



**Universidade do Minho**  
Licenciatura em Ciências da Computação

## **Unidade Curricular de Bases de Dados**

Ano Letivo de 2022/2023

### **Entrega ao domicílio**

**André Costa, Bruno Araújo, Tiago Teixeira, Filipe Castro**

Janeiro, 2023

**BD**

Data de Recepção	
Responsável	
Avaliação	
Observações	

## Entrega ao domicílio

GRUPO 15

**André Costa, Bruno Araújo, Tiago Teixeira, Filipe Castro**

**A95869,A97509,A97666,A96156**

Janeiro, 2023

## Resumo

Neste relatório temos como objetivo descrever a implementação de uma base de dados para uma loja de equipamentos eletrônicos, nomeadamente computadores e os seus componentes, que permita organizar o método de trabalho dos funcionários e ajudar a loja a receber e gerir o número constantemente crescente de pedidos. A loja com a qual se está a trabalhar integra-se na categoria de entregas ao domicílio, sendo esta categoria o principal molde da base de dados em questão.

Ao longo deste relatório vai-se descrever todas as fases de implementação da base de dados começando na Contextualização e acabando no Plano de Segurança e Recuperação de Dados.

Em primeira lugar apresentamos o caso que temos em mãos, dando a contextualização, análise da viabilidade, etc.

De seguida é feito o levantamento dos requisitos de descrição, exploração e controlo que foram definidos através de reuniões com o Sr. Bezos e os funcionários do estabelecimento.

Na fase seguinte foi construído o modelo conceptual da nossa base de dados usando o *brModelo*, a partir do qual, foi depois definido o modelo lógico feito em *mysql Workbench* que, como já estávamos confortáveis a usar, foi usado também mais à frente para a implementação física juntamente do povoamento e queries.

Para terminar, calculamos e fizemos a previsão para o espaço que a base de dados ocupa agora e vai ocupar daqui a 1 ano e apresentamos o plano de segurança e recuperação de dados que adotamos para o nosso trabalho.

Finalizando todas as etapas damos então por terminado o desenvolvimento do relatório.

**Área de Aplicação:** Criação e Implementação de uma Base de Dados para Gestão de Entrega ao Domicílio.

**Palavras-Chave:** Base de Dados, Entidade, Atributo, Relacionamento, Modelo Conceptual, Modelo Lógico, Modelo Físico, *MySQL*, *SQL*, Queries,

# Índice

1. Introdução	1
1.1. Contextualização	1
1.2. Apresentação do Caso de Estudo	1
1.3. Motivação e Objetivos do Trabalho	2
1.4. Análise da Viabilidade do processo	2
1.5. Recursos	3
1.6. Equipa de Trabalho	3
1.7. Plano de Execução do Projeto	3
2. Levantamento e Análise de Requisitos	4
2.1. Método de levantamento e análise de requisitos adotado	4
2.2. Organização dos requisitos levantados	4
2.2.1 Requisitos de descrição	4
2.2.2 Requisitos de exploração	5
2.2.3 Requisitos de controlo	5
2.3. Análise e validação geral dos requisitos	6
3. Modelação Conceptual	7
3.1. Apresentação da abordagem de modelação realizada	7
3.2. Identificação e caracterização das entidades	7
3.3. Identificação e caracterização dos relacionamentos	7
3.4. Identificação e caracterização da associação dos atributos com as entidades e relacionamentos	8
3.5. Apresentação e explicação do diagrama ER produzido	11
4. Modelação Lógica	12
4.1. Construção e validação do modelo de dados lógico	12
4.2. Normalização de Dados	14
4.3. Apresentação e explicação do modelo lógico produzido	15
4.4. Validação do modelo com interrogações do utilizador	15
5. Implementação Física	17
5.1. Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL	17
5.2. Tradução das interrogações do utilizador para SQL	20
5.3. Definição e caracterização das vistas de utilização em SQL	24
5.4. Cálculo do espaço da bases de dados	25
5.5. Plano de segurança e recuperação de dados	26
6. Conclusões e Trabalho Futuro	27
Referências	28

Lista de Siglas e Acrónimos	29
Anexos	30
I. Criação de tabelas	31
II. Povoamento	34
III. Queries, Vistas, Utilizadores e Procedimentos	36

## Índice de Figuras

Figura 1 - Diagrama de GANTT 4

Figura 2 - Modelo Conceptual 11

Figura 3 - Modelo Lógico 14

Figura 4 - Árvore de álgebra relacional que lista o histórico de pedidos feito por cada cliente

Figura 5 - Árvore de álgebra relacional que mostra a capacidade de um pedido ter vários itens.

Figura 6 - Implementação física da entidade Cliente

Figura 7 - Implementação física do atributo multivalorado Telemóvel

Figura 8 - Implementação física da entidade Item

Figura 9 - Implementação física da entidade Estafeta

Figura 10 - Implementação física da entidade Administrativo

Figura 11 - Implementação física da entidade Transporte

Figura 12 - Implementação física da entidade Pedido

Figura 13 - Implementação física do relacionamento Item\_has\_Pedido

Figura 14 - Query em SQL que traduz o requisito "Listar o top 3 de clientes da loja".

Figura 15 - Resultado da Query anterior, uma tabela (Pcliente, Nome, Valor, Numero\_de\_Compras).

Figura 16 - Query em SQL que traduz o requisito "Através das datas de receção e de envio, é possível calcular o tempo de entrega".

Figura 17 - Resultado da Query anterior, uma tabela (idPedido, DataPedido, DataReceção, TempoEnvio).

Figura 18 - Query em SQL que traduz o requisito "Listar o histórico de pedidos feito por cada Cliente".

Figura 19 - Resultado da query anterior, uma tabela (idCliente, Nome, IdPedido, DataPedido, Estado, Preço).

Figura 20 - Query em SQL que traduz o requisito "Listar os funcionários que estiveram envolvidos em cada pedido".

Figura 21 - Resultado da query anterior, uma tabela (idPedido, idAdministrativo, NomeAdministrativo, idEstafeta, NomeEstafeta, Veículo).

Figura 22 - Query em SQL que traduz o requisito "Através das datas em que os pedidos são feitos, é possível calcular o período de maior atividade"

Figura 23 - Resultado da query anterior, uma tabela (HoradeMaiorAtividade).

Figura 24 - Código SQL da Vista que mostra a informação completa de todos os pedidos já feitos.

Figura 25 - Primeira parte do resultado da vista anterior, a tabela (Id do Cliente, Nome, Apelido, Email, Nif, Id do Pedido, Data do Pedido, Data da Receção, Estado, Preço, Rua, ...

Figura 26 - Última parte do resultado da vista da figura 24, continuação da tabela da figura 25, ..., Localidade, Código Postal, Id do Estafeta, Nome Estafeta, Salário Estafeta, Estafeta Ativo?, Matrícula, Id do Administrativo, Nome Administrativo, Salário Administrativo, Administrativo Ativo?)

## Índice de Tabelas

Tabela 1 - Identificação e caracterização de todas as entidades

Tabela 2 - Identificação e caracterização do relacionamento Pedido - Cliente

Tabela 3 - Identificação e caracterização do relacionamento Pedido - Estafeta

Tabela 4 - Identificação e caracterização do relacionamento Pedido - Transporte

Tabela 5 - Identificação e caracterização do relacionamento Pedido - Item

Tabela 6 - Identificação e caracterização do relacionamento Pedido - Administrativo

Tabela 7 - Identificação e caracterização da associação dos atributos de Cliente com a entidade Cliente

Tabela 8 - Identificação e caracterização da associação dos atributos de Estafeta com a entidade Estafeta

Tabela 9 - Identificação e caracterização da associação dos atributos de Transporte com a entidade Transporte

Tabela 10 - Identificação e caracterização da associação dos atributos de Item com a entidade Item

Tabela 12 - Identificação e caracterização da associação dos atributos de Pedido com a entidade Pedido

Tabela 13 - Identificação e caracterização da associação dos atributos de Administrativo com a entidade Administrativo



# **1. Introdução**

## **1.1. Contextualização**

Com o avanço tecnológico e a crescente necessidade de equipamentos eletrónicos cada vez se torna mais lucrativa a sua distribuição.

Assim, mais estabelecimentos com essa finalidade surgem, e entre eles alguns que tomam proveito da web, os clientes através da internet escolhem o que pretendem comprar e o produto é enviado até à sua morada, a implementação de tal website pode ser a escolha de uma empresa já existente ou podem aparecer empresas apenas com esse serviço disponível, o que pode ser uma escolha mais favorável para as que têm um budget mais limitado.

## **1.2. Apresentação do Caso de Estudo**

Foi fundada em abril de 2020 por José Bezos uma loja online que vende equipamentos eletrónicos, nomeadamente computadores e os seus componentes, e faz entregas ao domicílio desses produtos com a intenção de inovar tal mercado em Braga, que é onde se situa a sede da empresa. Na sede existem 10 funcionários incluindo 4 administrativos, 1 gestor e 5 estafetas.

Devido ao número crescente de pedidos nos últimos meses, foi decidido que é necessária uma base de dados para garantir que a empresa consiga acompanhar o aumento de procura.

### **1.3. Motivação e Objetivos do Trabalho**

A principal razão para o Sr. Bezos querer a implementação de uma base de dados no seu negócio é para melhorar a eficiência e a utilidade do site da loja pois, é a partir do site que fazem a maior parte da faturação. Isto vê-se como provável pois componentes de computadores são muito procurados por jovens entre os 16-24 anos, que são um grupo demográfico com elevada presença online.

Depois de uma reunião com o Sr. Bezos fizemos a seguinte lista de objetivos de acordo com o que ele nos deu a entender que fossem os objetivos da empresa com a implementação da base de dados:

- 1) Organizar o modelo de negócio
- 2) Melhorar a capacidade e eficácia da receção de pedidos
- 3) Organização dos recursos materiais e humanos
- 4) Gestão mais eficiente da faturação e lucros
- 5) Conhecer os clientes e guardar as suas tendências
- 6) Consulta dos dados da empresa em tempo real
- 7) Gestão dos horários dos funcionários
- 8) Melhorar a qualidade de serviço

### **1.4. Análise da Viabilidade do processo**

- 1) Atualizar o registo eletrónico cada vez que se recebe um pedido
- 2) Gerir os horários dos estafetas de acordo com as horas de maior procura
- 3) Registar os pedidos de forma a organizar e distribuí-los de forma mais eficiente
- 4) Otimizar a gerência de recursos humanos usando as horas de menor procura para outras tarefas da empresa
- 5) Guardar os históricos de cada cliente traçando-lhe um perfil
- 6) Dar acesso aos históricos de pedidos, mercados e produtos disponíveis aos clientes usando o site

# 1.5. Recursos

Para satisfazer tudo o que é preciso para a construção da base de dados são necessários os seguintes recursos:

**Humanos:** Gestores, Administrativos, Estafetas, Clientes, Equipa de desenvolvimento

**Materiais:** Hardware(1 servidor, 3 máquinas), Software(SGBD + Software Gestão)

# 1.6. Equipa de Trabalho

Foram então escolhidas as pessoas que vão fazer parte da equipa de trabalho, estas serão:

**Pessoal Interno:** João Sousa - Estafeta, Ana Silva - Administrativo, José Bezos - Gestor

**Pessoal Externo:** André Costa, Bruno Araújo, Filipe Castro, Tiago Teixeira - Equipa de desenvolvimento

# 1.7. Plano de Execução do Projeto

De forma à execução do projeto ser mais suave e diminuir a probabilidade de haver problemas inesperados, decidimos fazer um plano para organizarmos o desenvolvimento da base de dados utilizando um diagrama de GANTT. Nesse diagrama apresentamos os dias em que trabalharemos em cada capítulo da execução do projeto e o tempo que vamos, em estimativa, demorar em cada um.

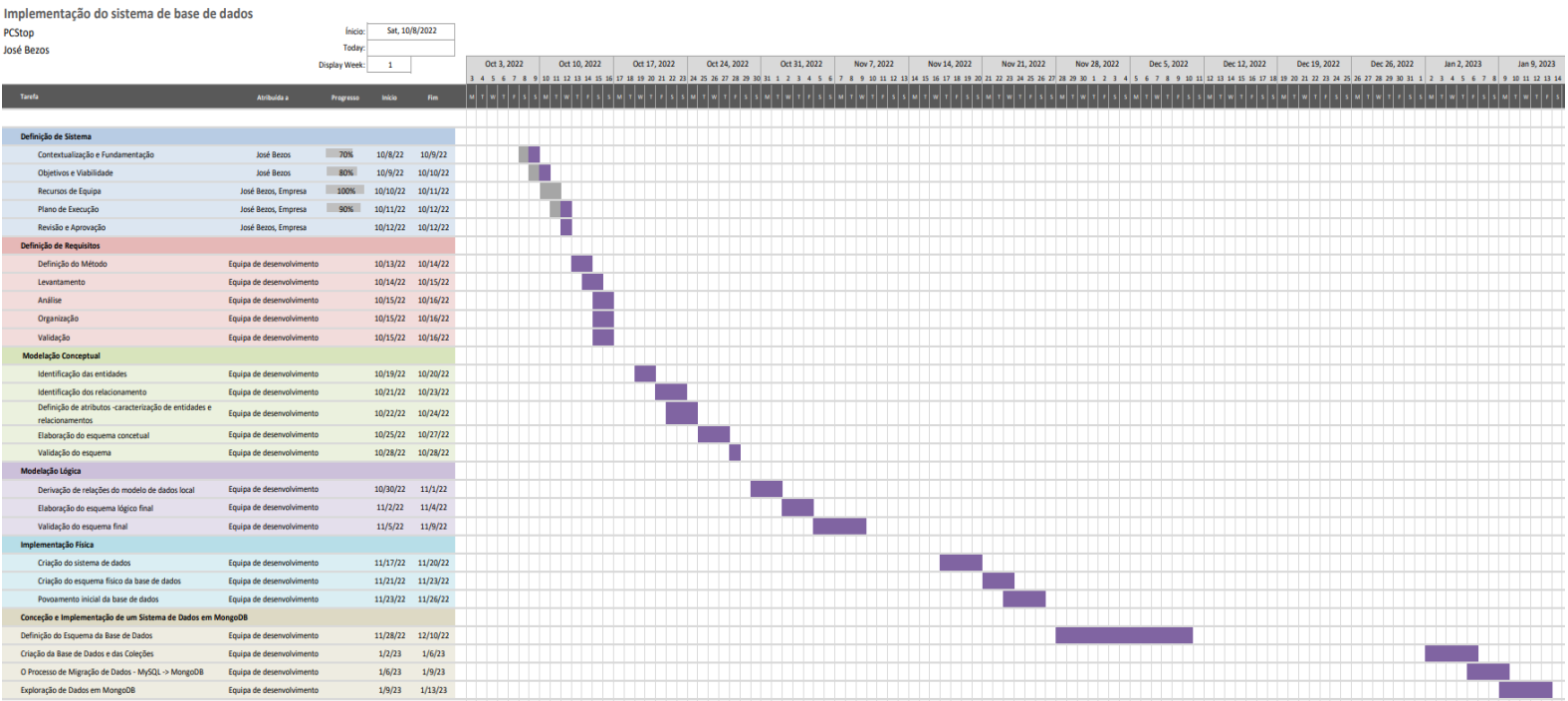


Figura 1

## **2. Levantamento e Análise de Requisitos**

### **2.1. Método de levantamento e análise de requisitos adotado**

Para entendermos o que a nossa base de dados deverá cumprir, decidimos planejar diversas reuniões com o Sr. Bezos e alguns dos seus funcionários, durante essas reuniões anotamos informações sobre o que nos era exigido. Também realizamos um inquérito entre passados clientes do serviço e fizemos uma pesquisa sobre o serviço da entrega ao domicílio.

Reunindo e processando toda esta informação acabamos com a lista de requisitos que serão apresentados nos próximos tópicos.

### **2.2. Organização dos requisitos levantados**

#### **2.2.1 Requisitos de descrição**

- Cada estafeta tem id, nome e contrato (salário e a situação do contrato).
- Cada administrativo tem id, nome e contrato (salário e a situação do contrato)
- Cada cliente tem id, nome, NIF, email e telemóvel.
- Cada pedido tem id, transporte, administrativo, estafeta, data de receção, data de envio, data em que foi feito o pedido, cliente, item, estado e endereço de entrega (rua, localidade e código postal).
- Cada transporte tem matrícula e custo dos portes.
- Cada item tem id, nome, preço, descrição, iva e stock.
- Sempre que um cliente realiza o seu primeiro pedido este é adicionado ao sistema, incluindo o seu NIF, email, telefone e nome.
- Sempre que é efetuado um novo pedido, este é registado no sistema com a data de receção, cliente, o(s) item(s), estafeta, administrativo, estado, endereço de entrega, data em que o pedido foi feito e transporte.
- Todos os itens devem estar catalogados no sistema.
- Quando a empresa adquire um novo transporte, este deve ser registado no sistema.

## 2.2.2 Requisitos de exploração

- Cada pedido pode ter vários itens.
- Quando um estafeta abandona a empresa, o atributo “Ativo” do seu contrato passa a ter o valor “Não” e este deixa de receber pedidos.
- Através das datas de receção e de envio, é possível calcular o tempo de entrega.
- Através das datas em que os pedidos são feitos, é possível calcular os períodos de maior atividade.
- A data de envio, a data de receção e NIF são atributos que podem ter o valor nulo.
- A data de envio tem um valor diferente de nulo atribuído quando o estado do pedido passa para “enviado”.
- A data de receção tem um valor diferente de nulo atribuído quando o estado do pedido passa para “entregue”.
- NIF é um atributo opcional que o cliente escolhe se vai ter valor diferente de nulo ou não.
- Cada estafeta pode apenas entregar um pedido de cada vez.
- Cada cliente tem 1 ou mais números de telemóvel a si associados.
- Listar o histórico de pedidos feito por cada cliente.
- Listar os funcionários que estiveram envolvidos em cada pedido.
- Listar o top 3 de clientes da loja
- Listar todos os itens que fazem parte de certo pedido.
- Deve ser possível ver os pedidos que foram feitos durante certo mês.
- Deve ser possível verificar o item que foi mais vendido.
- Deve ser possível verificar que administrativo registou menos pedidos.
- Um pedido só pode ser feito por um cliente.

## 2.2.3 Requisitos de controlo

- Os gestores têm permissões de inserção, modificação e remoção de dados.
- Os administrativos têm permissões para inserção e modificação de pedidos, clientes e itens.
- Os estafetas só podem ver os pedidos que lhes são atribuídos e não podem modificá-los de nenhuma forma.
- Os clientes têm acesso a algumas informações sobre os pedidos que fizeram através do site.
- Os clientes têm acesso a algumas informações sobre os pedidos que fizeram e podem mudar os seus dados de Cliente através do site

## **2.3. Análise e validação geral dos requisitos**

Foi feita uma reunião onde a equipa de desenvolvimento, juntamente com o proprietário, José Bezos, iriam analisar os requisitos tendo em consideração a sua compreensibilidade e consistência. Concluiu-se que tudo se encontra em ordem levando à validação deste e iniciou-se o processo de desenvolvimento e implementação do sistema de base de dados.

### 3. Modelação Conceptual

#### 3.1. Apresentação da abordagem de modelação realizada

A criação do modelo conceptual, sendo a base para grande parte da base de dados, teria de ter especial atenção e cuidado, por isso foi feita enquanto o Sr. Bezos era consultado o máximo de vezes possível. Para a sua realização utilizamos o software *brModelo*, onde começamos pela identificação das entidades e os seus atributos e depois das relações entre entidades.

#### 3.2. Identificação e caracterização das entidades

Entidade	Caracterização
Cliente	Termo com informação sobre as pessoas que fizeram pelo menos uma encomenda.
Pedido	Representa as encomendas feitas pelos clientes, englobando todos os dados necessários para ser feita a sua entrega.
Estafeta	Esta entidade representa os funcionários que fazem as entregas.
Transporte	Informação relacionada com os meios de transporte usados nas entregas.
Item	Engloba todos os produtos que serão vendidos pela loja.
Administrativo	Esta entidade representa os administrativos que vão receber os pedidos.

Tabela 1

#### 3.3. Identificação e caracterização dos relacionamentos

Entidade	Relacionamento	Cardinalidade	Entidade
Pedido	Feito_por	1:N	Cliente

Tabela 2

Um cliente poderá fazer **N** pedidos e cada pedido terá apenas **1** cliente.

Pedido	Entregue_por	1:N	Estafeta
--------	--------------	-----	----------

Tabela 3

Cada estafeta pode fazer parte de **N** pedidos e cada pedido tem apenas **1** estafeta.

Pedido	Transportado_por	1:N	Transporte
--------	------------------	-----	------------

Tabela 4

Cada transporte pode fazer parte de **N** pedidos e cada pedido tem associado apenas **1** transporte.

Pedido	Inclui	N:M	Item
--------	--------	-----	------

Tabela 5

Um pedido pode ter **N** itens e cada item pode fazer parte de **M** pedidos.

Pedido	Registado_por	1:N	Administrativo
--------	---------------	-----	----------------

Tabela 6

### 3.4. Identificação e caracterização da associação dos atributos com as entidades e relacionamentos

Cliente:

Atributo	Descrição	Domínio e Tamanho	Nulo	Exemplo	Tipo
IdCliente	Chave única que identifica o cliente.	INT	N	1	chave
Nome	Nome do cliente.	VARCHAR(45)	N	André Costa	simples
Email	Email do cliente.	VARCHAR(45)	N	basedados@gmail.com	simples
NIF	Número de Identificação Fiscal do cliente.	VARCHAR(45)	S	274935503	simples
Telemóvel(1..n)	Lista de números de telemóvel do cliente.	VARCHAR(45)	N	912222222	simples

Tabela 7



Estafeta:

Atributo	Descrição	Domínio e Tamanho	Nulo	Exemplo	Tipo
IdEstafeta	Chave única que identifica o estafeta.	INT	N	1	chave
Nome	Nome do estafeta.	VARCHAR(45)	N	Ramiro Gomes	simples
Contrato	Contrato do estafeta.	VARCHAR(45)	N	900€	simples
Salário	Salário mensal do estafeta.				
Ativo	Se o contrato está ativo ou não.				
		VARCHAR(1)	N	S	simples

Tabela 8

Transporte:

Atributo	Descrição	Domínio e tamanho	Nulo	Exemplo	Tipo
Matrícula	Matrícula do veículo.	VARCHAR(6)	N	AA08AM	chave
CustoPortes	O custo de cada entrega feita pelo veículo.	VARCHAR(45)	N	5€	simples

Tabela 9

Item:

Atributo	Descrição	Domínio e tamanho	Nulo	Exemplo	Tipo
IdItem	Chave única que identifica o item.	INT	N	1	chave
Nome	Nome do item.	VARCHAR(45)	N	GeForce RTX 3060	simples
Descrição	Descrição do item.	VARCHAR(45)	N	Placa gráfica nvidia GeForce RTX 3060	simples
Preço	Preço do item.	VARCHAR(45)	N	329€	simples
Stock	Stock do item.	INT	N	5	simples

Tabela 10

Pedido:

Atributo	Descrição	Domínio e tamanho	Nulo	Exemplo	Tipo
IdPedido	Chave única que identifica o pedido.	INT	N	1	chave
Estado	Qual a situação do pedido.	VARCHAR(45)	N	Processamento	simples
DataR	Data em que o cliente recebeu o pedido.	DATETIME	S	19/10/2022 16:30	simples
DataE	Data em que o pedido foi enviado.	DATETIME	S	19/10/2022 15:30	simples
DataP	Data em que o pedido foi feito.	DATETIME	N	19/10/2022 10:40	simples
Preço	Preço do pedido.	VARCHAR(45)	N	50€	simples
EndEntrega	Endereço de entrega do pedido.		N		
Rua	Nome da rua e número da porta.	VARCHAR(45)	N	Rua da Universidade, 30	simples
Localidade	Nome da localidade.	VARCHAR(45)	N	Braga	simples
CodPostal	Código Postal.	VARCHAR(45)	N	1234-567	simples

Tabela 11

Administrativo:

Atributo	Descrição	Domínio e Tamanho	Nulo	Exemplo	Tipo
IdAdministrativo	Chave única que identifica o administrativo.	INT	N	1	chave
Nome	Nome do administrativo.	VARCHAR(45)	N	Ramiro Gomes	simples
Contrato	Contrato do administrativo.				
Salário	Salário mensal do administrativo.	VARCHAR(45)	N	900€	simples
Ativo	Se o contrato está ativo ou não.	VARCHAR(1)	N	S	simples

Tabela 12

### 3.5. Apresentação e explicação do diagrama ER produzid

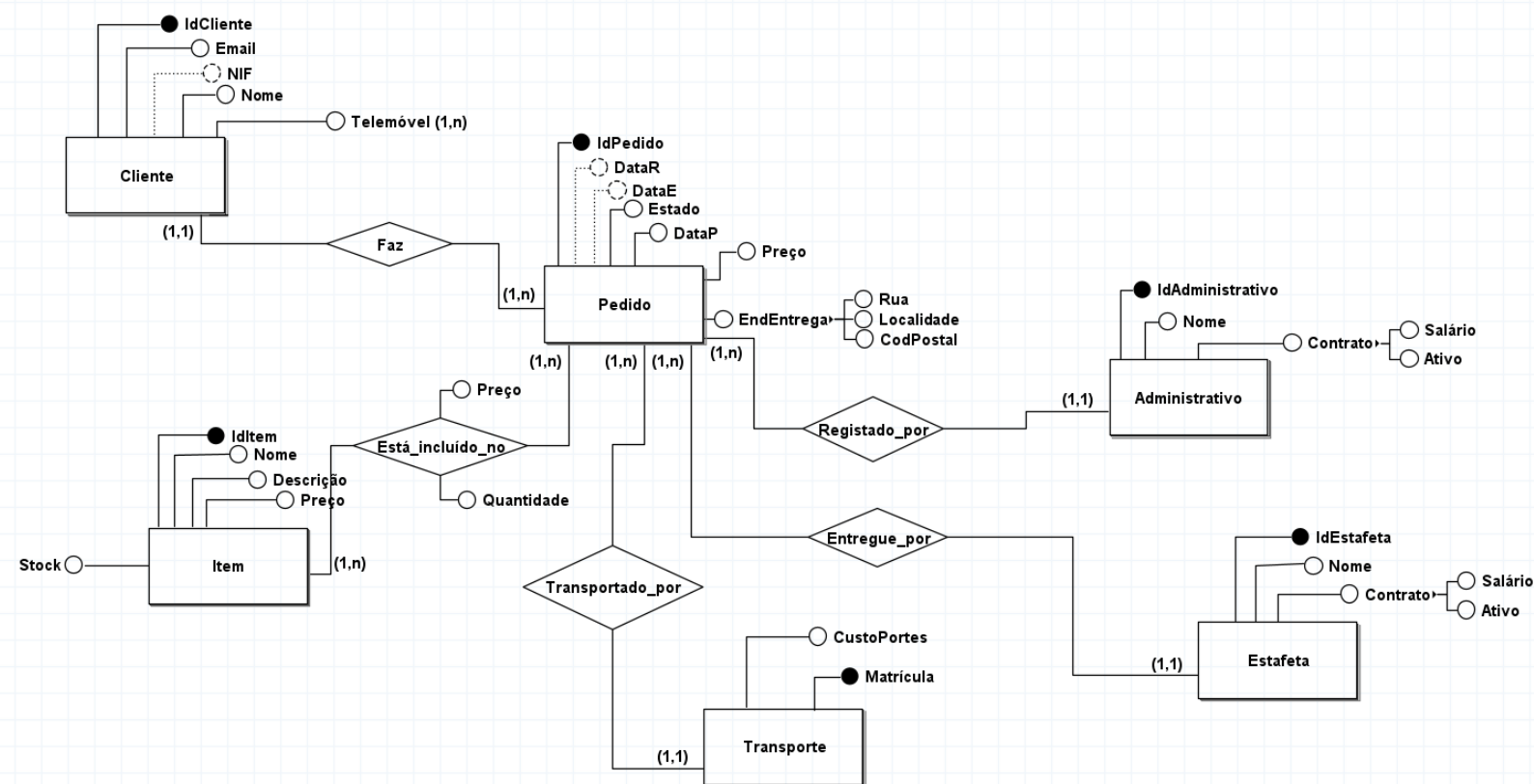


Figura 2

## 4. Modelação Lógica

### 4.1. Construção e validação do modelo de dados lógico

A partir do esquema conceptual obtido previamente, fizemos a sua conversão para esquema lógico da seguinte forma:

- **Cliente**

A entidade cliente deu origem a uma tabela com os atributos:

- idCliente
- Email
- NIF
- Nome

Com idCliente como chave primária.

E deu origem também à tabela **Telemóvel** que tem os atributos:

- NrTelemóvel
- TCliente

Com TCliente como chave estrangeira referenciando idCliente da tabela Cliente.

- **Estafeta**

A entidade estafeta deu origem a uma tabela com os atributos:

- idEstafeta
- Nome
- Salário
- Ativo

Com idEstafeta como chave primária.

- **Administrativo**

A entidade administrativo deu origem a uma tabela com os atributos:

- idAdministrativo
- Nome
- Salário
- Ativo

Com idAdministrativo como chave primária.

- **Transporte**

A entidade transporte deu origem a uma tabela com os atributos:

- Matrícula
- CustoPortes

Com Matrícula como chave primária.

- **Item**

A entidade item deu origem a uma tabela com os atributos:

- idItem
- Nome
- Descrição
- Preço
- Stock

Com idItem como chave primária.

- **Pedido**

A entidade pedido deu origem a uma tabela com os atributos:

- idPedido
- DataR
- DataE
- DataP
- Estado
- Preço
- Rua
- Localidade
- CodPostal

- PCliente
- PEstafeta
- PTransporte
- PAdministrativo

Com idPedido como chave primária e com PCliente, PEstafeta, PTransporte e PAdministrativo como chaves estrangeiras referenciando idCliente da tabela Cliente, idEstafeta da tabela Estafeta, Matrícula da tabela Transporte e idAdministrativo da tabela Administrativo respetivamente.

O relacionamento N para M entre Pedido e Item deu origem a uma nova tabela chamada de **Item\_has\_Pedido** que tem os seguintes atributos:

- IItem
- IPPedido
- Preço
- Quantidade

Com IItem e IPPedido como chaves estrangeira referenciando idItem da tabela Item e idPedido da tabela Pedido respetivamente.

## 4.2. Normalização de Dados

Como o esquema lógico apresenta um bom nível de controlo e consistência podemos concluir que este esquema se encontra normalizado até à terceira forma normal (3FN).

### 4.3. Apresentação e explicação do modelo lógico produzido

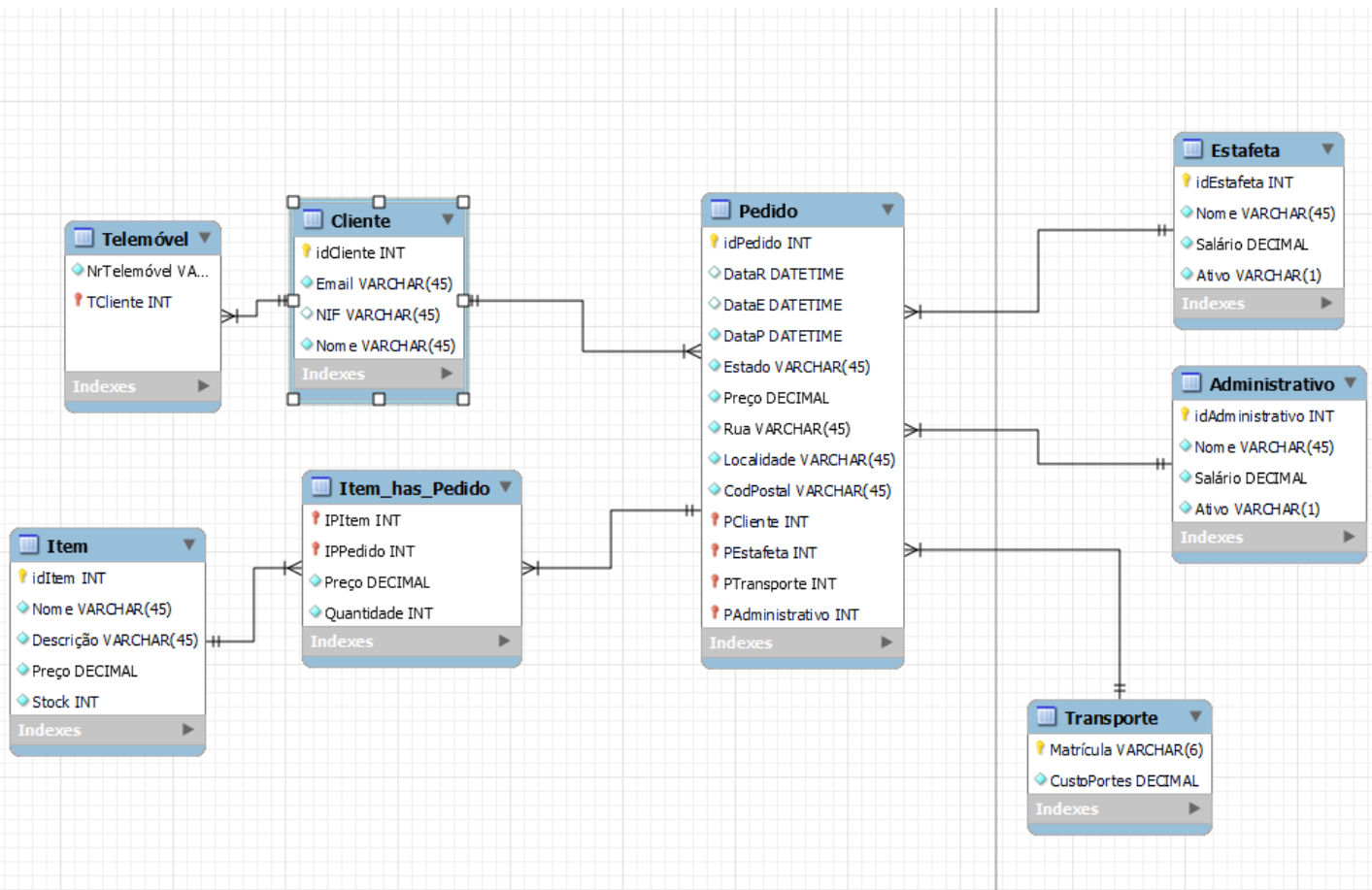
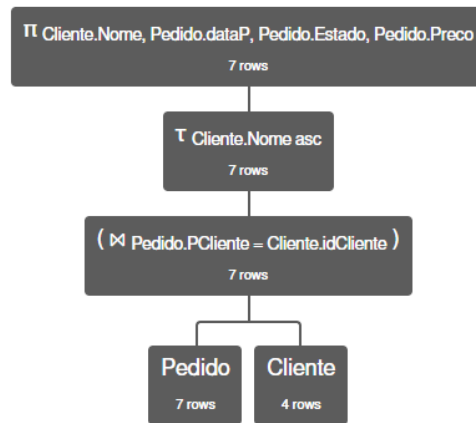


Figura 3

### 4.4. Validação do modelo com interrogações do utilizador

Para validar o nosso modelo lógico, testamos se alguns dos requisitos de exploração eram possíveis usando álgebra relacional com a ferramenta *RelaX – Relational Algebra Calculator*.

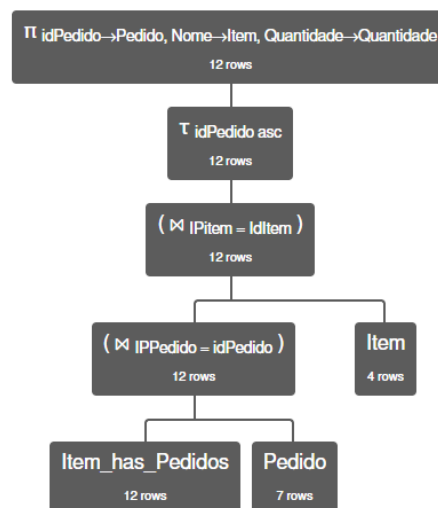
A partir do requisito de exploração “Listar o histórico de pedidos feito por cada cliente” obtemos a seguinte árvore:



$\pi_{\text{Cliente.Nome, Pedido.dataP, Pedido.Estado, Pedido.Preco}} ( \tau_{\text{Cliente.Nome asc}} ( ( \text{Pedido} ) \bowtie_{\text{Pedido.PCliente = Cliente.idCliente}} ( \text{Cliente} ) ) )$

Figura 4

A partir do requisito de exploração “Cada pedido pode ter vários itens” obtemos a seguinte árvore:



$\pi_{\text{idPedido} \rightarrow \text{Pedido, Nome} \rightarrow \text{Item, Quantidade} \rightarrow \text{Quantidade}} ( \tau_{\text{idPedido asc}} ( ( ( \text{Item\_has\_Pedidos} ) \bowtie_{\text{IPedido = idPedido}} ( \text{Pedido} ) ) \bowtie_{\text{IPitem = IdItem}} ( \text{Item} ) ) )$

Figura 5



## 5. Implementação Física

### 5.1. Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL

Para a gestão da base de dados deste projeto foi utilizado o MySQL já que, devimos à sua utilização durante as aulas, estamos de certa maneira familiarizados com este.

Na criação das tabelas na implementação física fizemos então o seguinte código SQL:

- Cliente

```
CREATE TABLE IF NOT EXISTS Cliente (  
    idCliente INT NOT NULL,  
    Email VARCHAR(45) NOT NULL,  
    Nif VARCHAR(45),  
    Nome VARCHAR(45) NOT NULL,  
    PRIMARY KEY(idCliente)  
);
```

Figura 6

- Telemóvel

```
CREATE TABLE IF NOT EXISTS Telemóvel (  
    NrTelemóvel VARCHAR(45) NOT NULL,  
    TCliente INT NOT NULL,  
    FOREIGN KEY(TCliente)  
        REFERENCES Cliente(idCliente)  
);
```

Figura 7

- Item

```
CREATE TABLE IF NOT EXISTS Item (
    idItem INT NOT NULL,
    Nome VARCHAR(45) NOT NULL,
    Descrição VARCHAR(100) NOT NULL,
    Preço DECIMAL(5,2) NOT NULL,
    Stock INT NULL,
    PRIMARY KEY(idItem)
);
```

Figura 8

- Estafeta

```
CREATE TABLE IF NOT EXISTS Estafeta (
    idEstafeta INT NOT NULL,
    Nome VARCHAR(45) NOT NULL,
    Salário DECIMAL(5,2) NOT NULL,
    Ativo VARCHAR(3) NOT NULL,
    CONSTRAINT cnt_Ativo_Estafeta CHECK (Ativo IN ('Sim','Não')),
    PRIMARY KEY(idEstafeta)
);
```

Figura 9

- Administrativo

```
CREATE TABLE IF NOT EXISTS Administrativo (
    idAdministrativo INT NOT NULL,
    Nome VARCHAR(45) NOT NULL,
    Salário DECIMAL(5,2) NOT NULL,
    Ativo VARCHAR(3) NOT NULL,
    CONSTRAINT cnt_Ativo_Administrativo CHECK (Ativo IN ('Sim','Não')),
    PRIMARY KEY(idAdministrativo)
);
```

Figura 10

- Transporte

```
CREATE TABLE IF NOT EXISTS Transporte (
    Matricula VARCHAR(8) NOT NULL,
    CustoPortes DECIMAL(5,2) NOT NULL,
    PRIMARY KEY(Matricula)
);
```

Figura 11

- Pedido

```
CREATE TABLE IF NOT EXISTS Pedido (
    idPedido INT NOT NULL,
    DataR DATETIME,
    DataE DATETIME,
    DataP DATETIME NOT NULL,
    Estado VARCHAR(45) NOT NULL,
    CONSTRAINT cnt_Estado CHECK (Estado IN ('Processamento','Enviado','Recebido')),
    Preço DECIMAL(10,2) NOT NULL,
    Rua VARCHAR(45) NOT NULL,
    Localidade VARCHAR(45) NOT NULL,
    CodPostal VARCHAR(45) NOT NULL,
    PCliente INT NOT NULL,
    PEstafeta INT NOT NULL,
    PTransporte VARCHAR(8) NOT NULL,
    PAdministrativo INT NOT NULL,
    PRIMARY KEY(idPedido),
    FOREIGN KEY (PCliente)
        REFERENCES Cliente(idCliente),
    FOREIGN KEY (PEstafeta)
        REFERENCES Estafeta(idEstafeta),
    FOREIGN KEY (PTransporte)
        REFERENCES Transporte(Matricula),
    FOREIGN KEY (PAdministrativo)
        REFERENCES Administrativo(idAdministrativo)
);
```

Figura 12

- Item\_has\_Pedido

```
CREATE TABLE IF NOT EXISTS Item_has_Pedidos (
    IPitem INT NOT NULL,
    IPPedido INT NOT NULL,
    PreçoTotal DECIMAL(6,2) NOT NULL,
    Quantidade INT NOT NULL,
    FOREIGN KEY(IPitem)
        REFERENCES Item(idItem),
    FOREIGN KEY(iPPedido)
        REFERENCES Pedido(idPedido)
);
```

Figura 13

## 5.2. Tradução das interrogações do utilizador para SQL

Neste ponto serão apresentadas queries que permitem responder a alguns requisitos de exploração e as tabelas que obteremos a partir das queries com o povoamento atual.

- “Listar o top 3 de clientes da loja”

```
SELECT Pcliente, Nome, SUM(Preço) AS Valor, COUNT(Preço) AS Numero_de_Compras
FROM Cliente
    INNER JOIN Pedido
    ON idCliente = PCliente
GROUP BY PCliente
ORDER BY Valor DESC
LIMIT 3;
```

Figura 14

Pcliente	Nome	Valor	Numero_de_Compras
2	Nuno Almeida	3086.00	2
3	Álvaro Cabral	2650.00	3
1	Carla Matos	1003.00	1

Figura 15

- “Através das datas de receção e de envio, é possível calcular o tempo de entrega”

```
SELECT IdPedido, dataP AS DataPedido, dataR AS DataReceção, CONCAT(DATEDIFF(dataR,dataP), " dias") AS TempoEntrega FROM Pedido;
```

Figura 16

IdPedido	DataPedido	DataReceção	TempoEntrega
1	2022-10-16 18:30:00	2022-10-19 16:30:00	3 dias
2	2022-10-13 18:30:00	2022-10-15 16:30:00	2 dias
3	2022-11-21 20:30:00	NULL	NULL
4	2022-12-03 20:30:00	NULL	NULL
5	2022-10-14 10:23:00	2022-10-18 14:21:00	4 dias
6	2022-12-15 15:59:00	2022-12-17 16:30:00	2 dias
7	2022-12-10 19:32:00	2022-12-13 18:43:00	3 dias

Figura 17

- “Listar o histórico de pedidos feito por cada cliente”

```
SELECT IdCliente, Nome, IdPedido, DataP AS DataPedido, Estado, Preço
FROM Cliente
INNER JOIN Pedido
ON PCliente = idCliente
ORDER BY IdCliente;
```

Figura 18

IdCliente	Nome	IdPedido	DataPedido	Estado	Preço
1	Carla Matos	1	2022-10-16 18:30:00	Recebido	1003.00
2	Nuno Almeida	2	2022-10-13 18:30:00	Recebido	1802.00
2	Nuno Almeida	5	2022-10-14 10:23:00	Recebido	1284.00
3	Álvaro Cabral	3	2022-11-21 20:30:00	Enviado	1744.00
3	Álvaro Cabral	6	2022-12-15 15:59:00	Recebido	103.00
3	Álvaro Cabral	7	2022-12-10 19:32:00	Recebido	803.00
4	Lídia Costa	4	2022-12-03 20:30:00	Processamento	203.00

Figura 19

- “Listar os funcionários que estiveram envolvidos em cada pedido”

```

SELECT idPedido,idAdministrativo,Administrativo.Nome AS NomeAdministrativo,
idEstafeta, Estafeta.Nome AS NomeEstafeta,
Matricula AS Veiculo FROM Pedido,Estafeta,Administrativo,Transporte
WHERE idEstafeta=PEstafeta AND
idAdministrativo=PAdministrativo AND
Matricula=PTransporte
ORDER BY idPedido;

```

Figura 20

idPedido	idAdministrativo	NomeAdministrativo	idEstafeta	NomeEstafeta	Veiculo
1	1	Joao	1	Vasco	AA-12-LK
2	2	Ricardo	2	Marta	VR-32-BA
3	3	Ruben	3	André	AE-21-FF
4	4	Martim	4	Renato	XD-76-XG
5	3	Ruben	3	André	AE-21-FF
6	3	Ruben	4	Renato	XD-76-XG
7	4	Martim	1	Vasco	GG-13-AJ

Figura 21

- “Através das datas em que os pedidos são feitos, é possível calcular o período de maior atividade”

```
SELECT CONCAT(HOUR(DataP), ":00") AS HoraDeMaiorAtividade FROM Pedido  
GROUP BY HOUR(DataP)  
ORDER BY COUNT(DataP) DESC  
LIMIT 1;
```

Figura 22

HoraDeMaiorAtividade
18:00

Figura 23

## 5.3. Definição e caracterização das vistas de utilização em SQL

Vista que mostra a informação completa de todos os pedidos já feitos

```
CREATE VIEW InfoGeral
AS
SELECT IdCliente AS "Id do Cliente",
       SUBSTRING_INDEX(Cliente.Nome, " ", 1) AS 'Nome',
       SUBSTRING(Cliente.Nome, (LENGTH(SUBSTRING_INDEX(Cliente.Nome, " ", 1)) + 1)) AS 'Apelido',
       Cliente.Email,
       Cliente.Nif,
       IdPedido AS 'Id do Pedido',
       dataP AS 'Data do Pedido',
       dataE AS 'Data do Envio',
       dataR AS 'Data da Receção',
       Estado,
       Preço,
       Rua,
       Localidade,
       CodPostal AS 'Código Postal',
       IdEstafeta AS 'Id do Estafeta',
       Estafeta.Nome AS 'Nome Estafeta',
       Estafeta.Salário AS 'Salário Estafeta',
       Estafeta.Ativo AS 'Estafeta Ativo?',
       Matricula,
       IdAdministrativo AS 'Id do Administrativo',
       Administrativo.Nome AS 'Nome Adminsitrativo',
       Administrativo.Salário AS 'Salário Administrativo',
       Administrativo.Ativo AS 'Administrativo Ativo?'

FROM Cliente
INNER JOIN Pedido
ON idCliente = PCliente
    INNER JOIN Estafeta
    ON idEstafeta = PEstafeta
        INNER JOIN Transporte
        ON Matricula = PTransporte
            INNER JOIN Administrativo
            ON idAdministrativo = PAdministrativo;
```

Figura 24



Id do Cliente	Nome	Apelido	Email	Nif	Id do Pedido	Data do Pedido	Data do Envio	Data da Receção	Estado	Preço	Rua
1	Carla	Matos	cm@gmail.com	236544365	1	2022-10-16 18:30:00	2022-10-19 15:00:00	2022-10-19 16:30:00	Recebido	1003.00	Rua da Universidade
2	Nuno	Almeida	na@gmail.com	214228355	2	2022-10-13 18:30:00	2022-10-15 15:00:00	2022-10-15 16:30:00	Recebido	1802.00	Rua deidadeidade Gualtar
2	Nuno	Almeida	na@gmail.com	214228355	5	2022-10-14 10:23:00	2022-10-18 14:00:00	2022-10-18 14:21:00	Recebido	1284.00	Rua da Liberdade
3	Álvaro	Cabral	ac@gmail.com	210579820	3	2022-11-21 20:30:00	2022-11-22 15:00:00	NULL	Enviado	1744.00	Rua Fernando Pessoa
3	Álvaro	Cabral	ac@gmail.com	210579820	6	2022-12-15 15:59:00	2022-12-17 16:03:00	2022-12-17 16:30:00	Recebido	103.00	Rua da Saudade
3	Álvaro	Cabral	ac@gmail.com	210579820	7	2022-12-10 19:32:00	2022-12-13 18:00:00	2022-12-13 18:43:00	Recebido	803.00	Rua do Pombal
4	Lidia	Costa	lc@gmail.com	285347276	4	2022-12-03 20:30:00	NULL	NULL	Processamento	203.00	Rua Alberto Caeiro

Figura 25

Localidade	Código Postal	Id do Estafeta	Nome Estafeta	Salário Estafeta	Estafeta Ativo?	Matricula	Id do Administrativo	Nome Administrativo	Salário Administrativo	Administrativo Ativo?
Braga	1111-111	1	Vasco	700.00	Sim	AA-12-LK	1	Joao	650.00	Sim
Braga	2222-222	2	Marta	750.00	Sim	VR-32-BA	2	Ricardo	800.00	Sim
Guimarães	4835-256	3	André	700.00	Sim	AE-21-FF	3	Ruben	700.00	Sim
Braga	3333-333	3	André	700.00	Sim	AE-21-FF	3	Ruben	700.00	Sim
Guimarães	4835-321	4	Renato	600.00	Sim	XD-76-XG	3	Ruben	700.00	Sim
Guimarães	4835-275	1	Vasco	700.00	Sim	GG-13-AJ	4	Martim	750.00	Sim
Braga	4444-444	4	Renato	600.00	Sim	XD-76-XG	4	Martim	750.00	Sim

Figura 26

## 5.4. Cálculo do espaço da bases de dados

O primeiro passo para se ter uma estimativa do espaço ocupado pela nossa base de dados, temos de ter em conta o tamanho em bytes de cada tipo de dados que utilizamos.

A partir de uma pesquisa rápida descobrimos que dados do tipo Int ocupam 4 bytes, Datetime 5 bytes, Decimal 8 bytes e Varchar(N) N + 2 bytes.

Tendo em conta estes valores, podemos atribuir um total de bytes que são utilizados por cada atributo e a partir daí por cada entidade e então:

Cada entrada de Cliente ocupará  $4 + (45 + 2) * 3 = 145$  bytes

Cada entrada de Telemóvel ocupará  $45 + 2 + 4 = 51$  bytes

Cada entrada de Item ocupará  $4 + 45 + 2 + 100 + 2 + 8 + 4 = 165$  bytes

Cada entrada de Estafeta ocupará  $4 + 45 + 2 + 8 + 3 + 2 = 64$  bytes

Cada entrada de Administrativo ocupará 64 bytes

Cada entrada de Transporte ocupará  $8 + 2 + 8 = 18$  bytes

Cada entrada de Pedido ocupará  $4 + 5 * 3 + (45 + 2) * 4 + 8 + 4 + 4 + 8 + 2 + 4 = 237$  bytes

Cada entrada de Item\_has\_Pedidos ocupará  $4 + 4 + 8 + 4 = 20$  bytes

Agora mesmo, com o povoamento atual, a base de dados ocupa um total de 4225 bytes.

Fazendo a projeção do próximo ano, como sabemos que por mês são feitos em média 60 pedidos, cada pedido tem em média 2 itens, cada cliente faz cerca de 2 pedidos, em média cada cliente tem 1 número de telemóvel, normalmente é adicionado 1 item novo a cada 2 semanas e a loja não tem interesse em contratar mais funcionários nem comprar mais veículos então em 1 ano a base de dados da loja do Sr.Bezos ocupará cerca de 326185 bytes.

## **5.5. Plano de segurança e recuperação de dados**

É extremamente importante estar preparado e pensar sempre num plano para restaurar os dados caso estes sejam perdidos de alguma maneira.

Para manter este projeto seguro, para além de termos 4 cópias guardadas em 4 computadores diferentes, também há um backup guardado num armazenamento externo.

## 6. Conclusões e Trabalho Futuro

Ao longo do desenvolvimento da base de dados que apresentamos neste relatório, surgiram, naturalmente, dificuldades. Algumas que vieram se mostrar como erros nossos, outras como falta de atenção, mas que no fim foram ultrapassadas, usando o checkpoint a nosso favor e retirando dúvidas com os docentes de Bases de Dados.

Uma dessas dificuldades e a que achamos a mais crucial para a evolução da base de dados, foi sem dúvida o levantamento de requisitos. É uma fase que dita o rumo do trabalho e que requiere um brainstorming por parte da equipa de desenvolvimento, o que nos deixava com questões como, "O que queremos fazer?", "É prático?", "Como implementamos isso no MySQL?", entre outras. E pode-se afirmar que esta fase foi a que mais nos retirou tempo livre. Além disso, outra dificuldade um pouco menos importante seria, validar o modelo lógico com os requisitos que se tinha estabelecido, porém foi só uma questão de entender o próprio conceito de validação.

No que toca a melhoramentos, evidentemente temos ainda muito por onde pegar.

Podíamos melhorar tabelas como a tabela "Transporte", adicionando atributos com o intuito de dar mais utilidade à tabela, que conseqüentemente aumentaria a variedade de queries. Implementar tabelas novas com a finalidade de melhorar a organização do sistema ou talvez transformar certos atributos que fariam mais sentido sendo a sua própria tabela.

Ou até colocar mais funções e triggers, para enriquecer a automação o existente sistema de base de dados.

Não obstante, concluímos que fomos ao encontro dos objetivos que se estipularam para a realização deste trabalho, visto que a base de dados que desenvolvemos providencia tudo o que a empresa do Sr. José Bezos necessita para ser mais eficiente.

## Referências

Gupta, R. (2019, December 9). An overview of SQL Server data types. SQL Shack - Articles About Database Auditing, Server Performance, Data Recovery, and More. <https://www.sqlshack.com/an-overview-of-sql-server-data-types/>

Indivíduos com 16 e mais anos que utilizam computador e Internet em % do total de indivíduos: por grupo etário. (n.d.). <https://www.pordata.pt/portugal/indivíduos+com+16+e+mais+anos+que+utilizam+computador+e+internet+em+percentagem+do+total+de+indivíduos+por+grupo+etario-1139>

Johannes, K. (n.d.). Relax - relational algebra calculator. <https://dbis-uibk.github.io/relax/>

## **Lista de Siglas e Acrónimos**

SGBD - Sistema de Gestão de Bases de Dados

FN - Forma Normal

BD - Base de Dados

## **Anexos**

## I. Criação de tabelas

```
DROP SCHEMA IF EXISTS Dom;
```

```
CREATE SCHEMA IF NOT EXISTS Dom;
```

```
USE Dom;
```

```
CREATE TABLE IF NOT EXISTS Cliente (
```

```
    idCliente INT NOT NULL,
```

```
    Email VARCHAR(45) NOT NULL,
```

```
    Nif VARCHAR(45),
```

```
    Nome VARCHAR(45) NOT NULL,
```

```
    PRIMARY KEY(idCliente)
```

```
);
```

```
CREATE TABLE IF NOT EXISTS Telemóvel (
```

```
    NrTelemóvel VARCHAR(45) NOT NULL,
```

```
    TCliente INT NOT NULL,
```

```
    FOREIGN KEY(TCliente)
```

```
        REFERENCES Cliente(idCliente)
```

```
);
```

```
CREATE TABLE IF NOT EXISTS Item (
```

```
    idItem INT NOT NULL,
```

```
    Nome VARCHAR(45) NOT NULL,
```

```
    Descrição VARCHAR(100) NOT NULL,
```

```
    Preço DECIMAL(5,2) NOT NULL,
```

```
    Stock INT NULL,
```

```
    PRIMARY KEY(idItem)
```

```
);
```

```
CREATE TABLE IF NOT EXISTS Estafeta (
```

```
    idEstafeta INT NOT NULL,
```

```
    Nome VARCHAR(45) NOT NULL,
```

```
    Salário DECIMAL(5,2) NOT NULL,
```

```
    Ativo VARCHAR(3) NOT NULL,
```

```

        CONSTRAINT cnt_Ativo_Estafeta CHECK (Ativo IN ('Sim','Não')),
        PRIMARY KEY(idEstafeta)
);

CREATE TABLE IF NOT EXISTS Administrativo (
    idAdministrativo INT NOT NULL,
        Nome VARCHAR(45) NOT NULL,
        Salário DECIMAL(5,2) NOT NULL,
        Ativo VARCHAR(3) NOT NULL,
        CONSTRAINT cnt_Ativo_Administrativo CHECK (Ativo IN ('Sim','Não')),
        PRIMARY KEY(idAdministrativo)
);

CREATE TABLE IF NOT EXISTS Transporte (
    Matricula VARCHAR(8) NOT NULL,
        CustoPortes DECIMAL(5,2) NOT NULL,
        PRIMARY KEY(Matricula)
);

CREATE TABLE IF NOT EXISTS Pedido (
    idPedido INT NOT NULL,
        DataR DATETIME,
        DataE DATETIME,
        DataP DATETIME NOT NULL,
        Estado VARCHAR(45) NOT NULL,
        CONSTRAINT cnt_Estado CHECK (Estado IN ('Processamento','Enviado','Recebido')),
        Preço DECIMAL(10,2) NOT NULL,
        Rua VARCHAR(45) NOT NULL,
        Localidade VARCHAR(45) NOT NULL,
        CodPostal VARCHAR(45) NOT NULL,
        PCliente INT NOT NULL,
        PEstafeta INT NOT NULL,
        PTransporte VARCHAR(8) NOT NULL,
        PAdministrativo INT NOT NULL,
        PRIMARY KEY(idPedido),
        FOREIGN KEY (PCliente)
            REFERENCES Cliente(idCliente),
        FOREIGN KEY (PEstafeta)
            REFERENCES Estafeta(idEstafeta),
        FOREIGN KEY (PTransporte)
            REFERENCES Transporte(Matricula),

```



```
        FOREIGN KEY (PAdministrativo)
            REFERENCES ADministrativo(idAdministrativo)
    );
```

```
CREATE TABLE IF NOT EXISTS Item_has_Pedidos (
    IPitem INT NOT NULL,
        IPPedido INT NOT NULL,
    PreçoTotal DECIMAL(6,2) NOT NULL,
    Quantidade INT NOT NULL,
        FOREIGN KEY(IPitem)
            REFERENCES Item(idItem),
        FOREIGN KEY(iPPedido)
            REFERENCES Pedido(idPedido)
    );
```

```
Set SQL_SAFE_Updates=1;
```

## II. Povoamento

USE Dom;

INSERT INTO Administrativo(idAdministrativo, Nome, Salário, Ativo)

VALUES (1,'Joao',650,'Sim'),(2,'Ricardo',800,'Sim'),(3,'Ruben',700,'Sim'),(4,'Martim',750,'Sim');

INSERT INTO Estafeta(idEstafeta, Nome, Salário, Ativo)

VALUES

(1,'Vasco',700,'Sim'),(2,'Marta',750,'Sim'),(3,'André',700,'Sim'),(4,'Renato',600,'Sim'),(5,'Carolina',800,'Sim');

INSERT INTO Transporte(Matricula, CustoPortes)

VALUES ('AA-12-LK',3),('VR-32-BA',2),('AE-21-FF',4),('XD-76-XG',3),('GG-13-AJ',3);

INSERT INTO Cliente(idCliente, Email, NIF, Nome)

VALUES (1,'cm@gmail.com',236544365,'Carla Matos'),(2,'na@gmail.com',214228355,'Nuno Almeida'),(3,'ac@gmail.com',210579820,'Álvaro Cabral'),(4,'lc@gmail.com',285347276,'Lídia Costa');

INSERT INTO Telemóvel(NrTelemóvel, TCliente)

VALUES ('911111111',1),('912222222',1),('911212122',2),('913333333',3),('917777777',4);

INSERT INTO Item(IdItem, Nome, Descrição, Preço, Stock)

VALUES (1,"GeForce GTX 1080","Placa gráfica NVIDIA - 8GB DDR5",600,20),(2,"IntelCore i7 6500","Processador IntelCore da 7ª geração",400,20),(3,"Memória RAM 16GB","DDR4-3200MHz",40,20),  
(4,"Motherboard B550M DS3H","Suporte para a 3ª Geração AMD Ryzen",100,20),(5,"Monitor 24 polegadas","144Hz, 1ms",250,20);

```

INSERT INTO
Pedido(idPedido,dataR,dataE,dataP,Estado,Preço,Rua,Localidade,CodPostal,PCliente,PEstafeta,PTranspo
rte,PAdministrativo)
VALUES (1,'2022-10-19 16:30','2022-10-19 15:00','2022-10-16 18:30','Recebido',0.0,'Rua da
Universidade','Braga','1111-111',1,1,'AA-12-LK',1),
(2,'2022-10-15 16:30','2022-10-15 15:00','2022-10-13 18:30','Recebido',0.0,'Rua
deidadeidade Gualtar','Braga','2222-222',2,2,'VR-32-BA',2),
(3,NULL,'2022-11-22 15:00','2022-11-21 20:30','Enviado',0.0,'Rua Fernando Pessoa','Braga','3333-
333',3,3,'AE-21-FF',3),
(4,NULL,NULL,'2022-12-03 20:30','Processamento',0.0,'Rua Alberto Caeiro','Braga','4444-
444',4,4,'XD-76-XG',4),
(5,'2022-10-18 14:21','2022-10-18 14:00','2022-10-14 10:23','Recebido',0.0,'Rua da
Liberdade','Guimarães','4835-256',2,3,'AE-21-FF',3),
(6,'2022-12-17 16:30','2022-12-17 16:03','2022-12-15 15:59','Recebido',0.0,'Rua da
Saude','Guimarães','4835-321',3,4,'XD-76-XG',3),
(7,'2022-12-13 18:43','2022-12-13 18:00','2022-12-10 19:32','Recebido',0.0,'Rua do
Pombal','Guimarães','4835-275',3,1,'GG-13-AJ',4);

```

```

INSERT INTO Item_has_Pedidos(IPitem,IPPedido,PreçoTotal,Quantidade)

```

```

VALUES
(1,1,0.0,1),(2,1,0.0,1),(1,2,0.0,3),(1,3,0.0,2),(2,3,0.0,1),(3,3,0.0,1),(4,3,0.0,1),(4,4,0.0,2),(1,5,0.0,2),(3,5,0.
0,2),(4,6,0.0,1),(2,7,0.0,2);

```

```

-- Atribuição dos preços ao Item_has_Pedidos e aos Pedidos

```

```

SET SQL_SAFE_UPDATES = 0;

```

```

CALL UpdateAllIHP();

```

```

CALL UpdateAllIP();

```

```

CALL UpdateAllStock();

```

### III. Queries, Vistas, Utilizadores e Procedimentos

/\*-----\*/

/\*TOP 3 de clientes da loja \*/

```
SELECT  Pcliente, Nome, SUM(Preço) AS Valor, COUNT(Preço) AS Numero_de_Compras FROM
Cliente, Pedido WHERE idCliente=Pcliente
        GROUP BY Pcliente
        ORDER BY Valor DESC
        LIMIT 3;
```

```
SELECT Pcliente, Nome, SUM(Preço) AS Valor, COUNT(Preço) AS Numero_de_Compras
        FROM Cliente
                INNER JOIN Pedido
        ON idCliente = Pcliente
        GROUP BY Pcliente
        ORDER BY Valor DESC
        LIMIT 3;
```

/\*Pedidos feitos em outubro de 2022 \*/

```
SELECT  Cliente.Nome, Pedido.Pcliente AS IdCliente, Pedido.IdPedido, DataP AS DataeHora FROM
Pedido, Cliente
        WHERE      MONTH(Pedido.DataP)="10"      AND      YEAR(Pedido.DataP)="2022"      AND
Pedido.Pcliente=Cliente.idCliente;
```

/\*Produto mais vendido\*/

```
SELECT IPitem AS IdItem, Nome, SUM(Quantidade)AS Quantidade FROM Item_has_Pedidos, Item WHERE
IPitem=idItem
```

```

GROUP BY IPitem
ORDER BY Quantidade DESC
LIMIT 1;

```

/\*Funcionários que estiveram envolvidos em cada pedido\*/

```

SELECT idPedido,idAdministrativo,Administrativo.Nome AS NomeAdministrativo,
idEstafeta, Estafeta.Nome AS NomeEstafeta,
Matricula AS Veiculo FROM Pedido,Estafeta,Administrativo,Transporte
WHERE idEstafeta=PEstafeta AND
idAdministrativo=PAdministrativo AND
Matricula=PTransporte
ORDER BY idPedido;

```

/\*Administrativo que registou menos pedidos\*/

```

SELECT idAdministrativo,Administrativo.Nome,COUNT(Administrativo.Nome) AS NumerodePedidos
FROM Administrativo,Pedido
WHERE PAdministrativo=idAdministrativo
GROUP BY idAdministrativo
ORDER BY NumerodePedidos
LIMIT 1;

```

/\*Todos os itens que fazem parte do pedido 3\*/

```

SELECT IPitem AS IdItem, Item.Nome, Descrição, Item.Preço
FROM Item_has_Pedidos
INNER JOIN Item
ON IPPedido = '3' AND IPitem=idItem;

```

/\*Através das datas de receção e de envio, é possível calcular o tempo de entrega.\*/

```

SELECT IdPedido, dataP AS DataPedido, dataR AS DataReceção, CONCAT(DATEDIFF(dataR,dataP), " dias")
AS TempoEntrega FROM Pedido;

```

/\*Cada cliente tem 1 ou mais números de telemóvel a si associados\*/

```

SELECT IdCliente, Nome, NrTelemóvel AS Números
      FROM Cliente
            INNER JOIN Telemóvel
                  ON TCliente = idCliente
            ORDER BY IdCliente;

```

/\*Listar o histórico de pedidos feito por cada cliente\*/

```

SELECT IdCliente, Nome, IdPedido, DataP AS DataPedido, Estado, Preço
      FROM Cliente
            INNER JOIN Pedido
                  ON PCliente = idCliente
            ORDER BY IdCliente;

```

/\*Através das datas em que os pedidos são feitos, é possível calcular o período de maior atividade.\*/

```

SELECT CONCAT(HOUR(DataP), ":00") AS HoraDeMaiorAtividade FROM Pedido
      GROUP BY HOUR(DataP)
      ORDER BY COUNT(DataP) DESC
      LIMIT 1;

```

/\*-----\*/

/\*Vista que mostra a informação completa de todos os pedidos já feitos\*/

```

CREATE VIEW InfoGeral
AS
      SELECT IdCliente AS "Id do Cliente",
            SUBSTRING_INDEX(Cliente.Nome, " ", 1) AS 'Nome',
            SUBSTRING(Cliente.Nome,(LENGTH(SUBSTRING_INDEX(Cliente.Nome, " ", 1)) + 1)) AS 'Apelido',
            Cliente.Email,
            Cliente.Nif,
            IdPedido AS 'Id do Pedido',
            dataP AS 'Data do Pedido',
            dataE AS 'Data do Envio',
            dataR AS 'Data da Receção',

```

```

Estado,
Preço,
Rua,
Localidade,
CodPostal AS 'Código Postal',
IdEstafeta AS 'Id do Estafeta',
Estafeta.Nome AS 'Nome Estafeta',
Estafeta.Salário AS 'Salário Estafeta',
Estafeta.Ativo AS 'Estafeta Ativo?',
Matricula,
IdAdministrativo AS 'Id do Administrativo',
Administrativo.Nome AS 'Nome Adminsitrativo',
Administrativo.Salário AS 'Salário Administrativo',
Administrativo.Ativo AS 'Administrativo Ativo?'

FROM Cliente
        INNER JOIN Pedido
        ON idCliente = PCliente
                INNER JOIN Estafeta
                ON idEstafeta = PEstafeta
                        INNER JOIN Transporte
                        ON Matricula = PTransporte
                                INNER JOIN Administrativo
                                ON idAdministrativo = PAdministrativo;

DROP VIEW InfoGeral;

SELECT *
        FROM InfoGeral;

/*-----*/

/*Utilizadores*/

-- Admin

CREATE USER 'admin'@'localhost';
SET PASSWORD FOR 'admin'@'localhost' = 'admin12';

```

```

GRANT ALL ON * TO 'admin'@'localhost';

-- Estaf

CREATE USER 'estaf'@'localhost';
SET PASSWORD FOR 'estaf'@'localhost' = 'estaf12';

GRANT SELECT, INSERT, UPDATE ON Pedido TO 'estaf'@'localhost';

-- User

CREATE USER 'user'@'localhost';
SET PASSWORD FOR 'user'@'localhost' = 'user12';

GRANT SELECT, INSERT ON Cliente TO 'user'@'localhost';

GRANT SELECT (DataP, DataE, DataR)
      ON Pedido TO 'user'@'localhost';

SELECT *
      FROM mysql.user;

SELECT User, Host
      FROM mysql.user
      ORDER BY User;

/*-----*/

/*Procedimentos*/

-- Atualiza os preços de todos os item_has_pedidos

DELIMITER $$
CREATE PROCEDURE UpdateAllIHP()
BEGIN
UPDATE Item_has_Pedidos
      SET PreçoTotal = (SELECT Preço FROM Item WHERE IdItem=IPitem) * Quantidade;
END $$

```



-- Atualiza os preços associados a um pedido em item\_has\_pedidos

DELIMITER \$\$

CREATE PROCEDURE UpdateOneIHP(IN Id INT)

BEGIN

UPDATE Item\_has\_Pedidos

SET PreçoTotal = (SELECT Preço FROM Item WHERE IdItem=IPitem) \* Quantidade WHERE  
IPPedido = Id;

END \$\$

DROP PROCEDURE UpdateOneIHP

CALL UpdateOneIHP(1)

-- Atualiza os preços de todos os pedidos

DELIMITER \$\$

CREATE PROCEDURE UpdateAllP()

BEGIN

UPDATE Pedido

SET Preço = (SELECT SUM(PreçoTotal) + CustoPortes FROM Item\_has\_Pedidos, Transporte  
WHERE IPPedido = idPedido AND PTransporte = Matricula);

END \$\$

-- Atualiza os preços de 1 pedido

DELIMITER \$\$

CREATE PROCEDURE UpdateOneP(IN Id INT)

BEGIN

UPDATE Pedido

SET Preço = (SELECT SUM(PreçoTotal) + CustoPortes FROM Item\_has\_Pedidos, Transporte  
WHERE IPPedido = idPedido AND PTransporte = Matricula) WHERE idPedido = Id;

END \$\$

-- Atualiza o stock de todos os itens

```

DELIMITER $$
CREATE PROCEDURE UpdateAllStock()
BEGIN
UPDATE Item
        SET Stock = (SELECT IF (IPitem = IdItem IS NOT NULL, (Stock - SUM(Quantidade)),Stock) FROM
Item_has_Pedidos WHERE IPitem = IdItem);
END $$

```

-- Atualiza o stock de 1 item

```

DELIMITER $$
CREATE PROCEDURE UpdateOneStock(IN Id INT)
BEGIN
UPDATE Item
        SET Stock = (SELECT IF (IPitem = IdItem IS NOT NULL, (Stock - SUM(Quantidade)),Stock) FROM
Item_has_Pedidos WHERE IPitem = IdItem AND IdItem = Id);
END $$

```

-- Atualiza o estado de 1 pedido

```

DELIMITER $$
CREATE PROCEDURE UpdateEstado(IN NewEstado VARCHAR(45), IN Id INT)
BEGIN
UPDATE Pedido
        SET Estado = NewEstado WHERE idPedido = Id;
UPDATE Pedido
        SET DataE = (SELECT IF (STRCMP(NewEstado, 'Enviado')=0, now(),DataE) WHERE idPedido = Id) WHERE
idPedido = Id;
UPDATE Pedido
        SET DataR = (SELECT IF (STRCMP(NewEstado, 'Recebido')=0, now(),DataR) WHERE idPedido = Id)
WHERE idPedido = Id;
END $$

```