHERE DATEDIFF(day, CF.Data_Contratar, getdate())<30
AND CF.ID_Funcionarios = F.ID_Funcionarios
AND CF.ID_Cidades = C.ID_Cidades
ORDER BY Num_Func ASC;
```

- 2.5. Quais os Jardins que não têm nenhum jardineiro na manutenção? [Jardins (Nome)]

```
SELECT J.Nome
FROM Jardins J, Manutencao M, Jardineiros Jar
WHERE J.ID_Jardins = M.ID_Jardins
AND M.ID_Jardineiros NOT IN (Jar.ID_Jardineiros)
```

- 2.6. Quais os Funcionários Temporários a trabalhar em cada Edifício? Ordene-os alfabeticamente. [Funcionários (Nome), Edifício (Código)]

```
select T.Nome, E.Codigo_Edificio, F.ID_Funcionarios
from Temporario T, Edificios E, Funcionarios F
where (T.Nome=E.Codigo_Edificio)
count (F.ID_Funcionarios) AS Num_Funcionarios
group by T.Nome
```

- 2.7. Qual o total gasto em contratações de Funcionários por cada cidade em 2022? [Cidade (Nome), TotalGasto]

```
select C.Nome
from Cidades C, Contratar_Func CF
count (CF.Preco) as PrecoTot
```

where C.ID_Cidades = CF.ID_Cidades

order by Cidade.Nome

3. Procedimento em SQL

“Assumindo que para os jardineiros as horas extra são pagas a 10€/h, crie um procedimento que para um dado mês atualize o TotalHoras na tabela Jardineiro e apresente uma tabela com os ID, nome e apelidos dos jardineiros e o total que cada um recebe nesse mês. O procedimento deve devolver o total da despesa com os jardineiros.”

```
CREATE PROCEDURE despesaJardineiros @Horas_Extra INTEGER
```

```
DECLARE @despesa MONEY;
```

```
DECLARE @Horas_Extra INTEGER;
```

```
DECLARE @ID_Jardineiro INTEGER;
```

```
SET @despesa = (SELECT Contratar_Jard.Salario  
                FROM Contratar_Jard  
                WHERE ID_Jardineiros = @ID_Jardineiros);
```

```
EXECUTE @despesa = getDespesa @Contratar_Jard.Salario;
```

```
UPDATE @Jardineiros.TotalHoras WHERE Jardineiros;
```

```
PRINT @ID_Jardineiros, @Jardineiros.Nome, @Jardineiros.Apelido @Contratar_Jard.Salario;
```

```
RETURN @despesa
```

4. Trigger em SQL

“Crie um trigger que apenas deixe inserir um novo registro no relacionamento Contratar -> Funcionário se o ID do indivíduo a contratar ainda não constar dos Funcionários.”

```
CREATE TRIGGER InserirRegisto
ON Contratar_Func
{
INSTEAD OF INSERT
AS
IF (select Funcionarios.ID_Funcionarios from Funcionarios where Funcionarios.ID_Funcionarios IS NOT
NULL)
end if;
}
```

Conclusão

Com a resolução deste projeto conseguimos adquirir conhecimento sobre o funcionamento do sistema de base de dados, e também sobre os processos que envolvem a gestão do mesmo.

A realização deste trabalho foi de uma certa maneira uma forma de desenvolver o conhecimento da unidade curricular, facilitando assim o nosso estudo e oferecendo-nos um “skill set” que será extremamente útil para o mercado de trabalho.

Webgrafia

- Slides da unidade curricular;
- <https://stackoverflow.com/>;
- <https://docs.microsoft.com/en-us/documentation/>;