



Universidade do Minho
Licenciatura em Ciências da Computação

Unidade Curricular de Bases de Dados

Ano Letivo de 2022/2023

Gestão de Entregas a Domicílio

Diogo Aires, Marcelo Sousa, Márlon Ferreira

Outubro, 2022

BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

Beerlivery

Diogo Aires, Marcelo Sousa, Márlon Ferreira

Outubro, 2022

Resumo

O presente trabalho tem como objetivo o planeamento e construção de um Sistema de Base de Dados Relacional para um serviço de Gestão de Entregas ao Domicílio, em particular escolhemos contruir um Sistema de Base de Dados para o ramo de entregas ao domicílio de uma empresa distribuição de cerveja artesanal.

Ao longo deste relatório é descrita a definição, contextualização, motivação e objetivos para a implementação de tal base de dados. É analisada a viabilidade do projeto bem como criado um plano de execução do projeto de implementação do Sistema de Base de Dados. Estão descritos o processo utilizado para o levantamento de requisitos e a caracterização dos diversos perfis de utilização envolvidos no problema, assim como a modelação conceptual e lógica. Posteriormente, apresentamos um plano de implementação física suportado por alguns excertos de código SQL, seguido de uma conclusão final.

Área de Aplicação: Desenho e arquitetura de Sistemas de Bases de Dados.

Palavras-Chave: SQL, MySQL, NoSQL, Base de Dados Relacional, Base de Dados Não Relacional, Modelo Conceptual, Modelo lógico, Modelo Físico.

Índice

RESUMO	I
ÍNDICE.....	II
ÍNDICE DE FIGURAS.....	IV
ÍNDICE DE TABELAS.....	V
1. DEFINIÇÃO DO SISTEMA.....	1
1.1. CONTEXTO DE APLICAÇÃO E FUNDAMENTAÇÃO DO SISTEMA	1
1.2. MOTIVAÇÃO E OBJETIVOS DO TRABALHO	2
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 E DE ANÁLISE DE REQUISITOS ADOTADO	4
2.2. ORGANIZAÇÃO DOS REQUISITOS LEVANTADOS	4
2.2.1. <i>Requisitos de descrição</i>	4
2.2.2. <i>Requisitos de exploração</i>	5
2.2.3. <i>Requisitos de controlo</i>	5
2.3. ANÁLISE E VALIDAÇÃO GERAL DOS REQUISITOS	6
3. MODELAÇÃO CONCEPTUAL.....	7
3.1. APRESENTAÇÃO DA ABORDAGEM DE MODELAÇÃO REALIZADA	7
3.2. IDENTIFICAÇÃO E CARACTERIZAÇÃO DAS ENTIDADES	7
3.3. IDENTIFICAÇÃO E CARACTERIZAÇÃO DOS RELACIONAMENTOS	8
3.4. IDENTIFICAÇÃO E CARACTERIZAÇÃO DA ASSOCIAÇÃO DOS ATRIBUTOS COM AS ENTIDADES E RELACIONAMENTOS.....	8
3.5. APRESENTAÇÃO E EXPLICAÇÃO DO DIAGRAMA ER PRODUZIDO	9
4. MODELAÇÃO LÓGICA	10
4.1 CONSTRUÇÃO E VALIDAÇÃO DO MODELO DE DADOS LÓGICO	10
4.2 DESENHO DO MODELO LÓGICO	12
4.3 VALIDAÇÃO DO MODELO COM INTERROGAÇÕES DO UTILIZADOR	13
5. IMPLEMENTAÇÃO FÍSICA	14
5.1 TRADUÇÃO DO ESQUEMA LÓGICO PARA O SISTEMA DE GESTÃO DE BASES DE DADOS ESCOLHIDO EM SQL.....	14
5.2 TRADUÇÃO DAS INTERROGAÇÕES DO UTILIZADOR PARA SQL (ALGUNS EXEMPLOS)	18
5.3 DEFINIÇÃO E CARACTERIZAÇÃO DAS VISTAS DE UTILIZAÇÃO EM SQL (ALGUNS EXEMPLOS).....	19

5.4 CÁLCULO DO ESPAÇO DA BASE DE DADOS (INICIAL E TAXA DE CRESCIMENTO ANUAL)	20
5.7 PLANO DE SEGURANÇA E RECUPERAÇÃO DE DADOS	23
6. CONCLUSÕES E TRABALHO FUTURO.....	24
7. REFERÊNCIAS BIBLIOGRÁFICAS	25

Índice de Figuras

Figura 1: Plano de Execução	3
Figura 2 Modelo Conceptual	9
Figura 3 Modelo Lógico	12

Índice de Tabelas

Tabela 1 Entidades	8
Tabela 2: Entidade Cliente	10
Tabela 3: Entidade Encomenda	10
Tabela 4: Entidade Armazém	10
Tabela 5: Entidade Veiculo	11
Tabela 6: Entidade Funcionaria Armazém	11
Tabela 7: Entidade Estafeta	11
Tabela 8 Cliente	20
Tabela 9 Encomenda	20
Tabela 10 Item.....	21
Tabela 11 Armazém	21
Tabela 12 Morada	21
Tabela 13 Tipo de Cliente	21
Tabela 14 Funcionário armazém.....	22
Tabela 15 Estafeta	22

1. Definição do Sistema

1.1. Contexto de aplicação e fundamentação do sistema

A cerveja artesanal tem vindo a crescer em popularidade na região do Minho ao longo dos anos e já é possível encontrar em muitos restaurantes e bares uma grande variedade de oferta. Nos últimos anos foram criadas marcas regionais como A LETRA em 2011, empresa criada por alunos da Universidade do Minho e a PCB – Portuguese Craft Beer, fundada em 2013 pelo humorista João Seabra que nos trouxe a Amphora e mais tarde a Alma Brasileira.

A Beerlivery, também minhota, é uma empresa de distribuição de cerveja artesanal e entra no ramo em meados de abril de 2019 fundada pelos sócios Martim Rodrigues e Eduardo Magalhães. Com a extensão do negócio, estes adquiriram 1 armazém, localizado em Gualtar.

A empresa dispõe de 8 funcionários, sendo 1 administrador, 3 funcionários fixos para o armazém e 4 estafetas.

Por este tipo de produto ser servido maioritariamente em bares e restaurantes, a chegada da obrigação de confinamento resultante da última pandemia dificultou a venda destes produtos.

De forma a aliviar esta perda de vendas a empresa Beerlivery, que já fazia a distribuição de cerveja às superfícies comerciais, cria um ramo de negócio destinado também à entrega de cerveja ao domicílio.

Numa fase inicial, tem intenção de implementar a entrega ao domicílio no centro de Braga, mas num futuro, caso venha a ser rentável, tem a intenção de expandir para novas cidades, com a criação de novos armazéns por todo o país.

Com a expansão do consumo de sidra artesanal em todo o país, a empresa equaciona também esse mercado num futuro próximo.

Com a introdução da entrega ao domicílio a Beerlivery procura reestruturar o funcionamento interno e pretende criar uma base de dados para que seja um processo mais simples, rápido e seguro.

A empresa recolhe as encomendas nos armazéns das diferentes empresas, armazena no seu próprio armazém e distribui para os seus clientes.

Existem estafetas destinados à entrega de grandes encomendas às superfícies comerciais pois possuem veículos adequados como carrinhas. No futuro prevê-se a necessidade de estafetas que farão entregas particulares ao domicílio utilizando carros e motocicletas.

1.2. Motivação e Objetivos do Trabalho

A Beerlivery utilizava um sistema de arquivos para o armazenamento de dados que lia e guardava dados consoante necessário. Este sistema funciona com utilizações em pequena escala, mas com a introdução das entregas ao domicílio a complexidade é demasiado grande para ser viável. O uso de um sistema de bases de dados tornou-se na melhor opção para um crescimento estruturado da empresa e para uma melhor gestão de recursos humanos e mercadoria.

1.3. Análise da viabilidade do processo

Com a implementação deste sistema na empresa espera-se que esta consiga atingir os objetivos pretendidos no serviço de entrega e assim conseguir oferecer o serviço de qualidade ao cliente.

Depois de uma reunião para analisar a proposta, concluímos que num momento inicial será feita apenas uma base de dados relativa a entregas e a gestão de custos de forma à empresa garantir financiamento para esta fase. Num futuro próximo será contruída a componente de gestão de stock do armazém.

Espera-se no primeiro ano recuperar 50% do investimento inicial.

1.4. Recursos e Equipa de Trabalho

Para implementar esta base de dados será necessário um computador para o armazém e outro para o escritório. Estes computadores irão servir de ponto de acesso a um servidor que armazena a base de dados com o objetivo de ler e modificar dados.

A equipa de trabalho é constituída por 3 técnicos que irão desenvolver a base de dados no decorrer dos próximos 3 meses.

1.5. Plano de Execução do Projeto

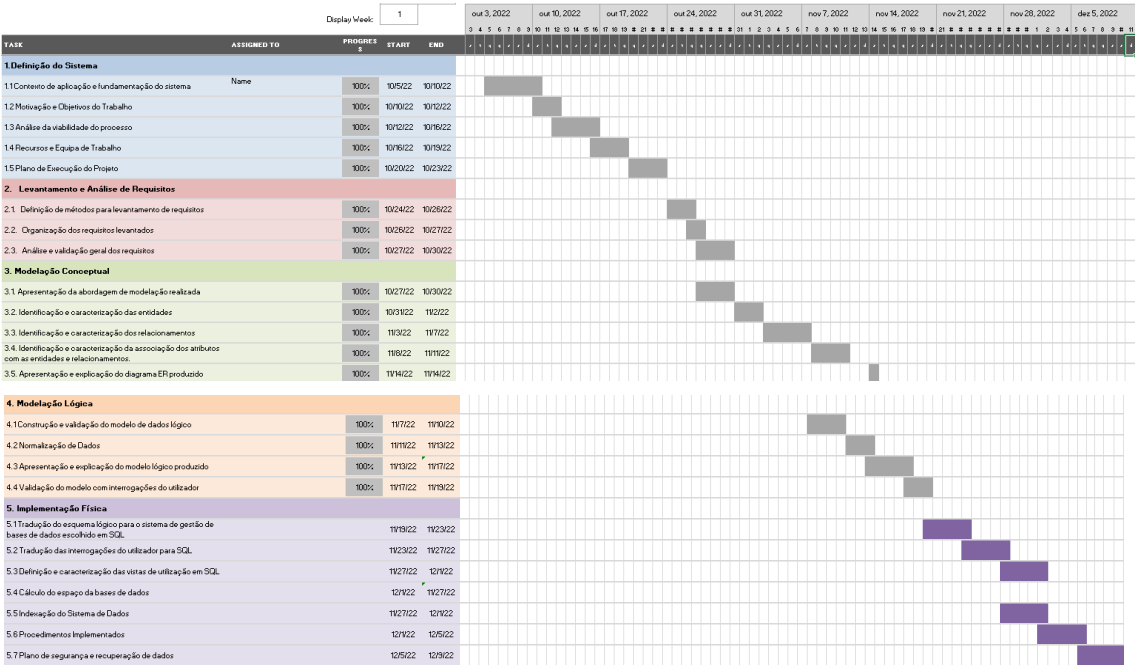


Figura 1: Plano de Execução

2. Levantamento e Análise de Requisitos

2.1. Método de levantamento e de análise de requisitos adotado

A principal forma de levantamento de requisitos foi através de três reuniões com todos os funcionários da empresa. Uma vez que a empresa não é de grandes dimensões foi possível juntar todos os funcionários no final de um dia de trabalho.

Com a análise das faturas já existentes foi possível concluir que dados são essenciais para a realização de uma encomenda.

A observação do método de trabalho no armazém foi essencial para perceber como implementar a base de dados sem correr o risco de prejudicar o ritmo dos trabalhadores, e até perceber como os tornar mais eficientes.

2.2. Organização dos requisitos levantados

De acordo com os métodos descritos acima levantaram-se os seguintes requisitos que são necessários para a criação desta base de dados.

2.2.1. Requisitos de descrição

1. No início apenas vamos contar com um armazém, mas com a possibilidade de expandir no futuro;
2. O registo do cliente tem de ter nome, data de nascimento, morada, email e número de telemóvel;
3. A entrega de cerveja ao domicílio será realizada através de estafetas;
4. Cada encomenda deve ter morada, identificador do cliente, número de identificação da encomenda, estafeta associado, método de pagamento;

5. O cliente terá a possibilidade de agendar uma entrega de cerveja, num dia e hora à sua escolha num período máximo de uma semana;
6. A encomenda será preparada por um funcionário do armazém;
7. Cada funcionário de armazém terá de ter no seu registo um número de identificação interno, NIF, data de nascimento e Nome;
8. A entrega terá um custo associado, que depende da distância com uma taxa de serviço base;
9. No início contaremos com quatro estafetas, mas caso seja necessário contratarão mais;
10. Possibilidade de expandir o negócio para outro tipo de bebidas, nomeadamente sidra. O item a ser entregue terá de ter um nome mais genérico que não cerveja;
11. A cada encomenda estará associado um funcionário do armazém, um estafeta, o respetivo cliente, e os respetivos itens;
12. No relatório mensal deve estar contido o tempo médio que demoraram as entregas;
13. Cada item terá uma descrição dos seus sabores, e %Álcool;
14. O estafeta, tem de ter no seu registo, nome, data de nascimento, NIF, N° da carta de condução;
15. O Cliente terá de ter um identificador a dizer se é empresa ou particular;
16. Para melhor produtividade no armazém, será necessário colocar a localização do item no armazém: (e.g., corredor, prateleira e nível).
17. Devido ao produto que é vendido (Álcool) o cliente tem de ter obrigatoriamente no seu registo a sua data de nascimento.

2.2.2. Requisitos de exploração

1. O cliente deve receber por email ou por mensagem, uma notificação de quando o estafeta sair do armazém;
2. Caso o pedido seja cancelado terá de ser aceite pelos funcionários do armazém;
3. Todos os meses o sistema deverá apresentar um relatório com o número de entregas e respetivo lucros;
4. O registo do cliente é obrigatório;
5. Deve ser recalculado o custo da encomenda sempre que é inserido um novo item utilizando uma fórmula que tem como parâmetros o preço dos itens e a distância de entrega.

2.2.3. Requisitos de controlo

1. Cada estafeta terá acesso apenas aos pedidos que vai entregar;
2. Cada funcionário de armazém terá acesso às encomendas e ao stock do armazém;

3. O administrador terá acesso as todas as entregas e terá possibilidade de cancelar os pedidos.

2.3. Análise e validação geral dos requisitos

O processo de análise provém do processo de levantamento de requisitos. Com os requisitos de descrição, focou-se em quais elementos seriam indispensáveis para que as vendas da Beerlivery e o armazenamento de informação sobre estas ocorram da mesma forma e como estes se podem relacionar entre si. Seguidamente, deliberou-se sobre quais informações seriam mais revelantes extrair, tendo em conta os elementos e relacionamentos, anteriormente mencionados.

Após a fase dos requisitos de exploração, entramos nos requisitos de controlo, onde estabelecemos as restrições de administração da base de dados.

No processo de validação verificamos a consistência, precisão e contextualização dos nossos requisitos expostos no processo de análise por forma a evitar falha na modelação dos requisitos.

3. Modelação Conceptual

3.1. Apresentação da abordagem de modelação realizada

Para a criação do modelo conceptual utilizamos o software brModelo. Começamos pela identificação das entidades, seguidas das relações com as respectivas restrições e caracterizações dos atributos.

3.2. Identificação e caracterização das entidades

Entidade	Descrição
Cliente	Termo genérico que representa a pessoa que efetua uma compra. O requisito que justifica a sua existência é 2. (" 2.2.1. Requisitos de descrição")
Encomenda	Termo genérico que representa a compra realizada pelo cliente. O requisito que justifica a sua existência é 4. (" 2.2.1. Requisitos de descrição")
Item	Termo genérico que representa cada item da encomenda. O requisito que justifica a sua existência é 9. (" 2.2.1. Requisitos de descrição")
Estafeta	Termo genérico que representa pessoa que entrega a encomenda. O requisito que justifica a sua existência é 14. (" 2.2.1. Requisitos de descrição")
Veículo	Termo genérico que representa o veículo usado pelo estafeta. O requisito que justifica a sua existência é 8. (" 2.2.1. Requisitos de descrição")
Funcionário Armazém	Termo genérico que representa pessoa que prepara a encomenda. O requisito que justifica a sua existência é 6. (" 2.2.1. Requisitos de descrição")

Armazém	Termo genérico que representa um armazém, o requisito que justifica a sua existência é 1. (“ 2.2.1. Requisitos de descrição”)
----------------	---

Tabela 1 Entidades

3.3. Identificação e caracterização dos relacionamentos

Os relacionamentos entre a encomenda <-> cliente, encomenda <-> item, encomenda <-> estafeta, encomenda <-> funcionário de armazém, são justificados pelo requisito 12. (“ 2.2.1. Requisitos de descrição”).

O relacionamento estafeta <-> veículo é justificado pelo requisito 10. (“ 2.2.1. Requisitos de descrição”).

O relacionamento encomenda <-> armazém é usado para saber qual armazém vai sair a encomenda.

O relacionamento armazém <-> funcionário de armazém é usado para saber em que armazém está a trabalhar.

O relacionamento armazém <-> estafeta é usado para saber em que armazém está a trabalhar.

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

Os principais atributos associados as entidades cliente, funcionário de armazém, e estafeta são nome, data de nascimento e NIF e um ID que esta associado a cada entidade, de forma a ser possível distinguir dentro da empresa. Devido a possibilidade de no futuro poder haver mais que um armazém os funcionários de armazém e os estafetas estão associados a um armazém. No caso do cliente tem mais alguns atributos, como morada (atributo multivalorado, código postal, rua e cidade), email, tipo de cliente e número de cliente estes são justificados pelo requisito 2. (“ 2.2.1. Requisitos de descrição”).

Na encomenda tem como atributos distancia, para poder se calcular o custo da entrega, a data do pedido e a data da entrega, para poder-se no final de cada mês garantir a qualidade de serviço, o custo total da encomenda e ainda o método de pagamento e o estado da encomenda. Para ser possível distinguir as encomendas, há um atributo ID.

Na entidade item, são guardados todos os itens disponíveis, como a empresa conta num futuro vender produtos alem da cerveja artesanal, foi adicionado os atributos, o nome, descrição e percentagem de álcool para se poder identificar e caracterizar o produto, e ainda o id produto para ser identificado facilmente pelos funcionários.

No caso do armazém tem apenas os atributos nome, morada e id.

3.5. Apresentação e explicação do diagrama ER produzido

Tendo em conta que o sistema de base dados a implementar gerir as encomendas de um sistema de entregas ao domicilio de cerveja, entre outros, para o seu bom funcionamento é necessário que esta tem capacidade de gerir as encomendas, de forma a ser possível o funcionários do armazém prepararem as encomendas, os estafetas terem a possibilidade de uma forma eficaz terem os todos dados necessários, desde do nome do cliente numero de telemóvel caso seja necessário contactar, morada para fazer a entrega, para fazer a entrega sem complicações.

Assim podemos concluir que uma encomenda obrigatoriamente tem como condições fundamentais, funcionário do armazém, estafeta, os itens atribuídos a encomenda e morada do cliente.

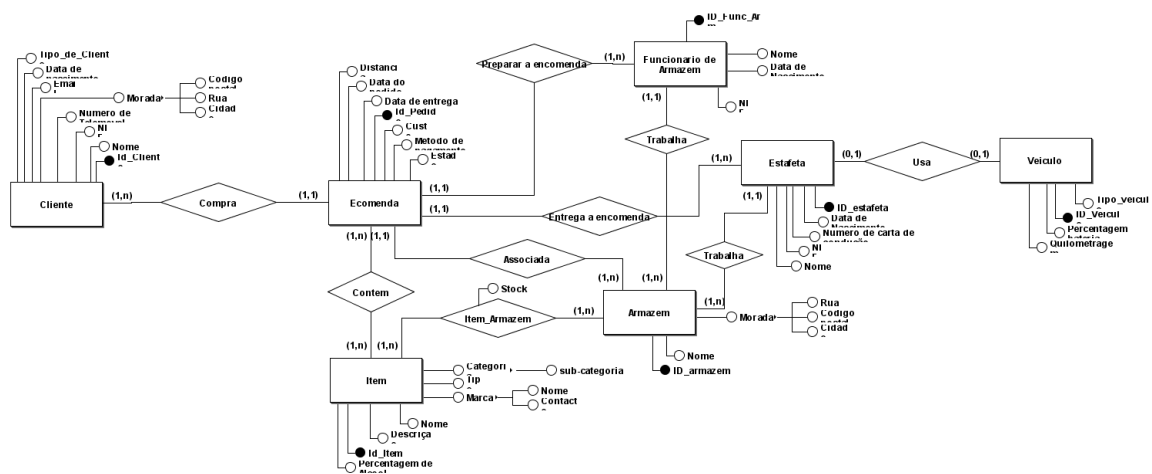


Figura 2 Modelo Conceptual

4. Modelação Lógica

4.1 Construção e validação do modelo de dados lógico

Cliente(id_cliente,nome,nif,email,telemovel,data_nascimento,id_morada,id_tipo_cliente,isDeleted,created_on,updated_on)
Chave Primaria: id_cliente
Chave Estrangeira: id_tipo_cliente, id_morada

Tabela 2: Entidade Cliente

Encomenda(id_encomenda,distancia,data_pedido,data_entrega,custo,id_metodo_pagamento,estado,id_cliente,id_funcionario_armazem,id_estafeta,id_armazem,isDeleted,created_on,updated_on)
Chave Primaria: id_encomenda
ChaveEstrangeira: id_metodo_pagamento,id_cliente,id_funcionario_armazem,id_estafeta,id_armazem,id_estado

Tabela 3: Entidade Encomenda

Armazem(id_armazem,nome,id_morada,isDeleted)
Chave Primaria:id_armazem
Chave Estrangeira: id_morada

Tabela 4: Entidade Armazém

Veiculo(id_veiculo,id_tipo_veiculo,nome,quilometragem,combustivel,isDeleted,created_on,updated_on)
Chave Primaria: id_veiculo
Chave Estrangeira: id_tipo_veiculo

Tabela 5: Entidade Veiculo

Funcionario_Armazem(id_funcionario_armazem,nome,data_nascimento,nif,id_armazem,isDeleted,created_on,updated_on)
Chave Primaria: id_funcionario_armazem
Chave Estrangeira: id_armazem

Tabela 6: Entidade Funcionaria Armazém

Estafeta(id_estafeta,nome,nr_conducao,nif,data_nascimento,id_armazem,id_veiculo,isDeleted,created_on,updated_on)
Chave Primaria: id_estafeta
Chave Estrangeira: id_armazem, id_veiculo

Tabela 7: Entidade Estafeta

4.2 Desenho do modelo lógico

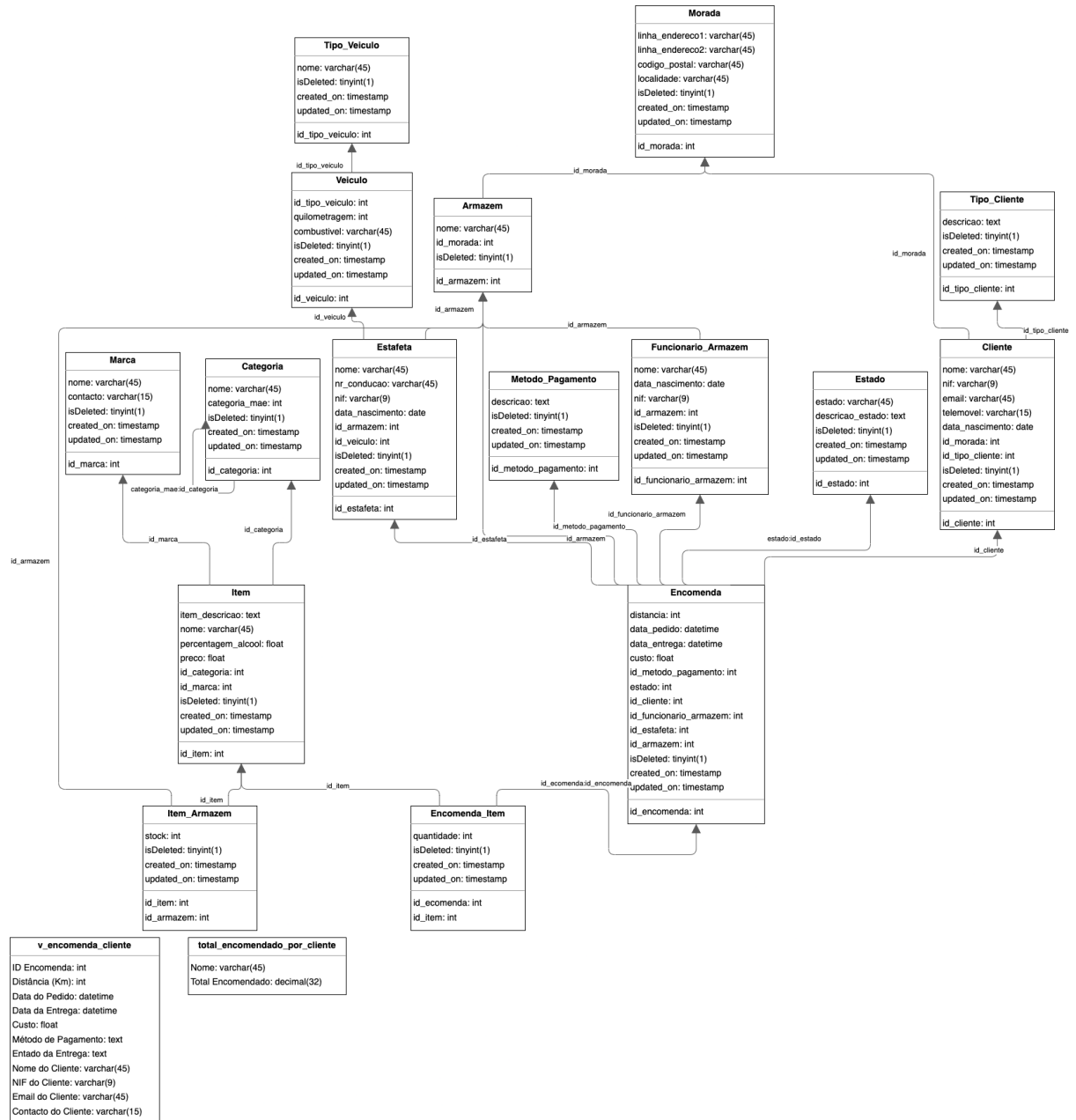


Figura 3 Modelo Lógico

4.3 Validação do modelo com interrogações do utilizador

De forma a validar o modelo logico testamos se seria possível satisfazer alguns requisitos de exploração, através de álgebra relacional usando a ferramenta Relax pois esta permite uma melhor visualização da forma como as Queries executam, através das arvores relacionais.

Em baixo apresentamos algumas dessas queries em álgebra relacional:

- $\pi \text{ nome, email, telemóvel } (\text{Cliente})$
- $\pi \text{ I.nome, M.nome } (I \bowtie \text{id_marca} = \text{M.id_marca } (Item \bowtie \text{Marca}))$
- $\pi \text{ id_encomenda, data_pedido } (\sigma \text{ data_pedido } \geq \text{DATE_SUB}(\text{NOW}(), \text{INTERVAL } 6 \text{ MONTH}) (\text{Encomenda}))$
- $\pi \text{ I.nome, C.nome } (I \bowtie \text{id_categoria} = \text{C.id_categoria } (Item \bowtie \text{Categoria}))$
- $\pi \text{ E.id_encomenda, C.nome, E.data_pedido } (E \bowtie \text{id_cliente} = \text{C.id_cliente } (\text{Encomenda} \bowtie \text{Cliente}))$
- $\pi \text{ E.id_encomenda, I.nome, C.nome } (\sigma \text{ C.nome} = \text{'Stout'} (E \bowtie \text{id_encomenda} = \text{EI.id_ecomenda } (\text{Encomenda}) \bowtie \text{id_item} = \text{I.id_item } (EI \bowtie \text{Item}) \bowtie \text{id_categoria} = \text{C.id_categoria } (I \bowtie \text{Categoria})))$

5. Implementação Física

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

Uma vez que o sistema de bases de dados abordado durante o semestre foi o MySQL decidimos que deveríamos adotá-lo para a implementação deste projeto. O MySQL é um Sistema de Gestão de Base de Dados *open-source* que assenta sobre um paradigma relacional. É um dos sistemas mais utilizados no mundo devido a facilidade de utilização, o seu elevado empenho e por ser uma ferramenta bastante intuitiva.

Alguma das tabelas criadas:

- Cliente

```
1. DROP TABLE IF EXISTS `beerlivery`.`Cliente` ;
2.
3. CREATE TABLE IF NOT EXISTS `beerlivery`.`Cliente` (
4.   `id_cliente` INT NOT NULL AUTO_INCREMENT,
5.   `nome` VARCHAR(45) NOT NULL,
6.   `nif` VARCHAR(9) NULL,
7.   `email` VARCHAR(45) NULL,
8.   `telemovel` VARCHAR(15) NOT NULL,
9.   `data_nascimento` DATE NOT NULL,
10.  `id_morada` INT NOT NULL,
11.  `id_tipo_cliente` INT NOT NULL,
12.  `isDeleted` BOOL NOT NULL DEFAULT 0,
13.  `created_on` TIMESTAMP DEFAULT now(),
14.  `updated_on` TIMESTAMP DEFAULT now() ON UPDATE now(),
15.  PRIMARY KEY (`id_cliente`),
16.  INDEX `fk_Cliente_Morada1_idx` (`id_morada` ASC) VISIBLE,
17.  INDEX `fk_Cliente_tipo_cliente1_idx` (`id_tipo_cliente` ASC) VISIBLE,
18.  CONSTRAINT `fk_Cliente_tipo_cliente1`
19.    FOREIGN KEY (`id_tipo_cliente`)
20.    REFERENCES `beerlivery`.`Tipo_Cliente` (`id_tipo_cliente`)
21.    ON DELETE NO ACTION
22.    ON UPDATE NO ACTION,
23.  CONSTRAINT `fk_Cliente_Morada1`
24.    FOREIGN KEY (`id_morada`)
25.    REFERENCES `beerlivery`.`Morada` (`id_morada`)
26.    ON DELETE NO ACTION
27.    ON UPDATE NO ACTION)
28. ENGINE = InnoDB;
```

- Armazém

```

1. DROP TABLE IF EXISTS `beerlivery`.`Armazem` ;
2.
3. CREATE TABLE IF NOT EXISTS `beerlivery`.`Armazem` (
4.   `id_armazem` INT NOT NULL AUTO_INCREMENT,
5.   `nome` VARCHAR(45) NOT NULL,
6.   `id_morada` INT NOT NULL,
7.   `isDeleted` BOOL NOT NULL DEFAULT 0,
8.   PRIMARY KEY (`id_armazem`),
9.   INDEX `fk_Armazem_Morada1_idx` (`id_morada` ASC) VISIBLE,
10.  CONSTRAINT `fk_Armazem_Morada1`
11.    FOREIGN KEY (`id_morada`)
12.    REFERENCES `beerlivery`.`Morada` (`id_morada`)
13.    ON DELETE NO ACTION
14.    ON UPDATE NO ACTION)
15. ENGINE = InnoDB;

```

- Funcionaria Armazém

```

1. DROP TABLE IF EXISTS `beerlivery`.`Funcionario_Armazem` ;
2.
3. CREATE TABLE IF NOT EXISTS `beerlivery`.`Funcionario_Armazem` (
4.   `id_funcionario_armazem` INT NOT NULL AUTO_INCREMENT,
5.   `nome` VARCHAR(45) NOT NULL,
6.   `data_nascimento` DATE NOT NULL,
7.   `nif` VARCHAR(9) NOT NULL,
8.   `id_armazem` INT NOT NULL,
9.   `isDeleted` BOOL NOT NULL DEFAULT 0,
10.  `created_on` TIMESTAMP DEFAULT now(),
11.  `updated_on` TIMESTAMP DEFAULT now() ON UPDATE now(),
12.  PRIMARY KEY (`id_funcionario_armazem`),
13.  INDEX `fk_funcionario_armazem_Armazem1_idx` (`id_armazem` ASC) VISIBLE,
14.  CONSTRAINT `fk_funcionario_armazem_Armazem1`
15.    FOREIGN KEY (`id_armazem`)
16.    REFERENCES `beerlivery`.`Armazem` (`id_armazem`)
17.    ON DELETE NO ACTION
18.    ON UPDATE NO ACTION)
19. ENGINE = InnoDB;

```

- Veículo

```

1. DROP TABLE IF EXISTS `beerlivery`.`Veiculo` ;
2.
3. CREATE TABLE IF NOT EXISTS `beerlivery`.`Veiculo` (
4.   `id_veiculo` INT NOT NULL AUTO_INCREMENT,
5.   `id_tipo_veiculo` INT NOT NULL,
6.   `quilometragem` INT NULL,
7.   `combustivel` VARCHAR(45) NULL,
8.   `isDeleted` BOOL NOT NULL DEFAULT 0,
9.   `created_on` TIMESTAMP DEFAULT now(),
10.  `updated_on` TIMESTAMP DEFAULT now() ON UPDATE now(),
11.  PRIMARY KEY (`id_veiculo`),
12.  INDEX `fk_Veiculo_id_tipo_veiculo_idx` (`id_tipo_veiculo` ASC) VISIBLE,
13.  CONSTRAINT `fk_Veiculo_id_tipo_veiculo`
14.    FOREIGN KEY (`id_tipo_veiculo`)
15.    REFERENCES `beerlivery`.`Tipo_Veiculo` (`id_tipo_veiculo`)
16.    ON DELETE NO ACTION
17.    ON UPDATE NO ACTION)
18. ENGINE = InnoDB;

```

- Encomenda

```

1. DROP TABLE IF EXISTS `beerlivery`.`Encomenda` ;
2.
3. CREATE TABLE IF NOT EXISTS `beerlivery`.`Encomenda` (
4.   `id_encomenda` INT NOT NULL AUTO_INCREMENT,
5.   `distancia` INT NOT NULL,
6.   `data_pedido` DATETIME NOT NULL,
7.   `data_entrega` DATETIME NULL,
8.   `custo` FLOAT NOT NULL,
9.   `id_metodo_pagamento` INT NOT NULL,
10.  `estado` INT NOT NULL,
11.  `id_cliente` INT NOT NULL,
12.  `id_funcionario_armazem` INT NOT NULL,
13.  `id_estafeta` INT NOT NULL,
14.  `id_armazem` INT NOT NULL,
15.  `isDeleted` BOOL NOT NULL DEFAULT 0,
16.  `created_on` TIMESTAMP DEFAULT now(),
17.  `updated_on` TIMESTAMP DEFAULT now() ON UPDATE now(),
18.  PRIMARY KEY (`id_encomenda`),
19.  INDEX `fk_Ecomenda_Cliente1_idx` (`id_cliente` ASC) VISIBLE,
20.  INDEX `fk_Ecomenda_Funcionario do Armazem1_idx` (`id_funcionario_armazem`
    ASC) VISIBLE,
21.  INDEX `fk_Ecomenda_Estafeta1_idx` (`id_estafeta` ASC) VISIBLE,
22.  INDEX `fk_Ecomenda_Armazem1_idx` (`id_armazem` ASC) VISIBLE,
23.  INDEX `fk_Ecomenda_Estado1_idx` (`estado` ASC) VISIBLE,
24.  INDEX `fk_Cliente_Metodo_Pagamento_idx` (`id_metodo_pagamento` ASC) VISIBLE,
25.  CONSTRAINT `fk_Cliente_Metodo_Pagamento_idx`
26.    FOREIGN KEY (`id_metodo_pagamento`)
27.    REFERENCES `beerlivery`.`Metodo_Pagamento` (`id_metodo_pagamento`)
28.    ON DELETE NO ACTION
29.    ON UPDATE NO ACTION,
30.  CONSTRAINT `fk_Ecomenda_Cliente1`
31.    FOREIGN KEY (`id_cliente`)
32.    REFERENCES `beerlivery`.`Cliente` (`id_cliente`)
33.    ON DELETE NO ACTION
34.    ON UPDATE NO ACTION,
35.  CONSTRAINT `fk_Ecomenda_Funcionario do Armazem1`
36.    FOREIGN KEY (`id_funcionario_armazem`)
37.    REFERENCES `beerlivery`.`Funcionario_Armazem` (`id_funcionario_armazem`)
38.    ON DELETE NO ACTION
39.    ON UPDATE NO ACTION,
40.  CONSTRAINT `fk_Ecomenda_Estafeta1`
41.    FOREIGN KEY (`id_estafeta`)
42.    REFERENCES `beerlivery`.`Estafeta` (`id_estafeta`)
43.    ON DELETE NO ACTION
44.    ON UPDATE NO ACTION,
45.  CONSTRAINT `fk_Ecomenda_Armazem1`
46.    FOREIGN KEY (`id_armazem`)
47.    REFERENCES `beerlivery`.`Armazem` (`id_armazem`)
48.    ON DELETE NO ACTION
49.    ON UPDATE NO ACTION,
50.  CONSTRAINT `fk_Ecomenda_Estado1`
51.    FOREIGN KEY (`estado`)
52.    REFERENCES `beerlivery`.`Estado` (`id_estado`)
53.    ON DELETE NO ACTION
54.    ON UPDATE NO ACTION)
55. ENGINE = InnoDB;

```

- Estafeta

```

1. DROP TABLE IF EXISTS `beerlivery`.`Estafeta` ;
2.
3. CREATE TABLE IF NOT EXISTS `beerlivery`.`Estafeta` (
4.   `id_estafeta` INT NOT NULL AUTO_INCREMENT,
5.   `nome` VARCHAR(45) NOT NULL,
6.   `nr_conducao` VARCHAR(45) NOT NULL,
7.   `nif` VARCHAR(9) NOT NULL,
8.   `data_nascimento` DATE NOT NULL,
9.   `id_armazem` INT NOT NULL,
10.  `id_veiculo` INT NOT NULL,
11.  `isDeleted` BOOL NOT NULL DEFAULT 0,
12.  `created_on` TIMESTAMP DEFAULT now(),
13.  `updated_on` TIMESTAMP DEFAULT now() ON UPDATE now(),
14.  PRIMARY KEY (`id_estafeta`),
15.  INDEX `fk_Estafeta_Veiculo1_idx` (`id_veiculo` ASC) VISIBLE,
16.  INDEX `fk_Estafeta_Armazem1_idx` (`id_armazem` ASC) VISIBLE,
17.  CONSTRAINT `fk_Estafeta_Armazem1`
18.    FOREIGN KEY (`id_armazem`)
19.    REFERENCES `beerlivery`.`Armazem` (`id_armazem`)
20.    ON DELETE NO ACTION
21.    ON UPDATE NO ACTION,
22.  CONSTRAINT `fk_Estafeta_Veiculo1`
23.    FOREIGN KEY (`id_veiculo`)
24.    REFERENCES `beerlivery`.`Veiculo` (`id_veiculo`)
25.    ON DELETE NO ACTION
26.    ON UPDATE NO ACTION)
27. ENGINE = InnoDB;

```


5.2 Tradução das interrogações do utilizador para SQL (alguns exemplos)

```
select E.id_encomenda, C.nome, E.data_pedido from Encomenda as E
inner join Cliente as C
on E.id_cliente = C.id_cliente;
```

Corresponde id da encomenda, nome do cliente e data do pedido.

```
select id_encomenda, data_pedido from Encomenda
where data_pedido >= DATE SUB(NOW(), INTERVAL 6 MONTH );
```

Devolve as encomendas com o pedido feito nos últimos 6 meses.

```
DELIMITER $$

CREATE PROCEDURE calculate_order_cost (IN id_encomenda_param INT)
BEGIN
    DECLARE custo_total FLOAT;
    DECLARE distancia_total INT;

    SET custo_total = 0;

    SELECT SUM(i.preco * ei.quantidade) INTO custo_total
    FROM Encomenda Item ei
    INNER JOIN Item i
    ON ei.id_item = i.id_item
    WHERE ei.id_encomenda = id_encomenda_param AND i.isDeleted = 0;

    SELECT SUM(e.distancia) INTO distancia_total
    FROM Encomenda e
    WHERE e.id_encomenda = id_encomenda_param AND e.isDeleted = 0;

    SET custo_total = custo_total + (distancia_total * 0.10);

    UPDATE Encomenda
    SET Encomenda.custo = custo_total
    WHERE Encomenda.id_encomenda = id_encomenda_param AND Encomenda.isDeleted = 0;
END$$
```

Soma o custo de cada item da encomenda e usa a distância também para atualizar o custo total da encomenda.

```
DELIMITER $$

CREATE PROCEDURE get_encomendas_mes (IN month INT, IN year INT)
BEGIN
    SELECT * FROM Encomenda
    WHERE MONTH(data_pedido) = month AND YEAR(data_pedido) = year AND Encomenda.isDeleted = 0
    ORDER BY data_pedido;
END$$
```

Mostra as encomendas com a data pedido de um certo mês e ano.

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

```
CREATE VIEW V_encomenda_cliente
AS
SELECT
    E.id_encomenda AS `ID Encomenda`,
    E.distancia AS `Distância (Km)`,
    E.data_pedido AS `Data do Pedido`,
    E.data_entrega AS `Data da Entrega`,
    E.custo AS `Custo`,
    MP.descricao AS `Método de Pagamento`,
    Estado.descricao_estado AS `Entado da Entrega`,
    C.nome AS `Nome do Cliente`,
    C.nif AS `NIF do Cliente`,
    C.email AS `Email do Cliente`,
    C.telemovel AS `Contacto do Cliente`
FROM
    Encomenda E
    INNER JOIN Cliente C
        ON E.id_cliente = C.id_cliente
    INNER JOIN Estado
        ON E.estado = Estado.id_estado
    INNER JOIN Metodo Pagamento MP
        on E.id metodo pagamento = MP.id metodo pagamento;
```

Mostra as informações das encomendas com o cliente na mesma linha

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

Atributo	Total em Bytes
ID (INT)	4
Email (VARCHAR (45))	45
Nome (VARCHAR (45))	45
Morada (INT)	4
Telemóvel (VARCHAR (15))	15
Tipo de Cliente (INT)	4
NIF (VARCHAR (9))	45
Data de nascimento (DATE)	3
Total por linha	165 Bytes

Tabela 8 Cliente

Atributo	Total em Bytes
ID (INT)	4
Distancia (INT)	4
Data de Pedido (DATETIME)	4
Data de Entrega (DATETIME)	4
Custo (FLOAT)	4
Método de Pagamento (INT)	4
Estafeta (INT)	4
Cliente (INT)	4
Itens (10*INT)	40
Funcionário de Armazém (INT)	4
Armazém (INT)	4
Total por linha	80 Bytes

Tabela 9 Encomenda

Atributo	Total em Bytes
ID (INT)	4
Descrição (TEXT)	120
Nome (VARCHAR (45))	45
Porcentagem de Álcool (FLOAT)	4
Preço (FLOAT))	4
Categoria (Total 50)	50
Marca (Total 60)	60
Total por linha	287 Bytes

Tabela 10 Item

Atributo	Total em Bytes
ID (INT)	4
Nome (VARCHAR (45))	40
Morada (INT)	4
Total por linha	48 Bytes

Tabela 11 Armazém

Atributo	Total em Bytes
ID (INT)	4
Linha endereço 1 (VARCHAR (45))	45
Linha endereço 2 (VARCHAR (45))	45
Código Postal (VARCHAR (20))	20
Localidade (VARCHAR (45))	45
Total por linha	159 Bytes

Tabela 12 Morada

Atributo	Total em Bytes
ID (INT)	4
Descrição (TEXT)	40
Total por linha	44 Bytes

Tabela 13 Tipo de Cliente

Atributo	Total em Bytes
ID (INT)	4
NIF (VARCHAR (9))	9
Nome (VARCHAR (45))	45
Data de Nascimento (DATE)	3
Armazém Associado (INT)	4
Total por linha	65 Bytes

Tabela 14 Funcionário armazém

Atributo	Total em Bytes
ID (INT)	4
NIF (VARCHAR (9))	9
Nr. Carta condução (VARCHAR (20))	20
Nome (VARCHAR (45))	45
Data de Nascimento (DATE)	3
Armazém Associado (INT)	4
Veículo Associado (Total 55 Bytes)	55
Total por linha	140 Bytes

Tabela 15 Estafeta

Inicialmente considerando que o armazém tem cerca de 140 itens 3 funcionários de armazém 4 estafetas 1 armazém e estão definidos 2 dos tipos de cliente o que no total seriam necessários 41068 bytes ($140 \cdot 287 + 3 \cdot 65 + 4 \cdot 140 + 45 + 2 \cdot 44$).

Espera-se que a empresa com o crescimento adicione todos os anos um funcionário de armazém e dois estafetas.

Por anos espera-se que surjam uma média de 20 novos itens/ano.

Espera-se que todos os meses haja uma média de 25 novos clientes, de forma a simplificar o os cálculos consideramos que todos os novos clientes têm novas moradas.

Diariamente deve haver uma média de 15 encomendas por dia.

Considerando os parâmetros acima podemos considerar que todos os anos precisaríamos de uma média de 541285 Bytes ($65 + 2 \cdot 140 + 20 \cdot 287 + 25 \cdot 12 \cdot (159 + 165) + 15 \cdot 365 \cdot 80$).

5.7 Plano de segurança e recuperação de dados

Uma vez que a base de dados é relativamente pequena, podem se fazer backups diários.

No que toca à segurança de dados, utilizamos uma *flag* '*isDeleted*', que nos permite "apagar um determinado registo/dado sem apagar definitivamente esse mesmo da base de dados, apenas marcando como estando apagado.

Os *timestamps* de criação e *update* são uteis para *debug*, isto é, aquando da necessidade de se fazer uma manutenção do sistema, é possível, saber quando os registos foram feitos.

6. Conclusões e Trabalho Futuro

Em conclusão, o sistema de base de dados construído foi bem-sucedido em atender às necessidades do projeto. O sistema foi projetado para ser escalável, seguro e fácil de usar. Trabalhamos diligentemente para garantir que todas as especificações fossem atendidas e que o prazo fosse cumprido. Realizamos também testes extensos para garantir a qualidade do sistema.

Existem alguns aspetos que podem ser melhorados e que poderiam ser implementados futuramente.

A utilização de índices é uma técnica comumente utilizada para melhorar a performance de uma base de dados e foi algo que acabou por não ser usado. Os índices permitem que as consultas sejam executadas de maneira mais rápida, uma vez que os dados já estão organizados de forma a facilitar a procura pelos registos desejados. Além disso, outras técnicas, como o uso de tabelas temporárias e otimização de consultas, também podem ser utilizadas para melhorar a performance de uma base de dados.

A separação de permissões de utilização também acabou por não ser implementado. Diferentes utilizadores como Cliente, Funcionário e até administrador teriam diferentes permissões de acesso e modificação da base de dados.

7. Referências Bibliográficas

BD	Base de Dados
DW	Data Warehouse
OLTP	<i>On-Line Analytical Processing</i>