

Índice

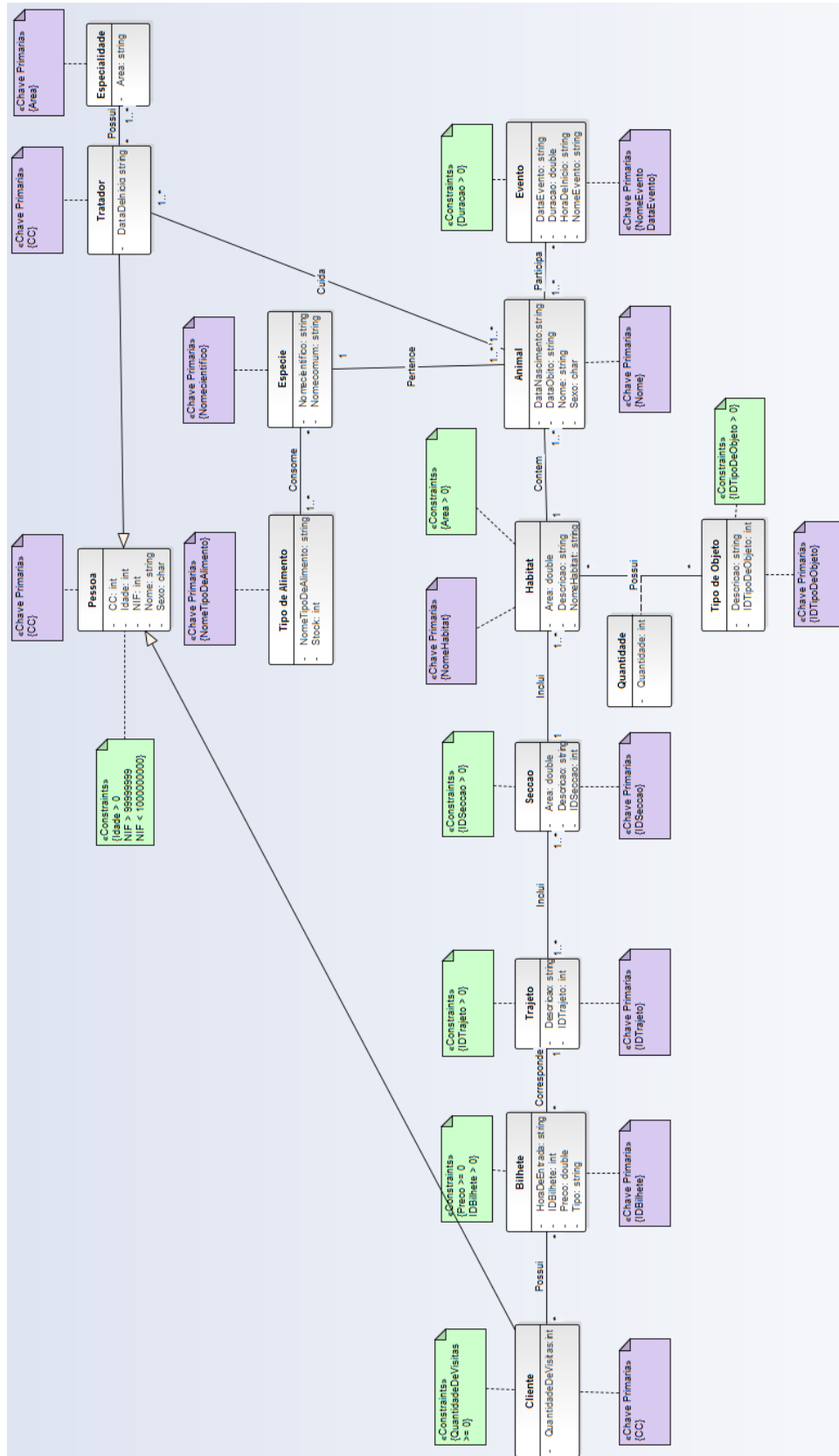
Descrição de contexto	2
Diagrama de classes UML	3
Esquema Relacional	4
Instruções LDD-SQL.....	5
Instruções LDM-SQL	11

DESCRIÇÃO DE CONTEXTO

Nesta segunda entrega continuámos a expansão do nosso projeto relativo ao jardim zoológico, desta vez com o esquema relacional, instruções LDD-SQL para a criação da base de dados e ainda algumas instruções LMD-SQL que permitem preencher a base de dados com informação fictícia.

Todo este trabalho foi relativo ao diagrama de classes UML que sofreu uma atualização, sendo que agora está mais completo com mais alguns atributos e restrições.

DIAGRAMA DE CLASSES UML



ESQUEMA RELACIONAL

Pessoa(CC , Idade, NIF, Nome, Sexo)

Cliente(CC, Quantidade de Visitas)

Bilhete(idBilhete, Hora de Entrada, Preço, Tipo, idTrajeto->Trajeto)

ClienteCompraBilhete(CC-> Cliente, idBilhete -> Bilhete)

Trajeto(idTrajeto, Descricao)

Seccao(idSeccao, Area, Descricao)

TrajetoIncluiSeccao(idTrajeto -> Trajeto, idSeccao -> Seccao)

Habitat(NomeHabitat, Area, Descricao, idSeccao -> Seccao)

TipodeObjeto(idTipodeObjeto, Descricao)

HabitatContemTipodeObjeto(NomeHabitat -> Habitat, idTipodeObjeto -> TipodeObjeto, Quantidade)

Animal(NomeAnimal, DataNascimento, DataObito, Sexo, NomeHabitat -> Habitat, NomeCientifico -> Especie)

Evento(NomeEvento, Data, Duracao, Hora de Inicio)

AnimalParticipaEvento(NomeAnimal -> Animal, [NomeEvento, DataEvento] -> Evento)

Especie(NomeCientifico, NomeComum)

TipodeAlimento(NomeTipodeAlimento, Stock)

EspecieComeTipodeAlimento(NomeCientifico -> Especie, NomeTipodeAlimento -> TipodeAlimento)

Tratador(CC, DataInicio)

Especialidade(Area)

TratadorTemEspecialidade(CC -> Tratador, Area -> Especialidade)

INSTRUÇÕES LDD-SQL

```
CREATE TABLE TratadorTemEspecialidade (  
    Tratador      TEXT,  
    Especialidade TEXT,  
    PRIMARY KEY(Tratador,Especialidade),  
    FOREIGN KEY(Tratador) REFERENCES Tratador(CC),  
    FOREIGN KEY(Especialidade) REFERENCES Especialidade(Area) );
```

```
CREATE TABLE Tratador (  
    Pessoa TEXT,  
    DataDeInicio NUMERIC NOT NULL,  
    PRIMARY KEY(Pessoa),  
    FOREIGN KEY(Pessoa) REFERENCES Pessoa(CC) );
```

```
CREATE TABLE TrajetoIncluiSeccao (  
    Trajeto INTEGER,  
    Seccao INTEGER,  
    PRIMARY KEY(Trajeto,Seccao),  
    FOREIGN KEY(Trajeto) REFERENCES Trajeto(IDTrajeto),  
    FOREIGN KEY(Seccao) REFERENCES Seccao(IDSeccao) );
```

```
CREATE TABLE Trajeto (  
    IDTrajeto      INTEGER,  
    Descricao      TEXT,  
    PRIMARY KEY(IDTrajeto),  
    CHECK(IDTrajeto > 0) );
```

```
CREATE TABLE TipoDeObjecto (
    IDTipoDeObjecto    INTEGER,
    Descricao          TEXT NOT NULL,
    PRIMARY KEY(IDTipoDeObjecto),
    CHECK(IDTipoDeObjecto > 0) );
```

```
CREATE TABLE TipoDeAlimento (
    NomeTipoDeAlimento TEXT,
    Stock    INTEGER NOT NULL,
    PRIMARY KEY(NomeTipoDeAlimento) );
```

```
CREATE TABLE Seccao (
    IDSeccao    INTEGER,
    Area        REAL NOT NULL,
    Descricao    TEXT NOT NULL,
    PRIMARY KEY(IDSeccao),
    CHECK(IDSeccao > 0) );
```

```
CREATE TABLE Pessoa (
    CC    TEXT,
    Nome  TEXT NOT NULL,
    Idade INTEGER NOT NULL,
    Sexo  TEXT NOT NULL,
    NIF    INTEGER NOT NULL UNIQUE,
    PRIMARY KEY(CC)
    CHECK(Idade > 0 AND NIF > 99999999 AND NIF < 1000000000) );
```

```
CREATE TABLE HabitatContemTipoDeObjecto (
Habitat      TEXT,
TipoDeObjecto INTEGER,
QuantidadeObjecto  INTEGER NOT NULL,
PRIMARY KEY(Habitat,TipoDeObjecto),
FOREIGN KEY(Habitat) REFERENCES Habitat(NomeHabitat),
FOREIGN KEY(TipoDeObjecto) REFERENCES TipoDeObjecto(IDTipoDeObjecto) );
```

```
CREATE TABLE Habitat (
NomeHabitat  TEXT,
Area  REAL NOT NULL,
Descricao  TEXT NOT NULL,
Seccao  INTEGER,
PRIMARY KEY(NomeHabitat),
FOREIGN KEY(Seccao) REFERENCES Seccao(IDSeccao),
CHECK(Area > 0) );
```

```
CREATE TABLE Evento (
NomeEvento  TEXT,
Data  NUMERIC,
Duracao  INTEGER NOT NULL,
HoraDeInicio  INTEGER NOT NULL,
PRIMARY KEY(NomeEvento,Data),
CHECK(Duracao > 0) );
```



```
CREATE TABLE EspecieComeTipoDeAlimento (
    Especie TEXT,
    TipoDeAlimento TEXT,
    PRIMARY KEY(Especie,TipoDeAlimento),
    FOREIGN KEY(Especie) REFERENCES Especie(NomeCientifico),
    FOREIGN KEY(TipoDeAlimento) REFERENCES
TipoDeAlimento(NomeTipoDeAlimento) );
```

```
CREATE TABLE Especie (
    NomeCientifico TEXT,
    NomeComum TEXT,
    PRIMARY KEY(NomeCientifico) );
```

```
CREATE TABLE Especialidade (
    Area TEXT,
    PRIMARY KEY(Area) );
```

```
CREATE TABLE Cliente (
    Pessoa TEXT,
    Quantidade_de_visitas INTEGER NOT NULL,
    PRIMARY KEY(Pessoa),
    FOREIGN KEY(Pessoa) REFERENCES Pessoa(CC)
    CHECK(Quantidade_de_visitas >= 0) );
```

```
CREATE TABLE Bilhete (
  IDBilhete      INTEGER,
  Hora_De_Entrada  NUMERIC,
  Preco  REAL NOT NULL,
  Tipo   TEXT NOT NULL,
  Trajeto INTEGER,
  PRIMARY KEY(IDBilhete),
  FOREIGN KEY(Trajeto) REFERENCES Trajeto(IDTrajeto),
  CHECK(Preco >= 0 AND IDBilhete > 0) );
```

```
CREATE TABLE AnimalParticipaEvento (
  Animal TEXT,
  NomeEvento  TEXT,
  DataEvento  NUMERIC,
  PRIMARY KEY(Animal, NomeEvento, DataEvento),
  FOREIGN KEY(Animal) REFERENCES Animal(NomeAnimal),
  FOREIGN KEY(NomeEvento, DataEvento) REFERENCES Evento(NomeEvento,
DataEvento) );
```

```
CREATE TABLE ClienteCompraBilhete (
  Cliente TEXT,
  Bilhete TEXT,
  PRIMARY KEY(Cliente, Bilhete),
  FOREIGN KEY(Cliente) REFERENCES Cliente(CC),
  FOREIGN KEY(Bilhete) REFERENCES Bilhete(IDBilhete) );
```

```
CREATE TABLE Animal (
NomeAnimal TEXT,
DataNascimento NUMERIC NOT NULL,
DataObito NUMERIC,
Sexo TEXT NOT NULL,
EspecieTEXT,
Habitat TEXT,
PRIMARY KEY(NomeAnimal),
FOREIGN KEY(Especie) REFERENCES Especie(NomeCientifico),
FOREIGN KEY(Habitat) REFERENCES Habitat(NomeHabitat) );
```

```
CREATE TABLE TratadorCuidaAnimal (
Tratador TEXT,
Animal TEXT,
PRIMARY KEY(Tratador, Animal)
FOREIGN KEY(Tratador) REFERENCES Tratador(CC),
FOREIGN KEY(Animal) REFERENCES Animal(NomeAnimal) );
```

```
CREATE TRIGGER IncrementBilhete
AFTER INSERT ON ClienteCompraBilhete
UPDATE Cliente SET Quantidade_de_visitas = Quantidade_de_visitas + 1,
[WHERE CC = Cliente];
```

INSTRUÇÕES LDM-SQL

```
INSERT INTO TipoDeObjecto VALUES (1, "Pedra <5cm");
INSERT INTO TipoDeAlimento VALUES ("Carne", 10);
INSERT INTO Seccao VALUES (1, 10, "Primeira seccao");
INSERT INTO Habitat VALUES ("Savana", 5, "Habitat Savana");
INSERT INTO HabitatContemTipoDeObjecto VALUES ("Savana", 1);
INSERT INTO Trajeto VALUES (1, "");
INSERT INTO TrajetoIncluiSeccao(1, 1);
INSERT INTO Especie VALUES ("Canis lupus", "Lobo");
INSERT INTO EspecieComeTipoDeAlimento ("Canis lupus", "Carne");
INSERT INTO Animal VALUES ("Goncalo", 1990, NULL, "M", "Canis lupus", "Savana");
INSERT INTO Pessoa VALUES ("1234", "Eduardo", 23, "M", 123456789);
INSERT INTO Cliente VALUES ("1234", 0);
INSERT INTO Bilhete VALUES (1, NULL, 1.5, "Especial", 1);
INSERT INTO ClienteCompraBilhete("1234", 1);
INSERT INTO Pessoa VALUES ("4321", "Mario", 20, "M", 987654321);
INSERT INTO Tratador("4321", "12-10-2015");
INSERT INTO TratadorTemEspecialidade("Alimentador");
INSERT INTO TratadorCuidaAnimal("4321", "Goncalo");
INSERT INTO Evento VALUES ("Primeiro evento", "11-02-2015")
INSERT INTO AnimalParticipaEvento VALUES ("Goncalo", "Primeiro evento", "11-02-2015");
```