



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

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

Ano Letivo de 2022/2023

### **Gestão de Entregas a Domicílio**

**Francisco Teófilo (93741), Pedro Gomes (91647),**

**Pedro Martins (91681), Rui Gonçalves (91652)**

**Grupo 14**

Janeiro, 2023

**BD**

Data de Recepção	
Responsável	
Avaliação	
Observações	

## Gestão de Entregas a Domicílio

**Francisco Teófilo (93741), Pedro Gomes (91647),  
Pedro Martins (91681), Rui Gonçalves (91652)**

Janeiro, 2023

## Resumo

Neste relatório é descrito o processo de implementação de uma base de dados para um estabelecimento com entregas a domicílio, permitindo facilitar a gestão do mesmo. O estabelecimento em questão é do ramo comercial, ou seja, uma loja onde existem vendas e entregas de produtos tecnológicos. Esta tem cada vez menos afluência de pessoas ao espaço físico e maior recorrência ao método de encomenda de produtos online e respetiva entrega ao domicílio devido à sua praticidade e comodidade. Desta forma, é necessário organizar de forma mais eficiente todos os dados deste processo, a fim de gerir bem este método.

Ao longo do relatório, são descritas todas as fases de implementação da base de dados, desde o seu planeamento até à sua implementação física.

Previamente, começamos por expor o contexto do problema em questão, seguido da sua fundamentação e da delineação dos seus objetivos que se pretendem cumprir e posteriormente certificar.

De seguida, é apresentado o levantamento e análise dos requisitos selecionados através das entrevistas à dona Graça, a proprietária do estabelecimento.

Concluído este procedimento, elaboramos o modelo concetual que, após estar definido, é transposto para o respetivo modelo lógico, implementado com recurso ao MySQL Workbench, que devido à sua versatilidade serviu também para a geração e implementação física da base de dados em questão.

Finalmente, calculamos o espaço ocupado pela base de dados e estimamos a forma como esse valor vai evoluir.

**Área de Aplicação:** Elaboração e implementação de uma Base de Dados para Gestão de Entregas ao Domicílio.

**Palavras-Chave:** Base de Dados, Modelo Conceptual, Modelo Lógico, Modelo Físico, *MySQL*, *SQL*.

# Índice

Resumo	i
Índice	ii
Índice de Figuras	iv
Índice de Tabelas	vi
1. Definição do Sistema	1
1.1. Contexto de aplicação do sistema	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 e Equipa de Trabalho	2
1.6. Plano de execução do Projeto	3
2. Levantamento e Análise de Requisitos	4
2.1. Método de levantamento e de 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/manipulação	5
2.2.3. Requisitos de controlo	5
2.3. Análise e validação dos requisitos	6
3. Modelo 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	8
3.4. Identificação e caracterização das associações dos atributos com as entidades e relacionamentos	9
3.5. Apresentação e explicação do diagrama ER	11
3.6. Validação do Modelo Conceptual	11
4. Modelo Lógico	12
4.1. Construção e validação do modelo de dados lógico	12
4.2. Desenho do modelo lógico	13
4.3. Validação do modelo com interrogações do utilizador	13
5. Implementação Física	7

5.1. Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL	7
5.2. Tradução das interrogações do utilizador para SQL	18
5.3. Definição e caracterização das vistas de utilização em SQL (alguns exemplos)	22
5.4. Cálculo do espaço da base de dados (inicial e taxa de crescimento anual)	24
5.5. Procedimentos Implementados	28
5.6. Plano de segurança e recuperação de dados	29
6. Conclusões e Trabalho Futuro	30
7. Referências	31
8. Lista de Siglas e Acrónimos	32
Anexos	33
I. Anexo 1 – <i>Script</i> da Criação da Base de Dados	34
II. Anexo 2 – <i>Script</i> do Povoamento	38
III. Anexo 3 – <i>Script</i> da Implementação das Interrogações	42
IV. Anexo 4 – <i>Script</i> da Criação das Views	45
V. Anexo 4 – <i>Script</i> da Criação do Procedimento	46

# Índice de Figuras

Figura 1 - Relação entre Cliente e Encomenda	8
Figura 2 - Relação entre Encomenda e Funcionário	8
Figura 3 - Relação entre Encomenda e Produto	8
Figura 4 - Relação entre Produto e Fornecedor	9
Figura 5 - Modelo Conceptual	11
Figura 6 - Modelo Lógico	13
Figura 7 - Árvore representativa da RE02	5
Figura 8 - Árvore representativa da RE03	5
Figura 9 - Árvore representativa da RE05	5
Figura 10 - Árvore representativa da RE08	6
Figura 11 - Código SQL para criar tabela Cliente	7
Figura 12 - Código SQL para criar tabela Fornecedor	7
Figura 13 - Código SQL para criar tabela Funcionario	17
Figura 14 - Código SQL para criar tabela Produto	17
Figura 15 - Código SQL para criar tabela Encomenda	17
Figura 16 - Código SQL para criar tabela Produto_Fornecedor	17
Figura 17 - Código SQL para criar tabela Encomenda_tem_Produto	17
Figura 18 - Código SQL para RE01	18
Figura 19 - Resultado da Query relativa à RE01	18
Figura 20 - Código SQL para RE02	18
Figura 21 - Resultado da Query relativa à RE02	18
Figura 22 - Código SQL para RE03	19
Figura 23 - Resultado da Query relativa à RE03	19
Figura 24 - Código SQL para RE04	19
Figura 25 - Resultado da Query relativa à RE04	19
Figura 26 - Código SQL para RE05	20
Figura 27 - Resultado da Query relativa à RE05	20
Figura 28 - Código SQL para RE06	20
Figura 29 - Resultado da Query relativa à RE06	20
Figura 30 - Código SQL para RE07	21
Figura 31 - Resultado da Query relativa à RE07	21

Figura 32 - Código SQL para RE08	21
Figura 33 - Resultado da Query relativa à RE08	21
Figura 34 - Código SQL para RE09	22
Figura 35 - Resultado da Query relativa à RE09	22
Figura 36 - Código SQL para RE10	22
Figura 37 - Resultado da Query relativa à RE10	22
Figura 38 - Código SQL para V01	22
Figura 39 - Resultado da V01	23
Figura 40 - Código SQL para V02	23
Figura 41 - Resultado para V02	23
Figura 42 - Código SQL para V03	24
Figura 43 - Resultado para V03	24
Figura 44 - Código SQL para SP3TopPagamentos	28
Figura 45 - Resultado para SP3TopPagamentos	28

## Índice de Tabelas

Tabela 1 – Caracterização das Entidades	7
Tabela 2 - Atributos da entidade Cliente	9
Tabela 3 - Atributos da entidade Encomenda	9
Tabela 4 - Atributos da entidade Funcionário	10
Tabela 5 - Atributos da entidade Produto	10
Tabela 6 - Atributos da entidade Fornecedor	10
Tabela 7 - Atributos e chaves das entidades Cliente, Encomenda e Produto	12
Tabela 8 - Atributos e chaves das entidades Funcionário, Fornecedor e respectivas relações de N para M	12
Tabela 9 - Tamanho de cada tipo de dado.	24
Tabela 10 - Tamanho (em bytes) dos atributos da entidade Cliente	25
Tabela 11 - Tamanho (em bytes) dos atributos da entidade Encomenda	25
Tabela 12 - Tamanho (em bytes) dos atributos da entidade Funcionario	26
Tabela 13 - Tamanho (em bytes) dos atributos da entidade Produto	26
Tabela 14 - Tamanho (em bytes) dos atributos da entidade Encomenda_tem_Produto	26
Tabela 15 - Tamanho (em bytes) dos atributos da entidade Fornecedor	27
Tabela 16 - Tamanho (em bytes) dos atributos da entidade Produto_Fornecedor	27



# 1. Definição do Sistema

## 1.1. Contexto de aplicação do sistema

Fruto da evolução tecnológica, é cada vez mais notória a intensificação do e-commerce, inclusive de aparelhos eletrónicos como smartphones, computadores, tablets, SmartTVs, entre outros.

Com o fácil acesso a todos os produtos online, o número de encomendas com pedidos de entregas ao domicílio tem vindo a aumentar, devido, sobretudo, à funcionalidade e conveniência deste método.

Uma das causas que mais contribuiu para o aumento do uso do mesmo teve origem no período pandémico, onde a deslocação ao espaço físico da loja se tornou menos frequente e muitas vezes impossibilitada.

Posto isto, um sistema de gestão e organização de dados num estabelecimento com entregas ao domicílio é essencial e imprescindível.

## 1.2. Apresentação do Caso de Estudo

A **MinhoTec**, fundada em 2001 pela senhora Graça Freitas, é uma empresa de venda de produtos tecnológicos, situada na Avenida da Liberdade da cidade de Braga. Com o período da pandemia, as deslocações à loja da dona Graça deixaram de ocorrer com a mesma afluência. Desde então, a dona Graça implementou um método de encomendas online e entregas ao domicílio. Na loja existem 1 gestor de armazém/encomendas (a dona Graça) e 3 funcionários para entregas. Com todos os produtos disponíveis online e a possibilidade de entregas em casa, o volume de encomendas aumentou de tal forma que a monitorização se tornou difícil de conceber. Por este motivo, a senhora Graça pretende investir num sistema de base de dados capaz de registar e gerir todas as encomendas e todas as entregas da sua loja.

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

Um dos motivos pelos quais a dona Graça pretende implementar uma base de dados na sua loja prende-se com a facilidade que esta ferramenta de recolha e organização de informações providencia. Para a dona Graça e os restantes funcionários torna-se mais fácil, seguro e eficiente aceder a toda a informação relativa aos registos da empresa, uma vez que o armazenamento dos dados é garantido pela base de dados. Desta forma, este sistema permite uma melhor gestão das encomendas e entregas dos produtos da loja. Os registos da base de dados possibilitam fazer o balanço económico da loja e dão uma melhor perspetiva de uma possível expansão da mesma.

Esta base de dados deve registar a informação de todos os clientes da loja e o histórico das suas encomendas. Deve também conter a informação dos funcionários da empresa, os registos dos seus transportes, as datas de envio e de receção e deve estar associado a uma viatura da empresa. No sistema deve ficar registada toda a informação das encomendas e estas devem estar relacionadas com os produtos da loja para que seja possível saber toda a informação dos produtos que a encomenda contém. Deve ainda existir uma ligação dos produtos aos fornecedores onde fica registado todo o processo e informação entre eles. O Cliente deve receber por email a confirmação de encomenda e uma previsão da data de entrega após a compra efetuada.

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

A implementação de uma base de dados relacional permitirá analisar e relacionar melhor os dados da loja da dona Graça Freitas e, conseqüentemente, melhorar os serviços da empresa, uma vez que no sistema será possível manipular e consultar os dados de forma segura e rápida.

Este processo será bastante trabalhoso, mas irá garantir mais eficiência do que o atual método, não só na atualização e consulta de dados, mas também em termos de fiabilidade e segurança.

### **1.5. Recursos e Equipa de Trabalho**

A equipa de trabalho para realizar a base de dados é constituída por 4 especialistas em base de dados e a dona Graça. Consultamos sites de vendedores semelhantes online: *Fnac*, *Worten*, *Rádio Popular*, *Media Markt*, entre outros e outros informativos relativamente ao funcionamento de lojas do tipo analisado.

## **1.6. Plano de execução do Projeto**

Começamos pela introdução onde é feita a contextualização do projeto, correspondente caso de estudo e as motivações e objetivos do mesmo. De seguida, no ponto dois, temos o levantamento e análise de requisitos, começando pelo processo adotado, seguido da organização dos requisitos por: descrição, exploração e controlo, terminando com a análise e validação dos mesmos.

No terceiro ponto é abordado o modelo concetual, começando pela apresentação da abordagem de modelação realizada, depois mostra-se a identificação e características das entidades, dos relacionamentos e da associação entre eles. No mesmo seguimento, temos a apresentação do diagrama ER e este ponto termina com a validação do modelo concetual.

Já no quarto ponto é tratada a modelação lógica transposta do modelo concetual, apresentação do desenho do modelo e validação do mesmo.

No ponto seguinte é feita a passagem do modelo lógico para a implementação física da base de dados em SQL, bem como o seu povoamento e desenvolvimento das suas potencialidades. O relatório terminará com a conclusão, com os vários recursos utilizados para a elaboração do mesmo e com o nosso parecer.

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

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

Foram realizadas várias entrevistas à dona Graça com a finalidade de elaborar um conjunto de requisitos adequados e necessários para a elaboração do projeto. Realizou-se ainda uma pesquisa sobre a área em questão para perceber todos os procedimentos realizados na gestão e funcionamento de um estabelecimento deste tipo, bem como na compra e entrega dos produtos da loja.

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

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

RD01 - Todos os produtos devem ter identificador numerado sequencialmente, nome, preço, uma breve descrição, o seu tipo, o Iva aplicado, o lucro obtido com a sua venda e o stock disponível.

RD02 - Todos os registos das encomendas devem ser numerados, sequencialmente, incluir uma referência ao cliente que a realizou, a data e hora do pedido e de entrega e o total do custo dos produtos adquiridos.

RD03 - O registo de um cliente deve conter um identificador único, o nome, endereço (rua, código postal e localidade), data de nascimento, número de telemóvel, e-mail e o histórico de encomendas realizadas na loja até ao momento.

RD04 - Cada encomenda deve ter associado o funcionário que a processou e os produtos que a constituem.

RD05 - Devem ficar registados os artigos vendidos, a quantidade, e o preço.

RD06 - Cada funcionário deve ter um número de identificação, o nome e o veículo associado a ele.

### **2.2.2. Requisitos de exploração/manipulação**

RE01 - O sistema deve apresentar o número de encomendas realizadas, a quantidade e o valor dos produtos vendidos, a determinado período.

RE02 - A cada momento, deverá ser possível obter uma lista de clientes, que contenha o identificador, nome, contacto e e-mail.

RE03 - Os clientes devem ser inseridos no sistema quando realizam a primeira encomenda na loja. A cada cliente é pedido o nome, endereço, número, e-mail, contribuinte e data de nascimento.

RE04 - Deve ser possível calcular o tempo de processamento entre o pedido da encomenda e a entrega da mesma.

RE05 - O sistema deve ser capaz de listar os produtos por ordem de maior percentagem lucro.

RE06 - Deve disponibilizar uma listagem decrescente dos funcionários que geraram mais receita (lucro das vendas que processou) a empresa.

RE07 - Listar os Clientes por maior valor total gasto na loja de todas as encomendas que fez.

RE08 - O sistema deve listar todos os identificadores dos produtos com stock disponível.

RE09 - Listar os fornecedores que vendem o maior número de produtos distintos a loja.

RE10 – Ver o histórico de compras de um cliente na loja.

### **2.2.3. Requisitos de controlo**

RC01 - Apenas o gerente da loja tem acesso aos dados dos fornecedores.

RC02 - Cada funcionário pode aceder e modificar a informação relativa às encomendas realizadas.

RC03 - Os funcionários, à exceção do gerente da loja, podem aceder às informações dos clientes e dos produtos, mas não podem modificá-las.

RC04 - Remoções de qualquer registo apenas podem ser realizadas pelo gestor da loja.

RC05 – Apenas o gestor tem acesso a tabela dos funcionários da loja.

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

O processo de análise e validação dos requisitos respeita um método específico, no qual o levantamento de requisitos é dividido em três partes: descrição, exploração e controle.

Nos requisitos de descrição, foram analisados os diversos atributos de cada entidade e o seu tipo, bem como os relacionamentos entre as diferentes entidades, elementos indispensáveis para o funcionamento da loja da dona Graça.

Relativamente aos requisitos de exploração/manipulação, neles foram descritos os pontos mais relevantes que o sistema deve compreender, nomeadamente operações de listagem, cálculo e consulta de informação, tendo em consideração os atributos e relacionamentos apresentados anteriormente.

Nos requisitos de controlo foram estabelecidas as restrições de administração da base de dados pelo gerente e funcionários, de forma a criar coerência entre todo o sistema.

No processo de validação verificou-se a consistência, a precisão e a contextualização dos requisitos, para que fossem evitados erros e falhas na modelação dos mesmos. Utilizamos todas as técnicas recomendadas e usuais, evitando repetição e redundância e mantendo as conformidades e consistência necessárias.

## 3. Modelo Conceptual

### 3.1. Apresentação da abordagem de modelação realizada

A concessão do modelo conceptual foi realizada com grande cuidado e rigor, para que na sua implementação fosse evitada a existência de erros, quer na parte lógica, quer na parte física.

Recorremos ao software *brModelo* para a esquematização do modelo, começando pela identificação das entidades sem dependência e só depois as dependentes e as relações entre elas, com as devidas restrições e os atributos de cada uma.

### 3.2. Identificação e caracterização das entidades

Entidade	Caracterização
Cliente	Termo que contém a informação sobre as pessoas que tenham feito compras na loja.
Encomenda	Termo que contém a informação sobre os produtos contidos e a identificação do cliente e do funcionário.
Funcionário	Termo que contém a informação sobre os funcionários e o veículo utilizado nas entregas.
Produto	Termo que contém a informação sobre os produtos.
Fornecedor	Termo que contém a informação sobre o fornecedor.

Tabela 1 – Caracterização das Entidades

### 3.3. Identificação e caracterização dos relacionamentos

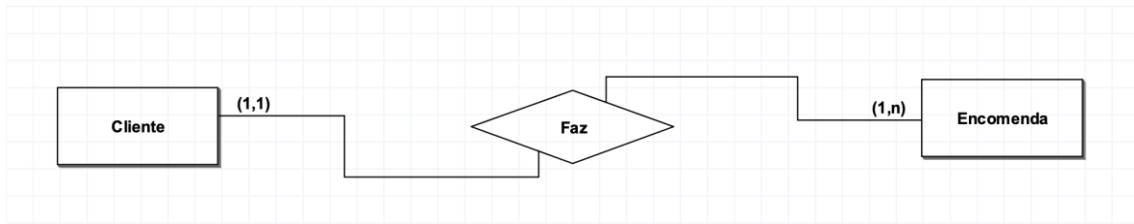


Figura 1 - Relação entre Cliente e Encomenda

Um cliente pode fazer 1 ou N Encomendas, mas uma encomenda só pode ser feita por um cliente. Logo temos uma cardinalidade de 1 para N.

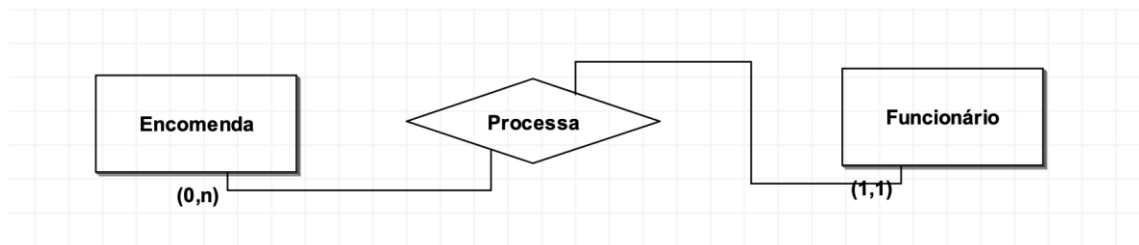


Figura 2 - Relação entre Encomenda e Funcionário

Um Funcionário pode processar 0 ou N Encomendas, mas as Encomendas só podem ser processadas por um único Funcionário. Logo, temos uma cardinalidade de 0 para N.

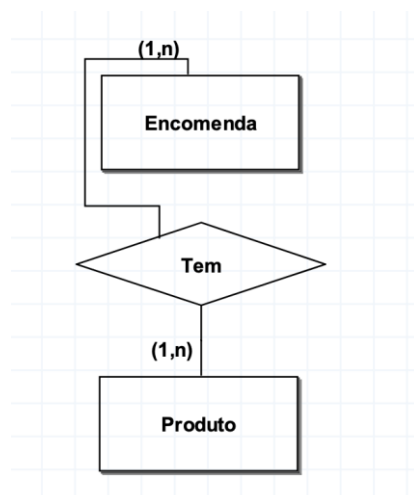


Figura 3 - Relação entre Encomenda e Produto

Uma encomenda pode ter 1 ou N produtos e um produto pode estar em 1 ou N encomendas. Logo temos uma cardinalidade de N para M.



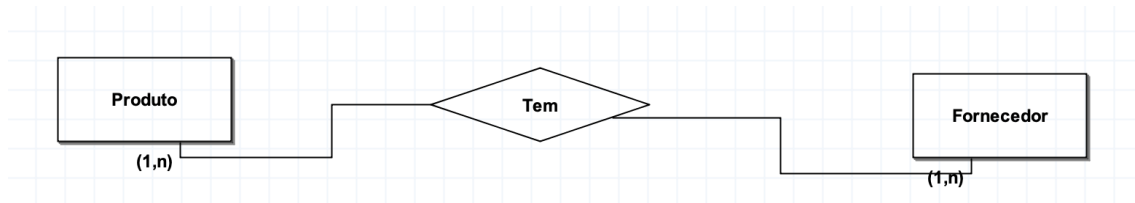


Figura 4 - Relação entre Produto e Fornecedor

Um produto pode ter 1 ou N fornecedores e um fornecedor pode fornecer 1 ou N produtos. Logo temos uma cardinalidade de N para M.

### 3.4. Identificação e caracterização das associações dos atributos com as entidades e relacionamentos

Atributo	Tipo	Descrição
<b>Id Cliente</b>	Chave	Número único que identifica o cliente
<b>Nome</b>	Simples	Nome do cliente
<b>Data de Nascimento</b>	Simples	Data de nascimento do cliente
<b>Email</b>	Simples	Email do cliente
<b>Telemóvel</b>	Simples	Número de telemóvel do cliente
<b>Histórico</b>	Simples	Histórico de encomendas do cliente
<b>Rua/Endereço</b>	Composto	Rua referente ao endereço do cliente
<b>Localidade/Endereço</b>	Composto	Localidade referente ao endereço do cliente
<b>Código Postal/Endereço</b>	Composto	Código postal referente ao endereço do cliente
<b>Contribuinte</b>	Simples	Número de Contribuinte do Cliente

Tabela 2 - Atributos da entidade Cliente

Atributo	Tipo	Descrição
<b>Id Encomenda</b>	Chave	Número único que identifica a encomenda
<b>Data/hora pedido</b>	Simples	Data em que a encomenda foi realizada
<b>Data/hora entrega</b>	Simples	Data de entrega da encomenda
<b>Custo</b>	Simples	Custo total da encomenda

Tabela 3 - Atributos da entidade Encomenda

Atributo	Tipo	Descrição
<b>Id Funcionário</b>	Chave	Número único que identifica o produto
<b>Nome</b>	Simples	Nome do funcionário
<b>Veículo</b>	Simples	Número que identifica o veículo atribuído ao funcionário
<b>Email</b>	Simples	Email do Funcionário
<b>Contacto</b>	Simples	Contacto Funcionário
<b>Contribuinte</b>	Simples	Contribuinte do Funcionário

Tabela 4 - Atributos da entidade Funcionário

Atributo	Tipo	Descrição
<b>Id Produto</b>	Chave	Número único que identifica o produto
<b>Preço</b>	Simples	Preço do produto
<b>Descrição</b>	Simples	Descrição do produto
<b>Nome</b>	Simples	Nome do produto
<b>Tipo</b>	Simples	Tipo do produto
<b>Iva</b>	Simples	Número decimal associado ao Iva do tipo do produto
<b>Lucro</b>	Simples	Número decimal associado ao lucro do produto
<b>Stock</b>	Simples	Número de unidades disponíveis do produto

Tabela 5 - Atributos da entidade Produto

Atributo	Tipo	Descrição
<b>Id Fornecedor</b>	Chave	Número único que identifica o fornecedor
<b>Designação</b>	Simples	Designação do fornecedor
<b>Contribuinte</b>	Simples	Contribuinte do fornecedor
<b>Contacto</b>	Simples	Contacto do fornecedor
<b>CodPostal/Endereço</b>	Composto	Código Postal da morada do fornecedor
<b>Localidade/Endereço</b>	Composto	Localidade da morada do fornecedor
<b>Rua/Endereço</b>	Composto	Rua da Morada do fornecedor
<b>Email</b>	Simples	Email do fornecedor

Tabela 6 - Atributos da entidade Fornecedor

### 3.5. Apresentação e explicação do diagrama ER

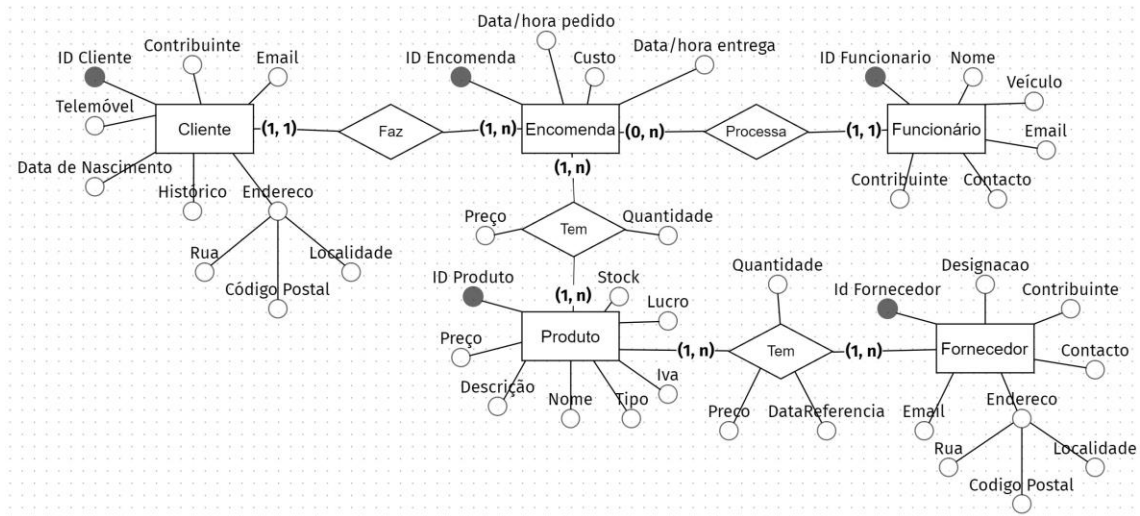


Figura 5 - Modelo Conceptual

### 3.6. Validação do Modelo Conceptual

Através de uma análise prolongada e detalhada, a equipa de trabalho chegou a uma concordância para o esquema final do modelo conceptual, onde todos os requisitos são satisfeitos. Após esta aprovação, prosseguimos para o modelo lógico.

## 4. Modelo Lógico

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

Para a elaboração do modelo lógico recorremos ao software *MySQL Workbench*, tendo por base o modelo conceptual, onde todas as normas foram seguidas com o maior rigor. Primeiramente, elaboramos as tabelas correspondentes a cada uma das entidades seguindo com especial atenção os relacionamentos de cada uma.

Cliente	Encomenda		Produto
(idCliente, nome, email, telemovel, endereco(rua, codigoPostal, localidade), dataNascimento, historico, contribuinte )	(idEncomenda, dataHora_pedido, custo, dataHora_entrega)		(idProduto, nome, preco, descricao, tipo, stock, iva, lucro)
Chave Primária: idCliente	Chave Primária: idEncomenda	Chaves Estrangeiras: idCliente, idFuncionario	Chave Primária: idProduto

Tabela 7 - Atributos e chaves das entidades Cliente, Encomenda e Produto

Funcionário	Encomenda_Tem_Produto	Produto_Fornecedor	Fornecedor
(idFuncionario, nome, veiculo, email, contacto, contribuinte)	(preco, quantidade)	(quantidade, preco, dataReferencia)	(idFornecedor, designacao, contribuinte, endereco(contacto, codPostal, localidade), email)
Chave Primária: idFuncionario	Chaves Estrangeiras: idProduto, idEncomenda	Chaves Estrangeiras: idFornecedor, idProduto	Chaves Estrangeiras: idProduto, idEncomenda, idCliente

Tabela 8 - Atributos e chaves das entidades Funcionário, Fornecedor e respectivas relações de N para M

## 4.2. Desenho do modelo lógico

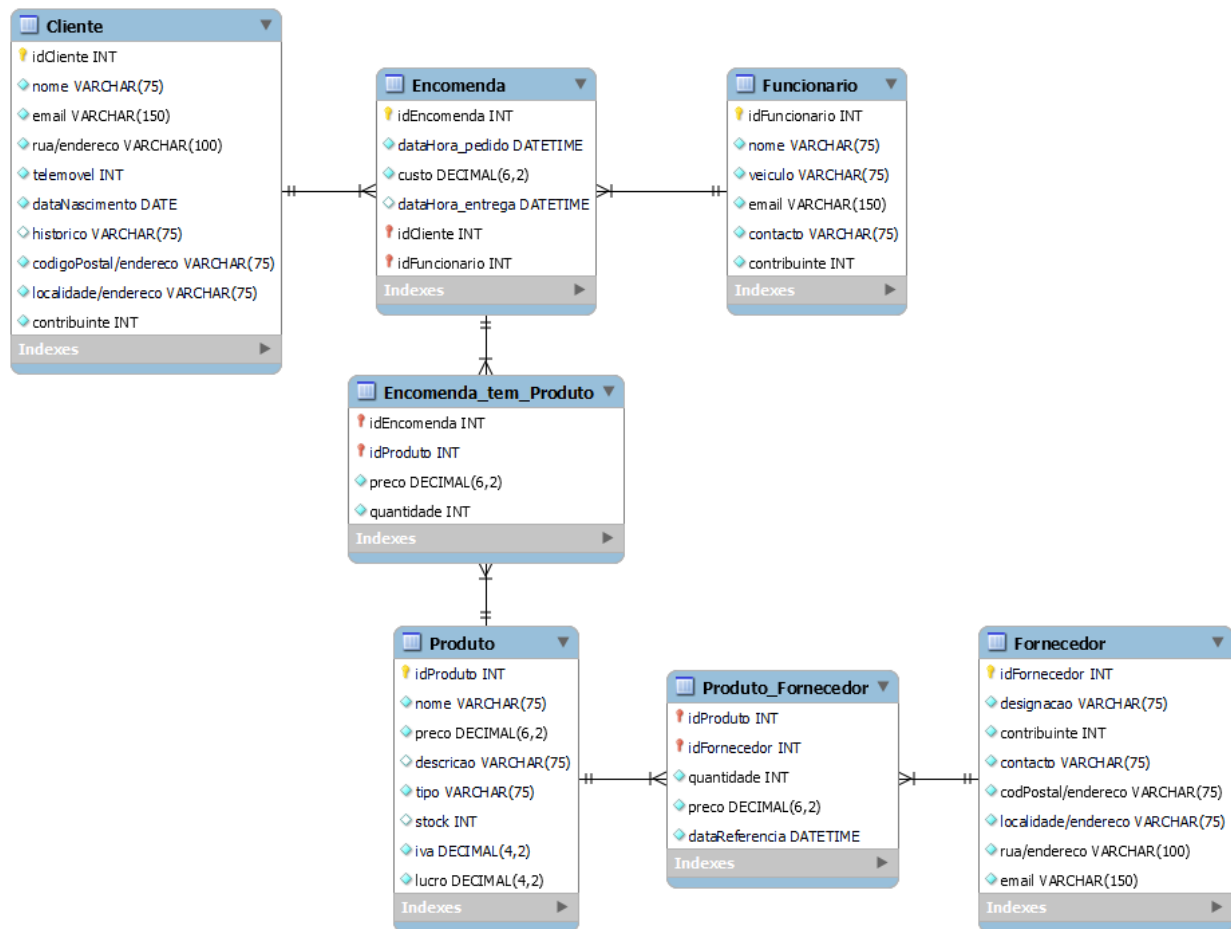
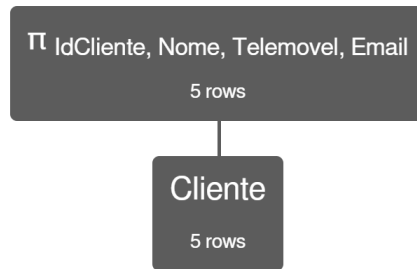


Figura 6 - Modelo Lógico

## 4.3. Validação do modelo com interrogações do utilizador

Para a validação do modelo lógico, implementamos alguns requisitos de exploração/manipulação que satisfazem as interrogações do utilizador na ferramenta *RelaX*, a fim de este as interpretar com sucesso. Devido as limitações que o programa possui no que toca à disponibilização de operadores de lógica estendida, não foi possível ilustrar algumas *queries* utilizadas na exploração/manipulação dos dados do sistema.



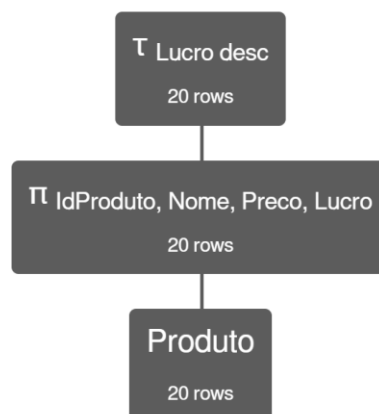
$\pi$  IdCliente, Nome, Telemovel, Email Cliente

Figura 7 - Árvore representativa da RE02



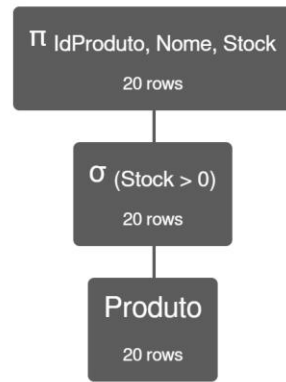
$\sigma$  (Historico = null) Cliente

Figura 8 - Árvore representativa da RE03



$\tau$  Lucro desc  $\pi$  IdProduto, Nome, Preço, Lucro Produto

Figura 9 - Árvore representativa da RE05



$\pi$  IdProduto, Nome, Stock  $\sigma$  (Stock > 0) Produto

Figura 10 - Árvore representativa da RE08

## 5. Implementação Física

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

A ferramenta, que utilizamos para a implementação física da base de dados foi a *MySQL Workbench*, a mesma que foi utilizada nas aulas. É bastante prática, intuitiva, tem um elevado número de recursos disponíveis e tem por base o paradigma relacional para a gestão de bases de dados.

Procedemos, agora, à apresentação das implementações físicas das tabelas do modelo da base de dados.

```
-- -----  
-- Criação da Tabela "Cliente"  
-- -----  
  
CREATE TABLE Cliente(  
  IdCliente INT NOT NULL,  
  Nome VARCHAR(75) NOT NULL,  
  Email VARCHAR(150) NOT NULL,  
  RuaEndereco VARCHAR(100) NOT NULL,  
  Telemovel INT NOT NULL,  
  DataNascimento DATE NOT NULL,  
  Historico VARCHAR(75),  
  CodPostalEndereco VARCHAR(75) NOT NULL,  
  LocalidadeEndereco VARCHAR(75) NOT NULL,  
  Contribuinte INT NOT NULL,  
  
  PRIMARY KEY(IdCliente),  
  UNIQUE KEY Email(Email),  
  UNIQUE KEY Contribuinte(Contribuinte));
```

Figura 11 - Código SQL para criar tabela  
Cliente

```
-- -----  
-- Criação da Tabela "Fornecedor"  
-- -----  
  
CREATE TABLE Fornecedor(  
  IdFornecedor INT NOT NULL,  
  Designacao VARCHAR(75) NOT NULL,  
  Contribuinte INT NOT NULL,  
  Contacto VARCHAR(75) NOT NULL,  
  CodPostalEndereco VARCHAR(75) NOT NULL,  
  LocalidadeEndereco VARCHAR(75) NOT NULL,  
  RuaEndereco VARCHAR(100) NOT NULL,  
  Email VARCHAR(150) NOT NULL,  
  
  PRIMARY KEY(IdFornecedor),  
  UNIQUE KEY Email(Email),  
  UNIQUE KEY Contribuinte(Contribuinte));
```

Figura 12 - Código SQL para criar tabela  
Fornecedor



```

-----
-- Criação da Tabela "Funcionario"
-----
CREATE TABLE Funcionario (
    IdFuncionario INT NOT NULL,
    Nome VARCHAR(75) NOT NULL,
    Veiculo VARCHAR(75) NOT NULL,
    Email VARCHAR(150) NOT NULL,
    Contacto VARCHAR(75) NOT NULL,
    Contribuinte INT NOT NULL,

    PRIMARY KEY (IdFuncionario),
    UNIQUE KEY Email(Email),
    UNIQUE KEY Contribuinte(Contribuinte));

```

Figura 13 - Código SQL para criar tabela Funcionario

```

-----
-- Criação da Tabela "Encomenda"
-----
CREATE TABLE Encomenda (
    IdEncomenda INT NOT NULL,
    DataHora_pedido DATETIME NOT NULL,
    Custo DECIMAL(6,2) NOT NULL,
    DataHora_entrega DATETIME DEFAULT NULL,
    IdCliente INT NOT NULL,
    IdFuncionario INT NOT NULL,

    PRIMARY KEY (IdEncomenda),
    FOREIGN KEY (IdCliente)
        REFERENCES Cliente(IdCliente),
    FOREIGN KEY (IdFuncionario)
        REFERENCES Funcionario(IdFuncionario));

```

Figura 14 - Código SQL para criar tabela Produto

```

-----
-- Criação da Tabela "Produto"
-----
CREATE TABLE Produto (
    IdProduto INT NOT NULL,
    Nome VARCHAR(75) NOT NULL,
    Preco DECIMAL(6,2) NOT NULL,
    Descricao VARCHAR(75),
    Tipo VARCHAR(75) NOT NULL,
    Stock INT,
    Iva DECIMAL(3,2) NOT NULL,
    Lucro DECIMAL(3,2) NOT NULL,

    PRIMARY KEY (IdProduto));

```

Figura 15 - Código SQL para criar tabela Encomenda

```

-----
-- Criação da Tabela "Produto_Fornecedor"
-----
CREATE TABLE Produto_Fornecedor (
    IdProduto INT NOT NULL,
    IdFornecedor INT NOT NULL,
    Quantidade INT NOT NULL,
    Preco DECIMAL(6,2) NOT NULL,
    DataReferencia DATETIME NOT NULL,

    FOREIGN KEY (IdProduto)
        REFERENCES Produto(IdProduto),
    FOREIGN KEY (IdFornecedor)
        REFERENCES Fornecedor(IdFornecedor));

```

Figura 16 - Código SQL para criar tabela Produto\_Fornecedor

```

-----
-- Criação da Tabela "Encomenda_tem_Produto"
-----
CREATE TABLE Encomenda_tem_Produto(
    IdEncomenda INT NOT NULL,
    IdProduto INT NOT NULL,
    Preco DECIMAL(6,2) NOT NULL,
    Quantidade INT NOT NULL,

    FOREIGN KEY (IdEncomenda)
        REFERENCES Encomenda(IdEncomenda),
    FOREIGN KEY (IdProduto)
        REFERENCES Produto(IdProduto));

```

Figura 17 - Código SQL para criar tabela Encomenda\_tem\_Produto

## 5.2. Tradução das interrogações do utilizador para SQL

Neste ponto, expomos o código SQL das *queries*, que representam a resposta às interrogações do utilizador feitas nos requisitos de exploração/manipulação, bem como a visualização do resultado que o programa apresenta para cada um com o povoamento atual.

**RE01 - O sistema deve apresentar o número de encomendas realizadas, a quantidade e o valor dos produtos vendidos, a determinado período.**

```
SELECT COUNT(distinct a.IdEncomenda) as Encomendas,  
       SUM(Quantidade) as Quantidades, SUM(Preco) as Preco_Bruto  
FROM Encomenda a, Encomenda_tem_produto b  
where (a.IdEncomenda = b.IdEncomenda and DataHora_pedido >= '2022-10-01');
```

Figura 18 - Código SQL para RE01

	Encomendas	Quantidades	Preco_Bruto
▶	4	7	3985.79

Figura 19 - Resultado da Query relativa à RE01

**RE02 - A cada momento, deverá ser possível obter uma lista de clientes, que contenha o identificador, nome, contacto e e-mail.**

```
SELECT IdCliente, Nome, Telemovel, Email FROM Cliente;
```

Figura 20 - Código SQL para RE02

	IdCliente	Nome	Telemovel	Email
▶	1	Luisa Fernandes	915283746	luisa.fernandes@email.com
	2	Bruno Almeida	914672938	bruno.almeida@email.com
	3	Sofia Gomes	963204789	sofia.gomes@email.com
	4	Daniel Pereira	930293847	daniel.pereira@email.com
	5	Isabel Costa	928364529	isabel.costa@email.com
*	NULL	NULL	NULL	NULL

Figura 21 - Resultado da Query relativa à RE02

RE03 - Os clientes devem ser inseridos no sistema quando realizam a primeira encomenda na loja. A cada cliente é pedido o nome, endereço, número, e-mail, contribuinte e data de nascimento.

```
SELECT * FROM Cliente
where (Historico is NULL);
```

Figura 22 - Código SQL para RE03

	IdCliente	Nome	Email	RuaEndereco	Telemovel	DataNascimento	Historico	CodPostalEndereco	LocalidadeEndereco	Contribuinte
▶	3	Sofia Gomes	sofia.gomes@email.com	Rua da Fonte	963204789	1995-11-09	NULL	4700-227	Braga	369258147
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 23 - Resultado da Query relativa à RE03

RE04 - Deve ser possível calcular o tempo de processamento entre o pedido da encomenda e a entrega da mesma.

```
SELECT IdEncomenda, (TIMESTAMPDIFF(day, DataHora_pedido, DataHora_entrega)) as 'Dias',
(TIMESTAMPDIFF(hour, DataHora_pedido, DataHora_entrega)
-(TIMESTAMPDIFF(day, DataHora_pedido, DataHora_entrega)*24)) as 'Horas' FROM Encomenda;
```

Figura 24 - Código SQL para RE04

	IdEncomenda	Dias	Horas
▶	1	2	2
	2	3	1
	3	1	9
	4	2	2
	5	2	1
	6	NULL	NULL
	7	NULL	NULL
	8	NULL	NULL
	9	NULL	NULL
	10	NULL	NULL

Figura 25 - Resultado da Query relativa à RE04

**RE05 - O sistema deve ser capaz de listar os produtos por ordem de maior percentagem lucro.**

```
SELECT IdProduto, Nome, Preço, Lucro FROM Produto
ORDER BY (Lucro) DESC;
```

Figura 26 - Código SQL para RE05

	IdProduto	Nome	Preço	Lucro
▶	14	Protetor ecra medida p	19.00	0.50
	15	Protetor ecra medida m	29.00	0.50
	16	Protetor ecra medida g	39.00	0.50
	9	Smartphone Marca B Medio	749.00	0.32
	8	Computador port 14 TSV	749.90	0.30
	17	Capa okk 8	19.95	0.30
	6	TV 50 TSV	899.00	0.28
	11	Leitor BRAYs STK	199.00	0.26
	4	Smartphone Marca A Medio	799.00	0.25
	5	Smartphone Marca B Top	1249.00	0.25
	18	Capa A Top	29.00	0.25
	1	Tv 55 XPTO 4K	1999.00	0.22
	2	Tablet 8 okk	399.00	0.20
	7	Tablet 10 okk	599.00	0.20
	10	TV 60 STK	1599.99	0.20
	12	Leitor CDs AUDVIS	79.00	0.20
	19	Capa B Medio	25.90	0.20
	20	Kit limp AUDVIS	29.00	0.20
	3	Smartphone Marca A Top	1299.00	0.15
	13	Leitor LPs AUDVIS	239.90	0.15
*	NULL	NULL	NULL	NULL

Figura 27 - Resultado da Query relativa à RE05

**RE06 - Deve disponibilizar uma listagem decrescente dos funcionários que geraram mais receita (lucro das vendas que processou) a empresa.**

```
SELECT IdFuncionario, sum(cast(c.Preço*Lucro as DECIMAL(6,2))) as 'Receita'
FROM Encomenda a, Encomenda_tem_produto b, Produto c
where (a.IdEncomenda = b.IdEncomenda and b.IdProduto = c.IdProduto)
GROUP BY (IdFuncionario)
ORDER BY (Receita) DESC;
```

Figura 28 - Código SQL para RE06

	IdFuncionario	Receita
▶	2	682.05
	3	661.38
	1	486.19
	4	171.54

Figura 29 - Resultado da Query relativa à RE06

**RE07 - Listar os Clientes por maior valor total gasto na loja de todas as encomendas que fez.**

```
SELECT b.IdCliente, Nome, Sum(Custo) as 'Total_Gasto' FROM Cliente a, Encomenda b
where(a.IdCliente = b.IdCliente)
GROUP BY (IdCliente)
ORDER BY (Total_Gasto) DESC;
```

Figura 30 - Código SQL para RE07

	IdCliente	Nome	Total_Gasto
►	2	Bruno Almeida	2917.70
	5	Isabel Costa	2267.90
	3	Sofia Gomes	1599.99
	4	Daniel Pereira	1446.00
	1	Luisa Fernandes	798.00

Figura 31 - Resultado da Query relativa à RE07

**RE08 - O sistema deve listar todos os identificadores dos produtos com stock disponível.**

```
SELECT IdProduto, Nome, Stock FROM Produto
where (Stock > 0);
```

Figura 32 - Código SQL para RE08

	IdProduto	Nome	Stock
►	1	Tv 55 XPTO 4K	5
	2	Tablet 8 okk	15
	3	Smartphone Marca A Top	50
	4	Smartphone Marca A Medio	80
	5	Smartphone Marca B Top	60
	6	TV 50 TSV	8
	7	Tablet 10 okk	35
	8	Computador port 14 TSV	40
	9	Smartphone Marca B Medio	90
	10	TV 60 STK	8
	11	Leitor BRAYs STK	35
	12	Leitor CDs AUDVIS	30
	13	Leitor LPs AUDVIS	19
	14	Protetor ecra medida p	130
	15	Protetor ecra medida m	100
	16	Protetor ecra medida g	80
	17	Capa okk 8	20
	18	Capa A Top	30
	19	Capa B Medio	35
	20	Kit limp AUDVIS	40
*	NULL	NULL	NULL

Figura 33 - Resultado da Query relativa à RE08

**RE09 - Listar os fornecedores que vendem o maior número de produtos distintos à loja.**

```
SELECT a.IdFornecedor, Designacao, Count(distinct IdProduto) as 'Produtos'
FROM Fornecedor a, Produto_Fornecedor b
where(a.IdFornecedor = b.IdFornecedor)
GROUP BY (IdFornecedor)
ORDER BY (Produtos) DESC;
```

Figura 34 - Código SQL para RE09

	IdFornecedor	Designacao	Produtos
►	5	Empresa telecomunicacoes	10
	1	Empresa tvs/video	4
	3	Empresa informatica	4
	2	Empresa som	2

Figura 35 - Resultado da Query relativa à RE09

**RE10 – Ver o histórico de compras de um cliente da loja.**

```
SELECT a.IdEncomenda, a.IdCliente, b.IdProduto, c.nome
FROM Encomenda a, Encomenda_tem_Produto b, Produto c
where(a.IdCliente = 2 and a.IdEncomenda = b.IdEncomenda
and b.IdProduto = c.IdProduto and a.DataHora_entrega is not null);
```

Figura 36 - Código SQL para RE10

	IdEncomenda	IdCliente	IdProduto	nome
►	1	2	4	Smartphone Marca A Medio
	1	2	19	Capa B Medio
	1	2	15	Protetor ecra medida m
	5	2	5	Smartphone Marca B Top
	5	2	16	Protetor ecra medida g
	5	2	15	Protetor ecra medida m

Figura 37 - Resultado da Query relativa à RE10

## 5.3. Definição e caracterização das vistas de utilização em SQL (alguns exemplos)

**V01 - Número de vezes que cada produto foi vendido**

```
CREATE VIEW V01 AS
SELECT a.IdProduto, Nome, Count(a.IdProduto) AS Total_Vendas FROM Encomenda_tem_Produto a, Produto b
where(a.IdProduto = b.IdProduto)
GROUP BY (IdProduto)
ORDER BY (Total_Vendas) DESC;
```

Figura 38 - Código SQL para V01

	IdProduto	Nome	Total_Vendas
►	15	Protetor ecra medida m	2
	16	Protetor ecra medida g	2
	1	Tv 55 XPTO 4K	1
	3	Smartphone Marca A Top	1
	4	Smartphone Marca A Medio	1
	5	Smartphone Marca B Top	1
	7	Tablet 10 okk	1
	8	Computador port 14 TSV	1
	10	TV 60 STK	1
	11	Leitor BRAYs STK	1
	12	Leitor CDs AUDVIS	1
	13	Leitor LPs AUDVIS	1
	18	Capa A Top	1
	19	Capa B Medio	1
	20	Kit limp AUDVIS	1

Figura 39 - Resultado da V01

## V02 - Unidades por encomenda

CREATE VIEW V02 AS

```
SELECT IdEncomenda, count(distinct IdProduto) AS 'Unidades' FROM Encomenda tem Produto
GROUP BY (IdEncomenda);
```

Figura 40 - Código SQL para V02

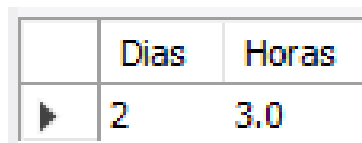
	IdEncomenda	Unidades
►	1	3
	2	1
	3	1
	4	1
	5	3
	6	1
	7	1
	8	1
	9	3
	10	2

Figura 41 - Resultado para V02

### V03 - Média de tempo de entrega de uma encomenda

```
CREATE VIEW V03 AS
SELECT Cast(Sum(TIMESTAMPDIFF(day, DataHora_pedido, DataHora_entrega))
/Count(DataHora_entrega) as DECIMAL(2,0)) as 'Dias',
Cast(Sum(TIMESTAMPDIFF(hour, DataHora_pedido, DataHora_entrega)
-(TIMESTAMPDIFF(day, DataHora_pedido, DataHora_entrega)*24))
/Count(DataHora_entrega) as DECIMAL(2,1)) as 'Horas' FROM Encomenda;
```

Figura 42 - Código SQL para V03



	Dias	Horas
▶	2	3.0

Figura 43 - Resultado para V03

## 5.4. Cálculo do espaço da base de dados (inicial e taxa de crescimento anual)

Com o intuito de estimar o espaço ocupado por uma entrada de cada entidade na nossa base de dados foi necessário calcular quantos bytes ocupam cada tipo de dados. Para tal, recorremos às secções 11.1, 11.2, 11.3 e 11.7 do *MySQL 8.0 Reference Manual*. Através desta pesquisa foi possível estabelecer as seguintes métricas de bytes para cada tipo de dado:

Tipo	Tamanho
INT	4 bytes
VARCHAR (N)	N+1 ou 2xN+1 ou 2xN+2*
DATE	3 bytes
DATETIME	5 bytes
DECIMAL (X, Y)	8 bytes**

Tabela 9 - Tamanho de cada tipo de dado.

\*N+1 bytes caso a *string* esteja codificada em *ASCII*, 2\*N + 1 caso a codificação seja mais complexa, mas o valor de 2\*N seja menor que 255 e 2\*N + 2 caso 2\*N maior que 255.

\*\*Quando X - Y = 9 e Y < 9, para valores superiores devemos consultar a tabela disponível na secção 11.7 do manual.



Tendo em consideração os valores fixados, são apresentados de seguida os valores máximos que uma entrada de cada entidade e relação com atributos ocupará.

Atributo	Tipo	Tamanho
<b>IdCliente</b>	INT	4 bytes
<b>Nome</b>	VARCHAR (75)	151 bytes
<b>Data de Nascimento</b>	DATE	3 bytes
<b>Email</b>	VARCHAR (150)	302 bytes
<b>Telemóvel</b>	INT	4 bytes
<b>Histórico</b>	VARCHAR (75)	151 bytes
<b>Rua/Endereço</b>	VARCHAR (100)	201 bytes
<b>Localidade/Endereço</b>	VARCHAR (75)	151 bytes
<b>Código Postal/Endereço</b>	VARCHAR (75)	151 bytes
<b>Contribuinte</b>	INT	4 bytes

Tabela 10 - Tamanho (em bytes) dos atributos da entidade Cliente

Uma entrada na tabela Cliente ocupa um total de 1122 bytes.

Atributo	Tipo	Tamanho
<b>IdEncomenda</b>	INT	4 bytes
<b>Data/hora pedido</b>	DATETIME	5 bytes
<b>Data/hora entrega</b>	DATETIME	5 bytes
<b>Custo</b>	DECIMAL (6,2)	8 bytes
<b>IdCliente</b>	INT	4 bytes
<b>IdFuncionario</b>	INT	4 bytes

Tabela 11 - Tamanho (em bytes) dos atributos da entidade Encomenda

Uma entrada na tabela Encomenda ocupa um total de 30 bytes.

Atributo	Tipo	Tamanho
IdFuncionario	INT	4 bytes
Nome	VARCHAR (75)	151 bytes
Veículo	VARCHAR (75)	151 bytes
Email	VARCHAR (150)	302 bytes
Contacto	VARCHAR (75)	151 bytes
Contribuinte	INT	4 bytes

Tabela 12 - Tamanho (em bytes) dos atributos da entidade Funcionario

Uma entrada na tabela Funcionário ocupa um total de 763 bytes.

Atributo	Tipo	Tamanho
IdProduto	INT	4 bytes
Preço	DECIMAL (6,2)	8 bytes
Descrição	VARCHAR (75)	151 bytes
Nome	VARCHAR (75)	151 bytes
Tipo	VARCHAR (75)	151 bytes
Iva	DECIMAL (4,2)	8 bytes
Lucro	DECIMAL (4,2)	8 bytes
Stock	INT	4 bytes

Tabela 13 - Tamanho (em bytes) dos atributos da entidade Produto

Uma entrada na tabela Produto ocupa um total de 485 bytes.

Atributo	Tipo	Tamanho
IdEncomenda	INT	4 bytes
IdProduto	INT	4 bytes
Preco	DECIMAL (6,2)	8 bytes
Quantidade	INT	4 bytes

Tabela 14 - Tamanho (em bytes) dos atributos da entidade Encomenda\_tem\_Produto

Uma entrada na tabela Encomenda\_tem\_Produto ocupa um total de 20 bytes.

Atributo	Tipo	Tamanho
<b>Id Fornecedor</b>	INT	4 bytes
<b>Designação</b>	VARCHAR (75)	151 bytes
<b>Contribuinte</b>	INT	4 bytes
<b>Contacto</b>	VARCHAR (75)	151 bytes
<b>CodPostal/Endereço</b>	VARCHAR (75)	151 bytes
<b>Localidade/Endereço</b>	VARCHAR (75)	151 bytes
<b>Rua/Endereço</b>	VARCHAR (100)	201 bytes
<b>Email</b>	VARCHAR (150)	302 bytes

Tabela 15 - Tamanho (em bytes) dos atributos da entidade Fornecedor

Uma entrada na tabela Fornecedor ocupa um total de 1115 bytes.

Atributo	Tipo	Tamanho
<b>IdProduto</b>	INT	4 bytes
<b>IdFornecedor</b>	INT	4 bytes
<b>Quantidade</b>	INT	4 bytes
<b>Preço</b>	DECIMAL (6,2)	8 bytes
<b>DataReferencia</b>	DATETIME	5 bytes

Tabela 16 - Tamanho (em bytes) dos atributos da entidade Produto\_Fornecedor

Uma entrada na tabela Produto\_Fornecedor ocupa um total de 25 bytes.

Usando os valores calculados foi possível obter uma estimativa do espaço ocupado pela base de dados sem povoamento de 3560 bytes e de 25840 bytes com o povoamento atual.

Tendo em consideração o contexto do nosso projeto, uma loja de equipamentos eletrónicos, é de esperar que as entidades que tenham uma maior quantidade de inserções na base de dados anualmente sejam Cliente e Encomenda.

Deste modo, se assumirmos uma taxa de crescimento anual de 60% no número de clientes, tendo atualmente 5, em espaço, isto traduzir-se-á num aumento de  $(5 \times 0.06) \times 1122 = 336.6$  bytes.

Considerando as encomendas de clientes existentes (10) e as de novos clientes e, desta forma, atribuindo uma taxa de crescimento anual de 80% ao número de encomendas, teremos um aumento de cerca de 240 bytes no espaço ocupado  $(10 \times 0.8) \times 30 = 240$ .

Relativamente aos funcionários, os crescimentos considerados não justificam novas entradas na tabela Funcionario.

Tendo em conta os novos clientes, poderá ocorrer um aumento na procura por diferentes produtos, implicando um aumento da variedade dos mesmos na loja. Consequentemente, isto representará acréscimos de entradas na tabela Produto e Produto\_Fornecedor. Contrariamente, as entradas na tabela Fornecedor poderão permanecer inalteradas, pelo facto dos novos produtos poderem vir dos mesmos fornecedores.

Considerando as taxas de crescimento anuais dos Clientes e Encomendas, o crescimento total anual do espaço ocupado pela base de dados seria de 576.6 bytes, perfazendo um total de 26416.6 bytes.

## 5.5. Procedimentos Implementados

```
-- -----  
-- Criacao de um procedimento para nos indicar os 3 maiores pagamentos realizados  
-- e respetivos clientes.  
DELIMITER $$  
CREATE PROCEDURE sp3TopPagamentos()  
BEGIN  
    SELECT b.IdCliente, Nome, Custo AS 'Pagamento' FROM Cliente a, Encomenda b  
        where(a.IdCliente = b.IdCliente)  
        ORDER BY Pagamento DESC  
        LIMIT 3;  
END $$  
  
CALL sp3TopPagamentos;
```

Figura 44 - Código SQL para SP3TopPagamentos

	IdCliente	Nome	Pagamento
►	5	Isabel Costa	1999.00
	3	Sofia Gomes	1599.99
	4	Daniel Pereira	1367.00

Figura 45 - Resultado para SP3TopPagamentos

## 5.6. Plano de segurança e recuperação de dados

A gestão de dados é uma parte fundamental de qualquer empresa, seja ela uma loja de venda de equipamentos tecnológicos ou outra, uma vez que os dados são utilizados para tomar decisões críticas para o alcance de objetivos e metas empresariais. Por este motivo, é necessário tomar precauções na gestão de dados, visto que, a sua perda poderá colocar em risco todo o negócio. Alguns riscos que a gestão de dados enfrenta passam pelo vazamento de dados, ataques cibernéticos, erros humanos, problemas de conformidade, episódios climáticos, entre outros. De forma a prevenir estas situações catastróficas ou minimizar os danos causados é necessário implementar medidas de segurança e diretrizes para assegurar o correto funcionamento da base de dados ou a sua recuperação eficiente.

As diretrizes que mais se destacam são:

- Uma medida de Backups Periódicos, para uma Cloud fornecida por serviços de terceiros, ou um Sistema Físico, com um grande aglomerado de Discos Rígidos, num local diferente do servidor da base de dados, prevenindo a destruição mútua, em caso de catástrofes ou acidentes nesse local.
- Monitoramento constante de segurança para detetar e prevenir qualquer tentativa de invasão ou violação da segurança.
- Um documento com a indicação do limite máximo de destruição de dados permitido e quais os meios mais afetados por esse incidente.
- Testes regulares de recuperação de desastres para garantir que as cópias de segurança e as medidas de recuperação estão a funcionar corretamente.
- Um documento que contenha uma lista de prioridades relativamente à informação que se encontra na base de dados.
- Controlo no acesso para garantir que apenas pessoas autorizadas tenham acesso à base de dados.
- Formação dos funcionários sobre boas práticas de segurança para assegurar que todos estão cientes dos procedimentos de recuperação em caso de desastres.
- Uso de software de segurança para proteger a base de dados contra malware, vírus e outras ameaças.
- Garantir acesso remoto à base de dados para permitir a recuperação dos mesmos em caso de catástrofe.

## 6. Conclusões e Trabalho Futuro

Após a realização deste trabalho, conseguimos perceber que a parte mais importante para que uma base de dados seja implementada com sucesso de forma funcional e prática e com todas as suas capacidades, é a da recolha detalhada e a organização dos requisitos específicos necessários, uma vez que é difícil construir uma base de dados coerente, robusta e segura, sem estes procedimentos. Para além disso, se esta fase não for realizada meticulosa e com o máximo de rigor possível, surgirão várias complicações e questões que poderiam ter sido evitadas, caso fossem realizadas todas essas etapas.

Com o trabalho completo, conseguimos perceber que, apesar de termos dado atenção a estes factos, ainda existe muito a melhorar no processo que realizamos, sendo, por isso, necessária bastante prática para que se consiga realizar o processo da forma mais eficaz possível. Concluimos que é necessário maior rigor no levantamento de requisitos aplicados ao projeto, nomeadamente nos requisitos de exploração/manipulação que foram posteriormente alterados a fim de potenciar as capacidades da base de dados para a empresa e nos requisitos de controlo, visto que não estavam de acordo com o lecionado na Unidade Curricular.

Foi necessária uma nova caracterização das várias entidades, com a criação da tabela Fornecedor e a tabela Produto\_Fornecedor, que faz a ligação entre ambas as tabelas Fornecedor e Produto, o que originou a criação de mais atributos, nestas e noutras entidades. Ao rever a estrutura da base de Dados, também tivemos de rever os tipos de dados dos atributos, uma vez que nos deparamos com dificuldades, no processo de criação de *queries* do sistema.

Podemos então concluir que os objetivos deste trabalho foram cumpridos, foram identificados e corrigidos os erros cometidos no processo, conseguindo assim o desenvolvimento de uma base de dados apta a sustentar e a corresponder a todas as necessidades do estabelecimento de Gestão de Entregas ao Domicílio da dona Graça Freitas, a **MinhoTec**.

## 7. Referências

Ferramenta RelaX, dezembro de 2022, < <https://dbis-uibk.github.io/relax/calc/local/uibk/local/0>>

Fnac, outubro de 2022, <<https://www.fnac.pt/>>

Jasmin, outubro de 2022, <<https://www.jasminsoftware.pt/blog/gestao-de-encomendas-loja/>>

Media Markt, outubro de 2022, <<https://mediamarkt.pt/>>

MySQL 8.0 Reference Manual, dezembro de 2022, <<https://dev.mysql.com/doc/refman/8.0/en/>>

MySQL Workbench, dezembro de 2022, < <https://dev.mysql.com/downloads/workbench/>>

Rádio Popular, outubro de 2022, <<https://www.radiopopular.pt/>>

Reduniq, outubro de 2022, <<https://www.reduniq.pt/blog/como-criar-um-sistema-de-entrega-e-passar-a-vender-online/>>

Software BR Modelo Web, novembro de 2022, < <https://app.brmodeloweb.com/#/>>

Universidade do Minho, e-learning+, Unidade Curricular de Base de Dados, outubro de 2022, < [https://elearning.uminho.pt/webapps/blackboard/content/listContent.jsp?course\\_id=\\_54554\\_1&content\\_id=\\_1250163\\_1&mode=reset](https://elearning.uminho.pt/webapps/blackboard/content/listContent.jsp?course_id=_54554_1&content_id=_1250163_1&mode=reset)>

Worten, outubro de 2022, <<https://www.worten.pt/>>

## 8. Lista de Siglas e Acrónimos

ER	Entidade Relacionamento
SQL	<i>Structured Query Language</i>
RelaX	<i>Relational Algebra Calculator</i>
RD	Requisitos de Descrição
RE	Requisitos de Exploração/Manipulação
RC	Requisitos de Controlo
SP	<i>Stored Procedure</i>



## **Anexos**

## I. Anexo 1 – *Script* da Criação da Base de Dados

```
CREATE DATABASE IF NOT EXISTS MinhoTec
DEFAULT CHARACTER SET utf8;
SHOW WARNINGS;
```

```
-- Indicação da base de dados de trabalho
USE MinhoTec;
```

```
-- -----
```

```
-- Apagar a Base de dados
-- DROP DATABASE minhotecc;
```

```
-- -----
```

```
-- Criação das tabelas da base de dados
```

```
-- -----
```

```
-- Criação da Tabela "Cliente"
-- DROP TABLE Cliente;
-- DESC Cliente;
```

```
-- -----
```

```
CREATE TABLE Cliente(
  IdCliente INT NOT NULL,
  Nome VARCHAR(75) NOT NULL,
  Email VARCHAR(150) NOT NULL,
  RuaEndereco VARCHAR(100) NOT NULL,
  Telemovel INT NOT NULL,
  DataNascimento DATE NOT NULL,
  Historico VARCHAR(75),
  CodPostalEndereco VARCHAR(75) NOT NULL,
  LocalidadeEndereco VARCHAR(75) NOT NULL,
  Contribuinte INT NOT NULL,
```

```
PRIMARY KEY(IdCliente),
UNIQUE KEY Email(Email),
UNIQUE KEY Contribuinte(Contribuinte));
```

```
-- -----
```

```
-- Criação da Tabela "Fornecedor"
-- DROP TABLE Fornecedor;
-- DESC Fornecedor;
```

```
-- -----
```

```

CREATE TABLE Fornecedor(
  IdFornecedor INT NOT NULL,
  Designacao VARCHAR(75) NOT NULL,
  Contribuinte INT NOT NULL,
  Contacto VARCHAR(75) NOT NULL,
  CodPostalEndereco VARCHAR(75) NOT NULL,
  LocalidadeEndereco VARCHAR(75) NOT NULL,
  RuaEndereco VARCHAR(100) NOT NULL,
  Email VARCHAR(150) NOT NULL,

```

```

  PRIMARY KEY(IdFornecedor),
  UNIQUE KEY Email(Email),
  UNIQUE KEY Contribuinte(Contribuinte));

```

```

-- -----
-- Criação da Tabela "Funcionario"
-- DROP TABLE Funcionario;
-- DESC Funcionario;
-- -----

```

```

CREATE TABLE Funcionario (
  IdFuncionario INT NOT NULL,
  Nome VARCHAR(75) NOT NULL,
  Veiculo VARCHAR(75) NOT NULL,
  Email VARCHAR(150) NOT NULL,
  Contacto VARCHAR(75) NOT NULL,
  Contribuinte INT NOT NULL,

  PRIMARY KEY (IdFuncionario),
  UNIQUE KEY Email(Email),
  UNIQUE KEY Contribuinte(Contribuinte));

```

```

-- -----
-- Criação da Tabela "Encomenda"
-- DROP TABLE Encomenda;
-- DESC Encomenda;
-- -----

```

```

CREATE TABLE Encomenda (
  IdEncomenda INT NOT NULL,
  DataHora_pedido DATETIME NOT NULL,
  Custo DECIMAL(6,2) NOT NULL,

```

```
DataHora_entrega DATETIME DEFAULT NULL,  
IdCliente INT NOT NULL,  
IdFuncionario INT NOT NULL,
```

```
PRIMARY KEY (IdEncomenda),  
FOREIGN KEY (IdCliente)  
  REFERENCES Cliente(IdCliente),  
FOREIGN KEY (IdFuncionario)  
  REFERENCES Funcionario(IdFuncionario));
```

```
-- -----  
-- Criação da Tabela "Produto"  
-- DROP TABLE Produto;  
-- DESC Produto;  
-- -----
```

```
CREATE TABLE Produto (  
  IdProduto INT NOT NULL,  
  Nome VARCHAR(75) NOT NULL,  
  Preco DECIMAL(6,2) NOT NULL,  
  Descricao VARCHAR(75),  
  Tipo VARCHAR(75) NOT NULL,  
  Stock INT,  
  Iva DECIMAL(3,2) NOT NULL,  
  Lucro DECIMAL(3,2) NOT NULL,  
  
  PRIMARY KEY (IdProduto));
```

```
-- -----  
-- Criação da Tabela "Produto_Fornecedor"  
-- DROP TABLE Produto_Fornecedor;  
-- DESC Produto_Fornecedor;  
-- -----
```

```
CREATE TABLE Produto_Fornecedor (  
  IdProduto INT NOT NULL,  
  IdFornecedor INT NOT NULL,  
  Quantidade INT NOT NULL,  
  Preco DECIMAL(6,2) NOT NULL,  
  DataReferencia DATETIME NOT NULL,  
  
  FOREIGN KEY (IdProduto)
```

```
REFERENCES Produto(IdProduto),  
FOREIGN KEY (IdFornecedor)  
REFERENCES Fornecedor(IdFornecedor));
```

```
-- -----  
-- Criação da Tabela "Encomenda_tem_Produto"  
-- DROP TABLE Encomenda_tem_Produto;  
-- DESC Encomenda_tem_Produto;  
-- -----
```

```
CREATE TABLE Encomenda_tem_Produto(  
    IdEncomenda INT NOT NULL,  
    IdProduto INT NOT NULL,  
    Preco DECIMAL(6,2) NOT NULL,  
    Quantidade INT NOT NULL,  
  
    FOREIGN KEY (IdEncomenda)  
        REFERENCES Encomenda(IdEncomenda),  
    FOREIGN KEY (IdProduto)  
        REFERENCES Produto(IdProduto));
```

## II. Anexo 2 – Script do Povoamento

-- Povoamento da Base de Dados

INSERT INTO Cliente

(IdCliente, Nome, Email, RuaEndereco, Telemovel, DataNascimento, Historico, CodPostalEndereco, LocalidadeEndereco, Contribuinte)

VALUES

('01', 'Luisa Fernandes', 'luisa.fernandes@email.com', 'Rua dos Pescadores', '915283746', '2000-01-01', '04', '4970-163', 'Vila Nova de Cerveira', '123987456'),

('02', 'Bruno Almeida', 'bruno.almeida@email.com', 'Rua do Comércio', '914672938', '1998-06-05', '01, 05', '4800-019', 'Barcelos', '741852963'),

('03', 'Sofia Gomes', 'sofia.gomes@email.com', 'Rua da Fonte', '963204789', '1995-11-09', NULL, '4700-227', 'Braga', '369258147'),

('04', 'Daniel Pereira', 'daniel.pereira@email.com', 'Rua do Calvário', '930293847', '1990-03-01', '02', '4810-422', 'Vila Verde', '258369147'),

('05', 'Isabel Costa', 'isabel.costa@email.com', 'Avenida da República', '928364529', '1972-08-06', '03', '4740-328', 'Esposende', '159483726');

INSERT INTO Funcionario

(IdFuncionario, Nome, Veiculo, Email, Contacto, Contribuinte)

VALUES

('01', 'Graça Freitas', 'AX-12-AB', 'graca.freitas@email.com', '912345678', '793485061'),

('02', 'Maria Santos', 'BI-52-ID', 'maria.santos@email.com', '918273645', '718304526'),

('03', 'João Rodrigues', 'CA-13-GC', 'joao.rodrigues@email.com', '910928374', '847691253'),

('04', 'Ana Carvalho', 'DJ-49-FE', 'ana.carvalho@email.com', '916403829', '547891236');

INSERT INTO Fornecedor

(IdFornecedor, Designacao, Contribuinte, Contacto, CodPostalEndereco, LocalidadeEndereco, RuaEndereco, Email)

VALUES

('01', 'Empresa tvs/video', '123436219', '912345678', '1234-567', 'Lisboa', 'Rua das Flores', 'empresatvsvideo@gmail.com'),

('02', 'Empresa som', '985654821', '923456789', '5678-123', 'Porto', 'Avenida dos Pássaros', 'empresasom@gmail.com'),

('03', 'Empresa informatica', '123957456', '934567801', '4567-812', 'Braga', 'Estrada das Árvores', 'empresainformatica@hotmail.com'),

('04', 'Empresa acessórios', '741802963', '945678912', '6789-123', 'Coimbra', 'Rua dos Bares', 'empresaacessorios@gmail.com'),  
 ('05', 'Empresa telecomunicacoes', '369281470', '956789012', '1234-567', 'Faro', 'Rua das Praias', 'empresatelecomunicacoes@email.com');

INSERT INTO Produto

(IdProduto, Nome, Preco, Descricao, Tipo, Stock, Iva, Lucro)  
 VALUES  
 ('01', 'Tv 55 XPTO 4K', '1999.00', 'tv 55, marca: xpto', 'Tv / Video', '5', '0.23', '0.22'),  
 ('02', 'Tablet 8 okk', '399.00', 'tablet 8, marca: okk', 'Telecomunicacoes', '15', '0.23', '0.20'),  
 ('03', 'Smartphone Marca A Top', '1299.00', 'smartphone top gama, marca: a', 'Informatica', '50', '0.23', '0.15'),  
 ('04', 'Smartphone Marca A Medio', '799.00', 'smartphone med gama, marca: a', 'Informatica', '80', '0.23', '0.25'),  
 ('05', 'Smartphone Marca B Top', '1249.00', 'smartphone top gama, marca: b', 'Informatica', '60', '0.23', '0.25'),  
 ('06', 'TV 50 TSV', '899.00', 'tv 50, marca: tsv', 'Tv / Video', '8', '0.23', '0.28'),  
 ('07', 'Tablet 10 okk', '599.00', 'tablet 10, marca: okk', 'Telecomunicacoes', '35', '0.23', '0.20'),  
 ('08', 'Computador port 14 TSV', '749.90', 'portatil 14, marca: tsv', 'Informatica', '40', '0.23', '0.30'),  
 ('09', 'Smartphone Marca B Medio', '749.00', 'smartphone med gama, marca: b', 'Telecomunicacoes', '90', '0.23', '0.32'),  
 ('10', 'TV 60 STK', '1599.99', 'tv 60, marca: stk', 'Tv / Video', '8', '0.23', '0.20'),  
 ('11', 'Leitor BRAYs STK', '199.00', 'leitor blueray, marca: stk', 'Tv / Video', '35', '0.23', '0.26'),  
 ('12', 'Leitor CDs AUDVIS', '79.00', 'leitor cds, marca: audvis', 'Som', '30', '0.23', '0.20'),  
 ('13', 'Leitor LPs AUDVIS', '239.90', 'leitor lps, marca: audvis', 'Som', '19', '0.23', '0.15'),  
 ('14', 'Protetor ecra medida p', '19.00', 'protetor ecra pequeno', 'Acessorios', '130', '0.23', '0.50'),  
 ('15', 'Protetor ecra medida m', '29.00', 'protetor ecra medio', 'Acessorios', '100', '0.23', '0.50'),  
 ('16', 'Protetor ecra medida g', '39.00', 'protetor ecra grande', 'Acessorios', '80', '0.23', '0.50'),

```

        ('17', 'Capa okk 8',          '19.95', 'capa para tablet okk 8',    'Acessorios',    '20',
'0.23', '0.30'),
        ('18', 'Capa A Top',          '29.00', 'capa para smart a top',    'Acessorios',    '30',
'0.23', '0.25'),
        ('19', 'Capa B Medio',          '25.90', 'capa para smart b medio',    'Acessorios',
'35', '0.23', '0.20'),
        ('20', 'Kit limp AUDVIS',          '29.00', 'kit limpeza aparelhos audvis', 'Acessorios',
'40', '0.23', '0.20');

```

INSERT INTO Encomenda

```

        (IdEncomenda, DataHora_pedido, Custo, DataHora_entrega, IdCliente, IdFuncionario)
VALUES

```

```

        ('01', '2022-01-03 10:04:21', '853.90', '2022-01-05 12:05:10', '02', '01'),
        ('02', '2022-02-04 14:30:50', '79.00', '2022-02-07 16:13:02', '04', '02'),
        ('03', '2022-04-12 09:00:01', '1999.00', '2022-04-13 18:30:40', '05', '03'),
        ('04', '2022-06-21 15:05:00', '599.00', '2022-06-23 17:06:30', '01', '04'),
        ('05', '2022-07-13 11:31:42', '1313.90', '2022-07-15 13:23:07', '02', '02'),
        ('06', '2022-09-26 10:56:30', '199.00', null, '01', '04'),
        ('07', '2022-10-07 14:42:07', '749.90', null, '02', '01'),
        ('08', '2022-11-15 09:01:23', '1599.99', null, '03', '02'),
        ('09', '2022-12-09 15:07:10', '1367.00', null, '04', '03'),
        ('10', '2022-12-10 11:30:00', '268.90', null, '05', '01');

```

INSERT INTO Encomenda\_tem\_Produto

```

        (IdEncomenda, IdProduto, Preco, Quantidade)
VALUES

```

```

        ('01', '04', '799.00', '1'),
        ('01', '19', '25.90', '1'),
        ('01', '15', '29.00', '1'),
        ('02', '12', '79.00', '1'),
        ('03', '01', '1999.00', '1'),
        ('04', '07', '599.00', '1'),
        ('05', '05', '1249.00', '1'),
        ('05', '16', '39.00', '1'),
        ('05', '15', '25.90', '1'),
        ('06', '11', '199.00', '1'),
        ('07', '08', '749.90', '1'),
        ('08', '10', '1599.99', '1'),
        ('09', '03', '1299.00', '1'),
        ('09', '16', '39.00', '1'),

```



```
('09', '18', '29.00', '1'),  
('10', '13', '239.90', '1'),  
('10', '20', '29.00', '1');
```

```
INSERT INTO Produto_Fornecedor
```

```
    (IdProduto, IdFornecedor, Quantidade, Preco, DataReferencia)
```

```
VALUES
```

```
    ('01', '01', '5', '1638.52', '2022-12-30 08:47:00'),  
    ('02', '05', '15', '332.50', '2022-12-30 14:11:00'),  
    ('03', '03', '50', '1129.57', '2022-12-30 19:34:00'),  
    ('04', '03', '80', '639.20', '2022-12-31 09:21:00'),  
    ('05', '03', '60', '999.20', '2022-12-31 15:55:00'),  
    ('06', '01', '8', '702.34', '2022-12-31 21:08:00'),  
    ('07', '05', '35', '499.17', '2023-01-01 10:42:00'),  
    ('08', '03', '40', '576.15', '2023-01-01 16:16:00'),  
    ('09', '05', '90', '567.42', '2023-01-01 21:29:00'),  
    ('10', '01', '8', '1290.83', '2023-01-02 11:03:00'),  
    ('11', '01', '35', '157.94', '2023-01-02 16:37:00'),  
    ('12', '02', '30', '65.83', '2023-01-02 21:50:00'),  
    ('13', '02', '19', '208.61', '2023-01-03 11:24:00'),  
    ('14', '05', '130', '12.67', '2023-01-03 16:58:00'),  
    ('15', '05', '100', '19.33', '2023-01-03 22:11:00'),  
    ('16', '05', '80', '26.00', '2023-01-04 11:45:00'),  
    ('17', '05', '20', '15.35', '2023-01-04 17:19:00'),  
    ('18', '05', '30', '23.2', '2023-01-04 22:32:00'),  
    ('19', '05', '35', '21.58', '2023-01-05 12:06:00'),  
    ('20', '05', '40', '24.17', '2023-01-05 17:40:00');
```

### III. Anexo 3 – Script da Implementação das Interrogações

-----  
-- Queries  
-----

-----  
-- RE01-Apresentar o número de encomendas realizadas,  
-- a quantidade e o valor dos produtos vendidos,  
-- a determinado período.  
-----

```
SELECT COUNT(distinct a.IdEncomenda) as Encomendas,  
       SUM(Quantidade) as Quantidades, SUM(Preco) as Preco_Bruto  
FROM Encomenda a, Encomenda_tem_produto b  
       where (a.IdEncomenda = b.IdEncomenda and DataHora_pedido >=  
'2022-10-01');
```

-----  
-- RE02-Lista de clientes, que contenha o identificador,  
-- nome, contacto e e-mail.  
-----

```
SELECT IdCliente, Nome, Telemovel, Email FROM Cliente;
```

-----  
-- RE03-Os clientes devem ser inseridos no sistema  
-- quando realizam a primeira encomenda na loja.  
-- A cada cliente é pedido o nome, endereço, número,  
-- e-mail, contribuinte e data de nascimento.  
-----

```
SELECT * FROM Cliente  
       where (Historico is NULL);
```

-----  
-- RE04-Deve ser possível calcular o tempo de processamento  
-- entre o pedido da encomenda e a entrega da mesma.  
-----

```
SELECT IdEncomenda, (TIMESTAMPDIFF(day, DataHora_pedido, DataHora_entrega)) as  
'Dias',  
       (TIMESTAMPDIFF(hour, DataHora_pedido, DataHora_entrega)
```

```
-(TIMESTAMPDIFF(day, DataHora_pedido, DataHora_entrega)*24)) as 'Horas' FROM  
Encomenda;
```

```
-- -----  
-- RE05-O sistema deve ser capaz de listar os produtos  
-- por ordem de maior percentagem de lucro.  
-- -----
```

```
SELECT IdProduto, Nome, Preco, Lucro FROM Produto  
ORDER BY (Lucro) DESC;
```

```
-- -----  
-- RE06-Listar os Funcionários que geraram mais receita (lucro das vendas) a empresa.  
-- -----
```

```
SELECT IdFuncionario, sum(cast(c.Preco*Lucro as DECIMAL(6,2))) as 'Receita'  
FROM Encomenda a, Encomenda_tem_produto b, Produto c  
where (a.IdEncomenda = b.IdEncomenda and b.IdProduto = c.IdProduto)  
GROUP BY (IdFuncionario)  
ORDER BY (Receita) DESC;
```

```
-- -----  
-- RE07-Listar os Clientes por maior valor total gasto na loja de todas as encomendas que fez.  
-- -----
```

```
SELECT b.IdCliente, Nome, Sum(Custo) as 'Total_Gasto' FROM Cliente a, Encomenda b  
where(a.IdCliente = b.IdCliente)  
GROUP BY (IdCliente)  
ORDER BY (Total_Gasto) DESC;
```

```
-- -----  
-- RE08-O sistema deve listar todos os identificadores dos produtos com stock disponível.  
-- -----
```

```
SELECT IdProduto, Nome, Stock FROM Produto  
where (Stock > 0);
```

```
-- -----  
-- RE09-Qual o fornecedor que vende mais produtos a loja.  
-- -----
```

```
SELECT a.IdFornecedor, Designacao, Count(distinct IdProduto) as 'Produtos'  
FROM Fornecedor a, Produto_Fornecedor b  
where(a.IdFornecedor = b.IdFornecedor)  
GROUP BY (IdFornecedor)
```

ORDER BY (Produtos) DESC;

-----

-- RE10-Ver o historico de compras de um cliente na loja.

-----

```
SELECT a.IdEncomenda, a.IdCliente, b.IdProduto, c.nome
      FROM Encomenda a, Encomenda_tem_Produto b, Produto c
where(a.IdCLiente = 2 and a.IdEncomenda = b.IdEncomenda
      and b.IdProduto = c.IdProduto and a.DataHora_entrega is not null);
```

## IV. Anexo 4 – Script da Criação das Views

```
-----  
-- Vistas  
-----  
  
-----  
-- V01 - Número de vezes que cada produto foi vendido  
-- DROP VIEW V01;  
-----  
  
CREATE VIEW V01 AS  
    SELECT  a.IdProduto,  Nome,  Count(a.IdProduto)  AS  Total_Vendas  FROM  
Encomenda_tem_Produto a, Produto b  
    where(a.IdProduto = b.IdProduto)  
    GROUP BY (IdProduto)  
    ORDER BY (Total_Vendas) DESC;  
  
-----  
-- V02 - Unidades por encomenda  
-- DROP VIEW V02;  
-----  
  
CREATE VIEW V02 AS  
    SELECT  IdEncomenda,  count(distinct  IdProduto)  AS  'Unidades'  FROM  
Encomenda_tem_Produto  
    GROUP BY (IdEncomenda);  
  
-----  
-- V03 - Média de tempo de entrega de uma encomenda  
-- DROP VIEW V03;  
-----  
  
CREATE VIEW V03 AS  
    SELECT  Cast(Sum(TIMESTAMPDIFF(day, DataHora_pedido, DataHora_entrega))  
    /Count(DataHora_entrega) as DECIMAL(2,0)) as 'Dias',  
    Cast(Sum(TIMESTAMPDIFF(hour, DataHora_pedido,  
DataHora_entrega)  
    -(TIMESTAMPDIFF(day, DataHora_pedido, DataHora_entrega)*24))  
    /Count(DataHora_entrega) as DECIMAL(2,1)) as 'Horas' FROM Encomenda;
```

## V. Anexo 4 – *Script* da Criação do Procedimento

```
-----  
-- Procedimentos  
-----  
-- Criacao de um procedimento para nos indicar os 3 maiores pagamentos realizados  
-- e respectivos clientes.  
DELIMITER $$  
CREATE PROCEDURE sp3TopPagamentos()  
BEGIN  
    SELECT b.IdCliente, Nome, Custo AS 'Pagamento' FROM Cliente a, Encomenda b  
        where(a.IdCliente = b.IdCliente)  
        ORDER BY Pagamento DESC  
        LIMIT 3;  
END $$  
  
CALL sp3TopPagamentos;  
-----  
-- DROP PROCEDURE sp3TopPagamentos;  
-----
```