



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

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

Ano Letivo de 2022/2023

### **Entrega de Medicamentos ao Domicílio**

**Bruno Fernandes, Carlos Costa, Tiago Silva, Tomás Pereira**

Janeiro, 2023

# BD

Data de Receção	
Responsável	
Avaliação	
Observações	

## Entrega de Medicamentos ao Domicílio

**Bruno Fernandes, Carlos Costa, Tiago Silva, Tomás Pereira**  
**A95972, A94543, A97450, A97402**

Janeiro, 2023

## Resumo

Neste relatório vamos apresentar o processo de criação de um Sistema de Base de Dados para uma empresa de entregas de medicamentos ao domicílio que facilite a criação, consulta e atualização dos pedidos efetuados pelos clientes. Nos capítulos seguintes descreveremos em detalhe todas as fases da implementação.

Inicialmente, realizamos uma investigação acerca da empresa para compreendermos o contexto e as necessidades do nosso cliente. Definimos então os objetivos e efetuamos a sua implementação. Em seguida, procedemos ao levantamento de requisitos com os quais elaboramos, através a ferramenta *brModelo*, o modelo conceptual que serve como estrutura para a Base de Dados. Levando em consideração este modelo, desenvolvemos o modelo lógico, utilizando a ferramenta *MySQL Workbench*, tendo em conta a forma de como os dados serão armazenados bem como os seus relacionamentos.

Posteriormente, na modelação física, traduzimos o modelo anterior para código SQL e implementamos todas as *queries* necessárias para uma fácil visualização e interpretação dos dados.

**Área de Aplicação:** Criação e Implementação de um Sistema de Base de Dados.

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

# Índice

Resumo	i
Índice	ii
Índice de Figuras	iii
Índice de Tabelas	iv
1. Definição do Sistema	1
1.1. Contexto de Aplicação e Fundamentação do Sistema	1
1.2. Motivação e Objetivos do Trabalho	1
2. Levantamento e Análise de Requisitos	3
3. Modelação Conceptual	5
4. Modelação Lógica	10
5. Implementação Física	13
6. Conclusões e Trabalho Futuro	27
7. Referências Bibliográficas	27
Anexos	28
I. <i>Script</i> da criação da Base de Dados	28
II. <i>Script</i> do povoamento	34
III. <i>Script</i> de <i>queries</i>	39

## Índice de Figuras

Figura 1 – Diagrama de GANTT	2
Figura 2 – Modelo conceptual	10
Figura 3 – Modelo lógico	11
Figura 4 - Query para visualizar o horário e os medicamentos disponíveis em cada farmácia	11
Figura 5 – Query para consultar todos os funcionários da empresa e o respetivo distrito onde trabalham	12
Figura 6 – Query para listar os clientes por ordem decrescente de despesa	12
Figura 7 - Código SQL para criar a tabela Gestor	13
Figura 8 - Código SQL para criar a tabela Administrador	13
Figura 9 - Código SQL para criar a tabela Estafeta	13
Figura 10 - Código SQL para criar a tabela Endereco	14
Figura 11 - Código SQL para criar a tabela Cliente	14
Figura 12 - Código SQL para criar a tabela Pedido	15
Figura 13 - Código SQL para criar a tabela Medicamento	15
Figura 14 - Código SQL para criar a tabela Pedido_inclui_Medicamento	16
Figura 15 - Código SQL para criar a tabela Farmacia	16
Figura 16 - Código SQL para criar a tabela ReceitaMedica	17
Figura 17 - Código SQL para criar a tabela Farmacia_fornece_Medicamento	17
Figura 18 - Código SQL para visualizar o horário e os medicamentos disponíveis em cada farmácia	18
Figura 19 - Resultado da query da Figura 18	18
Figura 20 - Código SQL para consultar todos os funcionários da empresa e o respetivo distrito onde trabalham	18
Figura 21 - Resultado da query da Figura 20	19
Figura 22 - Código SQL para listar os clientes por ordem decrescente de despesa	19
Figura 23 - Resultado da query da Figura 22	19
Figura 24 - Código SQL para a implementação da vista	19
Figura 25 – Resultado da vista	20
Figura 26 - Código SQL para criação do índice	23
Figura 27 – Tempo de execução antes da indexação (0.015 segundos)	23
Figura 28 – Tempo de execução após a indexação (0.0001 segundos)	23
Figura 29 - Procedimento que atualiza o custo de um medicamento	24
Figura 30 – Custo do medicamento antes de executar o procedimento	24
Figura 31 – Custo do medicamento após executar o procedimento	24
Figura 32 - Procedimento que indica os 3 clientes com mais compras	24
Figura 33 - Resultado do procedimento da Figura 32	24

Figura 34 - Procedimento que indica os estafetas que realizaram mais entregas	25
Figura 35 - Resultado do procedimento da Figura 34	25
Figura 36 – Procedimento que lista os pedidos que ainda não foram entregues num certo distrito	25
Figura 37 - Resultado do procedimento da Figura 36	25
Figura 38 - Procedimento que indica o total de pedidos por cada distrito	26
Figura 39 - Resultado do procedimento da Figura 38	26

## Índice de Tabelas

Tabela 1 - Caracterização das entidades	5
Tabela 2 - Caracterização dos relacionamentos	6
Tabela 3 - Caracterização dos atributos da entidade Cliente	6
Tabela 4 - Caracterização dos atributos da entidade Pedido	7
Tabela 5 - Caracterização dos atributos da entidade Medicamento	7
Tabela 6 - Caracterização dos atributos da entidade Farmácia	8
Tabela 7 - Caracterização dos atributos da entidade ReceitaMédica	8
Tabela 8 - Caracterização dos atributos da entidade Gestor	8
Tabela 9 - Caracterização dos atributos da entidade Administrador	8
Tabela 10 - Caracterização dos atributos da entidade Estafeta	9
Tabela 11 - Caracterização dos atributos do relacionamento N:M Pedido_inclui_Medicamento	9
Tabela 12 – Descrição do tamanho de cada tipo de dados	20
Tabela 13 - Tamanho dos atributos da entidade Cliente	21
Tabela 14 - Tamanho dos atributos da entidade Pedido	21
Tabela 15 - Tamanho dos atributos da entidade Medicamento	21
Tabela 16 - Tamanho dos atributos da entidade Farmácia	21
Tabela 17 - Tamanho dos atributos da entidade ReceitaMédica	22
Tabela 18 - Tamanho dos atributos da entidade Gestor	22
Tabela 19 - Tamanho dos atributos da entidade Administrador	22
Tabela 20 - Tamanho dos atributos da entidade Estafeta	22
Tabela 21 - Tamanho dos atributos do relacionamento N:M Pedido_inclui_Medicamento	23

# **1. Definição do Sistema**

## **1.1. Contexto de Aplicação e Fundamentação do Sistema**

Uma empresa de entregas ao domicílio, fundada em 2003 pelo Sr. Joaquim Pereira, começou por fazer distribuição de medicamentos no distrito de Braga. A sua empresa recebe os pedidos e, através da colaboração com várias farmácias da região, recebe e transporta os medicamentos ao domicílio. Face ao aumento da esperança média de vida e ao envelhecimento da população sentiu-se um aumento do número de pedidos nesta região e novos noutras regiões. Consequentemente, o Sr. Joaquim decidiu aumentar a área de distribuição para os distritos de Viana do Castelo, Porto e Aveiro.

Para cada distrito, a empresa tem 10 funcionários, dos quais 1 é gestor, 2 são administrativos e 7 fazem as entregas.

Devido ao aumento da área abrangida para entregas, houve um grande crescimento do número de funcionários e de pedidos. Assim, tornou-se difícil o controlo das encomendas a serem realizadas e da gestão dos turnos dos funcionários.

Por estes motivos, decidiram informatizar os seus serviços e, consequentemente, melhorar, de forma significativa, a qualidade do seu trabalho. Para tal, tiveram a necessidade de implementar um Sistema de Bases de Dados.

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

Numa reunião entre os gestores das várias áreas foram definidos os objetivos, que pretendem alcançar com a implementação de um Sistema de Bases de Dados:

- Organizar o modelo de negócio, melhorando a capacidade de gestão e registo de entregas;
- Controlar as entregas que foram feitas por distrito e por funcionário;
- Ter registo dos gastos e dos ganhos da empresa automaticamente;
- Organizar os recursos materiais e humanos com maior facilidade;
- Facilitar o acesso e rapidez com que se consultam os dados;
- Garantir a segurança de todos os dados;

- Guardar os dados de cada cliente, tal como as suas preferências e tendências, de forma a ser possível prever os seus pedidos futuros.

### 1.3. Análise da Viabilidade do Processo

Com a implementação da Base de Dados a gestão da empresa prevê:

- Informatizar os registos, com centralização diária dos dados;
- Gerir os recursos humanos consoante as horas em que há mais pedidos;
- Com a automatização dos registos, melhorar o tempo de entrega e consequentemente conseguir aumentar as entregas realizadas;
- Traçar o perfil de cada cliente de forma a entender as suas necessidades e ocasionalmente fazer descontos no custo da entrega;
- Registar a data e hora de cada entrega para posteriores análises com objetivo de aumentar a produtividade;
- Com o registo dos valores associados a cada entrega, gerir de forma eficiente a faturação, maximizando os lucros.

### 1.4. Recursos e Equipa de Trabalho

A equipa de trabalho reunida para a realização desta base de dados é constituída por 4 especialistas em bases de dados, o Sr. Joaquim, os 4 gestores, 4 administrativos (um por cada distrito), 2 estafetas por distrito para testar a implementação e clientes para realizarem inquéritos sobre a qualidade do serviço.

Para além disso, serão necessários 1 servidor, 4 máquinas, software de desenvolvimento e software de gestão em cada máquina.

### 1.5. Plano de Execução do Projeto

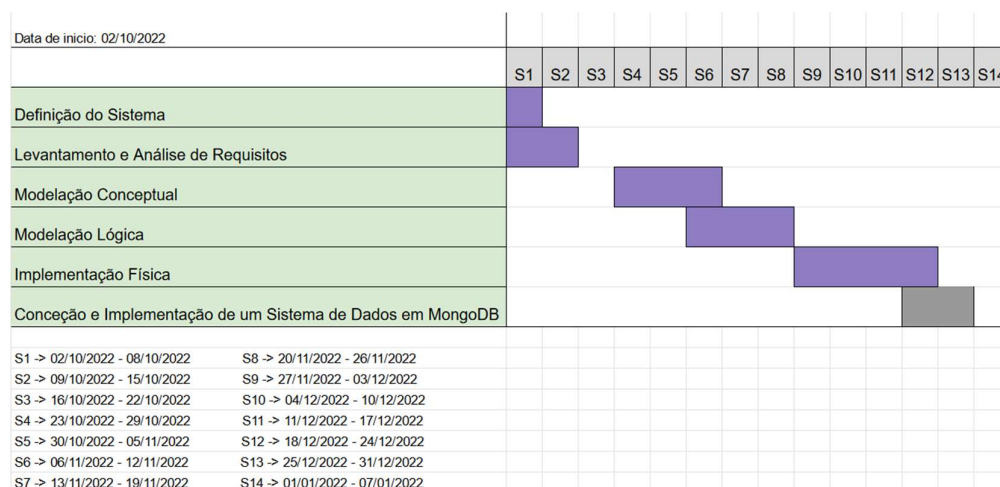


Figura 1 – Diagrama de GANTT



A definição, fundamentação, objetivos e viabilidade do sistema foram analisados, em reunião, pelos gestores, administradores, equipa de desenvolvimento e pelo Sr. Joaquim, que em conjunto validaram as decisões tomadas. Analisaram o diagrama de *GANTT* e retificaram-no. Posto isto, o gestor geral decidiu avançar com o processo de desenvolvimento e implementação do Sistema de Bases de Dados e contratou a equipa de desenvolvimento. O processo passou à fase seguinte.

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

### **2.1. Método de Levantamento e de Análise de Requisitos**

#### **Adotado**

A empresa foi observada durante uma semana por um membro da equipa de desenvolvimento. Após algumas entrevistas a funcionários e questionários a clientes foram estimados os requisitos necessários.

### **2.2. Organização dos Requisitos Levantados**

#### **2.2.1. Requisitos de Descrição**

- Existem 3 tipos de funcionários: administrativos, gestores e estafetas;
- Um funcionário deve ser identificado com um identificador único (número), nome e email;
- Cada funcionário deve estar associado a um distrito;
- Cada estafeta tem um veículo que é identificado por uma matrícula;
- A cada estafeta estão associados um ou mais pedidos;
- A cada pedido deve estar associado um identificador, o custo da entrega, o valor de faturação, a data do pedido e a data em que foi efetuada a entrega;
- Cada pedido está associado a um cliente, a um estafeta e a um ou mais medicamentos;
- Cada medicamento no pedido tem associado a quantidade, o custo, o preço e IVA;
- Cada pedido pode ser de carácter urgente ou não;
- Os medicamentos têm um identificador, a sua designação, uma descrição (opcional) e um custo;

- O cliente deve possuir receitas médicas com prescrição dos medicamentos, quando necessário;
- Cada receita médica tem um identificador e o nome do Médico que a prescreveu;
- O registo de um cliente deve conter um identificador único (número de contribuinte), o nome completo, um endereço postal (rua, localidade, distrito e código postal), data de nascimento, número do SNS, número de telefone e email (facultativo);
- Cada farmácia fornecedora deve ter um identificador, nome, número de telefone, email, endereço e horário de funcionamento.

### **2.2.2. Requisitos de Exploração**

- Todos os meses o sistema deve apresentar um relatório com todos os registos de entregas realizadas pelos estafetas;
- Esse relatório mensal deve conter ainda o tempo médio que a entrega demora;
- Quando é concluída a entrega regista-se a data e a hora e o veículo em que foi efetuada;
- O registo de cada cliente é obrigatório;
- Por cada cliente é obrigatório registar o seu nome, data de nascimento, endereço, contacto (email e telefónico), número do SNS, número de contribuinte;
- Calcular o número de pedidos realizados por cada estafeta;
- Listar os clientes por ordem decrescente de despesa;
- Consultar todos os funcionários da empresa e o respetivo distrito onde trabalham;
- Visualizar o horário e os medicamentos disponíveis em cada farmácia;
- Se um funcionário fizer mais do que 50 entregas por semana terá direito a um bónus de 10%;
- Se o pedido for urgente terá uma taxa adicional.

### **2.2.3. Requisitos de Controlo**

- Os administrativos apenas devem aceder aos dados relativos ao registo de cada cliente;
- Apenas os administrativos têm a possibilidade de registar e alterar as entregas;
- Apenas os gestores têm a possibilidade de fazer a requisição dos medicamentos a uma determinada farmácia;
- Os estafetas só têm acesso à visualização dos pedidos que lhes são atribuídos;
- No pedido são indicados os dados do cliente e do estafeta associado;
- A cada pedido não pode estar associado mais do que um estafeta;
- Cada gestor só tem acesso aos dados do distrito a que está associado.

### 2.3. Análise e Validação Geral dos Requisitos

Após uma reunião entre todos os intervenientes onde foram analisados os requisitos propostos, realizaram-se algumas correções e estes foram aprovados.

## 3. Modelação Conceptual

### 3.1. Apresentação da abordagem de modelação realizada

Para a criação de modelo conceptual utilizamos o software brModelo. Começamos por identificar as entidades, com os devidos atributos, e posteriormente as relações a partir dos requisitos levantados.

### 3.2. Identificação e caracterização das entidades

Entidade	Descrição
Pedido	Representa os pedidos realizados por cada cliente
Medicamento	Contém informação acerca do objeto de entrega
Cliente	Informação acerca das pessoas (clientes) que realizam os pedidos de entrega
Estafeta	Esta entidade reúne as informações pessoais de quem realiza a entrega, nomeadamente, o distrito em que atua
Administrador	É quem faz o registo dos pedidos de um determinado distrito em que atua
Gestor	Contém as suas informações pessoais, tais como o distrito em que atua, contactando as farmácias
Farmácia	Representa o fornecedor dos produtos, contem a sua identificação, horário, os seus contactos e o seu endereço

Tabela 1 - Caracterização das entidades

### 3.3. Identificação e caracterização dos relacionamentos

Entidade A	Relacionamento	Cardinalidade	Entidades B	Cardinalidade
Cliente	realiza	(1,1)	Pedido	(1,n)
Cliente	possui	(1,1)	ReceitaMédica	(0,n)
Estafeta	entrega	(1,1)	Pedido	(1,n)
Administrador	registra	(1,1)	Pedido	(1,n)
Gestor	contacta	(1,1)	Farmácia	(1,n)
Farmácia	fornece	(1,n)	Medicamento	(1,n)
Pedido	inclui	(1,n)	Medicamento	(1,n)
ReceitaMédica	prescreve	(0,n)	Medicamento	(0,1)

Tabela 2 - Caracterização dos relacionamentos

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

Atributo	Tipo de Atributo	Tipo de Dados	Descrição
NúmeroContribuinte	Chave	INT	Número que identifica o cliente
Nome	Simples	VARCHAR(45)	Nome do cliente
DataNascimento	Simples	VARCHAR(10)	Data de nascimento do cliente
NúmeroSNS	Simples	INT	Número de saúde do cliente
Rua\Endereço	Composto	VARCHAR(45)	Rua referente ao endereço do cliente
Localidade\Endereço	Composto	VARCHAR(45)	Localidade referente ao endereço do cliente
CodigoPostal\Endereço	Composto	VARCHAR(45)	Código postal referente ao endereço do cliente
Distrito\Endereço	Composto	VARCHAR(45)	Distrito referente ao endereço do cliente
Telefone	Simples	INT	Número de telefone do cliente
Email	Simples (opcional)	VARCHAR(45)	Endereço de email do cliente

Tabela 3 - Caracterização dos atributos da entidade Cliente

<b>Atributo</b>	<b>Tipo de Atributo</b>	<b>Tipo de Dados</b>	<b>Descrição</b>
Id	Chave	INT	Número que identifica o pedido
CustoEntrega	Simples	DECIMAL(5,2)	Custo referente à entrega
DataPedido	Simples	DATETIME	Data em que é efetuado o pedido
DataEntrega	Simples (opcional)	DATETIME	Data em que é efetuada a entrega
Urgente	Simples	BINARY(1)	Identifica se um pedido é ou não urgente
ValorFaturação	Simples	DECIMAL(5,2)	Valor da faturação de um pedido

Tabela 4 - Caracterização dos atributos da entidade Pedido

<b>Atributo</b>	<b>Tipo de Atributo</b>	<b>Tipo de Dados</b>	<b>Descrição</b>
Id	Chave	INT	Número que identifica o medicamento
Custo	Simples	DECIMAL(5,2)	Custo do medicamento
Designação	Simples	VARCHAR(45)	Nome do medicamento
Descrição	Simples(opcional)	VARCHAR(100)	Descrição do medicamento

Tabela 5 - Caracterização dos atributos da entidade Medicamento

<b>Atributo</b>	<b>Tipo de Atributo</b>	<b>Tipo de Dados</b>	<b>Descrição</b>
Id	Chave	INT	Número que identifica a farmácia
Nome	Simples	VARCHAR(45)	Nome da farmácia
Telefone	Simples	VARCHAR(45)	Número de telefone da farmácia
Email	Simples	VARCHAR(45)	Endereço de email da farmácia
Horário	Simples	VARCHAR(200)	Horário de funcionamento da farmácia

Rua\Endereço	Composto	VARCHAR(45)	Rua referente ao endereço da farmácia
Localidade\Endereço	Composto	VARCHAR(45)	Localidade referente ao endereço da farmácia
CodigoPostal\Endereço	Composto	VARCHAR(45)	Código Postal referente ao endereço da farmácia
Distrito\Endereço	Composto	VARCHAR(45)	Distrito referente ao endereço da farmácia

Tabela 6 - Caracterização dos atributos da entidade Farmácia

Atributo	Tipo de Atributo	Tipo de Dados	Descrição
Id	Chave	INT	Número que identifica a receita médica
Médico	Simples	VARCHAR(45)	Médico que prescreve a receita médica

Tabela 7 - Caracterização dos atributos da entidade ReceitaMédica

Atributo	Tipo de Atributo	Tipo de Dados	Descrição
Id	Chave	INT	Número que identifica o gestor
Nome	Simples	VARCHAR(45)	Nome do gestor
Email	Simples	VARCHAR(45)	Endereço de email do gestor
Distrito	Simples	VARCHAR(45)	Distrito onde atua o gestor

Tabela 8 - Caracterização dos atributos da entidade Gestor

Atributo	Tipo de Atributo	Tipo de Dados	Descrição
Id	Chave	INT	Número que identifica o administrador
Nome	Simples	VARCHAR(45)	Nome do administrador
Email	Simples	VARCHAR(45)	Endereço de email do administrador
Distrito	Simples	VARCHAR(45)	Distrito onde atua o administrador

Tabela 9 - Caracterização dos atributos da entidade Administrador

<b>Atributo</b>	<b>Tipo de Atributo</b>	<b>Tipo de Dados</b>	<b>Descrição</b>
Id	Chave	INT	Número que identifica o estafeta
Nome	Simples	VARCHAR(45)	Nome do estafeta
Email	Simples	VARCHAR(45)	Endereço de email do estafeta
Distrito	Simples	VARCHAR(45)	Distrito onde atua o estafeta
MatrículaVeículo	Simples	VARCHAR(8)	Matrícula do veículo utilizado pelo estafeta

Tabela 10 - Caracterização dos atributos da entidade Estafeta

<b>Atributo</b>	<b>Tipo de Atributo</b>	<b>Tipo de Dados</b>	<b>Descrição</b>
Quantidade	Simples	INT	Quantidade total de medicamentos
Custo	Simples	DECIMAL(5,2)	Custo dos medicamentos
Iva	Simples	INT	Taxa de iva aplicada aos medicamentos
Preço	Simples	DECIMAL(5,2)	Preço total dos medicamentos

Tabela 11 - Caracterização dos atributos do relacionamento N:M Pedido\_inclui\_Medicamento

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

O modelo conceptual foi criado utilizando o *software brModelo*, com base nos atributos, entidades e relacionamentos anteriormente descritos.

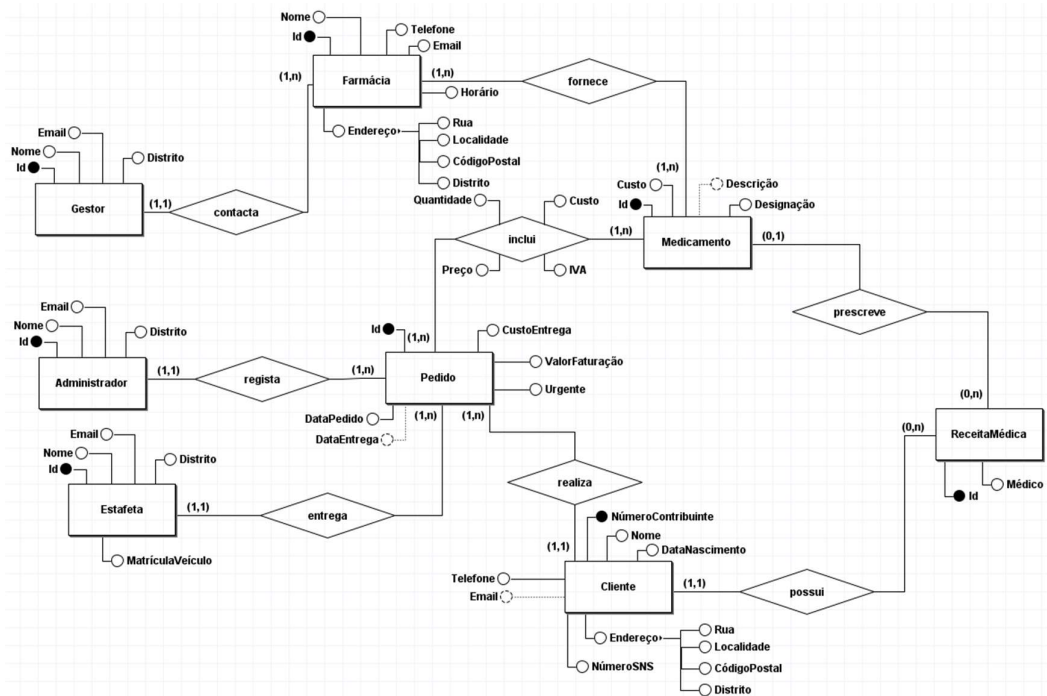


Figura 2 – Modelo conceptual

## 4. Modelação Lógica

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

Com base no modelo conceptual, usando a ferramenta MySQL, foram criadas tabelas, uma para cada entidade, relacionamento N:M e atributo composto.

### 4.2. Normalização de dados

Após a análise das tabelas do modelo lógico, verificamos que o esquema está normalizado até à terceira forma normal.



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

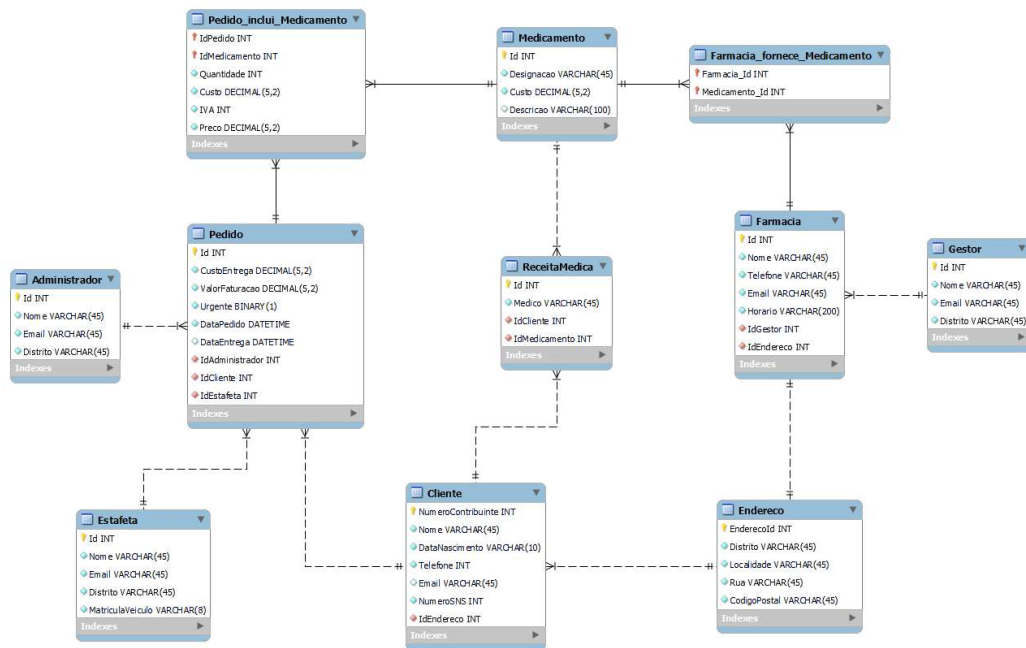


Figura 3 – Modelo lógico

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

Para validar o modelo lógico, convertemos alguns requisitos de exploração para álgebra relacional e testamos se as *queries* eram satisfeitas utilizando a ferramenta *RelaX*.

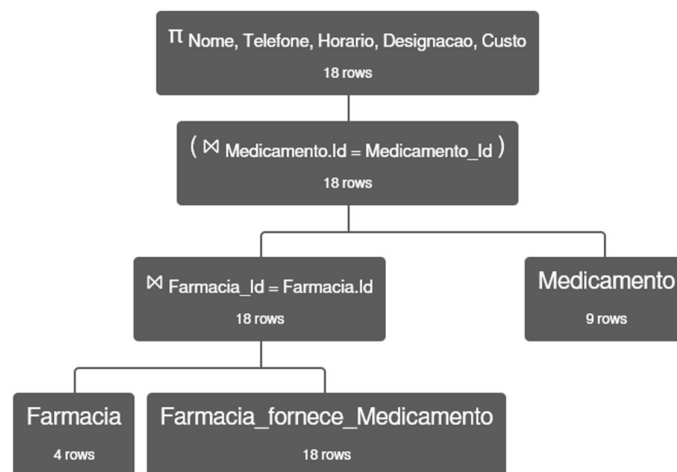


Figura 4 - Query para visualizar o horário e os medicamentos disponíveis em cada farmácia

$\pi_{\text{Nome, Telefone, Horário, Designação, Custo}}((\text{Farmacia}) \bowtie_{\text{Farmacia\_Id = Farmacia.Id}} (\text{Farmacia\_fornece\_Medicamento}) \bowtie_{\text{Medicamento.Id = Medicamento\_Id}} (\text{Medicamento}))$

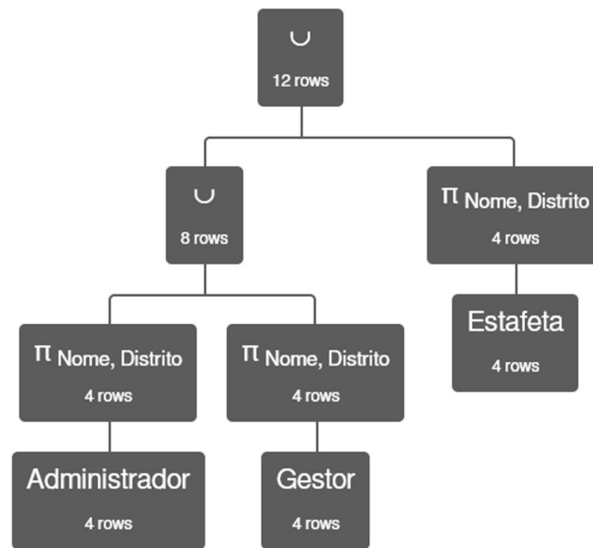


Figura 5 – Query para consultar todos os funcionários da empresa e o respectivo distrito onde trabalham

$\pi_{\text{Nome, Distrito}}(\text{Administrador}) \cup \pi_{\text{Nome, Distrito}}(\text{Gestor}) \cup \pi_{\text{Nome, Distrito}}(\text{Estafeta})$

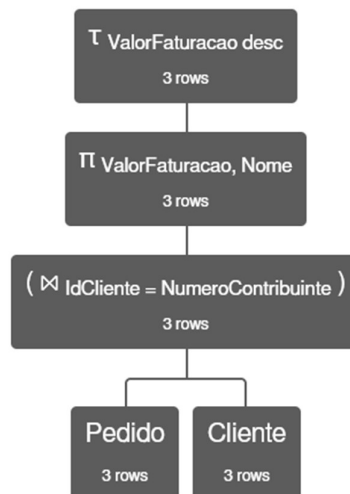


Figura 6 – Query para listar os clientes por ordem decrescente de despesa

$\tau_{\text{ValorFaturacao desc}} \pi_{\text{ValorFaturacao, Nome}} ((\text{Pedido}) \bowtie_{\text{IdCliente = NumeroContribuinte}} (\text{Cliente}))$

## 5. Implementação Física

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

```
CREATE TABLE IF NOT EXISTS `mydb`.`Gestor` (  
  `Id` INT NOT NULL,  
  `Nome` VARCHAR(45) NOT NULL,  
  `Email` VARCHAR(45) NOT NULL,  
  `Distrito` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`Id`),  
  UNIQUE INDEX `Id_UNIQUE` (`Id` ASC) VISIBLE)  
ENGINE = InnoDB;
```

Figura 7 - Código SQL para criar a tabela Gestor

```
CREATE TABLE IF NOT EXISTS `mydb`.`Administrador` (  
  `Id` INT NOT NULL,  
  `Nome` VARCHAR(45) NOT NULL,  
  `Email` VARCHAR(45) NOT NULL,  
  `Distrito` VARCHAR(45) NOT NULL,  
  UNIQUE INDEX `Id_UNIQUE` (`Id` ASC) VISIBLE,  
  PRIMARY KEY (`Id`))  
ENGINE = InnoDB;
```

Figura 8 - Código SQL para criar a tabela Administrador

```
CREATE TABLE IF NOT EXISTS `mydb`.`Estafeta` (  
  `Id` INT NOT NULL,  
  `Nome` VARCHAR(45) NOT NULL,  
  `Email` VARCHAR(45) NOT NULL,  
  `Distrito` VARCHAR(45) NOT NULL,  
  `MatriculaVeiculo` VARCHAR(8) NOT NULL,  
  PRIMARY KEY (`Id`),  
  UNIQUE INDEX `Id_UNIQUE` (`Id` ASC) VISIBLE)  
ENGINE = InnoDB;
```

Figura 9 - Código SQL para criar a tabela Estafeta

```

CREATE TABLE IF NOT EXISTS `mydb`.`Endereco` (
  `EnderecoId` INT NOT NULL,
  `Distrito` VARCHAR(45) NOT NULL,
  `Localidade` VARCHAR(45) NOT NULL,
  `Rua` VARCHAR(45) NOT NULL,
  `CodigoPostal` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`EnderecoId`),
  UNIQUE INDEX `EnderecoId_UNIQUE` (`EnderecoId` ASC) VISIBLE)
ENGINE = InnoDB;

```

Figura 10 - Código SQL para criar a tabela Endereco

```

CREATE TABLE IF NOT EXISTS `mydb`.`Cliente` (
  `NumeroContribuinte` INT NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `DataNascimento` VARCHAR(10) NOT NULL,
  `Telefone` INT NOT NULL,
  `Email` VARCHAR(45) NULL,
  `NumeroSNS` INT NOT NULL,
  `IdEndereco` INT NOT NULL,
  UNIQUE INDEX `NumeroContribuinte_UNIQUE` (`NumeroContribuinte` ASC) VISIBLE,
  PRIMARY KEY (`NumeroContribuinte`),
  INDEX `fk_Cliente_Endereco1_idx` (`IdEndereco` ASC) VISIBLE,
  CONSTRAINT `fk_Cliente_Endereco1`
    FOREIGN KEY (`IdEndereco`)
    REFERENCES `mydb`.`Endereco` (`EnderecoId`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

```

Figura 11 - Código SQL para criar a tabela Cliente

```

CREATE TABLE IF NOT EXISTS `mydb`.`Pedido` (
  `Id` INT NOT NULL,
  `CustoEntrega` DECIMAL(5,2) NOT NULL,
  `ValorFaturacao` DECIMAL(5,2) NOT NULL,
  `Urgente` BINARY(1) NOT NULL,
  `DataPedido` DATETIME NOT NULL,
  `DataEntrega` DATETIME NULL,
  `IdAdministrador` INT NOT NULL,
  `IdCliente` INT NOT NULL,
  `IdEstafeta` INT NOT NULL,
  PRIMARY KEY (`Id`),
  UNIQUE INDEX `Id_UNIQUE` (`Id` ASC) VISIBLE,
  INDEX `fk_Pedido_Administrador1_idx` (`IdAdministrador` ASC) VISIBLE,
  INDEX `fk_Pedido_Cliente1_idx` (`IdCliente` ASC) VISIBLE,
  INDEX `fk_Pedido_Estafeta1_idx` (`IdEstafeta` ASC) VISIBLE,
  CONSTRAINT `fk_Pedido_Administrador1`
    FOREIGN KEY (`IdAdministrador`)
      REFERENCES `mydb`.`Administrador` (`Id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_Pedido_Cliente1`
    FOREIGN KEY (`IdCliente`)
      REFERENCES `mydb`.`Cliente` (`NumeroContribuinte`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_Pedido_Estafeta1`
    FOREIGN KEY (`IdEstafeta`)
      REFERENCES `mydb`.`Estafeta` (`Id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Figura 12 - Código SQL para criar a tabela Pedido

```

CREATE TABLE IF NOT EXISTS `mydb`.`Medicamento` (
  `Id` INT NOT NULL,
  `Designacao` VARCHAR(45) NOT NULL,
  `Custo` DECIMAL(5,2) NOT NULL,
  `Descricao` VARCHAR(100) NULL,
  PRIMARY KEY (`Id`),
  UNIQUE INDEX `Id_UNIQUE` (`Id` ASC) VISIBLE)
ENGINE = InnoDB;

```

Figura 13 - Código SQL para criar a tabela Medicamento

```

CREATE TABLE IF NOT EXISTS `mydb`.`Pedido_inclui_Medicamento` (
  `IdPedido` INT NOT NULL,
  `IdMedicamento` INT NOT NULL,
  `Quantidade` INT NOT NULL,
  `Custo` DECIMAL(5,2) NOT NULL,
  `IVA` INT NOT NULL,
  `Preco` DECIMAL(5,2) NOT NULL,
  PRIMARY KEY (`IdMedicamento`, `IdPedido`),
  INDEX `fk_Pedido_has_Medicamento_Medicamento1_idx` (`IdMedicamento` ASC) VISIBLE,
  INDEX `fk_Pedido_has_Medicamento_Pedido1_idx` (`IdPedido` ASC) VISIBLE,
  CONSTRAINT `fk_Pedido_has_Medicamento_Pedido1`
    FOREIGN KEY (`IdPedido`)
      REFERENCES `mydb`.`Pedido` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Pedido_has_Medicamento_Medicamento1`
    FOREIGN KEY (`IdMedicamento`)
      REFERENCES `mydb`.`Medicamento` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Figura 14 - Código SQL para criar a tabela Pedido\_inclui\_Medicamento

```

CREATE TABLE IF NOT EXISTS `mydb`.`Farmacia` (
  `Id` INT NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `Telefone` VARCHAR(45) NOT NULL,
  `Email` VARCHAR(45) NOT NULL,
  `Horario` VARCHAR(200) NOT NULL,
  `IdGestor` INT NOT NULL,
  `IdEndereco` INT NOT NULL,
  PRIMARY KEY (`Id`),
  INDEX `fk_Farmacia_Gestor1_idx` (`IdGestor` ASC) VISIBLE,
  UNIQUE INDEX `Id_UNIQUE` (`Id` ASC) VISIBLE,
  INDEX `fk_Farmacia_Endereco1_idx` (`IdEndereco` ASC) VISIBLE,
  CONSTRAINT `fk_Farmacia_Gestor1`
    FOREIGN KEY (`IdGestor`)
      REFERENCES `mydb`.`Gestor` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Farmacia_Endereco1`
    FOREIGN KEY (`IdEndereco`)
      REFERENCES `mydb`.`Endereco` (`EnderecoId`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Figura 15 - Código SQL para criar a tabela Farmacia

```

CREATE TABLE IF NOT EXISTS `mydb`.`ReceitaMedica` (
  `Id` INT NOT NULL,
  `Medico` VARCHAR(45) NOT NULL,
  `IdCliente` INT NOT NULL,
  `IdMedicamento` INT NOT NULL,
  PRIMARY KEY (`Id`),
  UNIQUE INDEX `Id_UNIQUE` (`Id` ASC) VISIBLE,
  INDEX `fk_ReceitaMedica_Cliente1_idx` (`IdCliente` ASC) VISIBLE,
  INDEX `fk_ReceitaMedica_Medicamento1_idx` (`IdMedicamento` ASC) VISIBLE,
  CONSTRAINT `fk_ReceitaMedica_Cliente1`
    FOREIGN KEY (`IdCliente`)
      REFERENCES `mydb`.`Cliente` (`NumeroContribuinte`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_ReceitaMedica_Medicamento1`
    FOREIGN KEY (`IdMedicamento`)
      REFERENCES `mydb`.`Medicamento` (`Id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Figura 16 - Código SQL para criar a tabela ReceitaMedica

```

CREATE TABLE IF NOT EXISTS `mydb`.`Farmacia_fornece_Medicamento` (
  `Farmacia_Id` INT NOT NULL,
  `Medicamento_Id` INT NOT NULL,
  PRIMARY KEY (`Farmacia_Id`, `Medicamento_Id`),
  INDEX `fk_Farmacia_has_Medicamento_Medicamento1_idx` (`Medicamento_Id` ASC) VISIBLE,
  INDEX `fk_Farmacia_has_Medicamento_Farmacia1_idx` (`Farmacia_Id` ASC) VISIBLE,
  CONSTRAINT `fk_Farmacia_has_Medicamento_Farmacia1`
    FOREIGN KEY (`Farmacia_Id`)
      REFERENCES `mydb`.`Farmacia` (`Id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_Farmacia_has_Medicamento_Medicamento1`
    FOREIGN KEY (`Medicamento_Id`)
      REFERENCES `mydb`.`Medicamento` (`Id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Figura 17 - Código SQL para criar a tabela Farmacia\_fornece\_Medicamento



## 5.2 Tradução das interrogações do utilizador para SQL

Com base nos requisitos de exploração, criámos *queries* SQL para os resolver.

```
SELECT Nome, Telefone, Horario, Designacao, Custo FROM Farmacia
JOIN Farmacia_fornece_Medicamento ON Farmacia_Id = Farmacia.Id
JOIN Medicamento ON Medicamento.Id = Medicamento_Id;
```

Figura 18 - Código SQL para visualizar o horário e os medicamentos disponíveis em cada farmácia

Nome	Telefone	Horario	Designacao	Custo
Farmácia Monteiro	253123123	De segunda a sabado: 7h-23h; Domingos e feriados: 7h-20h	Brufen	8.40
Farmácia Monteiro	253123123	De segunda a sabado: 7h-23h; Domingos e feriados: 7h-20h	Ben-u-ron	2.71
Farmácia Monteiro	253123123	De segunda a sabado: 7h-23h; Domingos e feriados: 7h-20h	Caladryl	14.90
Farmácia Monteiro	253123123	De segunda a sabado: 7h-23h; Domingos e feriados: 7h-20h	Aerius	4.99
Farmácia Monteiro	253123123	De segunda a sabado: 7h-23h; Domingos e feriados: 7h-20h	Bephantere	12.85
Farmácia Monteiro	253123123	De segunda a sabado: 7h-23h; Domingos e feriados: 7h-20h	Strepsils	9.80
Farmácia Monteiro	253123123	De segunda a sabado: 7h-23h; Domingos e feriados: 7h-20h	Griponal	9.95
Farmácia Sao João	250124568	De segunda a sabado: 7h-22h; Domingos e feriados: 24h	Brufen	8.40
Farmácia Sao João	250124568	De segunda a sabado: 7h-22h; Domingos e feriados: 24h	Caladryl	14.90
Farmácia Sao João	250124568	De segunda a sabado: 7h-22h; Domingos e feriados: 24h	Bephantere	12.85
Farmácia Sao João	250124568	De segunda a sabado: 7h-22h; Domingos e feriados: 24h	Griponal	9.95
Farmácia Nova	240231567	De segunda a sabado: 24h; Domingos e feriados: 7h-13h	Ben-u-ron	2.71
Farmácia Nova	240231567	De segunda a sabado: 24h; Domingos e feriados: 7h-13h	Caladryl	14.90
Farmácia Nova	240231567	De segunda a sabado: 24h; Domingos e feriados: 7h-13h	Strepsils	9.80
Farmácia Popular	246852852	De segunda a sabado: 8h-24h; Domingos e feriados: 8h-14h	Brufen	8.40
Farmácia Popular	246852852	De segunda a sabado: 8h-24h; Domingos e feriados: 8h-14h	Ben-u-ron	2.71
Farmácia Popular	246852852	De segunda a sabado: 8h-24h; Domingos e feriados: 8h-14h	Caladryl	14.90
Farmácia Popular	246852852	De segunda a sabado: 8h-24h; Domingos e feriados: 8h-14h	Aerius	4.99

Figura 19 - Resultado da *query* da Figura 18

```
SELECT Nome, Distrito FROM Administrador
UNION
SELECT Nome, Distrito FROM Gestor
UNION
SELECT Nome, Distrito FROM Estafeta;
```

Figura 20 - Código SQL para consultar todos os funcionários da empresa e o respetivo distrito onde trabalham



Nome	Distrito
Alberto Oliveira	Braga
Bruna Guimarães	Braga
Lucia Enriques	Porto
Artur Silva	Porto
Rodrigo Salvador	Aveiro
Carla Nogueira	Aveiro
Nuno Martins	Viana do Castelo
Maria Martins	Viana do Castelo

Figura 21 - Resultado da *query* da Figura 20

```
SELECT ValorFaturacao, Nome FROM
Pedido JOIN Cliente ON IdCliente = NumeroContribuinte
ORDER BY ValorFaturacao DESC;
```

Figura 22 - Código SQL para listar os clientes por ordem decrescente de despesa

ValorFaturacao	Nome
76.73	Diego da Cunha
23.78	Silvio Freitas
23.03	Maria Manuela
15.32	João Moura
15.16	Rafael Gomes
13.55	Daniela Santos
13.39	Ricardo da Silva
13.31	Antonio Saraiva
11.90	Maria Augusta

Figura 23 - Resultado da *query* da Figura 22

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

Com o objetivo de facilitar a visualização de todos os funcionários da empresa, criamos uma vista que os lista, incluindo os respetivos identificadores, nomes, *e-mails* e distritos onde trabalham.

```
CREATE VIEW Funcionarios
AS
SELECT * FROM
(
SELECT Id, Nome, Email, Distrito, 'Gestor' AS 'Função' FROM Gestor
UNION
SELECT Id, Nome, Email, Distrito, 'Administrador' AS 'Função' FROM Administrador
UNION
SELECT Id, Nome, Email, Distrito, 'Estafeta' AS 'Função' FROM Estafeta
ORDER BY Função DESC, Distrito
) AS f;
```

Figura 24 - Código SQL para a implementação da vista

Id	Nome	Email	Distrito	Função
14	Rita Balado	rb@farmacia.com	Aveiro	Gestor
11	José Vilhena	jv@farmacia.com	Braga	Gestor
12	Maria Costa	mc@farmacia.com	Porto	Gestor
13	Joaquim Mirante	jm@farmacia.com	Viana do Castelo	Gestor
51	Carlos Lisboa	cl@farmacia.com	Aveiro	Estafeta
50	Rui Matos	rm@farmacia.com	Aveiro	Estafeta
49	Joana Sousa	js@farmacia.com	Aveiro	Estafeta
48	António Mota	am@farmacia.com	Aveiro	Estafeta
47	Pedro Silva	ps@farmacia.com	Aveiro	Estafeta
46	Mario Machado	mm@farmacia.com	Aveiro	Estafeta
45	João Costa	jc@farmacia.com	Aveiro	Estafeta
37	Martim Janota	mj@farmacia.com	Braga	Estafeta
31	Fernando Albino	fa@farmacia.com	Braga	Estafeta
32	Henrique Finorio	hf@farmacia.com	Braga	Estafeta

Figura 25 – Resultado da vista

## 5.4 Cálculo do espaço da base de dados

Calcularemos o espaço que será utilizado pela Base de Dados sabendo que os tipos de dados ocupam os tamanhos seguintes:

Tipo	Tamanho
INT	4 bytes
DECIMAL(5,2)	5 bytes
VARCHAR(N)	N+1 bytes
DATETIME	5 bytes
BINARY(1)	1 byte

Tabela 12 – Descrição do tamanho de cada tipo de dados

Tendo em conta estes tamanhos, de seguida calculamos o espaço ocupado por cada entrada de cada entidade.

Atributo	Tipo	Tamanho
NúmeroContribuinte	INT	4 bytes
Nome	VARCHAR(45)	46 bytes
DataNascimento	VARCHAR(10)	41 bytes
NúmeroSNS	INT	4 bytes
Rua\Endereço	VARCHAR(45)	46 bytes
Localidade\Endereço	VARCHAR(45)	46 bytes
CodigoPostal\Endereço	VARCHAR(45)	46 bytes

Distrito\Endereço	VARCHAR(45)	46 bytes
Telefone	INT	4 bytes
Email	VARCHAR(45)	46 bytes

Tabela 13 - Tamanho dos atributos da entidade Cliente

Cada entrada na tabela Cliente ocupa 329 *bytes*.

Atributo	Tipo	Tamanho
Id	INT	4 bytes
CustoEntrega	DECIMAL(5,2)	5 bytes
DataPedido	DATETIME	5 bytes
DataEntrega	DATETIME	5 bytes
Urgente	BINARY(1)	1 byte
ValorFaturação	DECIMAL(5,2)	5 bytes

Tabela 14 - Tamanho dos atributos da entidade Pedido

Cada entrada na tabela Pedido ocupa 25 *bytes*.

Atributo	Tipo	Tamanho
Id	INT	4 bytes
Custo	DECIMAL(5,2)	5 bytes
Designação	VARCHAR(45)	46 bytes
Descrição	VARCHAR(100)	101 bytes

Tabela 15 - Tamanho dos atributos da entidade Medicamento

Cada entrada na tabela Medicamento ocupa 156 *bytes*.

Atributo	Tipo	Tamanho
Id	INT	4 bytes
Nome	VARCHAR(45)	46 bytes
Telefone	VARCHAR(45)	46 bytes
Email	VARCHAR(45)	46 bytes
Horário	VARCHAR(200)	201 bytes
Rua\Endereço	VARCHAR(45)	46 bytes
Localidade\Endereço	VARCHAR(45)	46 bytes
CodigoPostal\Endereço	VARCHAR(45)	46 bytes
Distrito\Endereço	VARCHAR(45)	46 bytes

Tabela 16 - Tamanho dos atributos da entidade Farmácia

Cada entrada na tabela Farmácia ocupa 527 *bytes*.

Atributo	Tipo	Tamanho
Id	INT	4 bytes
Médico	VARCHAR(45)	46 bytes

Tabela 17 - Tamanho dos atributos da entidade ReceitaMédica

Cada entrada na tabela ReceitaMédica ocupa 50 *bytes*.

Atributo	Tipo	Tamanho
Id	INT	4 bytes
Nome	VARCHAR(45)	46 bytes
Email	VARCHAR(45)	46 bytes
Distrito	VARCHAR(45)	46 bytes

Tabela 18 - Tamanho dos atributos da entidade Gestor

Cada entrada na tabela Gestor ocupa 142 *bytes*.

Atributo	Tipo	Tamanho
Id	INT	4 bytes
Nome	VARCHAR(45)	46 bytes
Email	VARCHAR(45)	46 bytes
Distrito	VARCHAR(45)	46 bytes

Tabela 19 - Tamanho dos atributos da entidade Administrador

Cada entrada na tabela Administrador ocupa 142 *bytes*.

Atributo	Tipo	Tamanho
Id	INT	4 bytes
Nome	VARCHAR(45)	46 bytes
Email	VARCHAR(45)	46 bytes
Distrito	VARCHAR(45)	46 bytes
MatrículaVeículo	VARCHAR(8)	9 bytes

Tabela 20 - Tamanho dos atributos da entidade Estafeta

Cada entrada na tabela Estafeta ocupa 151 *bytes*.

Atributo	Tipo	Tamanho
Quantidade	INT	4 bytes
Custo	DECIMAL(5,2)	5 bytes
Iva	INT	4 bytes
Preço	DECIMAL(5,2)	5 bytes

Tabela 21 - Tamanho dos atributos do relacionamento N:M Pedido\_inclui\_Medicamento

Cada entrada na tabela Pedido\_inclui\_Medicamento ocupa 18 *bytes*.

Sendo assim, o espaço inicial (sem povoamento) ocupado pela Base de Dados é 1540 *bytes*.

Com o número inicial de 15 clientes e de 9 pedidos são necessários  $15 * 329 = 4935$  *bytes* para os dados dos clientes e  $9 * 25 = 225$  para os dados dos pedidos.

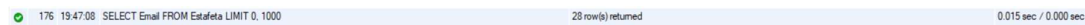
Assumindo que o crescimento anual esperado do número de clientes é de 15% e do número de pedidos é de 20%, prevê-se um crescimento de  $1.15 * 4935 = 5676$  *bytes* de clientes e  $1.20 * 225 = 270$  *bytes* de pedidos.

## 5.5 Indexação do Sistema de Dados

Para melhorar o desempenho de uma dada *query* sobre a Base de Dados utilizamos a indexação. Este processo melhora a eficiência temporal ao custo de algum espaço de armazenamento.

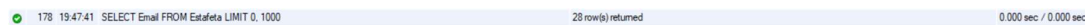
```
CREATE INDEX email ON Estafeta (Email);
```

Figura 26 - Código SQL para criação do índice



176 19:47:08 SELECT Email FROM Estafeta LIMIT 0, 1000 28 row(s) returned 0.015 sec / 0.000 sec

Figura 27 – Tempo de execução antes da indexação (0.015 segundos)



178 19:47:41 SELECT Email FROM Estafeta LIMIT 0, 1000 28 row(s) returned 0.000 sec / 0.000 sec

Figura 28 – Tempo de execução após a indexação (0.0001 segundos)

## 5.6 Procedimentos Implementados

De forma a facilitar o acesso a dados importantes e frequentemente necessários da Base de Dados criamos os seguintes procedimentos:

```
DELIMITER $$
CREATE PROCEDURE AtualizaPrecoMedicamento
    (NovoPreco DECIMAL(5,3), MedId INT)
BEGIN
    UPDATE Medicamento SET Custo = NovoPreco WHERE Id = MedId;
END $$

CALL AtualizaPrecoMedicamento(9.60,1) ;
```

Figura 29 - Procedimento que atualiza o custo de um medicamento

Id	Designacao	Custo	Descricao
1	Brufen	8.40	600mg, anti-inflamatório não esteroide

Figura 30 – Custo do medicamento antes de executar o procedimento

Id	Designacao	Custo	Descricao
1	Brufen	9.60	600mg, anti-inflamatório não esteroide

Figura 31 – Custo do medicamento após executar o procedimento

```
DELIMITER $$
CREATE PROCEDURE TopClientes()
BEGIN
    SELECT P.IdCliente, C.Nome, COUNT(P.IdCliente) AS `total_compras`
    FROM Pedido AS P INNER JOIN Cliente AS C
    ON P.IdCliente=C.NumeroContribuinte
    GROUP BY IdCliente
    ORDER BY `total_compras` DESC
    LIMIT 3;
END $$

CALL TopClientes;
```

Figura 32 - Procedimento que indica os 3 clientes com mais compras

	IdCliente	Nome	total_compras
►	251998161	Maria Augusta	2
	185449529	Maria Manuela	1
	191098178	Rafael Gomes	1
	244948120	Daniela Santos	1
	251098561	Diego da Cunha	1

Figura 33 - Resultado do procedimento da Figura 32

```

DELIMITER $$
CREATE PROCEDURE TopEstafetas()
BEGIN
    SELECT P.IdEstafeta, E.Nome, COUNT(P.IdEstafeta) AS `total_entregas`
    FROM Pedido AS P INNER JOIN Estafeta AS E
    ON P.IdEstafeta=E.Id
    WHERE P.DataEntrega BETWEEN DATE_SUB(NOW(), INTERVAL 30 DAY) AND NOW()
    GROUP BY IdEstafeta
    ORDER BY `total_entregas` DESC;
END $$

```

Figura 34 - Procedimento que indica os estafetas que realizaram mais entregas

IdEstafeta	Nome	total_entregas
31	Fernando Albino	2
45	João Costa	1
38	Ribeiro Mendes	1
32	Henrique Finorio	1
39	Kevin Rendeiro	1
53	Mariana Lima	1

Figura 35 - Resultado do procedimento da Figura 34

```

DELIMITER $$
CREATE PROCEDURE PedidosNaoEntregues
(Dist VARCHAR(45))
BEGIN
    SELECT Cliente.Nome, Endereco.Localidade, Endereco.CodigoPostal, Endereco.Rua, Pedido.Id AS `Id do Pedido` FROM
    Pedido INNER JOIN Cliente ON Cliente.NumeroContribuinte = Pedido.IdCliente
    INNER JOIN Endereco ON Cliente.IdEndereco = Endereco.EnderecoId
    WHERE Pedido.DataEntrega IS NULL AND Endereco.Distrito = Dist;
END $$

```

Figura 36 – Procedimento que lista os pedidos que ainda não foram entregues num certo distrito

Nome	Localidade	CodigoPostal	Rua	Id do Pedido
Antonio Saraiva	Cabeço	3800-531	Rua Tenente Coronel Afonso Lucas nº 24	8

Figura 37 - Resultado do procedimento da Figura 36

```

DELIMITER $$
CREATE PROCEDURE TotalPedidos()
BEGIN
    SELECT Endereco.Distrito, COUNT(Pedido.Id) AS `Número de Encomendas` FROM
        Pedido INNER JOIN Cliente ON Cliente.NumeroContribuinte = Pedido.IdCliente
        INNER JOIN Endereco ON Cliente.IdEndereco = Endereco.EnderecoId
    GROUP BY Endereco.Distrito
    ORDER BY `Número de Encomendas` DESC;
END $$

```

Figura 38 - Procedimento que indica o total de pedidos por cada distrito

Distrito	Número de Encomendas
Braga	3
Aveiro	2
Porto	2
Viana do Castelo	2

Figura 39 - Resultado do procedimento da Figura 38

## 5.7 Plano de segurança e recuperação de dados

Tendo em conta que estamos a trabalhar numa Base de Dados para uma empresa de entregas ao domicílio, e estando a trabalhar em prol da saúde pública, é indispensável a segurança de todos os dados e assegurar o bom funcionamento da Base de Dados. Como forma de garantir a proteção dos dados de ataques informáticos, erros humanos, falhas de energia, etc. decidiu-se então criar, diariamente, uma cópia de segurança para uma *Cloud* fornecida por terceiros.



## 6. Conclusões e Trabalho Futuro

Após o desenvolvimento desta Base de Dados percebemos a importância de todas as fases e como as executar de forma correta. Entendemos a importância da fase de levantamento e análise de requisitos pois esta serve como base para todos os passos seguintes. Para além disso, a implementação do modelo conceptual ajudou-nos a visualizar a estrutura que foi utilizada a partir daí, facilitando os processos seguintes de modelação lógica e física. Por fim, após o povoamento e criação de *queries* conseguimos, finalmente, observar a Base de Dados em funcionamento, dando então ao Sr. Joaquim a organização desejada para os dados da sua empresa.

Contudo, entendemos que existem melhorias a ser feitas que, no futuro, podiam ser implementadas, como por exemplo, a implementação de mais *queries* e procedimentos para tornar ainda mais dinâmica a consulta e atualização dos dados.

Concluimos, assim, que os objetivos do trabalho foram cumpridos e que implementamos uma Base de Dados capaz de suportar e otimizar o modelo de negócio da empresa de entregas do Sr. Joaquim.

## 7. Referências Bibliográficas

Connolly, T. M., & Begg, C. E. (2015). *Database systems: A practical Approach to Design, Implementation, and Management* (6th edition). Pearson Education.

DBIS University of Innsbruck. (n.d.). *RelaX - relational algebra calculator*. GitHub. Acedido a dezembro 17, 2022, em <https://dbis-uibk.github.io/relax/landing>

Oracle Corporation. (2023). *MySQL :: MySQL 8.0 Reference Manual*. Mysql.com. Acedido a janeiro 4, 2023, em <https://dev.mysql.com/doc/refman/8.0/en/>

SOFARMA. (n.d.). *Entrega de Medicamentos ao Domicílio*. Acedido a outubro 12, 2022, em <https://www.farmacia-servico.pt/prescription>

## Anexos

### I. *Script* da criação da Base de Dados

-- MySQL Workbench Forward Engineering

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO
_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```
-- -----
-- Schema mydb
-- -----
```

```
-- -----
-- Schema mydb
-- -----
```

```
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;
```

```
-- -----
-- Table `mydb`.`Gestor`
-- -----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Gestor` (
  `Id` INT NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `Email` VARCHAR(45) NOT NULL,
  `Distrito` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Id`),
  UNIQUE INDEX `Id_UNIQUE` (`Id` ASC) VISIBLE)
```

ENGINE = InnoDB;

-----  
-- Table `mydb`.`Administrador`  
-----

CREATE TABLE IF NOT EXISTS `mydb`.`Administrador` (  
 `Id` INT NOT NULL,  
 `Nome` VARCHAR(45) NOT NULL,  
 `Email` VARCHAR(45) NOT NULL,  
 `Distrito` VARCHAR(45) NOT NULL,  
 UNIQUE INDEX `Id\_UNIQUE` (`Id` ASC) VISIBLE,  
 PRIMARY KEY (`Id`))  
ENGINE = InnoDB;

-----  
-- Table `mydb`.`Estafeta`  
-----

CREATE TABLE IF NOT EXISTS `mydb`.`Estafeta` (  
 `Id` INT NOT NULL,  
 `Nome` VARCHAR(45) NOT NULL,  
 `Email` VARCHAR(45) NOT NULL,  
 `Distrito` VARCHAR(45) NOT NULL,  
 `MatriculaVeiculo` VARCHAR(8) NOT NULL,  
 PRIMARY KEY (`Id`),  
 UNIQUE INDEX `Id\_UNIQUE` (`Id` ASC) VISIBLE)  
ENGINE = InnoDB;

-----  
-- Table `mydb`.`Endereco`  
-----

CREATE TABLE IF NOT EXISTS `mydb`.`Endereco` (  
 `Enderecold` INT NOT NULL,  
 `Distrito` VARCHAR(45) NOT NULL,  
 `Localidade` VARCHAR(45) NOT NULL,  
 `Rua` VARCHAR(45) NOT NULL,  
 `CodigoPostal` VARCHAR(45) NOT NULL,  
 PRIMARY KEY (`Enderecold`),

```
    UNIQUE INDEX `EnderecoId_UNIQUE` (`EnderecoId` ASC) VISIBLE)
ENGINE = InnoDB;
```

```
-- -----
-- Table `mydb`.`Cliente`
-- -----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Cliente` (
  `NumeroContribuinte` INT NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `DataNascimento` VARCHAR(10) NOT NULL,
  `Telefone` INT NOT NULL,
  `Email` VARCHAR(45) NULL,
  `NumeroSNS` INT NOT NULL,
  `IdEndereco` INT NOT NULL,
  UNIQUE INDEX `NumeroContribuinte_UNIQUE` (`NumeroContribuinte` ASC) VISIBLE,
  PRIMARY KEY (`NumeroContribuinte`),
  INDEX `fk_Cliente_Endereco1_idx` (`IdEndereco` ASC) VISIBLE,
  CONSTRAINT `fk_Cliente_Endereco1`
    FOREIGN KEY (`IdEndereco`)
      REFERENCES `mydb`.`Endereco` (`EnderecoId`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----
-- Table `mydb`.`Pedido`
-- -----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Pedido` (
  `Id` INT NOT NULL,
  `CustoEntrega` DECIMAL(5,2) NOT NULL,
  `ValorFaturacao` DECIMAL(5,2) NOT NULL,
  `Urgente` BINARY(1) NOT NULL,
  `DataPedido` DATETIME NOT NULL,
  `DataEntrega` DATETIME NULL,
  `IdAdministrador` INT NOT NULL,
  `IdCliente` INT NOT NULL,
  `IdEstafeta` INT NOT NULL,
  PRIMARY KEY (`Id`),
```

```

UNIQUE INDEX `Id_UNIQUE` (`Id` ASC) VISIBLE,
INDEX `fk_Pedido_Administrador1_idx` (`IdAdministrador` ASC) VISIBLE,
INDEX `fk_Pedido_Cliente1_idx` (`IdCliente` ASC) VISIBLE,
INDEX `fk_Pedido_Estafeta1_idx` (`IdEstafeta` ASC) VISIBLE,
CONSTRAINT `fk_Pedido_Administrador1`
  FOREIGN KEY (`IdAdministrador`)
  REFERENCES `mydb`.`Administrador` (`Id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_Pedido_Cliente1`
  FOREIGN KEY (`IdCliente`)
  REFERENCES `mydb`.`Cliente` (`NumeroContribuinte`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_Pedido_Estafeta1`
  FOREIGN KEY (`IdEstafeta`)
  REFERENCES `mydb`.`Estafeta` (`Id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-- -----
-- Table `mydb`.`Medicamento`
-- -----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`Medicamento` (
  `Id` INT NOT NULL,
  `Designacao` VARCHAR(45) NOT NULL,
  `Custo` DECIMAL(5,2) NOT NULL,
  `Descricao` VARCHAR(100) NULL,
  PRIMARY KEY (`Id`),
  UNIQUE INDEX `Id_UNIQUE` (`Id` ASC) VISIBLE)
ENGINE = InnoDB;

```

```

-- -----
-- Table `mydb`.`Pedido_inclui_Medicamento`
-- -----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`Pedido_inclui_Medicamento` (
  `IdPedido` INT NOT NULL,

```

```

`IdMedicamento` INT NOT NULL,
`Quantidade` INT NOT NULL,
`Custo` DECIMAL(5,2) NOT NULL,
`IVA` INT NOT NULL,
`Preco` DECIMAL(5,2) NOT NULL,
PRIMARY KEY (`IdMedicamento`, `IdPedido`),
INDEX `fk_Pedido_has_Medicamento_Medicamento1_idx` (`IdMedicamento` ASC) VISIBLE,
INDEX `fk_Pedido_has_Medicamento_Pedido1_idx` (`IdPedido` ASC) VISIBLE,
CONSTRAINT `fk_Pedido_has_Medicamento_Pedido1`
  FOREIGN KEY (`IdPedido`)
    REFERENCES `mydb`.`Pedido` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_Pedido_has_Medicamento_Medicamento1`
  FOREIGN KEY (`IdMedicamento`)
    REFERENCES `mydb`.`Medicamento` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-- -----
-- Table `mydb`.`Farmacia`
-- -----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`Farmacia` (
  `Id` INT NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `Telefone` VARCHAR(45) NOT NULL,
  `Email` VARCHAR(45) NOT NULL,
  `Horario` VARCHAR(200) NOT NULL,
  `IdGestor` INT NOT NULL,
  `IdEndereco` INT NOT NULL,
  PRIMARY KEY (`Id`),
  INDEX `fk_Farmacia_Gestor1_idx` (`IdGestor` ASC) VISIBLE,
  UNIQUE INDEX `Id_UNIQUE` (`Id` ASC) VISIBLE,
  INDEX `fk_Farmacia_Endereco1_idx` (`IdEndereco` ASC) VISIBLE,
  CONSTRAINT `fk_Farmacia_Gestor1`
    FOREIGN KEY (`IdGestor`)
      REFERENCES `mydb`.`Gestor` (`Id`)
      ON DELETE NO ACTION

```

```

    ON UPDATE NO ACTION,
CONSTRAINT `fk_Farmacia_Endereço1`
    FOREIGN KEY (`IdEndereco`)
    REFERENCES `mydb`.`Endereco` (`Enderecold`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`ReceitaMedica`
-----

CREATE TABLE IF NOT EXISTS `mydb`.`ReceitaMedica` (
  `Id` INT NOT NULL,
  `Medico` VARCHAR(45) NOT NULL,
  `IdCliente` INT NOT NULL,
  `IdMedicamento` INT NOT NULL,
  PRIMARY KEY (`Id`),
  UNIQUE INDEX `Id_UNIQUE` (`Id` ASC) VISIBLE,
  INDEX `fk_ReceitaMedica_Cliente1_idx` (`IdCliente` ASC) VISIBLE,
  INDEX `fk_ReceitaMedica_Medicamento1_idx` (`IdMedicamento` ASC) VISIBLE,
  CONSTRAINT `fk_ReceitaMedica_Cliente1`
    FOREIGN KEY (`IdCliente`)
    REFERENCES `mydb`.`Cliente` (`NumeroContribuinte`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_ReceitaMedica_Medicamento1`
    FOREIGN KEY (`IdMedicamento`)
    REFERENCES `mydb`.`Medicamento` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`Farmacia_fornece_Medicamento`
-----

CREATE TABLE IF NOT EXISTS `mydb`.`Farmacia_fornece_Medicamento` (
  `Farmacia_Id` INT NOT NULL,
  `Medicamento_Id` INT NOT NULL,

```

```

PRIMARY KEY (`Farmacia_Id`, `Medicamento_Id`),
INDEX `fk_Farmacia_has_Medicamento_Medicamento1_idx` (`Medicamento_Id` ASC)
VISIBLE,
INDEX `fk_Farmacia_has_Medicamento_Farmacia1_idx` (`Farmacia_Id` ASC) VISIBLE,
CONSTRAINT `fk_Farmacia_has_Medicamento_Farmacia1`
FOREIGN KEY (`Farmacia_Id`)
REFERENCES `mydb`.`Farmacia` (`Id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Farmacia_has_Medicamento_Medicamento1`
FOREIGN KEY (`Medicamento_Id`)
REFERENCES `mydb`.`Medicamento` (`Id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

## II. *Script* do povoamento

```

USE mydb;

INSERT INTO Medicamento (Id, Designacao, Custo, Descricao) VALUES
(1, 'Brufen', 8.40, '600mg, anti-inflamatório não esteroide'),
(2, 'Ben-u-ron', 2.71, '500mg, analgésico e antipirético'),
(3, 'Caladryl', 14.90, '150ml'),
(4, 'Aerius', 4.99, 'Anti-histamínico tricíclico'),
(5, 'Bephantene', 12.85, '50g'),
(6, 'Strepsils', 9.80, 'Pastilhas antiinflamatórias de ação específica na garganta'),
(7, 'Griponal', 9.95, 'Alívio rápido de sintomas das gripes e constipações como dores e febre'),
(8, 'Vicks Vaporub', 10.50, 'Pomada para o tratamento de constipações e gripes'),
(9, 'Betaserc', 6.63, 'Fármaco antivertiginoso'),

```



```
(10, 'Aspirina', 5.95, 'É utilizado para tratar dores de cabeça'),
(11, 'Victan', 5.04, 'Medicamentos Sistema nervoso central');
-- SELECT * FROM Medicamento;
```

```
INSERT INTO Gestor (Id, Nome, Email, Distrito) VALUES
(11, 'José Vilhena', 'jv@farmacia.com', 'Braga'),
(12, 'Maria Costa', 'mc@farmacia.com', 'Porto'),
(13, 'Joaquim Mirante', 'jm@farmacia.com', 'Viana do Castelo'),
(14, 'Rita Balado', 'rb@farmacia.com', 'Aveiro');
-- SELECT * FROM Gestor;
```

```
INSERT INTO Administrador (Id, Nome, Email, Distrito) VALUES
(21, 'Alberto Oliveira', 'ao@farmacia.com', 'Braga'),
(22, 'Bruna Guimarães', 'bg@farmacia.com', 'Braga'),
(23, 'Lucia Enriques', 'le@farmacia.com', 'Porto'),
(24, 'Artur Silva', 'as@farmacia.com', 'Porto'),
(25, 'Rodrigo Salvador', 'rs@farmacia.com', 'Aveiro'),
(26, 'Carla Nogueira', 'cn@farmacia.com', 'Aveiro'),
(27, 'Nuno Martins', 'nm@farmacia.com', 'Viana do Castelo'),
(28, 'Maria Martins', 'mm@farmacia.com', 'Viana do Castelo');
-- SELECT * FROM Administrador;
```

```
INSERT INTO Endereco (Enderecoid, Distrito, Localidade, Rua, CodigPostal) VALUES
(1, 'Braga', 'Amares', 'Rua da Bela Vista nº 12', '4720-001'),
(2, 'Braga', 'Brito', 'Rua dos Oito Moinhos nº 134', '4805-630'),
(3, 'Braga', 'Braga', 'Rua do Paço nº 346', '4715-439'),
(4, 'Braga', 'Barco', 'Rua de São Martinho nº 87', '4809-005'),
(5, 'Braga', 'Guardizela', 'Rua da Granja nº 859', '4765-410'),
(6, 'Porto', 'São Félix Da Marinha', 'Rua Parque Gondarim nº 243', '4410-148'),
(7, 'Porto', 'Vila Chã', 'Rua Patrão Caramelho nº 276', '4485-861'),
(8, 'Porto', 'Rio Tinto', 'Rua Pádua Correia nº 765', '4435-011'),
(9, 'Porto', 'Maia', 'Rua Oliveirinhas nº 52', '4425-687'),
(10, 'Porto', 'Porto', 'Rua Comércio Porto nº 45', '4150-200'),
(11, 'Aveiro', 'Mealhada', 'Rua do Jardim nº 877', '3054-001'),
(12, 'Aveiro', 'Ovar', 'Rua Alexandre Herculano nº 634', '3884-005'),
(13, 'Aveiro', 'Cabeço', 'Rua Tenente Coronel Afonso Lucas nº 24', '3800-531'),
(14, 'Aveiro', 'Espinho', 'Rua Sales de Cima Anta nº 11', '4500-009'),
(15, 'Aveiro', 'Águeda', 'Rua Fernando Caldeira nº 97', '3754-501'),
(16, 'Viana do Castelo', 'Arão', 'Caminho de Alvarães nº 463', '4930-003'),
(17, 'Viana do Castelo', 'Vila Nova de Anha', 'Avenida de Viana do Castelo nº 33', '4935-415');
```

```
(18, 'Viana do Castelo', 'Viana do Castelo', 'Rua Santiago da Barra nº 444', '4904-882'),
(19, 'Viana do Castelo', 'Afife', 'Caminho do Pião nº 788', '4900-101'),
(20, 'Viana do Castelo', 'Caminha', 'Avenida Padre Pinheiro nº 999', '4910-106');
-- SELECT * FROM Endereco;
```

```
INSERT INTO Estafeta (Id, Nome, Email, Distrito, MatriculaVeiculo) VALUES
```

```
(31, 'Fernando Albino', 'fa@farmacia.com', 'Braga', '23-34-AR'),
(32, 'Henrique Finorio', 'hf@farmacia.com', 'Braga', '37-TU-56'),
(33, 'Almirante Melo', 'am@farmacia.com', 'Braga', '92-33-VR'),
(34, 'Alenquer Guerreiro', 'ag@farmacia.com', 'Braga', '55-34-GF'),
(35, 'Fernanda Taveira', 'ft@farmacia.com', 'Braga', '74-AB-37'),
(36, 'João Matos', 'jm@farmacia.com', 'Braga', 'HM-77-11'),
(37, 'Martim Janota', 'mj@farmacia.com', 'Braga', '00-12-MJ'),
(38, 'Ribeiro Mendes', 'rm@farmacia.com', 'Porto', '87-22-OL'),
(39, 'Kevin Rendeiro', 'kr@farmacia.com', 'Porto', '69-JN-26'),
(40, 'Yousuf Nabila', 'yn@farmacia.com', 'Porto', 'MJ-57-49'),
(41, 'Guilherme Dias', 'gd@farmacia.com', 'Porto', '78-LJ-25'),
(42, 'Diogo Bilhar', 'db@farmacia.com', 'Porto', '26-89-GI'),
(43, 'Passos Dias', 'pd@farmacia.com', 'Porto', 'GI-26-18'),
(44, 'Aguiar Mota', 'am@farmacia.com', 'Porto', 'FO-26-18'),
(45, 'João Costa', 'jc@farmacia.com', 'Aveiro', '12-42-FR'),
(46, 'Mario Machado', 'mm@farmacia.com', 'Aveiro', '76-TH-51'),
(47, 'Pedro Silva', 'ps@farmacia.com', 'Aveiro', '32-23-HG'),
(48, 'António Mota', 'am@farmacia.com', 'Aveiro', '43-94-MF'),
(49, 'Joana Sousa', 'js@farmacia.com', 'Aveiro', '54-AC-67'),
(50, 'Rui Matos', 'rm@farmacia.com', 'Aveiro', '90-FF-11'),
(51, 'Carlos Lisboa', 'cl@farmacia.com', 'Aveiro', '32-GH-26'),
(52, 'Paulo Campos', 'pc@farmacia.com', 'Viana do Castelo', '51-PA-56'),
(53, 'Mariana Lima', 'ml@farmacia.com', 'Viana do Castelo', 'AJ-53-41'),
(54, 'Miguel Mendes', 'mm@farmacia.com', 'Viana do Castelo', '23-MJ-19'),
(55, 'Rúben Andrade', 'ra@farmacia.com', 'Viana do Castelo', '14-91-PU'),
(56, 'Alice Brito', 'ab@farmacia.com', 'Viana do Castelo', 'AT-51-79'),
(57, 'Carlos Melo', 'cm@farmacia.com', 'Viana do Castelo', '81-SS-58'),
(58, 'Sílvia Freitas', 'sf@farmacia.com', 'Viana do Castelo', '71-34-58');
-- SELECT * FROM Estafeta;
```

```
INSERT INTO Cliente (NumeroContribuinte, Nome, DataNascimento, Telefone, Email,
NumeroSNS, IdEndereco) VALUES
```

```
(251998161, 'Maria Augusta', '1969-12-05', 964856174, NULL, 111111111, 2),
(257787161, 'João Moura', '1980-12-15', 912415230, 'johnmoura22@gmail.com', 222222222, 3),
```

```

(185449529,'Maria Manuela','2000-12-
26',925155432,'manuelamaria1989@gmail.com',333333333,4),
(239981691,'Luiz Miguel Almeida','1988-10-
13',931948912,'lmalmeida@gmail.com',444444444,5),
(244948120,'Daniela Santos','2022-11-
15',934840000,'danny_danny222@gmail.com',555555555,7),
(251098561,'Diego da Cunha','1988-09-
07',944448595,'DieguinhoCunha@gmail.com',666666666,8),
(221578164,'Emanuel Vieira','1975-08-
08',930001114,'EmanuelVSilva@gmail.com',777777777,9),
(185198162,'Alice Moreira','1976-02-14',966666667,'alimoreira123@gmail.com',888888888,10),
(191098178,'Rafael Gomes','1988-01-01',984848484,'rafinha449@gmail.com',999999999,12),
(874442799,'Antonio Saraiva','1975-03-11',958454874,'tonioS@gmail.com',121212121,13),
(305672973,'Albino José Costa','2022-05-
24',934748947,'binojosé543@gmail.com',131313131,15),
(747026991,'Silvio Freitas','1988-06-27',961978912,'silv44@gmail.com',141414141,16),
(443036645,'Ricardo da Silva','1988-05-05',911947350,NULL,151515151,18),
(904484256,'Sara Daniela','1984-11-20',961486312,'saritaa20@gmail.com',161616161,19),
(512444984,'Tania Silvéria','1983-10-10',919756910,NULL,171717171,20);
-- SELECT * FROM Cliente;

```

```

INSERT INTO ReceitaMedica (Id, Medico, IdCliente, IdMedicamento) VALUES
(50, 'Pedro Manuel Silva', 251998161, 1),
(51, 'Antonio Gaspar', 257787161, 1),
(52, 'Ana Maria Silveira', 185449529, 7),
(53, 'Rita Antunes', 239981691, 7),
(54, 'Francisca Carvalho Lima', 244948120, 9),
(55, 'Manuel Augusto dos Santos Cruz', 251098561, 9),
(56, 'Rogério Macedo de Oliveira', 221578164, 1),
(57, 'Maria Eduarda Ramos ', 185198162, 7),
(58, 'Isaac Farias', 191098178, 9);
-- SELECT * FROM ReceitaMedica;

```

```

INSERT INTO Farmacia (Id, Nome, Telefone, Email, Horario, IdGestor, IdEndereco) VALUES
(1, 'Farmácia Monteiro', 253123123, 'fm@farmacia.com', 'De segunda a sabado: 7h-23h;
Domingos e feriados: 7h-20h', 11, 1),
(2, 'Farmácia Sao João', 250124568, 'fsj@farmacia.com', 'De segunda a sabado: 7h-22h;
Domingos e feriados: 24h', 13, 6),
(3, 'Farmácia Nova', 240231567, 'fn@farmacia.com', 'De segunda a sabado: 24h; Domingos e
feriados: 7h-13h', 12, 11),

```

```
(4, 'Farmácia Popular', 246852852, 'fp@farmacia.com', 'De segunda a sabado: 8h-24h;
Domingos e feriados: 8h-14h', 14, 17);
-- SELECT * FROM Farmacia;
```

```
INSERT INTO Pedido (Id, CustoEntrega, ValorFaturacao, Urgente, DataPedido, DataEntrega,
IdAdministrador, IdCliente, IdEstafeta) VALUES
(1, 3.0, 11.90, 0, '2023-01-05 15:31', '2023-01-06 10:17', 21, 251998161, 31),
(2, 7.0, 15.16, 1, '2023-01-15 12:07', '2023-01-15 12:40', 25, 191098178, 45),
(3, 3.0, 23.78, 0, '2022-12-10 10:10', '2022-12-11 9:35', 27, 747026991, 52),
(4, 3.0, 13.55, 0, '2023-01-09 20:11', '2023-01-11 9:15', 23, 244948120, 38),
(5, 7.0, 23.03, 1, '2022-12-19 12:24', '2022-12-19 17:35', 22, 185449529, 32),
(6, 3.0, 76.73, 0, '2022-12-22 17:55', '2022-12-24 12:43', 24, 251098561, 39),
(7, 3.0, 13.39, 0, '2022-12-23 9:50', '2022-12-24 10:50', 28, 443036645, 53),
(8, 7.0, 13.31, 1, '2022-12-26 18:10', NULL, 26, 874442799, 46),
(9, 3.0, 15.32, 0, '2022-12-28 15:17', '2022-12-31 13:30', 21, 251998161, 31);
-- SELECT * FROM Pedido;
```

```
INSERT INTO Pedido_inclui_Medicamento (IdPedido, IdMedicamento, Quantidade, Custo, IVA,
Preco) VALUES
(1, 1, 1, 8.40, 6, 8.90),
(2, 2, 1, 2.71, 6, 2.87),
(2, 3, 1, 4.99, 6, 5.29),
(3, 6, 2, 9.80, 6, 20.78),
(4, 7, 1, 9.95, 6, 10.55),
(5, 11, 3, 5.04, 6, 16.03),
(6, 7, 1, 9.95, 6, 10.55),
(6, 3, 4, 14.90, 6, 63.18),
(6, 6, 1, 9.80, 6, 10.39),
(6, 10, 1, 5.95, 6, 6.31),
(6, 4, 1, 4.99, 6, 5.29),
(6, 9, 1, 6.63, 6, 7.03);
-- SELECT * FROM Pedido_inclui_Medicamento;
```

```
INSERT INTO Farmacia_fornece_Medicamento (Farmacia_Id, Medicamento_Id) VALUES
(1, 1),
(1, 2),
(1, 3),
(1, 4),
(1, 5),
(1, 6),
```

```

(1, 7),
(1, 9),
(1, 11),
(2, 1),
(2, 3),
(2, 5),
(2, 7),
(2, 8),
(2, 9),
(2, 10),
(3, 2),
(3, 3),
(3, 6),
(3, 7),
(3, 10),
(3, 11),
(4, 1),
(4, 2),
(4, 3),
(4, 4),
(4, 6),
(4, 7),
(4, 8),
(4, 11);
-- SELECT * FROM Farmacia_fornece_Medicamento;

```

### **III.      *Script de queries***

```

USE mydb;

-- Procedimento que lista os clientes com mais pedidos
DELIMITER $$
CREATE PROCEDURE TopClientes()
BEGIN
SELECT P.IdCliente, C.Nome, COUNT(P.IdCliente) AS `total_compras`

```

```

FROM Pedido AS P INNER JOIN Cliente AS C
ON P.IdCliente=C.NumeroContribuinte
GROUP BY IdCliente
ORDER BY `total_compras` DESC
LIMIT 5;
END $$

```

```

DROP PROCEDURE TopClientes;
CALL TopClientes;

```

```

-- Procedimento que lista os estafetas com mais entregas
DELIMITER $$
CREATE PROCEDURE TopEstafetas()
BEGIN
SELECT P.IdEstafeta, E.Nome, COUNT(P.IdEstafeta) AS `total_entregas`
FROM Pedido AS P INNER JOIN Estafeta AS E
ON P.IdEstafeta=E.Id
WHERE P.DataEntrega BETWEEN DATE_SUB(NOW(), INTERVAL 30 DAY) AND NOW()
GROUP BY IdEstafeta
ORDER BY `total_entregas` DESC;
END $$

```

```

DROP PROCEDURE TopEstafetas;
CALL TopEstafetas;

```

```

-- Procedimento para atualizar o preço dos medicamentos
DELIMITER $$
CREATE PROCEDURE AtualizaPrecoMedicamento
    (NovoPreco DECIMAL(5,2), MedId INT)
BEGIN
    UPDATE Medicamento SET Custo = NovoPreco WHERE Id = MedId;
END $$

```

```

DROP PROCEDURE AtualizaPrecoMedicamento;
CALL AtualizaPrecoMedicamento(9.60,1);

```

```

-- Procedimento que lista os pedidos que ainda não foram entregues num certo distrito
DELIMITER $$
CREATE PROCEDURE PedidosNaoEntregues
    (Dist VARCHAR(45))

```

```

BEGIN
SELECT Cliente.Nome, Endereco.Localidade, Endereco.CodigoPostal, Endereco.Rua,
Pedido.Id AS `Id do Pedido` FROM
Pedido INNER JOIN Cliente ON Cliente.NumeroContribuinte = Pedido.IdCliente
      INNER JOIN Endereco ON Cliente.IdEndereco = Endereco.Enderecold
      WHERE Pedido.DataEntrega IS NULL AND Endereco.Distrito = Dist;
END $$

```

```

DROP PROCEDURE PedidosNaoEntregues;
CALL PedidosNaoEntregues('Aveiro');

```

```

-- Procedimento que mostra o total de pedidos por cada distrito
DELIMITER $$
CREATE PROCEDURE TotalPedidos()
BEGIN
SELECT Endereco.Distrito, COUNT(Pedido.Id) AS `Número de Encomendas` FROM
Pedido INNER JOIN Cliente ON Cliente.NumeroContribuinte = Pedido.IdCliente
INNER JOIN Endereco ON Cliente.IdEndereco = Endereco.Enderecold
GROUP BY Endereco.Distrito
      ORDER BY `Número de Encomendas` DESC;
END $$

```

```

DROP PROCEDURE TotalPedidos;
CALL TotalPedidos;

```

```

-- Função que, dado um pedido, mostra o número de telefone do cliente que o realizou
DELIMITER $$
CREATE FUNCTION TelefoneCliente
      (IdPed INT)
      RETURNS INT
      DETERMINISTIC
BEGIN
DECLARE telefoneCliente INT;
SELECT Cliente.Telefone INTO telefoneCliente FROM
Pedido INNER JOIN Cliente ON Pedido.IdCliente = Cliente.NumeroContribuinte
      WHERE Pedido.Id = IdPed;
RETURN telefoneCliente;
END $$

```

```

DROP FUNCTION TelefoneCliente;

```

```
SELECT TelefoneCliente(5);
```

```
-- Query que mostra os dados de uma farmácia
```

```
SELECT Nome, Telefone, Horario, Designacao, Custo FROM Farmacia  
JOIN Farmacia_fornece_Medicamento ON Farmacia_Id = Farmacia.Id  
JOIN Medicamento ON Medicamento.Id = Medicamento_Id;
```

```
-- Query que lista todos os funcionários e o distrito em que trabalham
```

```
SELECT Nome, Distrito FROM Administrador  
UNION  
SELECT Nome, Distrito FROM Gestor  
UNION  
SELECT Nome, Distrito FROM Estafeta;
```

```
-- Query que mostra os clientes com maior valor de faturação
```

```
SELECT ValorFaturacao, Nome FROM  
Pedido JOIN Cliente ON IdCliente = NumeroContribuinte  
ORDER BY ValorFaturacao DESC
```

```
-- Índice
```

```
CREATE INDEX email ON Estafeta (Email);
```

```
SHOW INDEX FROM Estafeta;
```

```
SELECT Email FROM Estafeta;
```

```
DROP INDEX email ON Estafeta;
```

```
-- Vista que lista todos os funcionários da empresa
```

```
CREATE VIEW Funcionarios  
AS  
SELECT * FROM  
(  
SELECT Id, Nome, Email, Distrito, 'Gestor' AS 'Função' FROM Gestor  
UNION  
SELECT Id, Nome, Email, Distrito, 'Administrador' AS 'Função' FROM Administrador  
UNION  
SELECT Id, Nome, Email, Distrito, 'Estafeta' AS 'Função' FROM Estafeta  
ORDER BY Função DESC, Distrito  
) AS f;
```



```
DROP VIEW Funcionarios;  
SELECT * FROM Funcionarios;
```