



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

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

Ano Lectivo de 2022/2023

### **Empresa UbMeds**

**Ana Silva (A91678), Catarina Quintas (A91650),  
Francisco Paulino (A91666), Henrique Faria  
(A91637)**

Janeiro, 2023

# **BD**

Data de Recepção	
Responsável	
Avaliação	
Observações	

## Empresa UbMeds

**Ana Silva (91678), Catarina Quintas (A91650),  
Francisco Paulino (A91666), Henrique Faria  
(A91637)**

Janeiro, 2023

## Resumo

Este relatório foi realizado para ajudar na compreensão e desenvolvimento de uma base de dados. O nosso sistema pretende fazer a criação de uma base de dados para uma pequena empresa de distribuição de medicamentos de forma a ajudar os que destes precisam, e não se podem deslocar para os comprar.

De modo a poder estruturar o nosso sistema da melhor forma, o processo foi feito em três etapas, sendo todas elas alvo de validação antes de progredirmos para a etapa seguinte. Primeiramente, estabelecemos os requisitos que a nossa base de dados teria de ser capaz de suportar. De seguida fizemos uma modelação conceptual, em que estabelecemos as entidades essenciais e os relacionamentos existentes entre si, de acordo com os requisitos definidos. Posteriormente, fizemos a validação do nosso modelo conceptual. Após este passo importante, montamos o modelo lógico partindo do conceptual, fizemos a sua análise e validação através da normalização e das interrogações do utilizador. Quando consideramos que a modelação lógica estava pronta passamos à terceira fase. Nesta terceira fase, a partir do modelo lógico geramos um modelo físico no MySQL Workbench garantindo assim, que as modelações feitas anteriormente convergiram para a obtenção de uma base de dados correta e segura. Após concluída esta fase dá-se por terminado o projeto.

**Área de Aplicação:** Desenvolvimento e Implementação de Sistemas de Bases de Dados.

**Palavras-Chave:** Bases de Dados, Bases de Dados Relacionais, Análise de Requisitos, Entidades, Atributos, Relacionamentos, Modelo Conceptual, Diagrama ER, Modelo Lógico, Modelo Físico, brModel, MySQL Workbench, SQL.

# Índice

Resumo	i
Índice	ii
Índice de Figuras	iv
Índice de Tabelas	v
1. Introdução	1
1.1. Contextualização	1
1.2. Motivação e Objetivos do Trabalho	1
1.3. Análise da Viabilidade do Processo	2
1.4. Recursos e Equipa de Trabalho	2
1.5. Plano de Execução do Projeto	3
2. Levantamento e Análise de Requisitos	4
2.1. Método de Levantamento de Requisitos	4
2.2. Organização dos Requisitos	4
2.2.1 Requisitos de Descrição	4
2.2.2. Requisitos de Exploração	5
2.2.3. Requisitos de Controlo	5
2.3. Análise e Validação Geral dos Requisitos	5
3. Modelação Conceptual	6
3.1. Apresentação da abordagem de modelação realizada	6
3.2. Identificação e Caraterização das Entidades	6
3.3. Identificação e Caraterização dos Relacionamentos	7
3.4. Identificação e Caraterização da Associação dos Atributos com as Entidades e os Relacionamentos	9
3.5. Apresentação e Explicação do Diagrama ER	11
4. Modelação Lógico	12
4.1. Construção e Validação do Modelo de Dados Lógico	12
4.2. Normalização dos Dados	13
4.3. Apresentação e Explicação do Modelo Lógico Produzido	14
4.4. Validação do Modelo com Interrogações do Utilizador	14
5. Implementação Física	16
5.1. Tradução do Esquema lógico para o Sistema de Gestão de Base de Dados escolhido em SQL	16
5.2. Tradução das Interrogações do Utilizador para SQL	16

5.3. Definição e Caracterização das Vistas de Utilização em SQL	18
5.4. Cálculo do Espaço da Base de Dados	20
5.5. Indexação do Sistema de Dados	22
5.6. Procedimentos Implementados	22
5.7. Plano de Segurança e Recuperação de Dados	24
6. Conclusões e Trabalho Futuro	25
Referências	26
Lista de Siglas e Acrónimos	27
Anexos	28
Anexo 1	28

## Índice de Figuras

Figura 1 – Diagrama de Gant	3
Figura 2- Relacionamento Utilizador-Encomenda	7
Figura 3 – Relacionamento Encomenda-Medicamento	7
Figura 4 – Relacionamento Encomenda-Veículo	8
Figura 5 – Relacionamento Encomenda-Estafeta	8
Figura 6 – Relacionamento Estafeta-Veículo	9
Figura 7 – Diagrama ER	11
Figura 8 – Modelo Lógico	14
Figura 9 – Query 1	16
Figura 10 – Query 2	17
Figura 11 – Query 3	17
Figura 12 – Query 4	17
Figura 13 – Query 5	18
Figura 14 – View 1	18
Figura 15 – View 2	19
Figura 16 – View 3	20
Figura 17 – Procedure 1	22
Figura 18 – Procedure 2	23
Figura 19 – Before Procedure 3	23
Figura 20 – After Procedure 3	23
Figura 21 – Data Export	24
Figura 22 – Data Import	24

## Índice de Tabelas

Tabela 1 – Caraterização dos Atributos da entidade Encomenda	9
Tabela 2 - Caraterização dos Atributos da entidade Estafeta	9
Tabela 3 - Caraterização dos Atributos da entidade Veículo	10
Tabela 4 - Caraterização dos Atributos da entidade Medicamento	10
Tabela 5 - Caraterização dos Atributos da entidade Utilizador	10
Tabela 6 – Tamanho (em bytes) dos atributos da Encomenda	20
Tabela 7 – Tamanho (em bytes) dos atributos do Estafeta	21
Tabela 8 - Tamanho (em bytes) dos atributos do Veículo	21
Tabela 9 - Tamanho (em bytes) dos atributos do Medicamento	21
Tabela 10 - Tamanho (em bytes) dos atributos do Utilizador	21

# **1. Introdução**

## **1.1. Contextualização**

Desde há alguns anos que em certas zonas de Braga, os habitantes tem sentido dificuldade em aceder a diversos serviços, sendo um deles o acesso a farmácias e medicamentos.

Como um grupo de 4 pequenos empreendedores (Henrique Faria, Ana Silva, Francisco Paulino e Catarina Quintas), decidimos montar uma empresa chamada UbMeds com o propósito de ajudar os mais debilitados, assim como aqueles com dificuldades a chegar a farmácias. Esta ideia surgiu durante a primeira fase da pandemia pois os idosos, similarmente aos doentes crónicos, não tinham a possibilidade de sair de casa e, no entanto, tinham de arranjar forma de adquirir tratamentos. Assim, devido a todos estes fatores, decidimos informatizar um serviço de entrega ao domicílio que fornece um bem essencial para estas pessoas.

Começamos apenas pelos nossos bairros, mas devido ao aumento de utilizadores, encontramos-nos de momento com 20 estafetas e estamos já a operar na cidade de Braga, com o objetivo de, no final do ano, termos a capacidade de progredir para os arredores da cidade.

Vimos por isso a necessidade de criar uma Base de Dados, cujo sistema deve estar preparado para, a qualquer momento, adicionar novos clientes.

É necessária uma distribuição eficiente dos medicamentos, de forma a satisfazer as necessidades das populações, sempre que estas os requisitem. A recolha e distribuição é feita por estafetas, que após receberem os pedidos, se deslocam às farmácias para recolher a devida encomenda, e efetuam a entrega. Este transporte é registado com a data e hora de recolha do produto, bem como da receção deste do cliente, e é também adicionado ao registo os medicamentos que foram entregues, e o seu destino. A cada pedido pode estar ou não associada uma prescrição-médica, consoante o tipo de medicamento, e a informação relativa ao paciente; caso a prescrição-médica não seja necessária, são apenas dados os nomes dos medicamentos. Cada entrega tem também um valor de faturação, e o valor de lucro associado. Há também uma forma de manter o registo das viaturas e dos seus respetivos condutores de forma a garantir, não só a segurança do medicamento, como dos utilizadores. Para terminar, há também uma maneira de deixar os nossos estimados clientes colocar sugestões, ou até mesmo queixas sobre o nosso trabalho, de forma que estas possam ser analisadas, com o objetivo de melhorar os nossos serviços.

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

Devido ao elevado número de pessoas que não conseguiam ou não deviam sair de casa durante a pandemia, a nossa ideia, que começou com os nossos vizinhos, rapidamente de expandiu para as nossas freguesias, o que levou a que tivéssemos de adaptar um modelo de registo de entregas, que começou com os nossos próprios computadores, onde escrevíamos num ficheiro Excel a informação que precisávamos. Com o aumento da procura do nosso



serviço, vieram cargos maiores como a necessidade de garantir os pagamentos a longas distâncias, um maior número de distribuidores, um maior número de fornecedores, e uma forma de garantir que conseguíamos lidar com toda esta informação.

No início, quando se deu a primeira expansão, sentimos de imediato as repercussões da falta de um sistema de gestão de dados, visto que os clientes se queixavam de atrasos, ou até mesmo de não ter ocorrido entregas. Potenciais clientes não podiam aderir à nossa empresa por ser demasiado longe, e chegamos mesmo a ter casos de pessoas que se faziam passar por trabalhadores da nossa pequena empresa. Resolvemos por isso desenvolver uma aplicação chamada UbMeds que por sua vez precisava de uma forma de guardar toda a informação alusiva à gestão e manutenção. Assim começamos a pensar em desenvolver a nossa base de dados.

Com isto os objetivos principais da implementação da Base de Dados assentariam na:

- Organização do modelo de negócio;
- Melhoraria da gestão do registo de pedidos efetuados;
- Organização dos fornecedores e dos estafetas da empresa;
- Organização dos clientes e a criação de um perfil dos mesmos;
- Gestão da faturação, bem como os custos e lucros efetuados pela empresa;
- Melhoria da velocidade e eficácia dos transportes na entrega do pedido.

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

Este projeto assenta na implementação de uma Base de Dados que permite estudar e relacionar os dados que a UbMeds possui, de modo a compreender e melhorar as condições de distribuição, quer para os utilizadores da app, quer para os integrantes da empresa. A utilização de uma Base de Dados Relacional é importantíssima, pois permite a manipulação e a remoção de dados sobre o nosso sistema de forma rápida, e com a complexidade que é exigida. Este investimento, embora considerável, vai significar ganhos superiores no futuro, pois ter-se-á uma melhor noção das necessidades dos clientes e dos trabalhadores. Outra vantagem, e esta mais importante, é a de garantir que os medicamentos chegam a todas as pessoas que deles precisam, através da inserção criteriosa na Base de Dados dos pedidos feitos.

A abordagem atual, que utiliza um ficheiro para guardar toda esta informação, é algo que não é centralizado, e que, portanto, obriga a que cada inserção tenha de ser feita em vários locais. Tal sistema, não só não garante que a informação esteja disponível quando é necessária, em cada situação, como também promove o erro, uma vez que não há qualquer forma de garantir que a informação é inserida corretamente em todos os ficheiros.

Com uma Base de Dados implementada num Sistema de Gestão de Base de Dados, vai ser possível fazer todas as operações de gestão da aplicação de forma simultânea, sem necessidade de reescrita, e com a segurança de que todos os intervenientes dispõem da informação necessária, tornando o sistema mais rápido e seguro.

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

Para a implementação desta base de dados a equipa de trabalho, constituída por 4 empreendedores, com conhecimentos na área de gestão e criação de bases de dados, utilizou como recursos as ferramentas BrModel, MySQL Server e MySQL Workbench.

# 1.5. Plano de Execução do Projeto

## Implementação do Sistema de Base de Dados

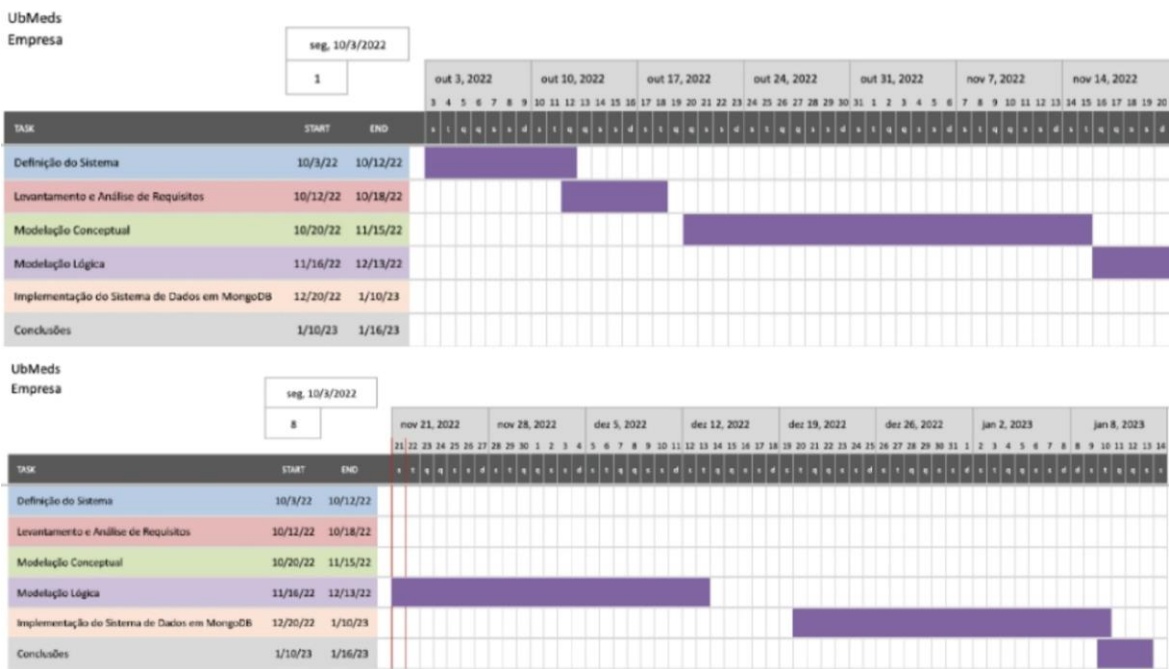


Figura 1 – Diagrama de Gant

## 2. Levantamento e Análise de Requisitos

### 2.1. Método de Levantamento de Requisitos

Após a ocorrência de alguns incidentes nas entregas, e chegarem queixas de problemas, decidimos fazer questionários de forma a perceber o que estava mal, e de que forma podíamos melhorar. Fizemos então a distribuição manual destes inquéritos pelas pessoas que estavam a usufruir dos nossos serviços, sendo estes posteriormente analisados em conjunto pela equipa. Para além disso, foram efetuadas reuniões onde foi analisada a documentação existente até ao momento, tendo em vista o que poderia ser melhorado, considerando a forma como os registos de pedidos eram efetuados, a forma como os estafetas eram registados, a lista de clientes aderentes ao serviço e os fornecedores envolvidos.

Uma vez que, durante a fase de questionários foram apontadas falhas nos serviços prestados, disponibilizamos 2 membros da equipa para, durante uma semana, acompanharem alguns dos estafetas da empresa de forma a perceber como as entregas estavam a ser efetuadas, e a seriedade com que eles executavam o seu serviço.

### 2.2. Organização dos Requisitos

Com o propósito de facilitar a organização dos nossos requisitos, tendo em conta a estrutura estabelecida pela linguagem SQL, procedemos à organização dos requisitos levantados da seguinte forma: **requisitos de descrição**, **requisitos de exploração** e **requisitos de controlo**.

#### 2.2.1 Requisitos de Descrição

1. Cada cliente tem de ter no seu registo um identificador próprio, nome, NIF, endereço com código postal, localidade, rua, porta e andar(opcional);
2. Todos os estafetas deverão possuir no seu registo um identificador próprio, nome e contacto;
3. Nos medicamentos é esperado ter acesso ao seu identificador próprio, preço, nome, venda livre, dosagem e quantidade stock;
4. Cada encomenda realizada deve ter uma data de compra e data de entrega;
5. Uma entrega contém várias encomendas e é realizada por um ou mais estafetas;
6. Um estafeta faz várias entregas, podendo conduzir diferentes veículos;
7. Cada encomenda que o cliente efetua tem um custo associado.

## **2.2.2. Requisitos de Exploração**

1. Listar todos os pedidos;
2. Listar todos os medicamentos que um cliente já comprou;
3. Listar os pedidos de um estafeta;
4. Listar os pedidos realizadas num dia;
5. Lista o nome de todos os clientes de uma determinada localidade;
6. Pretende-se também, que seja possível verificar qual o utilizador que trouxe mais lucro à empresa;
7. Deve ser possível identificar o lucro mensal de cada estafeta.

## **2.2.3. Requisitos de Controlo**

1. O pedido tem de ser realizado por um utilizador;
2. O utilizador apenas consegue gerir a sua informação e pedido;
3. Os funcionários fazem não só a entrega do pedido como a recolha nas farmácias e inserem a data de entrega do mesmo;
4. O utilizador no fim consegue introduzir a sua satisfação com a eficácia da entrega;
5. Os gerentes têm acesso a toda a base de dados;
6. Cada encomenda pode ser realizada por vários veículos.

## **2.3. Análise e Validação Geral dos Requisitos**

Tendo em conta o processo de levantamento aplicado neste projeto, os requisitos levantados foram divididos em 3 categorias: requisitos de descrição, de exploração, e de controlo.

Na primeira fase de levantamento dos requisitos de descrição, foram analisados, principalmente, os elementos essenciais para o funcionamento da aplicação.

Na fase de levantamento de requisitos de exploração, foram analisadas as informações que se poderiam obter, através dos vários elementos e relacionamentos presentes na base de dados.

Na última fase, correspondente ao levantamento dos requisitos de controlo, foram definidas quais as restrições necessárias para a administração da base de dados de forma a tornar o sistema conciso.

Após tomar em consideração estes 3 grupos de requisitos, a equipa de trabalho analisou a viabilidade dos mesmos, e validou a sua aplicação de forma a construir uma base de dados adequada face às exigências apresentadas.

## 3. Modelação Conceptual

### 3.1. Apresentação da abordagem de modelação realizada

Considerando os requisitos levantados e validados anteriormente, vamos agora proceder à explicação do processo de planeamento da estrutura da Base de Dados implementada. Para alcançar isto, vamos construir um Diagrama ER que representará as diversas entidades, incluindo os atributos que as compõem e os relacionamentos existentes entre as mesmas. Sendo assim, vamos para apresentar os diferentes componentes desta modelação, simulando o processo que decorreu. Ou seja, começamos pela identificação e caracterização de cada entidade, seguida pela caracterização e identificação de cada relacionamento e, por fim, o diagrama obtido.

### 3.2. Identificação e Caracterização das Entidades

Tendo em conta a informação obtida durante o processo de levantamento dos requisitos, as entidades que consideramos essenciais são: Encomenda, Utilizador, Medicamento, Estafeta e Veículo. Assim, de forma a cumprir os requisitos, apresentamos a seguir as diferentes entidades com os seus respetivos atributos.

- **Encomenda:** Esta entidade corresponde à encomenda efetuada por cada utilizador, sendo identificada por um Id da encomenda. Contém informações sobre a data de compra, data de entrega, estado da encomenda, preço total e avaliação (opcional);
- **Utilizador:** Esta entidade corresponde aos diversos Utilizadores da Aplicação, sendo identificada por um id de utilizador. Contém informações sobre o nome do usuário, NIF e Endereço que inclui a localidade, código postal, rua, porta e em certos casos andar;
- **Medicamento:** Esta entidade corresponde aos medicamentos que serão requisitados pelos utilizadores deste serviço, sendo identificada por um id do medicamento. Contém informações sobre o nome dos medicamentos, a quantidade pedida, o preço associado a cada medicamento, a dosagem recomendada e se é de venda livre;
- **Estafeta:** Esta entidade corresponde aos funcionários da empresa que realizam as entregas, sendo identificada por um Id do estafeta. Para além disso contém informações sobre o nome do funcionário e o seu contacto;
- **Veículo:** Esta entidade corresponde aos veículos associados à empresa por parte dos estafetas, sendo identificada por um ID do veículo. Contem informações sobre o tipo de veículo utilizado e dependendo do veículo a sua matrícula.

### 3.3. Identificação e Caracterização dos Relacionamentos

- **Relacionamento Utilizador-Encomenda**

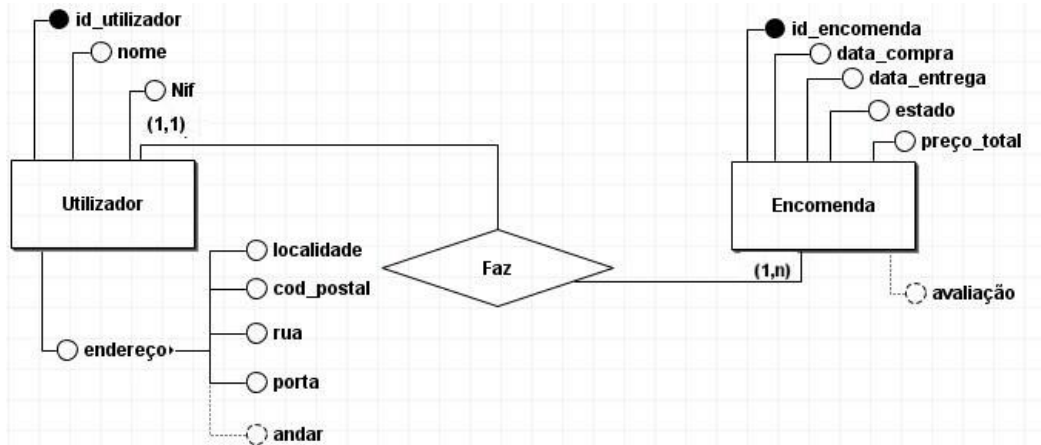


Figura 2- Relacionamento Utilizador-Encomenda

Relacionamento: Utilizador faz Encomenda

Descrição: Cada utilizador pode fazer uma, ou mais, encomendas

Cardinalidade: Utilizador (1, 1) – Encomenda (1,N)

Atributos: Este relacionamento não possui atributos.

- **Relacionamento Encomenda-Medicamento**

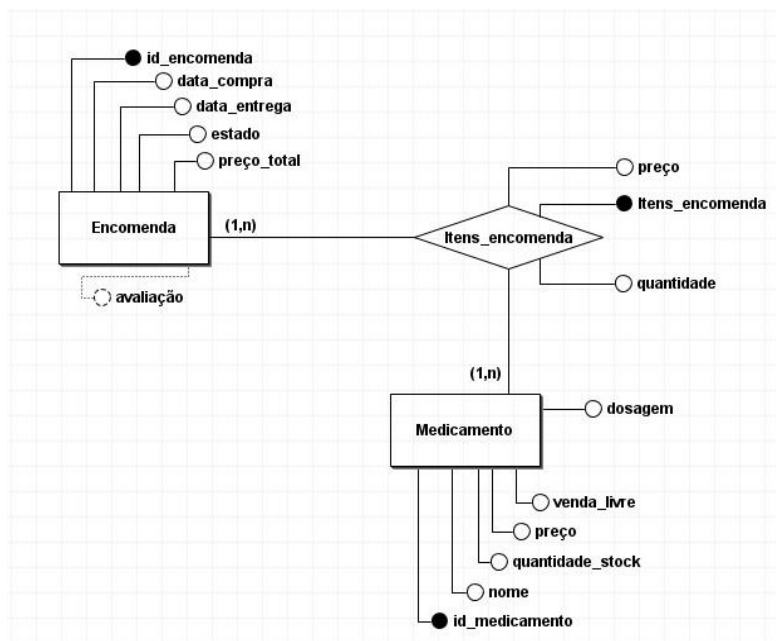


Figura 3 – Relacionamento Encomenda-Medicamento

Relacionamento: medicamentos associados à encomenda

Descrição: cada encomenda pode conter um, ou mais, medicamentos

Cardinalidade: Encomenda (1, N) – Medicamento (1, N)

Atributos: preço, quantidade e itens\_encomenda.

- **Relacionamento Encomenda-Veículo**

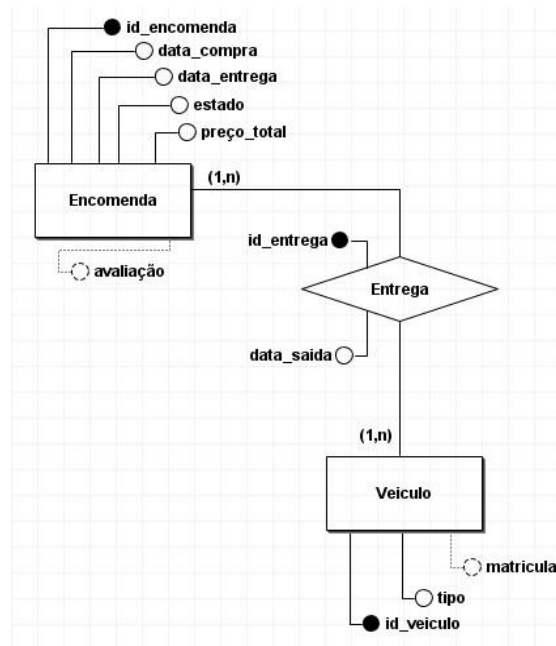


Figura 4 – Relacionamento Encomenda-Veículo

Relacionamento: veículo usado no transporte da encomenda.

Descrição: um veículo pode levar uma ou mais encomendas.

Cardinalidade: Encomenda (1, N) – Veículo (1, N)

Atributos: id\_entrega, data\_saida.

- **Relacionamento Encomenda-Estafeta**

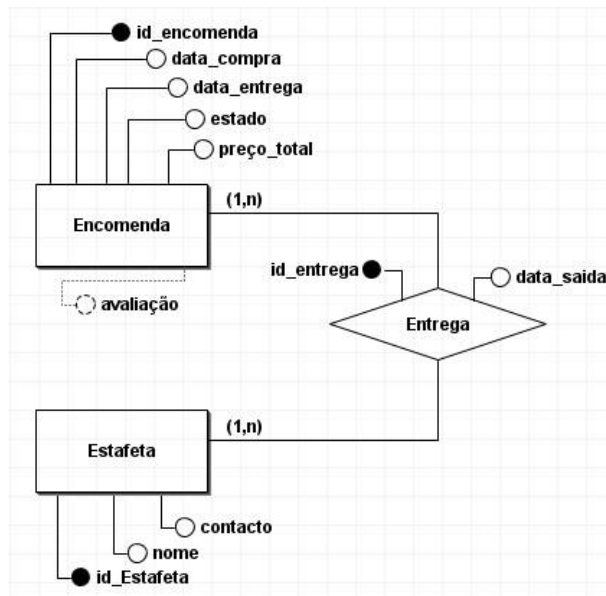


Figura 5 – Relacionamento Encomenda-Estafeta

Relacionamento: estafeta associado ao transporte da encomenda.

Descrição: um estafeta pode estar encarregue do transporte de uma ou mais encomendas.

Cardinalidade: Encomenda (1, N) – Estafeta (1, N)

Atributos: id\_entrega, data\_saida.

- **Relacionamento Estafeta-Veículo**

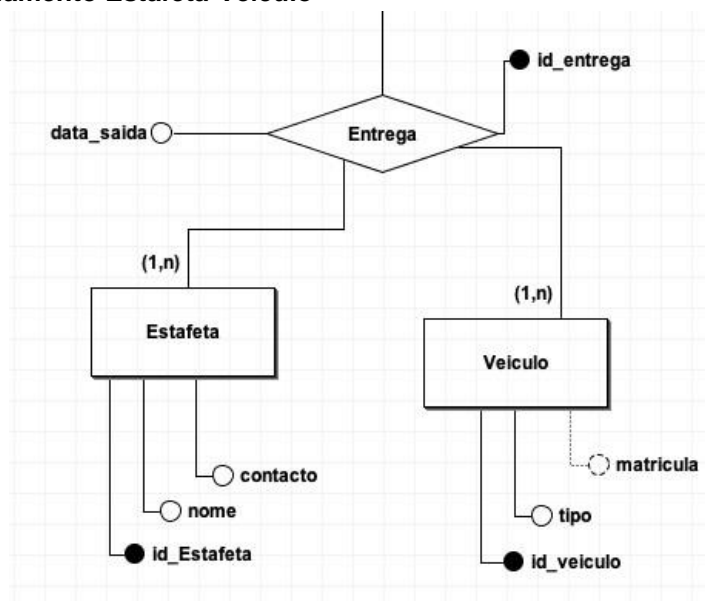


Figura 6 – Relacionamento Estafeta-Veículo

Relacionamento: Cada estafeta efetua a entrega associada ao veículo.

Descrição: cada estafeta vai usar um dos seus veículos para efetuar a entrega.

Cardinalidade: Veículo (1, N) – Estafeta (1, N)

Atributos: id\_entrega, data\_saida.

### 3.4. Identificação e Caracterização da Associação dos Atributos com as Entidades e os Relacionamentos

- **Encomenda**

Atributos	Tipo de Dados	Nulo	Composto	Multivalorado	Derivado	Chave Primária
id_encomenda	INT	Não	Não	Não	Não	Sim
data_compra	DATE	Não	Não	Não	Não	Não
data_entrega	DATETIME	Não	Não	Não	Não	Não
estado	VARCHAR(45)	Não	Não	Não	Não	Não
preço_total	DECIMAL(4)	Não	Não	Não	Não	Não
avaliação	TEXT(90)	Sim	Não	Não	Não	Não

Tabela 1 – Caracterização dos Atributos da entidade Encomenda

- **Estafeta**

Atributos	Tipo de Dados	Nulo	Composto	Multivalorado	Derivado	Chave Primária
id_estafeta	INT	Não	Não	Não	Não	Sim
nome	VARCHAR(45)	Não	Não	Não	Não	Não
contacto	VARCHAR(45)	Não	Não	Não	Não	Não

Tabela 2 - Caracterização dos Atributos da entidade Estafeta



- **Veículo**

Atributos	Tipo de Dados	Nulo	Composto	Multivalorado	Derivado	Chave Primária
id_estafeta	INT	Não	Não	Não	Não	Sim
tipo	VARCHAR(45)	Não	Não	Não	Não	Não
matrícula	VARCHAR(45)	Sim	Não	Não	Não	Não

Tabela 3 - Caracterização dos Atributos da entidade Veículo

- **Medicamento**

Atributos	Tipo de Dados	Nulo	Composto	Multivalorado	Derivado	Chave Primária
id_medicamento	INT	Não	Não	Não	Não	Sim
nome	VARCHAR(45)	Não	Não	Não	Não	Não
quantidade	INT	Não	Não	Não	Não	Não
preço	DECIMAL(4)	Não	Não	Não	Não	Não
dosagem	INT	Não	Não	Não	Não	Não
venda_livre	TINYINT	Não	Não	Não	Não	Não

Tabela 4 - Caracterização dos Atributos da entidade Medicamento

- **Utilizador**

Atributos	Tipo de Dados	Nulo	Composto	Multivalorado	Derivado	Chave Primária
id_utilizador	INT	Não	Não	Não	Não	Sim
nome	VARCHAR(45)	Não	Não	Não	Não	Não
Nif	VARCHAR(45)	Sim	Não	Não	Não	Não
rua	VARCHAR(45)	Não	Sim	Não	Não	Não
cod_Postal	VARCHAR(45)	Não	Sim	Não	Não	Não
porta	VARCHAR(45)	Não	Sim	Não	Não	Não
andar	VARCHAR(45)	Sim	Sim	Não	Não	Não
localidade	VARCHAR(45)	Não	Sim	Não	Não	Não

Tabela 5 - Caracterização dos Atributos da entidade Utilizador

### 3.5. Apresentação e Explicação do Diagrama ER

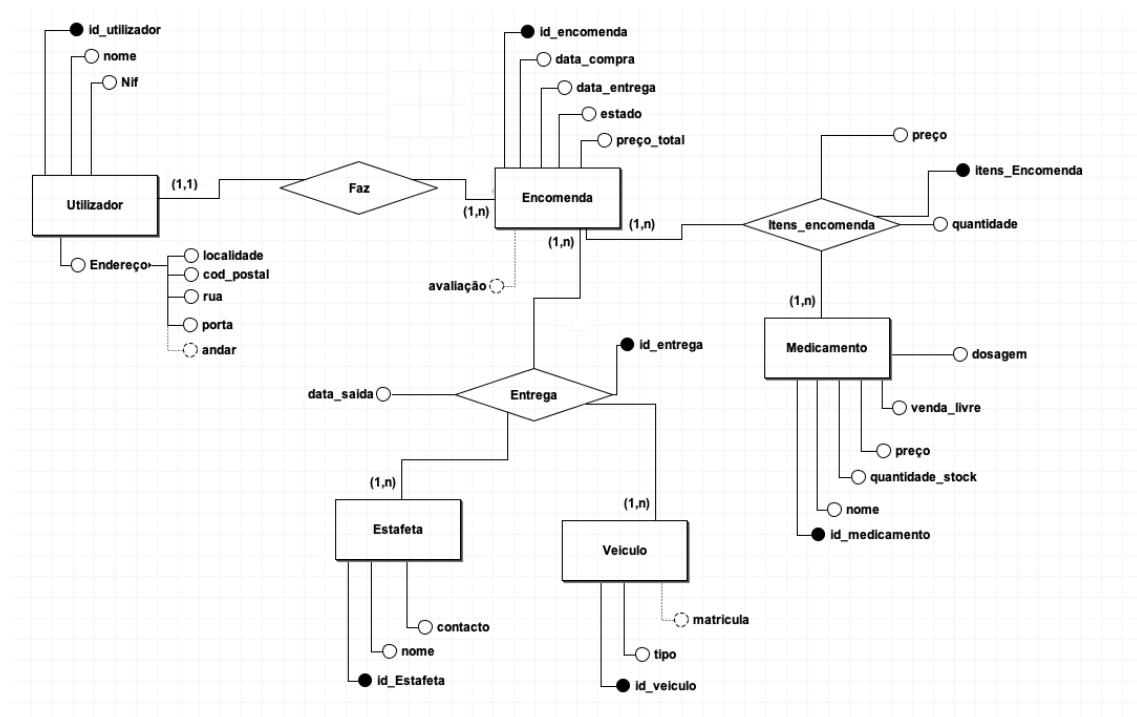


Figura 7 – Diagrama ER

Este diagrama pode ser interpretado como um utilizador pode realizar uma ou mais encomendas, cada encomenda vai estar associada a uma lista de medicamentos, e cada encomenda vai ser entregue por um dado estafeta, e um dado veículo associado a esse estafeta.

## 4. Modelação Lógico

### 4.1. Construção e Validação do Modelo de Dados Lógico

O modelo lógico é baseado no modelo conceptual. As entidades e atributos compostos multivalorados são convertidos em tabelas. Os identificadores próprios de cada entidade vão passar a ser PRIMARY KEYS, e os identificadores de outras entidades com a qual possuam uma relação vão ser FOREIGN KEYS.

- **Encomenda**  
Primary Key: id\_encomenda(INT);  
Atributos: data\_compra(DATE), data\_entrega(DATETIME), estado(VARCHAR(45)), preco\_total (DECIMAL(4,2)), avaliação(TEXT);  
Foreign Key: id\_utilizador (INT), id\_Entrega(INT)
- **Utilizador**  
Primary Key: id\_utilizador (INT)  
Atributos: nome (VARCHAR(45)), Nif (VARVHAR(45))  
Foreign Key: id\_endereço (INT)
- **Veiculo**  
Primary Key: id\_Veiculo (INT)  
Atributos: tipo (VARVHAR(45)), matrícula (VARCHAR(45))  
Foreign Key: Não possui
- **Estafeta**  
Primary Key: id\_estafeta (INT)  
Atributos: nome (VARCHAR(45)), contacto (VARCHAR(45))  
Foreign Key: Não possui
- **Medicamento**  
Primary Key: id\_medicamento (INT)  
Atributos: nome (VARCHAR(45)), quantidade\_stock (INT), preço (DECIMAL(4,2)), dosagem (INT), venda\_livre (TINYINT)  
Foreign Key: Não possui

#### Atributo composto:

- **Endereço**  
Primary Key: id\_endereço  
Atributos: rua, cod\_postal, porta, andar, localidade  
Foreign Key: Não possui

#### Relacionamentos:

- **Entrega**  
Primary Key: id\_Entrega  
Atributos: data\_saida  
Foreign Key: Veiculo\_id, Estafeta\_id
- **Itens\_Encomenda**  
Primary Key: itens\_Encomenda  
Atributos: quantidade, preço  
Foreign Key: id\_medicamento, id\_encomenda

## 4.2. Normalização dos Dados

O processo de normalização é uma técnica formal de verificação das diversas tabelas base de um esquema de uma base de dados, que se baseia na análise das chaves primárias e das dependências funcionais de todos os seus atributos. É um processo progressivo, que assenta na execução de uma série de etapas, cada uma delas correspondendo a uma forma normal específica, que vai transformando o esquema de uma base de dados sucessivamente em direção a um estado mais robusto e menos vulnerável.

A **primeira forma normal (1FN)** acontece quando todos os seus atributos são atômicos, ou seja, não são multivalorados, e não contêm grupos de dados repetidos. Adicionalmente, para uma dada tabela estar na primeira forma normal, esta tem de possuir uma chave primária.

Na prática, podemos dizer que uma relação na qual as interseções entre colunas (atributos) e linhas (registos) apenas contêm um único valor – um valor atômico – está na 1FN.

A **segunda forma normal (2FN)** ocorre se, para além de estar previamente normalizada na primeira forma, todos os atributos não-primos (que não fazem parte de uma chave primária) de uma tabela forem totalmente dependentes da sua chave primária.

A **terceira forma normal (3FN)** verifica-se quando temos uma tabela que se encontra já nas primeira e segunda formas normalizadas, e adicionalmente, onde todos os atributos que não são chaves primárias, sejam mutuamente independentes, não havendo assim dependências funcionais transitivas.

As tabelas nesta Base de Dados estão na terceira forma normal (3NF). As chaves primárias são únicas e não nulas, e cada atributo depende funcionalmente da chave primária. As restrições da chave estrangeira também garantem que cada atributo faça referência a uma chave primária noutra tabela e que não haja dependências parciais.

### 4.3. Apresentação e Explicação do Modelo Lógico Produzido

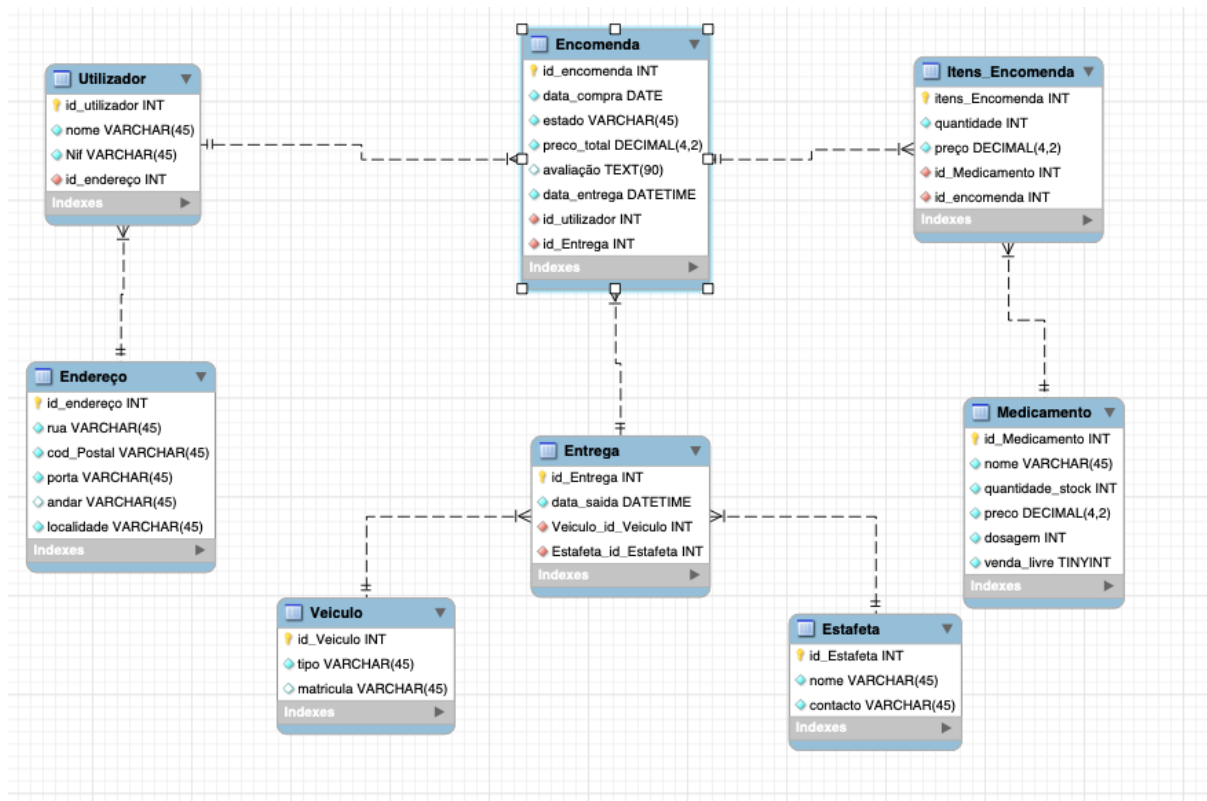


Figura 8 – Modelo Lógico

### 4.4. Validação do Modelo com Interrogações do Utilizador

1. Tabela de todas as encomendas realizadas:  

$$\pi * (\text{Encomenda})$$
2. Compras realizadas num determinado dia, neste caso, no '2022-01-02':  

$$\Pi \text{id\_encomenda}(\sigma \text{data\_compra}='2022-01-02'(\text{Encomenda}))$$
3. Tabela com o ranking de entregas;  

$$\pi \text{Quantidade, Veiculo.tipo}(\sigma \text{Veiculo.id\_Veiculo} = \text{Entrega.Veiculo\_id\_Veiculo}(\text{Veiculo} \bowtie \text{Entrega}))$$
4. Tabela de clientes com um código postal estabelecido, que realizaram uma encomenda num certo dia;  

$$\pi \text{Id, Nome, Nif, Id\_encomenda}(\sigma \text{data\_compra}='2022-01-02' \wedge \text{cod\_Postal}='4710-383'(\text{Utilizador} \bowtie \text{Endereço} \bowtie \text{Encomenda}))$$

5. Tabela dos medicamentos transportados por um determinado veículo:  
 $\pi_{\text{nome}}(\sigma_{\text{matricula}='AA-00-00'}(\text{Medicamento} \bowtie \text{Itens\_Encomenda} \bowtie \text{Encomenda} \bowtie \text{Entrega} \bowtie \text{Veiculo}))$

## 5. Implementação Física

### 5.1. Tradução do Esquema lógico para o Sistema de Gestão de Base de Dados escolhido em SQL

O sistema de gestão de bases de dados utilizado para a implementação deste projeto foi o MySQL, visto que este foi fortemente recomendado. Este consiste num sistema de gerenciamento de bases de dados relacional de código aberto, que tem a linguagem SQL como interface.

Ao longo da realização deste projeto, foram usadas as ferramentas MySQL, e também MySQL Workbench que, graças à simples manipulação de dados que estas garantem, tornaram o processo de criação de uma base de dados mais acessível, o que, por sua vez enriqueceu a nossa experiência.

### 5.2. Tradução das Interrogações do Utilizador para SQL

1. Resulta em uma tabela com todas as encomendas realizadas;

```
SELECT * FROM Encomenda;
```

	id_encomenda	data_compra	estado	preco_total	avaliação	data_entrega	id_utilizad...	id_Entrega
▶	1	2022-01-02	Entregue	10.32	Bom serviço	2022-01-10 12:59:34	1	3
	2	2022-01-20	Entregue	21.87		2022-01-31 18:21:34	2	6
	3	2021-02-24	Entregue	36.92	Serviço Rápido	2021-02-25 12:01:34	15	5
	4	2022-01-02	Entregue	11.74		2022-01-06 09:31:03	13	2
	5	2022-01-09	Entregue	3.21	Demora na entrega	2022-01-31 18:21:34	12	7
	6	2022-12-01	Entregue	17.30	Otimo serviço	2022-12-05 14:45:34	12	1
	7	2022-01-31	Entregue	5.09	Qualidade e rapidez	2022-02-01 11:25:58	9	4
	8	2022-01-29	Entregue	67.32		2022-01-31 14:45:34	2	8
	9	2021-11-13	Entregue	5.09	Qualidade e rapidez	2021-11-16 19:40:00	12	9
	10	2022-07-09	Entregue	27.75		2022-07-12 17:21:56	9	10
	11	2021-05-04	Entregue	28.09	Bastante satisfeito	2021-05-13 18:21:34	8	11
	12	2022-01-02	Entregue	13.32	Bom serviço	2022-03-10 12:59:34	1	12
	13	2021-10-07	Entregue	21.87		2021-10-28 18:21:34	2	13
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 9 – Query 1

2. Mostra todas as compras realizadas no dia '2022-01-02';

```
SELECT Encomenda.id_encomenda FROM Encomenda
WHERE Encomenda.data_compra = '2022-01-02';
```

id_encomenda
1
4
12
NULO

Figura 10 – Query 2

3. Dá o ranking das entregas realizadas;

```
SELECT count(*) as Quantidade, Veiculo.tipo FROM Veiculo
INNER JOIN Entrega
ON Veiculo.id_Veiculo = Entrega.Veiculo_id_Veiculo
GROUP BY Veiculo.tipo;
```

	Quantidade	tipo
▶	7	Carro
	3	Mota
	3	Bicicleta

Figura 11 – Query 3

4. Origina a tabela de clientes com o código postal '4710-383', que realizaram uma compra no dia '2022-01-02';

```
SELECT Utilizador.id_utilizador AS Id,
Utilizador.nome AS Nome,
Utilizador.Nif AS Nif,
Encomenda.id_encomenda AS Id_encomenda
FROM Utilizador
INNER JOIN Endereço
ON Utilizador.id_endereço = Endereço.id_endereço
INNER JOIN Encomenda
ON Utilizador.id_utilizador = Encomenda.id_utilizador
WHERE Encomenda.data_compra = '2022-01-02' AND Endereço.cod_Postal = '4710-383';
```

	Id	Nome	Nif	Id_encomenda
▶	1	João António	250223110	1
	1	João António	250223110	12

Figura 12 – Query 4



- Determina quais os medicamentos transportados pelo veículo com a matrícula 'AA-00-00'.

```
SELECT Medicamento.nome
FROM Medicamento
  INNER JOIN Itens_Encomenda
    ON Medicamento.id_Medicamento = Itens_Encomenda.id_Medicamento
  INNER JOIN Encomenda
    ON Itens_Encomenda.id_encomenda = Encomenda.id_encomenda
  INNER JOIN Entrega
    ON Encomenda.id_Entrega = Entrega.id_Entrega
  INNER JOIN Veiculo
    ON Entrega.Veiculo_id_Veiculo = Veiculo.id_Veiculo
WHERE Veiculo.matricula = 'AA-00-00';
```

nome
▶ Ácido acetilsalicílico

Figura 13 – Query 5

### 5.3. Definição e Caracterização das Vistas de Utilização em SQL

- Todos os Medicamentos sem venda livre;

```
CREATE VIEW sem_venda_livre AS
SELECT * from Medicamento WHERE Medicamento.venda_livre = '0';
```

	id_Medicamen...	nome	quantidade_sto...	preco	dosagem	venda_livre
▶	2	Ácido acetilsalicílico	20	10.00	500	0
	4	Irbesartan	40	6.50	75	0
	NULL	NULL	NULL	NULL	NULL	NULL

Figura 14 – View 1

2. Demora das Entregas;

```
CREATE VIEW demora_entregas AS
SELECT Entrega.id_Entrega,
Entrega.data_saida, Encomenda.data_entrega,
datediff(Encomenda.data_entrega, Entrega.data_saida)as TIME
FROM Entrega
INNER JOIN Encomenda
ON Entrega.id_entrega = Encomenda.id_entrega
ORDER BY TIME DESC;
```

	id_Entrega	data_saida	data_entrega	TIME
▶	13	2021-10-21 13:21:21	2021-10-28 18:21:34	7
	11	2021-05-08 16:29:47	2021-05-13 18:21:34	5
	3	2022-01-07 15:50:34	2022-01-10 12:59:34	3
	2	2022-01-04 17:21:56	2022-01-06 09:31:03	2
	1	2022-12-03 18:21:34	2022-12-05 14:45:34	2
	9	2021-11-14 12:00:01	2021-11-16 19:40:00	2
	5	2021-02-24 09:20:05	2021-02-25 12:01:34	1
	7	2022-01-30 09:10:01	2022-01-31 18:21:34	1
	4	2022-01-31 12:10:01	2022-02-01 11:25:58	1
	8	2022-01-30 14:10:20	2022-01-31 14:45:34	1
	10	2022-07-11 08:30:20	2022-07-12 17:21:56	1
	12	2022-03-09 15:35:52	2022-03-10 12:59:34	1
	6	2022-01-31 12:10:01	2022-01-31 18:21:34	0

Figura 15 – View 2

### 3. Ranking do lucro dos Estafetas.

```
CREATE VIEW Ranking_Estafetas AS
SELECT Estafeta.nome, SUM(Encomenda.preco_total) AS Lucro
FROM Encomenda
INNER JOIN Entrega
ON Encomenda.id_Entrega = Entrega.id_Entrega
INNER JOIN Estafeta
ON Entrega.Estafeta_id_Estafeta = Estafeta.id_Estafeta
GROUP BY Estafeta.nome
ORDER BY Lucro DESC;
```

	nome	Lucro
►	Vitor Joaquim	67.32
	Debora Martins	49.62
	Maria Antunes	36.92
	Nelson Matos	35.19
	Nuno Machado	29.04
	Mariana Sousa	28.09
	João Guimarães	10.32
	António Mendes	5.09
	Amélia Cruz	5.09
	Márcia Espirito Santo	3.21

Figura 16 – View 3

## 5.4. Cálculo do Espaço da Base de Dados

Para obtermos o espaço que é ocupado pela Base de Dados, vamos começar por calcular o espaço de cada entidade.

Assim sendo, sabemos que um **Int** corresponde a 4 bytes, um **Date** corresponde a 3 bytes e um **Varchar(N)** ocupa  $2*N + 1$  caso a sua codificação seja mais complexa do que ASCII, mas o valor de  $2*N$  é menor que 255 e ocupa  $2*N + 2$  caso  $2*N > 255$ . Temos ainda o **Tinyint**, que corresponde a 1 byte, o **Text(N)** que ocupa  $N+2$  bytes quando  $N < 2^{16}$ , e por fim, ambos o **Decimal** e o **Datetime** correspondem a 8 bytes.

- **Encomenda**

Atributos	Tipo de Dados	Tamanho (em bytes)
id_encomenda	INT	4 bytes
data_entrega	DATETIME	8 bytes
data_compra	DATETIME	8 bytes
estado	VARCHAR(45)	$2*45+1 = 91$ bytes
preço_total	DECIMAL(4,2)	8 bytes
Avaliação	TEXT(90)	$90+2 = 92$ bytes

Tabela 6 – Tamanho (em bytes) dos atributos da Encomenda

- **Estafeta**

Atributo	Tipo de Dados	Tamanho (em bytes)
id_Estafeta	INT	4 bytes
nome	VARCHAR(45)	$2 \cdot 45 + 1 = 91$ bytes
contacto	VARCHAR(45)	$2 \cdot 45 + 1 = 91$ bytes

Tabela 7 – Tamanho (em bytes) dos atributos do Estafeta

- **Veículo**

Atributos	Tipo de Dados	Tamanho (em bytes)
id_veiculo	INT	4 bytes
tipo	VARCHAR(45)	$2 \cdot 45 + 1 = 91$ bytes
matricula	VARCHAR(45)	$2 \cdot 45 + 1 = 91$ bytes

Tabela 8 - Tamanho (em bytes) dos atributos do Veículo

- **Medicamento**

Atributos	Tipo de Dados	Tamanho (em bytes)
id_medicamento	INT	4 bytes
nome	VARCHAR(45)	$2 \cdot 45 + 1 = 91$ bytes
quantidade_stock	INT	4 bytes
preço	DECIMAL(4,2)	8 bytes
venda_livre	TINYINT	1 byte
dosagem	INT	4 bytes

Tabela 9 - Tamanho (em bytes) dos atributos do Medicamento

- **Utilizador**

Atributos	Tipo de Dados	Tamanho (em bytes)
id_utilizador	INT	4 bytes
nome	VARCHAR(45)	$2 \cdot 45 + 1 = 91$ bytes
NIF	VARCHAR(45)	$2 \cdot 45 + 1 = 91$ bytes
localidade	VARCHAR(45)	$2 \cdot 45 + 1 = 91$ bytes
cod_postal	VARCHAR(45)	$2 \cdot 45 + 1 = 91$ bytes
rua	VARCHAR(45)	$2 \cdot 45 + 1 = 91$ bytes
porta	VARCHAR(45)	$2 \cdot 45 + 1 = 91$ bytes
andar	VARCHAR(45)	$2 \cdot 45 + 1 = 91$ bytes

Tabela 10 - Tamanho (em bytes) dos atributos do Utilizador

## 5.5. Indexação do Sistema de Dados

Em SQL , a utilização de índices numa Base de Dados permite que o tempo de procura de informação seja reduzido, no entanto, leva a um maior consumo de espaço de armazenamento. O SQL , por definição, gera automaticamente índices para cada objeto criado que contém uma Primary Key, denominados por índices implícitos. Contudo, além destes, é possível criar novos índices, conhecidos como índices explícitos.

Para a base de dados da UbMeds, consideramos pertinente a criação dos seguintes índices:

1.

```
CREATE INDEX NomeMedicamento ON Medicamento(nome);
```

2.

```
CREATE INDEX Data_Entrega on Encomenda(data_entrega);
```

## 5.6. Procedimentos Implementados

1. Procedimento correspondente ao top 3 das encomendas realizadas e respetivos clientes


```
DELIMITER $$  
CREATE PROCEDURE Top3Pagamentos()  
BEGIN  
SELECT Utilizador.nome as Nome, Encomenda.preco_total as Preço  
FROM Utilizador INNER JOIN Encomenda  
ON Utilizador.id_utilizador = Encomenda.id_utilizador  
ORDER BY Encomenda.preco_total DESC  
LIMIT 3;  
END $$  
CALL Top3Pagamentos;
```

Nome	Preço
José Maria	67.32
Manuel Brandão	36.92
Lucélia Martins	28.09

Figura 17 – Procedure 1

2. Consultar o número total de entregas num determinado ano, neste caso, 2022

```
DELIMITER $$
CREATE PROCEDURE n_total_ano (IN ano INT)
BEGIN
SELECT count(id_Entrega) as n_Entregas FROM Entrega WHERE
YEAR(data_saida)=ano;
END $$
CALL n_total_ano(2022);
```



n_Entregas
9

Figura 18 – Procedure 2

3. Atualização do preço de um medicamento devido à inflação

```
DELIMITER $$
CREATE PROCEDURE Update_Preço_Med (IN med INT, IN new_preco
DECIMAL(4,2))
BEGIN
UPDATE Medicamento SET preco = new_preco where
id_Medicamento=med;
END $$
call Update_Preço_Med (1,20.04);
SELECT * FROM Medicamento;
```

id_Medicamen...	nome	quantidade_sto...	preco	dosagem	venda_livre
1	Azitromicina	10	7.65	150	1
2	Ácido acetilsalicílico	20	10.00	500	0
3	Retinol	30	11.10	120	1
4	Irbesartan	40	6.50	75	0
5	Brufen	50	9.90	300	1
NULL	NULL	NULL	NULL	NULL	NULL

Figura 19 – Before Procedure 3

id_Medicamen...	nome	quantidade_sto...	preco	dosagem	venda_livre
1	Azitromicina	10	20.04	150	1
2	Ácido acetilsalicílico	20	10.00	500	0
3	Retinol	30	11.10	120	1
4	Irbesartan	40	6.50	75	0
5	Brufen	50	9.90	300	1
NULL	NULL	NULL	NULL	NULL	NULL

Figura 20 – After Procedure 3

## 5.7. Plano de Segurança e Recuperação de Dados

O plano de segurança, contra a potencial perda de dados ou, até mesmo, contra a existência de erros que possam corromper a base de dados, consiste na realização de backups frequentes do projeto.

No MySQL recorremos às seguintes funcionalidades: Data export -> selecionar a base de dados -> Self contained file -> Start export e, após realizados estes passos, a base de dados encontra-se guardada num script SQL.

Assim, caso seja necessário efetuar a recuperação da base de dados, seguimos os seguintes passos: Data Import/restore -> Selecionar a script pretendida -> Start import e, depois disto, fica concluída a recuperação da base de dados.

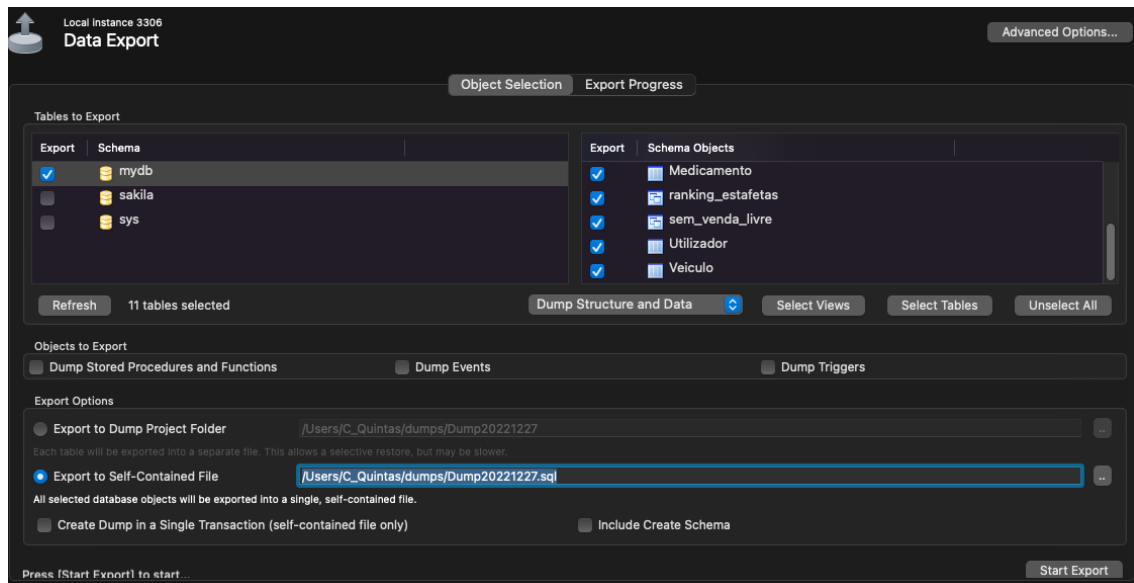


Figura 21 – Data Export

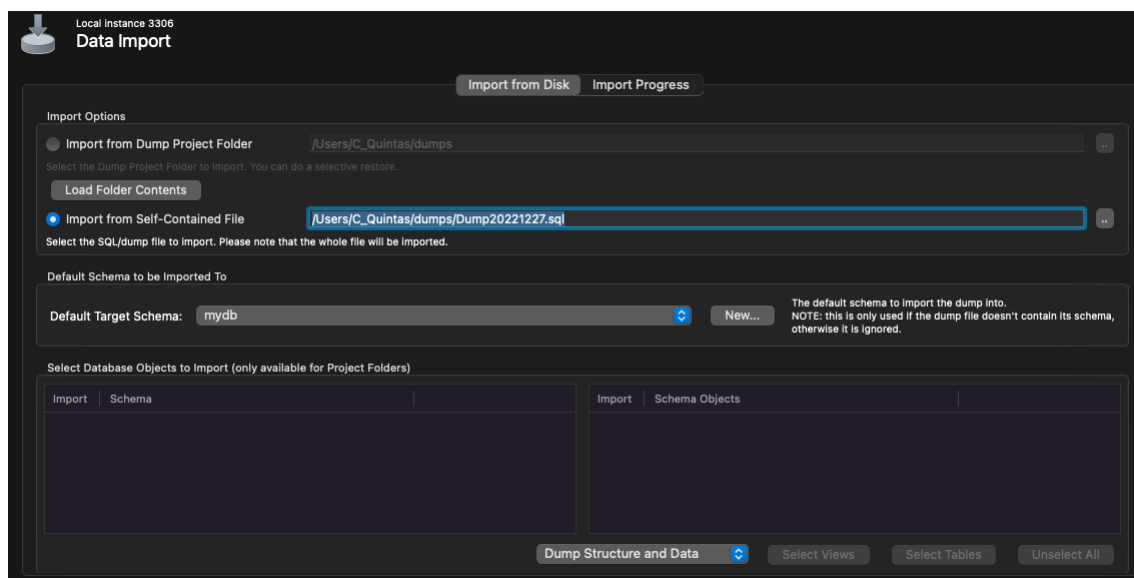


Figura 22 – Data Import

## **6. Conclusões e Trabalho Futuro**

Com a implementação desta base de dados, verificamos que trará diversos benefícios para o nosso serviço. Entre eles, estão, um sistema mais robusto capaz de gerir melhor o negócio, tendo por base uma boa gestão dos pedidos efetuados e a sua entrega por parte dos estafetas que trabalham para a empresa, com também a capacidade para expandir o serviço futuramente. Apesar da solução desenvolvida ser eficaz e de fácil implementação para as necessidades da empresa, o grupo de trabalho também identificou algumas melhorias que poderiam ser implementadas, tais como: a melhor gestão de lucro e despesas de empresa, e um melhor controlo do estado da entrega da encomenda.

Assim, tendo em conta a capacidade de expansão da base de dados construída, podemos reconhecer que estamos perante um projeto que permitirá resolver os desafios identificados para oferecer um melhor serviço. bem como algar a outras necessidades do negócio no futuro.



## Referências

Connolly, T., Begg, C., 2014. Database Systems: A Practical Approach to Design, Implementation, and Management. Addison-Wesley, Global Edition.

Belo, O., 2021. Bases de Dados Relacionais: Implementação com MySQL. FCA, Editora de Informática. Gouveia, F., 2021.

MySQL, 2021. Reference Manual. [online] Available at: <https://dev.mysql.com/doc/refman/8.0/en/>

## **Lista de Siglas e Acrónimos**

NIF – Número Identificação Fiscal

SQL – Structured Query Language

# Anexos

## Anexo 1

```
-- 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` ;
USE `mydb` ;

-- -----
-- Table `mydb`.`Endereço`
-- -----

CREATE TABLE IF NOT EXISTS `mydb`.`Endereço` (
  `id_endereço` INT NOT NULL,
  `rua` VARCHAR(45) NOT NULL,
  `cod_Postal` VARCHAR(45) NOT NULL,
  `porta` VARCHAR(45) NOT NULL,
  `andar` VARCHAR(45) NULL,
  `localidade` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id_endereço`))
ENGINE = InnoDB;

-- -----
-- Table `mydb`.`Utilizador`
-- -----

CREATE TABLE IF NOT EXISTS `mydb`.`Utilizador` (
  `id_utilizador` INT NOT NULL,
  `nome` VARCHAR(45) NOT NULL,
```

```

`Nif` VARCHAR(45) NOT NULL,
`id_endereço` INT NOT NULL,
PRIMARY KEY (`id_utilizador`),
INDEX `fk_utilizador_endereço_idx` (`id_endereço` ASC) VISIBLE,
CONSTRAINT `fk_utilizador_endereço`
  FOREIGN KEY (`id_endereço`)
  REFERENCES `mydb`.`Endereço` (`id_endereço`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `mydb`.`Veiculo`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`Veiculo` (
  `id_Veiculo` INT NOT NULL,
  `tipo` VARCHAR(45) NOT NULL,
  `matricula` VARCHAR(45) NULL,
  PRIMARY KEY (`id_Veiculo`))
ENGINE = InnoDB;

-- -----
-- Table `mydb`.`Estafeta`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`Estafeta` (
  `id_Estafeta` INT NOT NULL,
  `nome` VARCHAR(45) NOT NULL,
  `contacto` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id_Estafeta`))
ENGINE = InnoDB;

-- -----
-- Table `mydb`.`Entrega`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`Entrega` (
  `id_Entrega` INT NOT NULL,
  `data_saida` DATETIME NOT NULL,
  `Veiculo_id_Veiculo` INT NOT NULL,
  `Estafeta_id_Estafeta` INT NOT NULL,
  PRIMARY KEY (`id_Entrega`),
  INDEX `fk_Entrega_Veiculo1_idx` (`Veiculo_id_Veiculo` ASC) VISIBLE,
  INDEX `fk_Entrega_Estafeta1_idx` (`Estafeta_id_Estafeta` ASC)
  VISIBLE,
  CONSTRAINT `fk_Entrega_Veiculo1`
    FOREIGN KEY (`Veiculo_id_Veiculo`)
    REFERENCES `mydb`.`Veiculo` (`id_Veiculo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Entrega_Estafeta1`
    FOREIGN KEY (`Estafeta_id_Estafeta`)

```

```

REFERENCES `mydb`.`Estafeta` (`id_Estafeta`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `mydb`.`Encomenda`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`Encomenda` (
  `id_encomenda` INT NOT NULL,
  `data_compra` DATE NOT NULL,
  `estado` VARCHAR(45) NOT NULL,
  `preco_total` DECIMAL(4,2) NOT NULL,
  `avaliação` TEXT(90) NULL,
  `data_entrega` DATETIME NOT NULL,
  `id_utilizador` INT NOT NULL,
  `id_Entrega` INT NOT NULL,
  PRIMARY KEY (`id_encomenda`),
  INDEX `fk_encomenda_utilizador1_idx` (`id_utilizador` ASC) VISIBLE,
  INDEX `fk_Encomenda_Entregal_idx` (`id_Entrega` ASC) VISIBLE,
  CONSTRAINT `fk_encomenda_utilizador1`
    FOREIGN KEY (`id_utilizador`)
      REFERENCES `mydb`.`Utilizador` (`id_utilizador`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `fk_Encomenda_Entregal`
    FOREIGN KEY (`id_Entrega`)
      REFERENCES `mydb`.`Entrega` (`id_Entrega`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `mydb`.`Medicamento`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`Medicamento` (
  `id_Medicamento` INT NOT NULL,
  `nome` VARCHAR(45) NOT NULL,
  `quantidade_stock` INT NOT NULL,
  `preco` DECIMAL(4,2) NOT NULL,
  `dosagem` INT NOT NULL,
  `venda_livre` TINYINT NOT NULL,
  PRIMARY KEY (`id_Medicamento`))
ENGINE = InnoDB;

-- -----
-- Table `mydb`.`Itens_Encomenda`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`Itens_Encomenda` (
  `itens_Encomenda` INT NOT NULL,

```

```

`quantidade` INT NOT NULL,
`preço` DECIMAL(4,2) NOT NULL,
`id_Medicamento` INT NOT NULL,
`id_encomenda` INT NOT NULL,
PRIMARY KEY (`itens_Encomenda`),
INDEX `fk_Itens_Encomenda_Medicamento1_idx` (`id_Medicamento` ASC)
VISIBLE,
INDEX `fk_Itens_Encomenda_Encomenda1_idx` (`id_encomenda` ASC)
VISIBLE,
CONSTRAINT `fk_Itens_Encomenda_Medicamento1`
FOREIGN KEY (`id_Medicamento`)
REFERENCES `mydb`.`Medicamento` (`id_Medicamento`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Itens_Encomenda_Encomenda1`
FOREIGN KEY (`id_encomenda`)
REFERENCES `mydb`.`Encomenda` (`id_encomenda`)
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;

```

```
SELECT * FROM Veiculo;
```

```

INSERT INTO Veiculo
(id_Veiculo, tipo, matricula)
Values ('1','Carro', 'AA-00-00'),
('2','Carro', 'ZZ-00-10'),
('3','Mota', 'QX-01-10'),
('4','Bicicleta', 'HJ-20-30'),
('5','Carro', '00-AA-30');

```

```
SELECT * FROM Medicamento;
```

```

INSERT INTO Medicamento
(id_Medicamento, nome, quantidade_stock, preco, dosagem,
venda_livre)
Values ('1','Azitromicina', '10', '07.65', '150', 1),
('2','Ácido acetilsalicílico', '20', '10', '500', '0'),
('3','Retinol', '30', '11.10', '120', '1'),
('4','Irbesartan', '40', '06.50', '75', '0'),
('5','Brufen', '50', '09.90', '300', '1');

```

```
SELECT * FROM Endereço;
```

```

INSERT INTO Endereço
(id_endereço, rua, cod_Postal, porta, andar, localidade)
Values ('1', 'R. Dr. José de Sousa Machado', '4710-383', '14', '3Dto',
'Braga'),

```

```
( '2', 'R. do Taxa', '4710-383', '11', '2Dto', 'Braga'),
( '3', 'R. Santa Margarida', '4710-100', '1', '3Esq', 'Braga'),
( '4', 'R. Santa Catarina', '4710-404', '5', '1Ctr', 'Braga'),
( '5', 'R. Marchal Gomes da Costa', '4710-383', '10', '3Esq', 'Braga');
```

```
SELECT * FROM Estafeta;
```

```
INSERT INTO Estafeta
(id_Estafeta, nome ,contacto)
Values ( '1', ' Nuno Machado', '934056871'),
( '2', 'Maria Antunes', '964021871'),
( '3', 'João Guimarães', '913487090'),
( '4', ' Debora Martins', '936947320'),
( '5', 'António Mendes ', '914096371'),
( '6', 'Mariana Sousa', '929391873'),
( '7', 'Amélia Cruz ', '914021654'),
( '8', 'Vitor Joaquim', '964021871'),
( '9', 'Márcia Espirito Santo', '916783214'),
( '10', ' Nelson Matos', '937689423');
```

```
SELECT * FROM Entrega;
```

```
INSERT INTO Entrega
(id_Entrega, data_saida, Veiculo_id_Veiculo, Estafeta_id_Estafeta)
Values( '1', '2022-12-03 18:21:34', '5', '1'),
( '2', '2022-01-04 17:21:56', '1', '1'),
( '3', '2022-01-07 15:50:34', '2', '3'),
( '4', '2022-01-31 12:10:01', '4', '5'),
( '5', '2021-02-24 09:20:05', '1', '2'),
( '6', '2022-01-31 12:10:01', '1', '10'),
( '7', '2022-01-30 09:10:01', '5', '9'),
( '8', '2022-01-30 14:10:20', '3', '8'),
( '9', '2021-11-14 12:00:01', '3', '7'),
( '10', '2022-07-11 08:30:20', '4', '4'),
( '11', '2021-05-08 16:29:47', '2', '6'),
( '12', '2022-03-09 15:35:52', '3', '10'),
( '13', '2021-10-21 13:21:21', '4', '4');
```

```
SELECT * FROM Utilizador;
```

```
INSERT INTO Utilizador
(id_utilizador, nome ,Nif, id_endereço)
Values ( '1', ' João António', '250223110', '1'),
( '2', 'José Maria ', '251330412', '2'),
( '3', 'Ana Rosa', '32242002', '3'),
( '4', 'Pedro Miguel', '250224712', '4'),
( '5', 'Maria Martins', '483594097', '5'),
( '6', 'Eduardo Moniz', '325387396', '5'),
( '7', 'Rosa Silva', '217163918', '4'),
( '8', 'Lucélia Martins', '325387396', '3'),
( '9', 'Catarina Quintas', '665125340', '2'),
( '10', 'João Carlos', '576340098', '1'),
( '11', 'Ricardo Pereira', '690680222', '1'),
```

```

('12', 'Carlos Costa', '116543870', '2'),
('13', 'Fátima Antunes', '932564861', '3'),
('14', 'José Filipe', '742678687', '4'),
('15', 'Manuel Brandão', '287594323', '5');

SELECT * FROM Encomenda;

INSERT INTO Encomenda
(id_encomenda, data_compra ,estado, preco_total, avaliação,
data_entrega, id_utilizador, id_Entrega)
Values ('1','2022-01-02', 'Entregue', '10.32', 'Bom serviço',
'2022-01-10 12:59:34','1', '3'),
('2','2022-01-20', 'Entregue', '21.87', ' ', '2022-01-31
18:21:34','2', '6'),
('3','2021-02-24', 'Entregue', '36.92', 'Serviço Rápido', '2021-
02-25 12:01:34','15', '5'),
('4','2022-01-02', 'Entregue', '11.74', ' ', '2022-01-06
09:31:03','13', '2'),
('5','2022-01-09', 'Entregue', '3.21', 'Demora na entrega', '2022-
01-31 18:21:34','12', '7'),
('6','2022-12-01', 'Entregue', '17.30', 'Ótimo serviço ', '2022-
12-05 14:45:34','12', '1'),
('7','2022-01-31', 'Entregue', '5.09', 'Qualidade e rapidez',
'2022-02-01 11:25:58','9', '4'),
('8','2022-01-29', 'Entregue', '67.32', ' ', '2022-01-31
14:45:34','2', '8'),
('9','2021-11-13', 'Entregue', '5.09', 'Qualidade e rapidez',
'2021-11-16 19:40:00','12', '9'),
('10','2022-07-09', 'Entregue', '27.75', ' ', '2022-07-12
17:21:56','9', '10'),
('11','2021-05-04', 'Entregue', '28.09', 'Bastante satisfeito',
'2021-05-13 18:21:34','8', '11'),
('12','2022-01-02', 'Entregue', '13.32', 'Bom serviço', '2022-03-
10 12:59:34','1', '12'),
('13','2021-10-07', 'Entregue', '21.87', ' ', '2021-10-28
18:21:34','2', '13');

SELECT * FROM Itens_Encomenda;

INSERT INTO Itens_Encomenda
(itens_Encomenda, quantidade, preço, id_Medicamento,
id_encomenda)
Values ('1', '1', '07.65','5', '10'),
('2', '1', '30','4', '5'),
('3', '5', '55.50','3', '9'),
('4', '7', '45.50','2', '4'),
('5', '2', '19.80','1', '1');

#-----QUERIES-----
#tabela dos medicamentos transportados por um veiculo
SELECT Medicamento.nome
FROM Medicamento
INNER JOIN Itens_Encomenda

```



```

ON Medicamento.id_Medicamento = Itens_Encomenda.id_Medicamento
INNER JOIN Encomenda
ON Itens_Encomenda.id_encomenda = Encomenda.id_encomenda
INNER JOIN Entrega
ON Encomenda.id_Entrega = Entrega.id_Entrega
INNER JOIN Veiculo
ON Entrega.Veiculo_id_Veiculo = Veiculo.id_Veiculo
WHERE Veiculo.matricula = 'AA-00-00';

#compras de um determinado dia
SELECT Encomenda.id_encomenda FROM Encomenda
WHERE Encomenda.data_compra = '2022-01-02';

#tabela de todas as encomendas
SELECT * FROM Encomenda;

#tabela dos clientes com o codigo postal "x" q fizeram encomenda num
certo dia -----(1,5,2)
SELECT Utilizador.id_utilizador AS Id,
Utilizador.nome AS Nome,
Utilizador.Nif AS Nif,
Encomenda.id_encomenda AS Id_encomenda
FROM Utilizador
    INNER JOIN Endereço
        ON Utilizador.id_endereço = Endereço.id_endereço
    INNER JOIN Encomenda
        ON Utilizador.id_utilizador = Encomenda.id_utilizador
WHERE Encomenda.data_compra = '2022-01-02' AND
Endereço.cod_Postal = '4710-383';

#tabela de ranking de entregas dos veículos (o tipo de veiculo com
mais entregas )
SELECT count(*) as Quantidade, Veiculo.tipo FROM Veiculo
    INNER JOIN Entrega
        ON Veiculo.id_Veiculo = Entrega.Veiculo_id_Veiculo
GROUP BY Veiculo.tipo;

#VISTAS
#vista do ranking do lucro dos estafetas
CREATE VIEW Ranking_Estafetas AS
SELECT Estafeta.nome, SUM(Encomenda.preco_total) AS Lucro
FROM Encomenda
    INNER JOIN Entrega
        ON Encomenda.id_Entrega = Entrega.id_Entrega
    INNER JOIN Estafeta
        ON Entrega.Estafeta_id_Estafeta = Estafeta.id_Estafeta
GROUP BY Estafeta.nome
ORDER BY Lucro DESC;

#vista da demora das entregas
CREATE VIEW demora_entregas AS
SELECT Entrega.id_Entrega,
Entrega.data_saida, Encomenda.data_entrega,

```

```

datediff(Encomenda.data_entrega, Entrega.data_saida)as TIME
FROM Entrega
INNER JOIN Encomenda
ON Entrega.id_entrega = Encomenda.id_entrega
ORDER BY TIME DESC;

#vista de todos os medicamentos sem venda_livre
CREATE VIEW sem_venda_livre AS
SELECT * from Medicamento WHERE Medicamento.venda_livre = '0';

#INDEX
CREATE INDEX NomeMedicamento ON Medicamento(nome);
#SHOW INDEX FROM Medicamento ;

CREATE INDEX Data_Entrega on Encomenda(data_entrega);

#PROCEDIMENTOS
#Procedimento do top 3 do valor das encomendas realizadas e respectivos
clientes.
DELIMITER $$
CREATE PROCEDURE Top3Pagamentos()
BEGIN
    SELECT Utilizador.nome as Nome, Encomenda.preco_total as Preço
    FROM Utilizador INNER JOIN Encomenda
    ON Utilizador.id_utilizador = Encomenda.id_utilizador
    ORDER BY Encomenda.preco_total DESC
    LIMIT 3;
END $$
CALL Top3Pagamentos;

#Consultar o número total de entregas num determinado ano (2022)
DELIMITER $$
CREATE PROCEDURE n_total_ano (IN ano INT)
BEGIN
SELECT    count(id_Entrega)    as    n_Entregas    FROM    Entrega    WHERE
YEAR(data_saida)=ano;
END $$
CALL n_total_ano(2022);

#Atualização do preço de um medicamento devido a inflação
DELIMITER $$
CREATE PROCEDURE Update_Preço_Med (IN med INT, IN new_preco
DECIMAL(4,2))
BEGIN
UPDATE Medicamento SET preco = new_preco where id_Medicamento=med;
END $$
call Update_Preço_Med (1,20.04);

SELECT * FROM Medicamento;

```