

# LEI – Base de Dados Gestão de uma Loja de Vestuário

Grupo 205:

Tiago Pacheco – 8200421

Rui Campos– 8190352

Lúcia Pereira – 8200278

## Índice

|  |           |
|--|-----------|
| <b>1. INTRODUÇÃO.....</b>  | <b>3</b>  |
| 1.1 CONTEXTUALIZAÇÃO.....  | 3         |
| 1.2 APRESENTAÇÃO DO CASO DE ESTUDO .....                                     | 3         |
| 1.3 MOTIVAÇÃO E OBJETIVOS.....   | 4         |
| 1.4 ESTRUTURA DO RELATÓRIO.....  | 4         |
| <b>2. CONSTRUÇÃO DO MODELO CONCEPTUAL .....</b>                              | <b>4</b>  |
| 2.1. IDENTIFICAÇÃO DAS ENTIDADES.....  | 5         |
| 2.2. IDENTIFICAÇÃO DOS RELACIONAMENTOS.....                                  | 6         |
| 2.3. IDENTIFICAÇÃO E ATRIBUIÇÃO DE ATRIBUTOS ÀS ENTIDADES .....              | 6         |
| 2.4. DETERMINAR O RESPECTIVO DOMÍNIO PARA CADA ATRIBUTO .....                | 8         |
| 2.5. IDENTIFICAÇÃO DAS CHAVES PRIMÁRIAS E DAS CHAVES CANDIDATAS.....         | 9         |
| 2.6. DESENHO CONCEPTUAL.....   | 9         |
| <b>3. DESENHO LÓGICO .....</b>   | <b>10</b> |
| 3.1. CONSTRUÇÃO E VALIDAÇÃO DO MODELO LÓGICO DE DADOS .....                  | 10        |
| 3.2. OBTER AS RELAÇÕES PARA O MODELO LÓGICO DE DADOS .....                   | 10        |
| 3.3. DIAGRAMA ER COM A IDENTIFICAÇÃO DAS CHAVES ESTRANGEIRAS .....           | 11        |
| 3.4. NORMALIZAÇÃO.....   | 12        |
| 3.5. CONCLUSÃO DA NORMALIZAÇÃO.....  | 13        |
| 3.6. DEFINIR RESTRIÇÕES DE INTEGRIDADE – REGRAS DE NEGÓCIO .....             | 14        |
| <b>4. IMPLEMENTAR A BASE DE DADOS EM SQL SERVER 2022 (OU POSTERIOR).....</b> | <b>15</b> |
| 4.1. CRIAÇÃO DE TABELAS .....  | 15        |
| 4.2. RESTRIÇÕES.....   | 16        |
| 4.3. TRIGGERS.....   | 17        |
| 4.4. QUERIES.....  | 17        |
| 4.5. VIEWS.....  | 19        |
| <b>5. CONCLUSÕES E TRABALHO FUTURO .....</b>                                 | <b>21</b> |
| <b>6. BIBLIOGRAFIA .....</b>   | <b>21</b> |

## 1. Introdução

### 1.1 Contextualização

No âmbito da disciplina de Base de Dados inserida no curso de Licenciatura de Informática, do segundo ano do segundo semestre, foi-nos proposta a realização de um trabalho prático que procura aprofundar os conhecimentos adquiridos ao longo das aulas desta Unidade Curricular.

A proposta consiste na criação de uma base de dados capaz satisfazer as necessidades ao nível da gestão e do tratamento dos processos de uma loja de vestuário.

Deste modo, no decorrer deste trabalho prático seguimos a metodologia proposta pelo docente da UC, composta essencialmente por três fase:

1. A construção do modelo conceptual;
2. A construção do modelo lógico;
3. A construção do modelo físico.

### 1.2 Apresentação do Caso de Estudo

#### **“Gestão de uma loja de vestuário”**

O surgimento de problemas ao nível da qualidade dos serviços prestados numa loja de vestuário, deu origem à tomada de decisão por parte da gerência de contratar uma empresa que prestasse consultoria e que desenhasse uma base de dados que permita uma melhor e mais eficiente gestão da loja de vestuário.

A loja está fisicamente dividida em 3 secções (Homem, Mulher, Criança), existindo em cada secção 10 escaparates para expor as peças. Na loja tem incluído um pequeno armazém onde se encontram guardadas peças para a reposição de stock. O horário de funcionamento da loja é das 10h até 22h, com os funcionários a fazer turnos de 6h por dia, tendo como principais tarefas atender os clientes, registar as vendas na aplicação e no caso de encontrar uma peça defeituosa deve de a catalogar e armazenar no armazém como uma peça defeituosa.

Os processos acima descritos passam pela criação de uma base de dados para dar suporte à informação e deverá também ser capaz de gerar as seguintes respostas a estas questões:

1. Qual o total faturado no dia anterior, semana anterior, mês anterior?
2. Lista das peças vendidas no dia anterior?
3. Lista dos funcionários por ordem decrescente do valor de vendas do mês anterior?
4. Qual a peça mais vendida?
5. Qual o valor de peças defeituosas identificadas numa determinada semana?

### 1.3 Motivação e Objetivos

Como motivação compreendemos que este trabalho visa antes de tudo obter conhecimentos de forma prática com base nos princípios fundamentais da elaboração de uma base de dados e de que maneira este se torna relevante para o mundo real de trabalho, em específico para este problema apresentado. Os objetivos resumem-se a nível macro, na aquisição de conhecimentos sobre esta UC e como aplicar num contexto específico de trabalho. A nível micro os objetivos passam por ultrapassar o desafio de adquirir com sucesso competências ao nível do desenho de uma base de dados, saber interpretar os requisitos apresentados na proposta pelo docente da UC e desenvolver o espírito crítico na procura da melhor solução para o problema em questão.

### 1.4 Estrutura do Relatório

De acordo com a metodologia apresentada em contexto de sala de aula, a estrutura deste trabalho compreende os seguintes tópicos:

1. Estudo do domínio sobre lojas de venda de vestuário;
2. Identificação das entidades envolvidas no negócio, dos seus atributos e relacionamentos;
3. Construção do diagrama Entidade-Relacionamento (ER) que representa a base de dados;
4. Aplicação do processo de normalização, com a identificação das dependências funcionais;
5. Implementação em base de dados em SQL Server 2017(ou posterior);
6. Elaboração do relatório detalhando todos os passos e implementações feitas sobre este tema.

## 2. Construção do Modelo Conceptual

Após o estudo do domínio, o modelo conceptual deve ser o passo seguinte a desenvolver, de acordo com o estudado em contexto de aula.

É de ressaltar que todo o percurso aqui desenvolvido vai de encontro à bibliografia fornecida pelo docente de Base de Dados que consta no livro de *C. Thomas e B. Carolyn in Database Systems, 6th Edition, Part 4 methodology*.

## 2.1. Identificação das Entidades

Aqui procedemos à identificação das entidades e a tabela seguinte mostra ao detalhe esse propósito - nome das entidades; descrição associada a cada uma delas e a ocorrência.

| Entidade     | Descrição   | Nome Alternativo  | Ocorrência   |
|--------------|---|-------------------|--|
| Funcionário  | Um funcionário presta um serviço de venda de roupa numa loja ao cliente desempenhando o seguinte papel: regista as vendas, atende os clientes, cataloga as peças defeituosas e valida a existência da peça em loja. | Empregado, staff  | Um funcionário pode atender zero ou mais clientes por dia  |
| RegistoVenda | Contém informação de uma ou mais peças vendidas ao cliente. São registadas por um funcionário numa determinada data (dia/hora) e contém o preço de venda ao público (PVP) da peça individual e o total da compra.   | Serviço prestado  | O mesmo registo de venda pode conter uma ou mais peças efetuadas a um cliente. Pode ocorrer zero ou mais registos por dia. |
| Cliente      | O cliente identifica a peça que quer comprar e paga pelo serviço de venda em loja.  | Utilizador        | O cliente pode comprar zero ou mais peças.   |
| Artigo       | O artigo é o define de forma geral uma peça de roupa, com os seus atributos: marca, modelo, tamanho, cor e preço.   | Produto           | Cada artigo é composto por zero ou n quantidade de peças.  |
| Peça         | A peça é o detalhe do Artigo e pode conter características diferenciadoras.   | Produto detalhado | Cada peça pode existir em zero ou mais unidades.   |
| Secção       | A secção representa-se como uma divisão física da loja e pode ser de três tipos: Homem, Mulher e Criança.   | Repartição        | Cada secção contém 10 escaparates.   |
| Escaparate   | O escaparate expõe as peças para os clientes  | Montra            | Cada escaparate pode conter zero ou mais peças.  |

|  |  |  |  |
|--|--|--|--|
|  | poderem identificar o que pretendem comprar. |  |  |
|--|--|--|--|

Tabela 2 – Identificação de Identidade

## 2.2. Identificação dos Relacionamentos

| Entidade     | Multiplicidade | Relacionamento   | Entidade     | Multiplicidade |
|--------------|----------------|------------------|--------------|----------------|
| Funcionário  | 1..1           | Regista          | RegistoVenda | 0..*           |
| RegistoVenda | 0..*           | É registado      | Funcionário  | 1..1           |
|              | 1..*           | Contém           | Cliente      | 1..1           |
|              | 0..1           | Contém           | Peça         | 1..*           |
| Cliente      | 1..1           | Está contido     | RegistoVenda | 1..*           |
| Artigo       | 1..1           | Contém           | Peça         | 0..*           |
| Peça         | 1..*           | Está contida     | RegistoVenda | 0..1           |
|              | 0..*           | Está contida     | Artigo       | 1..1           |
|              | 0..*           | Pode ser Exposta | Escaparate   | 0..1           |
| Secção       | 1..1           | Contém           | Escaparate   | 1..*           |
| Escaparate   | 1..*           | Está contido     | Secção       | 1..1           |
|              | 0..1           | Pode Conter      | Peça         | 0..*           |

Tabela 3 – Identificação dos relacionamentos entre entidades

## 2.3. Identificação e atribuição de Atributos às Entidades

| Entidade     | Atributos                    | Descrição  | Tipo dado & Tamanho                | Nulls | Multi-valued |
|--------------|------------------------------|--|------------------------------------|-------|--------------|
| Funcionário  | <b><u>IdFuncionario</u></b>  | Identifica univocamente um Funcionário                             | 2 dígitos                          | Não   | Não          |
|              | NomeFuncionario              | Nome do Funcionário  | 150 caracteres                     | Não   | Não          |
|              | MoradaFuncionario            | Morada do Funcionário  | 200 caracteres                     | Não   | Não          |
|              | Turno                        | Identifica o turno que o funcionário realiza                       | Bit                                | Não   | Não          |
|              | NIF                          | Número de Identificação Fiscal do Cliente exatamente com 9 dígitos | 9 dígitos                          | Não   | Não          |
|              | ContactoFuncionario          | Contacto do Funcionário entre 9 e 13 caracteres                    | 13 caracteres                      | Não   | Não          |
| RegistoVenda | <b><u>IdRegistoVenda</u></b> | Identifica univocamente um Pedido                                  | 5 dígitos                          | Não   | Não          |
|              | IdFuncionario                | Identifica univocamente um Funcionário                             | 2 dígitos                          | Não   | Não          |
|              | DataVenda                    | Data da venda  | Data e Hora no formato AAAA/MM/ DD | Não   | Não          |

|            |                              |  |                                    |     |     |
|------------|------------------------------|--|------------------------------------|-----|-----|
|            | NIF                          | Número de Identificação Fiscal do Cliente exactamente com 9 dígitos    | 9 dígitos                          | Não | Não |
|            | TotalPagar                   | Total a pagar c/iva  | 6 dígitos decimais                 | Não | Não |
|            | Obs                          | Campo para observações   | 255 caracteres                     | Sim | Não |
| Cliente    | NIF                          | Número de Identificação Fiscal do Cliente exactamente com 9 dígitos    | 9 dígitos                          | Não | Não |
|            | NomeCliente                  | Nome do cliente (prato/artigo)   | 50 caracteres                      | Não | Não |
|            | ContactoCliente              | Contacto do Funcionário entre 9 e 13 caracteres                        | 13 caracteres                      | Não | Não |
| Artigo     | <b><u>IdArtigo</u></b>       | Identifica univocamente um artigo                                      | 5 dígitos                          | Não | Não |
|            | NomeArtigo                   | Nome do artigo   | 50 caracteres                      | Não | Não |
|            | MarcaArtigo                  | Marca do artigo  | 50 caracteres                      | Não | Não |
|            | Modelo                       | Modelo das peças de roupa  | 50 caracteres                      | Não | Não |
|            | Tamaho                       | Tamanho das peças de roupa   | 50 caracteres                      | Não | Não |
|            | Preco                        | Preço do artigo  | 6 dígitos decimais                 |     |     |
|            | Cor                          | Cor do artigo  | 50 caracteres                      | Não | Não |
| Peca       | <b><u>IdPeca</u></b>         | Identifica univocamente uma Peça                                       | 5 dígitos                          | Não | Não |
|            | IdArtigo                     | Identifica univocamente um artigo                                      | 5 dígitos                          | Não | Não |
|            | IdEscaparate                 | Identifica univocamente um Escaparate                                  | 2 dígitos                          | Não | Não |
|            | <b><u>IdRegistoVenda</u></b> | Identifica univocamente um Pedido                                      | 5 dígitos                          | Sim | Não |
|            | DataDefeito                  | Data da identificação do defeito                                       | Data e Hora no formato AAAA/MM/ DD | Sim | Não |
|            | Estado                       | Identifica a localização da peça e se for null, se esta já foi vendida | 50 caracteres                      | Sim | Não |
| Seccao     | <b><u>IdSeccao</u></b>       | Identifica univocamente uma Secção                                     | 2 dígitos                          | Não | Não |
|            | Nome                         | Nome da secção   | 50 caracteres                      | Não | Não |
| Escaparate | <b><u>IdEscaparate</u></b>   | Identifica univocamente um Escaparate                                  | 2 dígitos                          | Não | Não |
|            | IdSeccao                     | Identifica univocamente uma Secção                                     | 2 dígitos                          | Não | Não |

Tabela 4 – Identificação e descrição dos atributos às entidades

## 2.4. Determinar o respetivo Domínio para cada Atributo

| Entidade     | Atributos                    | Domínio   |
|--------------|------------------------------|---|
| Funcionario  | <b><u>IdFuncionario</u></b>  | Número inteiro positivo com 2 dígitos entre 01 e 99 |
|              | NomeFuncionario              | Campo com 150 caracteres                            |
|              | MoradaFuncionario            | Campo com 200 caracteres                            |
|              | Turno                        | Bit: 1: manhã 0: tarde                              |
|              | NIF                          | Número com 9 dígitos entre 100000000 e 999999999    |
|              | ContactoFuncionario          | Contacto do funcionário no mínimo com 9 caracteres  |
| RegistoVenda | <b><u>IdRegistoVenda</u></b> | Número com 5 dígitos entre 00001 e 99999            |
|              | IdFuncionario                | Número inteiro positivo com 2 dígitos entre 01 e 99 |
|              | DataVenda                    | Data  |
|              | NIF                          | Número com 9 dígitos entre 100000000 e 999999999    |
|              | TotalPagar                   | Número decimal positivo com 6 dígitos               |
|              | Obs                          | Campo com 255 caracteres                            |
| Cliente      | NIF                          | Número com 9 dígitos entre 100000000 e 999999999    |
|              | NomeCliente                  | Campo com 50 caracteres                             |
|              | ContactoCliente              | Contacto do cliente no mínimo com 9 caracteres      |
| Artigo       | <b><u>IdArtigo</u></b>       | Número com 5 dígitos entre 00001 e 99999            |
|              | NomeArtigo                   | Campo com 50 caracteres                             |
|              | MarcaArtigo                  | Campo com 50 caracteres                             |
|              | Modelo                       | Campo com 50 caracteres                             |
|              | Tamaho                       | Campo com 50 caracteres                             |
|              | Preco                        | Número decimal positivo com 6 dígitos               |
|              | Cor                          | Campo com 50 caracteres                             |
| Peca         | <b><u>IdPeca</u></b>         | Número com 5 dígitos entre 00001 e 99999            |
|              | IdArtigo                     | Número com 5 dígitos entre 00001 e 99999            |
|              | IdEscaparate                 | Número inteiro positivo com 2 dígitos entre 01 e 99 |
|              | IdRegistoVenda               | Número com 5 dígitos entre 00001 e 99999            |
|              | DataDefeito                  | Data  |
|              | Estado                       | Campo com 50 caracteres                             |
| Seccao       | <b><u>IdSeccao</u></b>       | Número inteiro positivo com 2 dígitos entre 01 e 99 |
|              | Nome                         | Campo com 50 caracteres                             |
| Escaparate   | <b><u>IdEscaparate</u></b>   | Número inteiro positivo com 2 dígitos entre 01 e 99 |
|              | IdSeccao                     | Número inteiro positivo com 2 dígitos entre 01 e 99 |

Tabela 5 – Domínio dos atributos



## 2.5. Identificação das Chaves Primárias e das Chaves Candidatas

| Entidade     | Chave Primária        | Chave Candidata |
|--------------|-----------------------|-----------------|
| Funcionario  | <u>IdFuncionario</u>  | NIF, Contato    |
| RegistoVenda | <u>IdRegistoVenda</u> | NIF             |
| Cliente      | <u>NIF</u>            | ContactoCliente |
| Artigo       | <u>IdArtigo</u>       | N/A             |
| Peca         | <u>IdPeca</u>         | N/A             |
| Seccao       | <u>IdSeccao</u>       | Nome            |
| Escaparate   | <u>IdEscaparate</u>   | N/A             |

Tabela 6 – Identificação das chaves primárias e candidatas

## 2.6. Desenho Conceptual

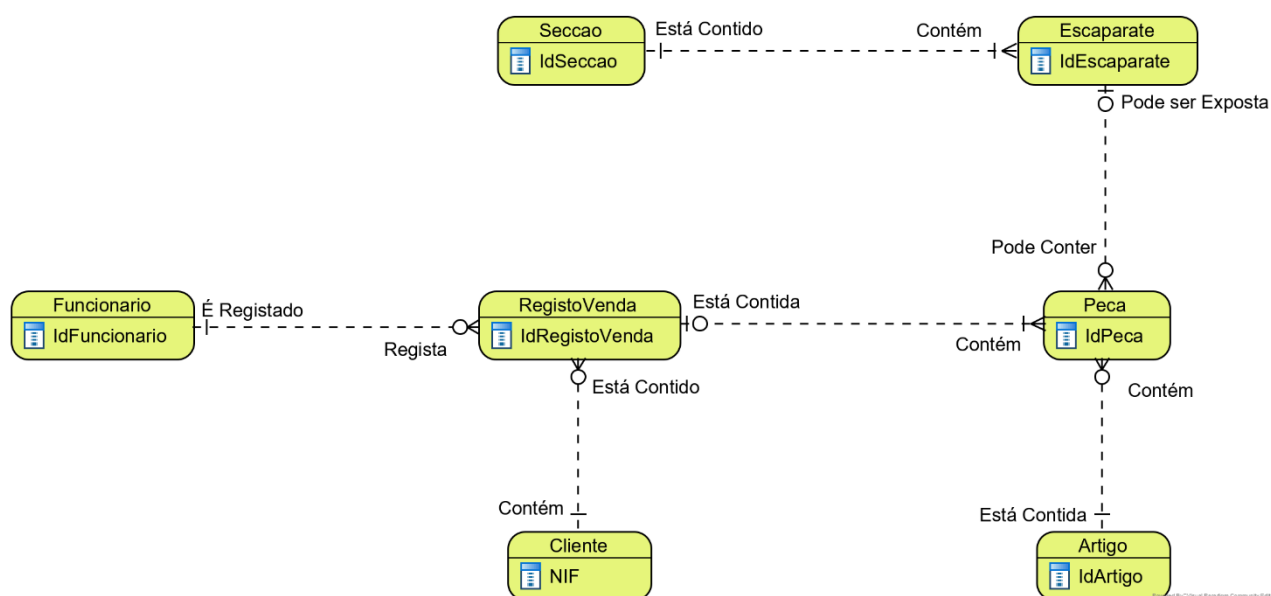


Figura 1 – Diagrama conceptual

### 3. Desenho Lógico

#### 3.1. Construção e validação do Modelo Lógico de Dados

Conforme aprendemos nas aulas vamos a partir do primeiro modelo conceptual construir o nosso modelo lógico. Mediante os dados, definidos e de acordo com as boas práticas e do referencial do livro proposto, damos seguimento a construção deste modelo lógico.

#### 3.2. Obter as relações para o Modelo Lógico de Dados

Nesta etapa pretendemos a identificação e criação das relações para o modelo lógico de dados. Deste modo, passamos a representar todas as entidades, seus atributos e relacionamentos que foram sendo definidos no anterior desenho conceptual.

De seguida, apresentamos a lista de relações por entidade:

**Funcionario:** IdFuncionario, NomeFuncionario, MoradaFuncionario, Turno, NIF, ContactoFuncionario

**Chave Primária:** IdFuncionario

**Chave Estrangeira:** NIF, Contato

**RegistoVenda:** IdRegistoVenda, IdFuncionario, NIF, DataVenda, TotalPagar, Obs

**Chave Primária:** IdRegistoVenda

**Chave Estrangeira:** NIF

**Cliente:** NIF, NomeCliente, ContactoCliente

**Chave Primária:** NIF

**Chave Estrangeira:** ContactoCliente

**Artigo:** IdArtigo, NomeArtigo, MarcaArtigo, Modelo, Tamaho, Preco, Cor

**Chave Primária:** IdArtigo

**Chave Estrangeira:** N.A.

**Peca:** IdPeca, IdArtigo, IdEscaparete, IdRegistoVenda, DataDefeito

**Chave Primária:** IdPeca

**Chave Estrangeira:** N.A.

**Seccao:** IdSeccao, Nome

**Chave Primária:** IdSeccao

**Chave Estrangeira:** N.A.

**Escaparete:** IdEscaparete, IdSeccao

**Chave Primária:** IdEscaparete

**Chave Estrangeira:** N.A.

### 3.3. Diagrama ER com a identificação das Chaves Estrangeiras

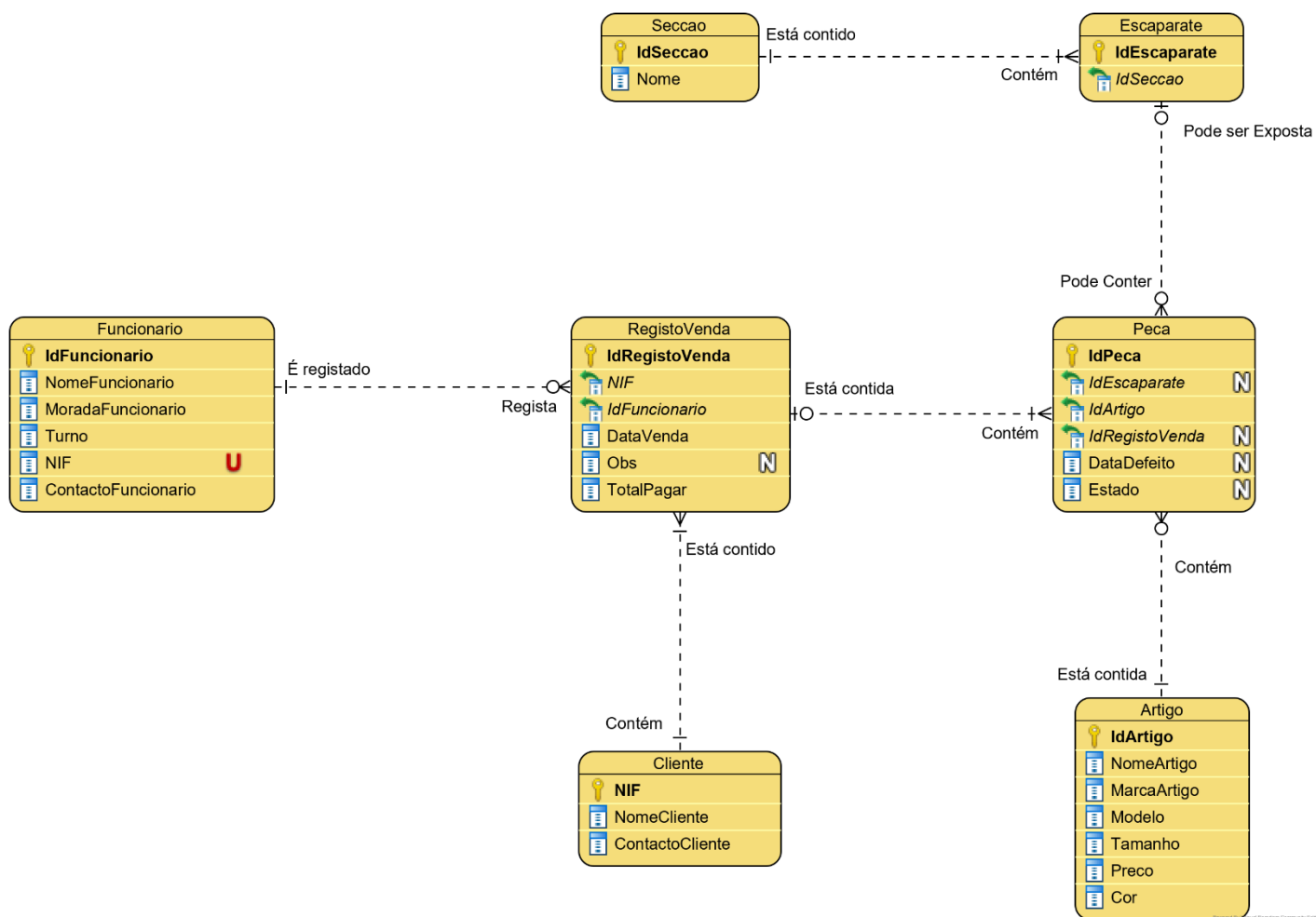


Figura 2 – Diagrama Lógico

### 3.4. Normalização

#### 3.4.1- Documento Registo de Venda

Loja XPTO

Morada XYZT

NIPC

+++++

Tipo de documento: FA

Nr. Do Documento: FA0001

Data de venda: 20-12-2020

NIF Cliente: 239943260

+++++

Artigo 1.....10.00

Artigo 2.....15.00

Artigo 3.....30.00

Total45.00

+++++

Funcionario: 253

Obs: Devoluçao até 30 dias

##### 1.ª Forma Normal

IdRegistoVenda, DataVenda, NIF, NomeArtigo, IdPeca, Preco, TotalPagar, IdFuncionario, Obs

##### Dependencia parciais

IdRegistoVenda -> DataVenda, TotalPagar, Obs

IdPeca -> IdArtigo

##### 2ª Forma Normal

Registo de Venda: IdRegistoVenda, NIF, IdFuncionario, DataVenda, TotalPagar, Obs

Peça: IdPeca, IdArtigo

##### Dependências transitivas

IdArtigo-> NomeArtigo, Preco

##### 3ª Forma Normal

Registo de Venda: IdRegistoVenda, NIF, IdFuncionario, DataVenda, TotalPagar, Obs,

Peça: IdPeca, IdArtigo,

Artigo: IdArtigo, NomeArtigo, Preco,

### 3.4.2- Validação das relações com uso da normalização

#### 1) Identificação de dependências funcionais

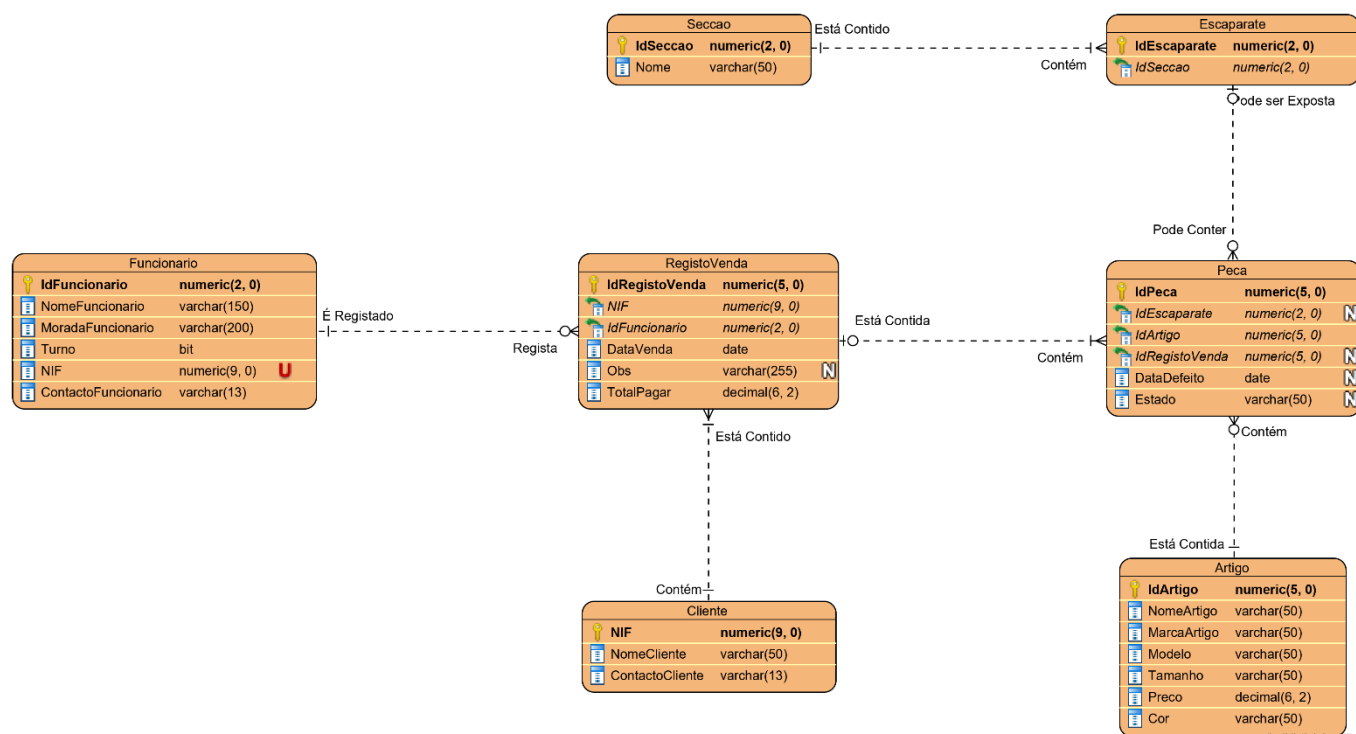
- IdSeccao -> Nome;
- IdEscapareate -> IdSeccao;
- IdPeca -> IdEscapareate, IdArtigo, IdRegistoVenda, DataDefeito;
- IdArtigo -> NomeArtigo, MarcaArtigo, Modelo, Tamanho, Preco, Cor;
- IdRegistoVenda -> NIF, IdFuncionario, DataVenda, Obs, TotalPagar;
- NIF -> NomeCliente, ContactoCliente;
- IdFuncionario -> NomeFuncionario, MoradaFuncionario, Turno, NIF, ContactoFuncionario;

#### 2) Identificação de dependências transitivas

- Não aplicável

### 3.5. Conclusão da Normalização

Verifica-se então que o seguinte modelo está completamente na 3FN devido ao facto de não existir dependências transitivas, que seriam removidas no processo de normalização



### 3.6. Definir restrições de integridade – Regras de Negócio

#### Requisitos/Restrições empresa consultadoria

- ✓ Artigo será a descrição geral de cada peça, isto é, a tabela peça tem como objetivo identificar. peças de roupas únicas com X características definidas em Artigo.
- ✓ Os clientes serão registados a partir do seu NIF e este não pode ter mais do que 9 dígitos.
- ✓ Cada escaparate pode ter várias peças.
- ✓ Cada peça possui um código de referência, um modelo e um preço em euros.
- ✓ Cada funcionário tem um turno definido sendo este, manhã ou tarde.
- ✓ Um registo de venda contém o NIF do cliente, a lista de peças, o funcionário que recebeu o pedido, a data, o total a pagar e um conjunto de observações.
- ✓ Não podem existir preços negativos.
- ✓ Não podem existir quantidades negativas.

#### Partimos do pressuposto que:

- ✓ A loja está dividida em 3 secções, sendo estas homem, mulher e criança.
- ✓ Cada secção tem a partida 10 escaparates.
- ✓ Toda a roupa presente nos escaparates terá semelhantes de diferentes tamanhos no stock da loja cujo funcionários terão acesso.
- ✓ Todas as peças poderão estar no stock da loja, no armazém ou vendidas.
- ✓ É preciso gestão de funcionários para registar nos pedidos.
- ✓ Após o pagamento os pedidos devem ser guardados num histórico.
- ✓ Ao pedir o recibo tem que ter NIF.

```
CREATE PROCEDURE CalcularTotalPagar
AS
BEGIN
    UPDATE dbo.RegistoVenda
    SET TotalPagar = t.Total
    FROM (
        SELECT
            dbo.Peca.IdRegistoVenda AS IdRegistoVenda,
            SUM(dbo.Artigo.Preco) AS Total
        FROM
            dbo.Artigo
            JOIN dbo.Peca ON dbo.Artigo.IdArtigo = dbo.Peca.IdArtigo
        GROUP BY
            dbo.Peca.IdRegistoVenda
    ) AS t
    WHERE dbo.RegistoVenda.IdRegistoVenda = t.IdRegistoVenda;
END;
```

Este stored procedure ao ser executado vai calcular automaticamente o preço total de cada Registo de venda.

#### 4. Implementar a Base de Dados em SQL Server 2022 (ou posterior)

##### 4.1. Criação de tabelas

- Seccao

```
CREATE TABLE Seccao
(
    IdSeccao NUMERIC(2,0) NOT NULL IDENTITY(0,1),
    Nome VARCHAR(50) NOT NULL,

    PRIMARY KEY (IdSeccao)
);
```

- Escaparate

```
CREATE TABLE Escaparate
(
    IdEscaparate NUMERIC(2,0) NOT NULL IDENTITY(0,1),
    IdSeccao NUMERIC(2,0) NOT NULL,

    PRIMARY KEY (IdEscaparate),
    FOREIGN KEY (IdSeccao) REFERENCES Seccao(IdSeccao)
);
```

- Artigo

```
CREATE TABLE Artigo
(
    IdArtigo NUMERIC(5,0) NOT NULL IDENTITY(0,1),
    NomeArtigo VARCHAR(50) NOT NULL,
    MarcaArtigo VARCHAR(50) NOT NULL,
    Modelo VARCHAR(50) NOT NULL,
    Tamanho VARCHAR(50) NOT NULL,
    Preco DECIMAL(6,2) NOT NULL DEFAULT 0.00 CONSTRAINT artigoPreco CHECK (Preco >= 0),
    Cor VARCHAR(50) NOT NULL,

    PRIMARY KEY (IdArtigo)
);
```

- Funcionario

```
CREATE TABLE Funcionario
(
    IdFuncionario NUMERIC(2,0) NOT NULL IDENTITY(0,1),
    NomeFuncionario VARCHAR(150) NOT NULL,
    MoradaFuncionario VARCHAR(200) NOT NULL,
    Turno BIT NOT NULL,
    NIF NUMERIC(9,0) NOT NULL UNIQUE,
    ContactoFuncionario VARCHAR(13) NOT NULL,

    PRIMARY KEY (IdFuncionario)
);
```

- Cliente

```
CREATE TABLE Cliente
(
    NIF NUMERIC(9,0) NOT NULL IDENTITY(0,0),
    NomeCliente VARCHAR(50) NOT NULL,
    ContactoCliente VARCHAR(13) NOT NULL,

    PRIMARY KEY (NIF)
);
```

- RegistoVenda

```
CREATE TABLE RegistoVenda
(
    IdRegistoVenda NUMERIC(5,0) NOT NULL IDENTITY(0,1),
    NIF NUMERIC(9,0) NOT NULL,
    IdFuncionario NUMERIC(2,0) NOT NULL,
    DataVenda DATETIME NOT NULL,
    Obs VARCHAR(255) NULL,
    TotalPagar DECIMAL(6,2) NOT NULL DEFAULT 0.00 CONSTRAINT registoVendaTotal CHECK (TotalPagar >= 0),

    PRIMARY KEY (IdRegistoVenda),
    FOREIGN KEY (NIF) REFERENCES Cliente(NIF),
    FOREIGN KEY (IdFuncionario) REFERENCES Funcionario(IdFuncionario)
);
```

- Peca

```
CREATE TABLE Peca
(
    IdPeca NUMERIC(5,0) NOT NULL IDENTITY(0,1),
    IdEscaparete NUMERIC(2,0) NULL,
    IdArtigo NUMERIC(5,0) NOT NULL,
    IdRegistoVenda NUMERIC(5,0) NULL,
    DataDefeito DATETIME NULL,

    PRIMARY KEY (IdPeca),
    FOREIGN KEY (IdEscaparete) REFERENCES Escaparete(IdEscaparete),
    FOREIGN KEY (IdArtigo) REFERENCES Artigo(IdArtigo),
    FOREIGN KEY (IdRegistoVenda) REFERENCES RegistoVenda(IdRegistoVenda)
);
```

## 4.2. Restrições

- Artigo

A restrição presente no atributo “Preco” obriga que os valores sejam superiores a 0.00

- RegistoVenda

A restrição presente no atributo “TotalPagar” obriga que os valores sejam superiores a 0.00



### 4.3. Triggers

```
CREATE TRIGGER trg_SetEstadoToNull
ON [dbo].[Peca]
AFTER INSERT, UPDATE
AS
BEGIN
    IF EXISTS (SELECT 1 FROM inserted WHERE [IdRegistoVenda] IS NOT NULL)
    BEGIN
        UPDATE [dbo].[Peca]
        SET [Estado] = NULL
        WHERE [IdPeca] IN (SELECT [IdPeca] FROM inserted WHERE [IdRegistoVenda] IS NOT NULL)
    END
END
```

Este trigger tem como função, ao inserir ou modificar uma peça, automaticamente modificar o Estado desta para Null caso esta tenha sido vendida. Pois o Estado indica se a peça estava no stock da loja ou no armazém.

### 4.4. Queries

1.

```
USE ClothingStore;
GO
|
SELECT
    SUM(dbo.RegistoVenda.TotalPagar) AS Total
FROM
    dbo.RegistoVenda
WHERE
    DAY(dbo.RegistoVenda.DataVenda) = DAY(DATEADD(DAY,-1,GETDATE()))

USE ClothingStore;
GO
|
SELECT
    SUM(dbo.RegistoVenda.TotalPagar) AS Total
FROM
    dbo.RegistoVenda
WHERE
    DATEPART(WEEK, dbo.RegistoVenda.DataVenda) = DATEPART(WEEK, DATEADD(WEEK, -1, GETDATE()));

USE ClothingStore;
GO
|
SELECT
    SUM(dbo.RegistoVenda.TotalPagar) AS Total
FROM
    dbo.RegistoVenda
WHERE
    MONTH(CAST(dbo.RegistoVenda.DataVenda AS DATE)) = MONTH(CAST(DATEADD(MONTH,-1,GETDATE()) AS DATE))
```

2.

```
USE ClothingStore;
GO

SELECT
    dbo.Artigo.*
FROM
    (SELECT
        dbo.RegistoVenda.IdRegistoVenda
    FROM
        dbo.RegistoVenda
    WHERE
        DAY(dbo.RegistoVenda.DataVenda) = DAY(DATEADD(DAY,-1,GETDATE()))
    GROUP BY
        dbo.RegistoVenda.IdRegistoVenda
    ) AS t , dbo.Peca, dbo.Artigo
WHERE
    t.IdRegistoVenda = dbo.Peca.IdRegistoVenda AND dbo.Peca.IdArtigo = dbo.Artigo.IdArtigo
```

3.

```
USE ClothingStore;
GO

SELECT
    dbo.Artigo.*
FROM
    (SELECT
        dbo.RegistoVenda.IdRegistoVenda
    FROM
        dbo.RegistoVenda
    WHERE
        DAY(dbo.RegistoVenda.DataVenda) = DAY(DATEADD(DAY,-1,GETDATE()))
    GROUP BY
        dbo.RegistoVenda.IdRegistoVenda
    ) AS t , dbo.Peca, dbo.Artigo
WHERE
    t.IdRegistoVenda = dbo.Peca.IdRegistoVenda AND dbo.Peca.IdArtigo = dbo.Artigo.IdArtigo
```

4.

```
USE ClothingStore;
GO

SELECT DISTINCT TOP 1
    dbo.Artigo.*,
    COUNT(dbo.Peca.IdRegistoVenda) AS Total
FROM
    dbo.Artigo,
    dbo.Peca
WHERE
    dbo.Artigo.IdArtigo = dbo.Peca.IdArtigo
GROUP BY
    dbo.Artigo.IdArtigo,
    dbo.Artigo.NomeArtigo,
    dbo.Artigo.MarcaArtigo,
    dbo.Artigo.Modelo,
    dbo.Artigo.Tamanho,
    dbo.Artigo.Preco,
    dbo.Artigo.Cor
ORDER BY
    Total DESC
```

5.

```
USE ClothingStore;
GO

SELECT
    dbo.Artigo.*,
    COUNT(dbo.Peca.IdArtigo) AS TotalPecasDefeituosas
FROM
    dbo.Peca,
    dbo.Artigo
WHERE
    DATEPART(WEEK, dbo.Peca.DataDefeito) = DATEPART(WEEK, DATEADD(WEEK, -1, '2023-06-04')) AND dbo.Peca.IdArtigo = dbo.Artigo.IdArtigo
GROUP BY
    dbo.Artigo.IdArtigo,
    dbo.Artigo.NomeArtigo,
    dbo.Artigo.MarcaArtigo,
    dbo.Artigo.Modelo,
    dbo.Artigo.Tamanho,
    dbo.Artigo.Preco,
    dbo.Artigo.Cor
```

## 4.5. Views

1.

```
CREATE VIEW ArtigosSemDefeito
AS
SELECT
    dbo.Artigo.IdArtigo,
    dbo.Artigo.NomeArtigo,
    dbo.Artigo.MarcaArtigo,
    dbo.Artigo.Modelo,
    dbo.Artigo.Tamanho,
    dbo.Artigo.Preco,
    dbo.Artigo.Cor
FROM
    dbo.Artigo INNER JOIN
    dbo.Peca ON dbo.Artigo.IdArtigo = dbo.Peca.IdPeca AND dbo.Peca.DataDefeito = NULL
;
GO
```

2.

```
CREATE VIEW ArtigosComDefeito
AS
SELECT
    dbo.Artigo.IdArtigo,
    dbo.Artigo.NomeArtigo,
    dbo.Artigo.MarcaArtigo,
    dbo.Artigo.Modelo,
    dbo.Artigo.Tamanho,
    dbo.Artigo.Preco,
    dbo.Artigo.Cor,
    dbo.Peca.DataDefeito
FROM
    dbo.Artigo INNER JOIN
    dbo.Peca ON dbo.Artigo.IdArtigo = dbo.Peca.IdPeca AND dbo.Peca.DataDefeito != NULL
;
GO
```

3.

```
CREATE VIEW RegistosVenda
AS
SELECT
    dbo.RegistoVenda.IdRegistoVenda,
    dbo.Cliente.NomeCliente,
    dbo.Funcionario.NomeFuncionario,
    dbo.RegistoVenda.DataVenda,
    dbo.RegistoVenda.Obs,
    dbo.RegistoVenda.TotalPagar
FROM
    dbo.RegistoVenda INNER JOIN
    dbo.Cliente ON dbo.RegistoVenda.NIF = dbo.Cliente.NIF INNER JOIN
    dbo.Funcionario ON dbo.RegistoVenda.IdFuncionario = dbo.Funcionario.IdFuncionario
;
GO
```

4.

```
CREATE VIEW RegistosVendaDetalhes
AS
SELECT
    dbo.RegistoVenda.IdRegistoVenda,
    dbo.Artigo.IdArtigo,
    dbo.Artigo.NomeArtigo,
    dbo.Artigo.MarcaArtigo,
    dbo.Artigo.Modelo,
    dbo.Artigo.Tamanho,
    dbo.Artigo.Preco,
    dbo.Artigo.Cor,
    dbo.RegistoVenda.DataVenda,
    dbo.RegistoVenda.TotalPagar AS TotalRegistoVenda
FROM
    dbo.RegistoVenda INNER JOIN
    dbo.Peca ON dbo.RegistoVenda.IdRegistoVenda = dbo.Peca.IdRegistoVenda INNER JOIN
    dbo.Artigo ON dbo.Peca.IdArtigo = dbo.Artigo.IdArtigo
;
GO
```

5.

```
CREATE VIEW ArtigosNoEscaparete
AS
SELECT
    dbo.Escaparete.IdEscaparete,
    dbo.Artigo.IdArtigo,
    dbo.Artigo.NomeArtigo,
    dbo.Artigo.MarcaArtigo,
    dbo.Artigo.Modelo,
    dbo.Artigo.Tamanho,
    dbo.Artigo.Preco,
    dbo.Artigo.Cor
FROM
    dbo.Escaparete INNER JOIN
    dbo.Peca ON dbo.Escaparete.IdEscaparete = dbo.Peca.IdEscaparete INNER JOIN
    dbo.Artigo ON dbo.Peca.IdArtigo = dbo.Artigo.IdArtigo
;
GO
```

6.

```
CREATE VIEW EscaparetesPorSeccao
AS
SELECT
    dbo.Seccao.IdSeccao,
    dbo.Seccao.Nome,
    dbo.Escaparete.IdEscaparete
FROM
    dbo.Seccao INNER JOIN
    dbo.Escaparete ON dbo.Seccao.IdSeccao = dbo.Escaparete.IdSeccao
;
GO
```

## 5. Conclusões e Trabalho Futuro

Nesta fase, procedemos a algumas considerações finais sobre a realização deste trabalho e de que forma este contribuiu para a consolidação dos nossos conhecimentos, obtidos no âmbito desta unidade curricular.

De acordo, com a descrição dos objetivos gerais e específicos apontados anteriormente, podemos concluir de forma unânime que estes foram obtidos de forma satisfatória, sedimentando os nossos conhecimentos sobre o sistema de gestão de base de dados.

Numa primeira etapa, procedemos à identificação das entidades, dos relacionamentos e posteriormente da atribuição de atributos a estas mesmas entidades. Este processo envolveu um aprofundamento dos conhecimentos deste processo de negócio bem como de alguns módulos que o compõem, a determinação do respetivo domínio para cada atributo e a identificação das chaves primárias e candidatas. Ainda nesta etapa procedeu-se ao desenho do modelo conceptual.

Seguidamente, desencadeou-se todo o processo Lógico que envolvia a construção e validação, a obtenção de relações, a realização do diagrama ER, bem como a sua normalização tendo em conta as restrições de integridade verificadas.

Mediante as premissas inicialmente elaboradas, e após o trabalho todo desenvolvido, procedemos ao manuseamento da ferramenta SQL Server através da criação de tabelas, de restrições nos atributos e a realização de Triggers.

Consensualmente, a realização deste trabalho abre caminho para uma melhor compreensão de conceitos chave sobre este tipo de negócio bem como uma compreensão elevada dos conteúdos programáticos desta Unidade Curricular, onde apesar das dificuldades sentidas, todos os membros do grupo pensamos que elaboramos um bom trabalho que satisfaça todas as condições pretendidas pelo docente desta UC.

## 6. Bibliografia

- Diapositivos fornecidos pelo docente da Unidade Curricular
- C. Thomas e B. Carolyn in *Database Systems*, 6th Edition, Part 4 methodology.