Universidade do Minho
Mestrado Integrado em Engenharia Informática
Licenciatura em Ciências da Computação

Unidade Curricular de Bases de Dados

Ano Letivo de 2020/2021

Gestão de uma cadeia de Minimercados

Bruna Micaela Rodrigues Araújo a84914,

Filipe Matias Antas Viana a85948,

João Ricardo Rodrigues Nogueira a87973,

Marco Alexandre Félix de Lima a86030

Dezembro, 2020

BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

Gestão de uma cadeia de Minimercados

Bruna Micaela Rodrigues Araújo a84914,

Filipe Matias Antas Viana a85948,

João Ricardo Rodrigues Nogueira a87973,

Marco Alexandre Félix de Lima a86030

Dezembro,2020

Resumo

Este trabalho prático foi proposto no âmbito da unidade curricular de Base de Dados, tendo como por objetivo a análise, planeamento, modelação, arquitetura e implementação de um Sistema de Base de Dados relacional, com o tema "Gestão de uma cadeia de Minimercados".

O presente relatório visa detalhar toda a elaboração da base de dados em questão além de detalhar todas as etapas da sua criação baseada em modelos criados a partir de um levantamento bastante sucinto de dados relativos ao tema em questão, modelos tais como o conceptual e o lógico. Também será possível observar todo o processo da implementação física deste sistema terminando com uma breve estimativa do espaço em disco da base de dados e a sua taxa de crescimento anual.

Área de Aplicação: Modelação e implementação de Sistemas de Bases de Dados.

Palavras-Chave: SQL, Bases de Dados Relacionais, Modelo Conceptual, Modelo Lógico, Modelo Físico.

Índice

Índice de Figuras e Tabelas

1. Definição do Sistema

1.1. Contextualização

Uma base de dados é um conjunto de arquivos relacionados entre si sobre pessoas, lugares, etc. Uma cadeia de minimercados lida diariamente com diferentes tipos de entidades, entre os quais encontramos os clientes, os funcionários, os dados de faturação, os produtos, os fornecedores, a localização de cada minimercado que pertence a esta cadeia, etc.

Com a implementação de uma base de dados iremos conseguir lidar com todos os minimercados, que integram a cadeia, de um modo mais geral e com isto, facilitar o tratamento dos dados. Deste modo, achamos pertinente a criação de uma base de dados para conseguir gerir de forma mais eficiente a cadeia de minimercados.

1.2. Apresentação do Caso de Estudo

No nosso quotidiano verificamos cada vez mais uma expansão notória de várias cadeias de minimercados, em diversas localizações, não só para se dar a conhecer ao máximo de população possível, mas também, e principalmente, com vista a aumentar os seus lucros.

A cadeia de Minimercados SuperPrice é amplamente conhecida pela qualidade dos produtos que vende e a sua vasta escolha, organização dos produtos nos minimercados, estruturas exteriores inovadoras que atraem os cidadãos visualmente e, atendimento de excelência. Para garantir que tudo isto se cumpra torna-se indispensável a criação de uma base de dados para organizar da melhor maneira a parte “interna” e que os clientes não vêm, mas que é preciosa para o bom funcionamento desta cadeia.

1.3. Motivação e Objetivos

A afluência de clientes nos minimercados SuperPrice faz com que seja possível a expansão desta cadeia, e tudo o que nela se insere, ou seja, um maior número de minimercados SuperPrice tem como consequência a necessidade de mais funcionários, mais gerentes, mais faturação, etc.

Com a criação de uma base de dados torna-se mais simples a inserção de um novo minimercado SuperPrice e a análise do fluxo de clientes, funcionários, faturação, quantidade de produtos em stock e deste modo a cadeira irá ficar cada vez mais eficiente uma vez que existirá uma melhor evolução na gestão de todas as entidades que pertencem a cada minimercado SuperPrice. A implementação de uma base de dados também irá permitir fazer uma análise financeira mais detalhada de cada minimercado e deste modo os lucros irão subir, assim como a gestão de funcionários em cada minimercado irá ser mais eficiente uma vez que irá haver mais controlo sobre o número de clientes por dia que frequentam o minimercado em questão e deste modo contratar-se-ão ou despedir-se-ão funcionários em função do número de clientes diários.

1.4. Estrutura do Relatório

Serve este presente relatório para ilustrar o processo de uma criação de uma base de dados e em consequência disso, podemos encontrar já descrito anteriormente uma breve contextualização e apresentação do caso em estudo assim como os seus objetivos.

No Capítulo 2 é feito um levantamento de requisitos e uma consecutiva análise dos mesmos com vista a serem validados para assim ser possível prosseguir para a Modelação Conceptual, descrita no Capítulo 3, onde são identificadas e caraterizadas várias Entidades, Relacionamentos e os seus Atributos associados, terminando assim este capítulo com o Diagrama ER e a validação do modelo de dados.

No capítulo seguinte é descrita a modelação lógica através da construção de um Modelo Lógico e validação do mesmo, para assim ser possível a Implementação Física que é estudada no Capítulo 5. É neste capítulo que se selecionada um sistema de gestão de bases de dados e é feita a tradução do esquema lógico para o sistema escolhido, em SQL, assim como se traduzem algumas interrogações do utilizador para SQL, definem-se índices e ainda é feita uma estimativa do espaço em disco da base de dados e o seu possível crescimento anual.

Por fim, é feita uma última revisão do sistema implementado, terminando assim com algumas conclusões e devidas previsões para o trabalho futuro, não esquecendo os anexos que sustentam alguns pontos deste projeto.

2. Levantamento e Análise de Requisitos

2.1. Método de levantamento e de análise de requisitos

O levantamento de requisitos é sempre visto como um ponto de partida de qualquer projeto, neste caso não fugimos à regra e tentamos perceber quais os pontos a solucionar pelo sistema de base de dados e a compreender e identificar as necessidades que o responsável pela gerência da cadeia de minimercados SuperPrice quer ver resolvidas.

Com vista a recolhermos da forma mais explícita estas informações agendamos então uma primeira reunião com o responsável pela gerência da cadeia e assim pudemos coletar vários requisitos, os quais apresentamos de seguida, para o sistema de base de dados por meio de perguntas e observações feitas tanto pela nossa equipa como pelo próprio responsável da cadeia.

2.2. Requisitos Levantados

2.2.1 Requisitos de Descrição

Minimercado: localização, contactos, funcionários, clientes;

Funcionário: Nome, data de nascimento, função, residência, contactos, minimercado empregador;

Cliente: Nome, NIF, contactos, residência, minimercado frequentado.

2.2.2 Requisitos de Exploração

- Procurar clientes;
- Procurar funcionários;
- Procurar contactos de cliente/funcionário/minimercado;
- Procurar residência de cliente/funcionário;
- Procurar faturas;
- Procurar localização de minimercado;
- Procurar preços de produtos;
- Calcular total gasto por cliente;
- Procurar compras feitas por cliente;
- Procurar faturas emitidas por funcionário.

2.2.3 Requisitos de Controlo

- Adicionar Clientes e Funcionários;
- Adicionar contactos adicionais aos clientes, funcionários e ao minimercado;
- Alterar montante gasto por cliente.

2.3. Análise e validação geral dos requisitos

Após a primeira reunião realizada entre a nossa equipa e o responsável pela gerência da cadeia de minimercados SuperPrice, trabalhamos as informações recolhidas e obtivemos uma listagem de requisitos bem composta adicionando previamente algumas propostas de requisitos que o responsável pela cadeia não mencionou na primeira reunião, mas que nos pareceram benéficas para este sistema de base de dados.

Assim, para validarmos a nossa proposta de requisitos decidimos agendar uma segunda reunião com vista a analisar e descrever todos os requisitos e então, poder prosseguir com a criação da base de dados.

Em suma, neste segundo encontro ficaram validados os requisitos por nós propostos, após uma descrição clara e concisa dos mesmos e com uma prévia visão do cenário futuro para a base de dados em questão.

3. Modelação Conceptual

3.1 Abordagem da Modelação

Com auxílio dos requisitos obtidos no capítulo anterior tentámos integrar os conceitos de maneira a definir entidades, relacionamentos e atributos coesos.

Para a modelação utilizamos um programa recomendado pelo professor, o brmodelo, no qual definimos as entidades, os atributos e as relações entre cada entidade para a concretização do modelo conceptual da cadeia de minimercados SuperPrice.

3.2 Identificação e Caraterização das Entidades

Entidade	Descrição	Informação
Cliente	Informação do cliente	Nome, NIF, residência, contactos
Funcionário	Informação do funcionário	Nome, data de nascimento, função, contactos, residência
Minimercado	Informação do minimercado	Nome, localização, contactos
Lista de contactos	Conjunto de contactos	Emails, números telefónicos
Localização	Descrição de uma localização	País, Rua, cidade, código Postal
Fatura	Informação de uma fatura	Número da fatura, valor pago, data de emissão, produtos
Produto	Descrição de produto	Preço, Nome

Tabela 1 – Entidades e respetiva Caracterização

3.3 Identificação e Caraterização dos Relacionamentos

Entidade	Multiplicidade	Descrição	Multiplicidade	Entidade
Cliente	1	Frequenta	N	Minimercado
Cliente	N	Habita	1	Localização
Cliente	1	Pertence	N	Fatura
Cliente	1	Tem	1	Lista de Contactos
Funcionário	N	Trabalha	1	Minimercado
Funcionário	1	Tem	1	Lista de Contactos
Funcionário	N	Habita	1	Localização
Funcionário	1	Emite	N	Fatura
Fatura	1	Tem	N	Produto
Minimercado	1	Localiza-se	1	Localização
Minimercado	1	Tem	1	Lista de Contactos

Tabela 2 – Relacionamentos das Entidades

3.4 Identificação e Caraterização da associação dos Atributos com as Entidades e Relacionamentos

Entidade	Relacionamento	Atributos
Cliente	Localização, Lista de Contactos	Nome, NIF, Residência, Contactos
Funcionário	Localização, Lista de Contactos	Nome, Data de nascimento, Função, Residência, Contactos
Minimercado	Lista de Contactos, Localização	Nome, Contactos, Localização
Lista de Contactos	----------	Número de Telemóvel, Email
Localização	----------	País, Cidade, Rua, Código Postal
Fatura	Produto	Número, Data, Valor, Quantidade
Produtos	----------	Preço, Descrição

Tabela 3 – Associação de Atributos com as Entidades e Relacionamentos

3.5 Generalização de Entidades

Numa base de dados para uma cadeia de minimercados é fundamental considerar o Minimercado como uma entidade. Porém, um Minimercado precisa de muitas outras entidades para que este possa ser funcional e, deste modo, alcançar uma base de dados otimizada.

Cada Minimercado terá uma quantidade de Produtos, sendo que esta quantidade varia de minimercado para minimercado e, deste modo iremos considerar os Produtos como uma entidade também. Esta variação de quantidade de Produtos varia de minimercado para minimercado uma vez que o número de compras feitas por Cliente irá, como é óbvio, variar em cada Minimercado. Desta forma, é essencial considerar os Clientes como uma entidade assim como as Faturas para então registar a compra dos Produtos e deste modo conseguir analisar melhor a quantidade de produtos existentes no Minimercado.

Para o bom funcionamento de um Minimercado será elementar considerar os Funcionários como uma entidade pois, são eles quem faz o registo das Faturas e uma "gestão" do Minimercado. Como modo de identificação iremos considerar a Localização e uma Lista de Contactos como entidades uma vez que cada Minimercado irá estar localizado num certo local e cada um terá os seus devidos contactos, assim como os Clientes e os Funcionários que possuirão uma localização corresponde à sua morada e uma lista de contactos.

3.6 Diagrama ER

No diagrama a seguir apresentado constam todas as entidades, atributos e relacionamentos apresentados anteriormente.

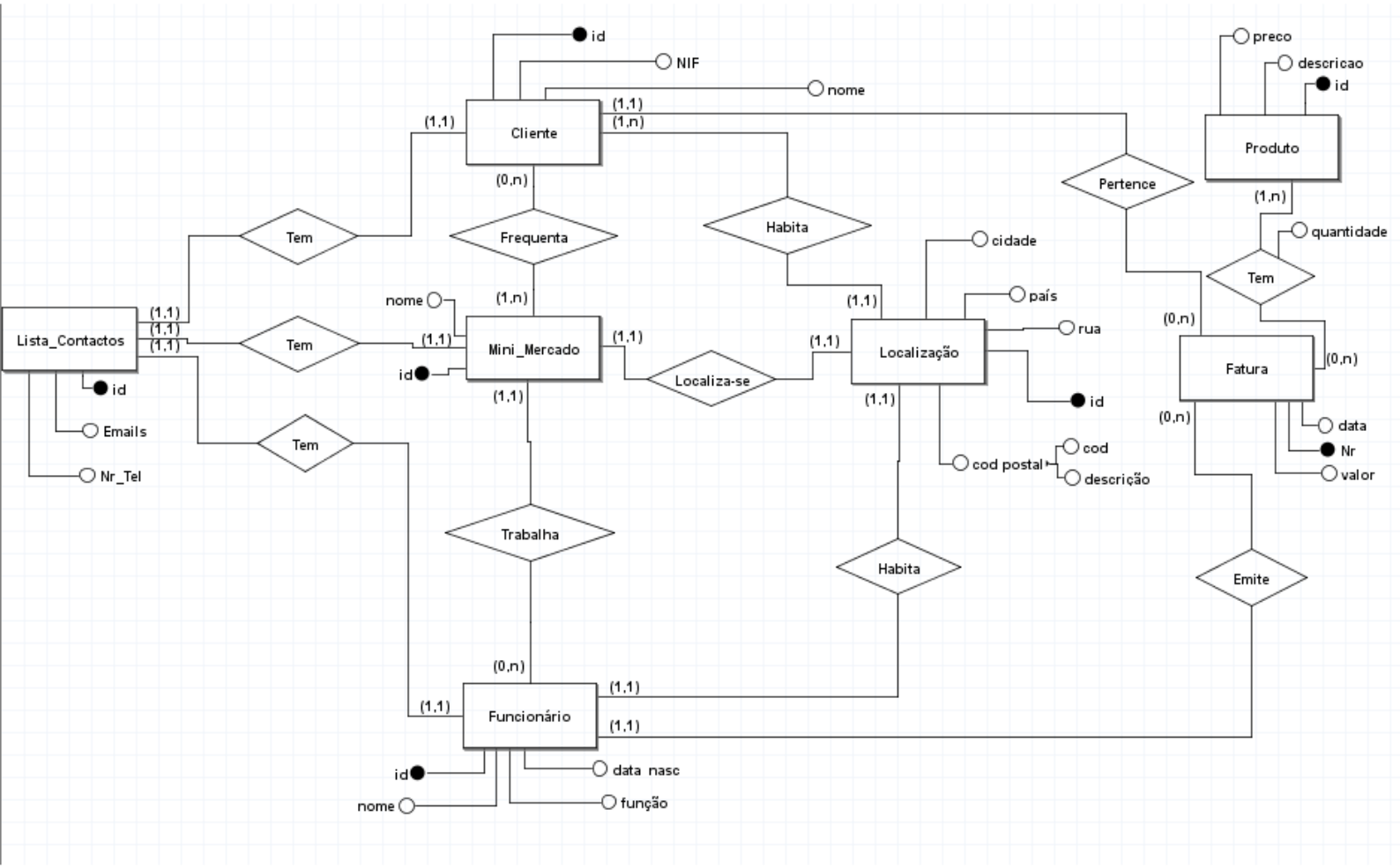

Figura 1 – Diagrama ER

3.7 Validação do Modelo de Dados

Para validarmos o modelo de dados apresentado anteriormente, este foi devidamente enviado para o responsável da gerência da cadeia de minimercados para o próprio o poder analisar e tirar as suas conclusões.

Após a verificação do modelo por parte do responsável pela cadeia este pediu alguns esclarecimentos acerca de alguns relacionamentos estabelecidos. Então, para uma melhor explicação foi agendada uma terceira reunião com esse objetivo, explicar o modelo conceptual apresentado e assim validar o próprio.

Desta terceira reunião resultou então a validação do modelo de dados apresentado, uma vez que este engloba todos os requisitos levantados e cumpre a utilidade pretendida pelo responsável da gerência da cadeia de minimercados SuperPrice.

4. Modelação Lógica

4.1 Construção e Validação do Modelo de Dados Lógico

Para derivar a base para o modelo lógico do modelo conceptual seguimos as seguintes regras:

- Cada Entidade deve ser traduzida para uma tabela onde os seus atributos serão as colunas;

- Para atributos multivalorados deve ser criada uma nova tabela que contem a chave estrangeira referente à entidade a que pertence;

- Relacionamentos 1-1 : a chave primária de uma das entidades deve tornar-se uma chave estrangeira na tabela da outra entidade conforme coerente;

- Relacionamentos 1-n : a chave primária da entidade com multiplicidade 1 será a chave estrangeira para a entidade com multiplicidade n.

Após derivar a base para o modelo podemos fazer decisões adicionais de maneira a otimizar os recursos de memória utilizada, nomeadamente criar tabelas para guardar valores que possivelmente seriam repetidos várias vezes, para o caso específico deste problema, temos como exemplo os atributos “Rua”, “Cidade”, “País” e “Código Postal” da entidade “Localização”, e ainda os atributos “Emails” e “Nr_Telemovel” da entidade “Lista de Contactos”.

4.2 Desenho do Modelo Lógico

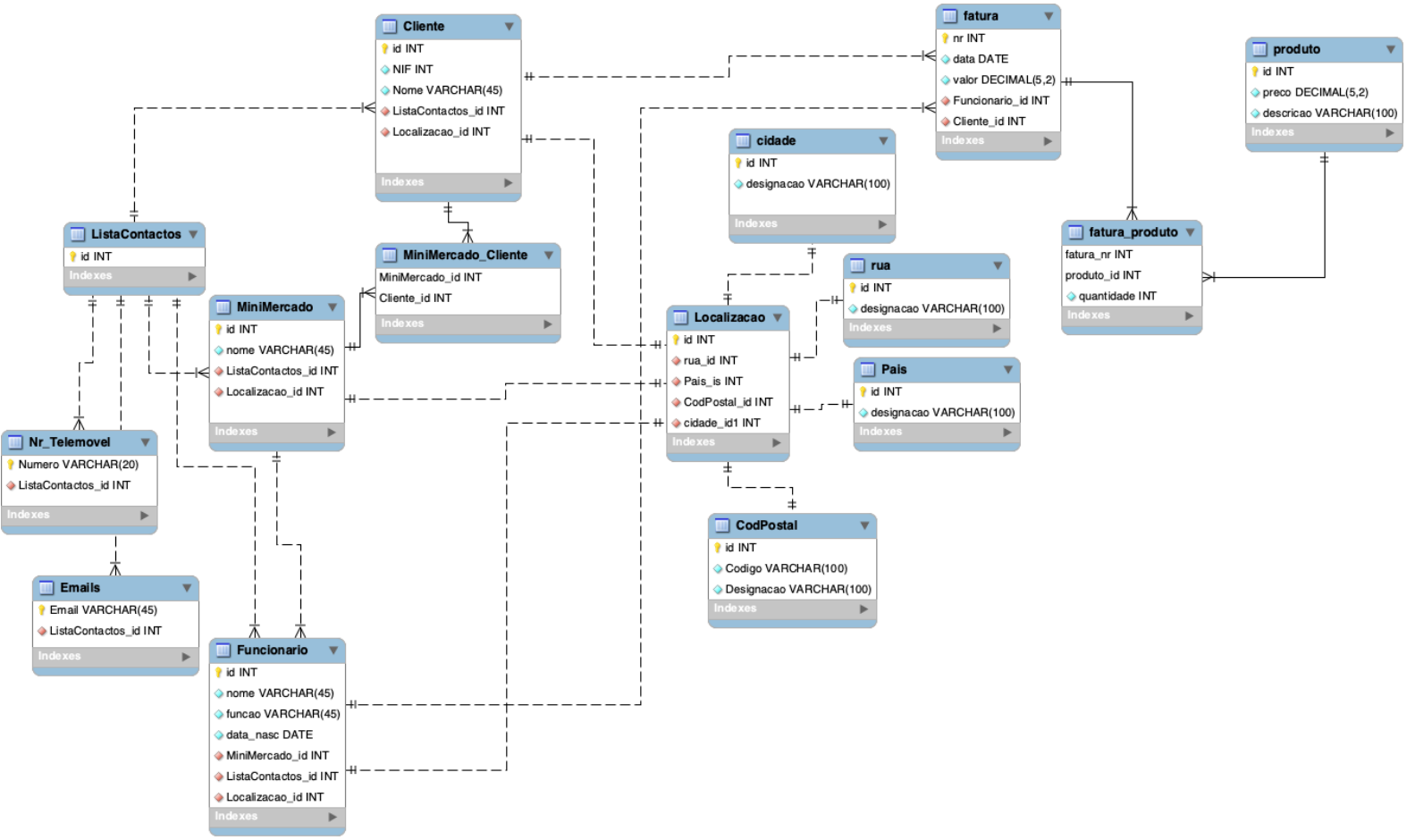

Figura 2 – Modelo Lógico

4.3 Validação do Modelo através da Normalização

O modelo lógico apresentado anteriormente é válido tendo em conta as regras da normalização, seguindo-se a análise de cada uma das regras:

- 1ª Forma Normal (1FN): todos os atributos são atómicos, ou seja, não divisíveis.

No nosso modelo lógico verifica-se que é cumprida 1FN, uma vez que, por exemplo, se a entidade "Localizacao" tivesse como atributos "Cidade"," Rua", "Pais" e "CodPostal" teria de ser decomposta para que os atributos passassem a ser atómicos.

- 2ª Forma Normal (2FN): o esquema relacional está na 1FN e todos os atributos não-chave dependem por completo da chave primária.

Como o nosso modelo lógico está na 1FN então já cumpre um dos "requisitos" da 2FN, falta só verificar se os atributos não-chave dependem da chave primária. Visto que todas as nossas tabelas têm apenas uma chave primária então podemos concluir que o modelo lógico está na 2FN pois nenhum atributo depende de uma parte da chave pois, a chave não é composta por partes.

- 3ª Forma Normal (3FN): o esquema relacional está na 2FN e nenhum atributo não-chave depende, por transitividade, da chave primária.

Visto que o modelo lógico apresentado cumpre a 2FN então falta analisar se nenhum atributo não-chave depende, por transitividade, da chave primária. Dado que, nas tabelas que constituem o modelo lógico não existe nenhuma dependência entre os atributos não-chave concluímos então que o modelo cumpre a 3FN.

4.4 Validação do Modelo com Interrogações do Utilizador

Para obter uma validação do Modelo com interrogações do utilizador podemos, por exemplo, escolher alguns requisitos de exploração, envolvendo entidades com chaves estrangeiras, para explicar como os acessos acontecem.

Por exemplo, na procedure "residência_func" usa-se como input o id de um funcionário, ou seja, esta procedure vai procurar o id da "localizacao" correspondente ao id do "funcionario" e devolverá o nome da rua, da cidade, do país e o código postal, estando estes armazenados na entidade "localizacao" através de foreign keys e que, sabendo o id da localizacao (que se relaciona ao id do funcionario dado no input) ficam-se a conhecer os ids de cada um dos parâmetros da localizacao.

Um outro exemplo, na procedure "lista_nrtelemovel_cliente" (que procura o número de telemóvel de um cliente), para aceder à entidade "Nr_Telemovel" tem-se uma outra entidade ("ListaContactos") para unir estas duas, e assim para fazer a procura chama-se a esta procedure, sendo o input o id do cliente cujo número se quer saber, este id vai então corresponder a um id na Lista de Contactos que é armazenado na entidade "Nr_telemovel" como foreign key.

4.5 Revisão do Modelo Lógico Produzido

Após a construção do modelo lógico é importante fazer uma revisão para assim poder verificar-se se realmente efetua todos os requisitos a que se propunha e cumpre a utilidade pretendida.

Depois da revisão do modelo lógico, podemos então validá-lo uma vez que todos os requisitos propostos são executados, também através do modelo consegue prever-se a devida utilidade e, ainda, está organizado de maneira simplificada.

5. Implementação Física

5.1 Seleção do Sistema de gestão de Bases de Dados

O sistema selecionado para a implementação da base de dados foi o MySQL uma vez que a nível de popularidade está muito bem posicionado o que origina uma maior quantidade de recursos disponíveis e compatibilidade com outros programas e linguagens.
O MySQL oferece também suporte de diversas medidas que melhoram a segurança e ainda é licenciado com GPL para o caso de desenvolvimento de software livre.
Tudo isto são razões que nos levaram a escolher este sistema de gestão de bases de dados.

5.2 Tradução do Modelo Lógico

O programa utilizado para a modelação lógica foi o "MySQL Workbench", e, como consequência desta escolha foi possível utilizar a ferramenta "Forward Engeneer" por ele disponibilizada para a obtenção do sistema físico.
O script obtido está apresentado no <<Anexo 1>>.

5.3 Tradução das interrogações do utilizador em SQL

```
-- Procurar Contactos de Cliente
DELIMITER //
CREATE PROCEDURE lista_nrtelmovel_cliente (IN id_cliente INT)
BEGIN
        SELECT cliente.id, cliente.Nome, nr_telemovel.Numero
  FROM nr_telemovel, listacontactos, cliente
```

```
        WHERE cliente.id = id_cliente
  AND cliente.ListaContactos_id = listacontactos.id
  AND listacontactos.id = nr_telemovel.ListaContactos_id;
END; //
DELIMITER ;

DELIMITER //
CREATE PROCEDURE lista_email_cliente (IN id_client INT)
BEGIN
        SELECT cliente.id, cliente.Nome, emails.Email
  FROM emails, listacontactos, cliente
        WHERE cliente.id = id_client
  AND cliente.ListaContactos_id = listacontactos.id
  AND listacontactos.id = emails.ListaContactos_id;
END; //
DELIMITER ;

-- Procurar Residência de Funcionário
DELIMITER //
CREATE PROCEDURE residencia_func (IN id_func INT)
BEGIN
        SELECT funcionario.id, funcionario.Nome, rua.designacao, cidade.designacao,
codpostal.Codigo, codpostal.Designacao, pais.designacao
  FROM funcionario, localizacao, rua, cidade, codpostal, pais
        WHERE funcionario.id=id_func
  AND funcionario.localizacao_id = localizacao.id
  AND localizacao.rua_id = rua.id
  AND localizacao.cidade_id1 = cidade.id
  AND localizacao.CodPostal_id = codpostal.id
  AND localizacao.Pais_is = pais.id;
END; //
DELIMITER ;


-- Total gasto por Cliente
DELIMITER //
create procedure gasto_por_cliente (in id_total INT)
begin
```

```
   select cliente.id, cliente.nome, sum(fatura.valor)
   from fatura,cliente
   where id_total = cliente.id
   and fatura.cliente_id = cliente.id;
end; //
DELIMITER ;
```

5.4 Estimativa do Espaço em disco da base de dados e Taxa de crescimento anual

Após uma reflexão da nossa equipa com o responsável pela gerência da cadeia de minimercados, foi-nos informado que num futuro próximo não está prevista qualquer expansão desta cadeia visto que o negócio está bem encaminhado e quer alguma estabilidade financeira antes de apostar numa expansão. Portanto, o espaço em disco da base de dados não é muito significativo.

5.5 Revisão do Sistema Implementado

Com uma revisão de todo o trabalho feito e analisando todos os procedimentos, será seguro dizer que o sistema foi bem implementado uma vez que com este conseguimos reproduzir uma boa gestão de todas as entidades pertencentes a uma cadeia de minimercados. Conseguimos também fazer uma análise geral de todos os ganhos e despesas de cada minimercado pertencentes a esta cadeia e, deste modo, atingir o objetivo principal de um sistema de base de dados para uma cadeia de minimercados, que é conseguir aumentar o lucro da cadeia e deste modo fazer com que cresça cada vez mais.
No futuro poderá ser possível o melhoramento da eficiência do sistema tornando-o mais leve e com melhor e maior eficácia.

6. Conclusões e Trabalho Futuro

De um modo geral podemos afirmar que os objetivos traçados para a elaboração deste projeto foram satisfatoriamente atingidos uma vez que conseguimos criar a base de dados em questão e simultaneamente fazer a gestão e apresentar dados da mesma.

A nível de competências adquiridas, podemos destacar a experiência da criação e gestão de uma base de dados relacional, a criação de modelos lógicos e físicos assim como a importância de um levantamento de dados e estruturação dos mesmos, o aperfeiçoamento dos conhecimentos da linguagem MySQL e as suas potencialidades.

Por outro lado, somos da opinião que deveria ser feito um estudo mais abrangente do tema de modo a aperfeiçoar o conteúdo da base de dados e assim obter melhores resultados.

Na realização do presente relatório surgiram algumas dificuldades, nomeadamente na escolha, definição e caraterização de índices em SQL e na definição e caraterização das vistas de utilização em SQL. Dificuldades essas que não conseguimos ver resolvidas.

Em suma, podemos concluir que o objetivo foi atingido assim como foram adquiridas competências essênciais para aplicar no trabalho futuro.

Anexos

I. Anexo 1: Script gerado pelo MySQL Workbench

```
-- 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,ERRO
R_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----------------------------------------------------
-- Schema MiniMercado
-- -----------------------------------------------------

-- -----------------------------------------------------
-- Schema MiniMercado
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `MiniMercado` DEFAULT CHARACTER SET utf8 ;
USE `MiniMercado` ;

-- -----------------------------------------------------
-- Table `MiniMercado`.`ListaContactos`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `MiniMercado`.`ListaContactos` (
  `id` INT NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;


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

```
-- Table `MiniMercado`.`rua`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `MiniMercado`.`rua` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `designacao` VARCHAR(100) NOT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `MiniMercado`.`Pais`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `MiniMercado`.`Pais` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `designacao` VARCHAR(100) NOT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `MiniMercado`.`CodPostal`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `MiniMercado`.`CodPostal` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `Codigo` VARCHAR(100) NOT NULL,
  `Designacao` VARCHAR(100) NOT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `MiniMercado`.`cidade`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `MiniMercado`.`cidade` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `designacao` VARCHAR(100) NOT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `MiniMercado`.`Localizacao`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `MiniMercado`.`Localizacao` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `rua_id` INT NOT NULL,
  `Pais_is` INT NOT NULL,
  `CodPostal_id` INT NOT NULL,
  `cidade_id1` INT NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_Localizacao_rua1_idx` (`rua_id` ASC) VISIBLE,
  INDEX `fk_Localizacao_Pais1_idx` (`Pais_is` ASC) VISIBLE,
  INDEX `fk_Localizacao_CodPostal1_idx` (`CodPostal_id` ASC) VISIBLE,
  INDEX `fk_Localizacao_cidade2_idx` (`cidade_id1` ASC) VISIBLE,
  CONSTRAINT `fk_Localizacao_rua1`
    FOREIGN KEY (`rua_id`)
    REFERENCES `MiniMercado`.`rua` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Localizacao_Pais1`
    FOREIGN KEY (`Pais_is`)
    REFERENCES `MiniMercado`.`Pais` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Localizacao_CodPostal1`
    FOREIGN KEY (`CodPostal_id`)
    REFERENCES `MiniMercado`.`CodPostal` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Localizacao_cidade2`
    FOREIGN KEY (`cidade_id1`)
    REFERENCES `MiniMercado`.`cidade` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `MiniMercado`.`MiniMercado`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `MiniMercado`.`MiniMercado` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `nome` VARCHAR(45) NOT NULL,
  `ListaContactos_id` INT NOT NULL,
  `Localizacao_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_MiniMercado_ListaContactos1_idx` (`ListaContactos_id` ASC) VISIBLE,
  INDEX `fk_MiniMercado_Localizacao1_idx` (`Localizacao_id` ASC) VISIBLE,
  CONSTRAINT `fk_MiniMercado_ListaContactos1`
    FOREIGN KEY (`ListaContactos_id`)
    REFERENCES `MiniMercado`.`ListaContactos` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_MiniMercado_Localizacao1`
    FOREIGN KEY (`Localizacao_id`)
    REFERENCES `MiniMercado`.`Localizacao` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `MiniMercado`.`Cliente`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `MiniMercado`.`Cliente` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `NIF` INT NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `ListaContactos_id` INT NOT NULL,
  `Localizacao_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_Cliente_ListaContactos1_idx` (`ListaContactos_id` ASC) VISIBLE,
  INDEX `fk_Cliente_Localizacao1_idx` (`Localizacao_id` ASC) VISIBLE,
  CONSTRAINT `fk_Cliente_ListaContactos1`
```

```
    FOREIGN KEY (`ListaContactos_id`)
    REFERENCES `MiniMercado`.`ListaContactos` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Cliente_Localizacao1`
    FOREIGN KEY (`Localizacao_id`)
    REFERENCES `MiniMercado`.`Localizacao` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `MiniMercado`.`MiniMercado_Cliente`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `MiniMercado`.`MiniMercado_Cliente` (
  `MiniMercado_id` INT NOT NULL,
  `Cliente_id` INT NOT NULL,
  PRIMARY KEY (`MiniMercado_id`, `Cliente_id`),
  INDEX `fk_MiniMercado_has_Cliente_Cliente1_idx` (`Cliente_id` ASC) VISIBLE,
  INDEX `fk_MiniMercado_has_Cliente_MiniMercado1_idx` (`MiniMercado_id` ASC) VISIBLE,
  CONSTRAINT `fk_MiniMercado_has_Cliente_MiniMercado1`
    FOREIGN KEY (`MiniMercado_id`)
    REFERENCES `MiniMercado`.`MiniMercado` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_MiniMercado_has_Cliente_Cliente1`
    FOREIGN KEY (`Cliente_id`)
    REFERENCES `MiniMercado`.`Cliente` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `MiniMercado`.`Funcionario`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `MiniMercado`.`Funcionario` (
```

```
  `id` INT NOT NULL AUTO_INCREMENT,
  `nome` VARCHAR(45) NOT NULL,
  `funcao` VARCHAR(45) NOT NULL,
  `data_nasc` DATE NOT NULL,
  `MiniMercado_id` INT NOT NULL,
  `ListaContactos_id` INT NOT NULL,
  `Localizacao_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_Funcionario_MiniMercado1_idx` (`MiniMercado_id` ASC) VISIBLE,
  INDEX `fk_Funcionario_ListaContactos1_idx` (`ListaContactos_id` ASC) VISIBLE,
  INDEX `fk_Funcionario_Localizacao1_idx` (`Localizacao_id` ASC) VISIBLE,
  CONSTRAINT `fk_Funcionario_MiniMercado1`
    FOREIGN KEY (`MiniMercado_id`)
    REFERENCES `MiniMercado`.`MiniMercado` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Funcionario_ListaContactos1`
    FOREIGN KEY (`ListaContactos_id`)
    REFERENCES `MiniMercado`.`ListaContactos` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Funcionario_Localizacao1`
    FOREIGN KEY (`Localizacao_id`)
    REFERENCES `MiniMercado`.`Localizacao` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `MiniMercado`.`Emails`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `MiniMercado`.`Emails` (
  `Email` VARCHAR(45) NOT NULL,
  `ListaContactos_id` INT NOT NULL,
  PRIMARY KEY (`Email`),
  INDEX `fk_Emails_ListaContactos1_idx` (`ListaContactos_id` ASC) VISIBLE,
  CONSTRAINT `fk_Emails_ListaContactos1`
```

```
  FOREIGN KEY (`ListaContactos_id`)
  REFERENCES `MiniMercado`.`ListaContactos` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `MiniMercado`.`Nr_Telemovel`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `MiniMercado`.`Nr_Telemovel` (
  `Numero` VARCHAR(20) NOT NULL,
  `ListaContactos_id` INT NOT NULL,
  PRIMARY KEY (`Numero`),
  INDEX `fk_Nr_Telemovel_ListaContactos1_idx` (`ListaContactos_id` ASC) VISIBLE,
  CONSTRAINT `fk_Nr_Telemovel_ListaContactos1`
    FOREIGN KEY (`ListaContactos_id`)
    REFERENCES `MiniMercado`.`ListaContactos` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `MiniMercado`.`produto`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `MiniMercado`.`produto` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `preco` DECIMAL(5,2) NOT NULL,
  `descricao` VARCHAR(100) NOT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `MiniMercado`.`fatura`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `MiniMercado`.`fatura` (
```

```
  `nr` INT NOT NULL AUTO_INCREMENT,
  `data` DATE NOT NULL,
  `valor` DECIMAL(5,2) NOT NULL,
  `Funcionario_id` INT NOT NULL,
  `Cliente_id` INT NOT NULL,
  PRIMARY KEY (`nr`),
  INDEX `fk_fatura_Funcionario1_idx` (`Funcionario_id` ASC) VISIBLE,
  INDEX `fk_fatura_Cliente1_idx` (`Cliente_id` ASC) VISIBLE,
  CONSTRAINT `fk_fatura_Funcionario1`
    FOREIGN KEY (`Funcionario_id`)
    REFERENCES `MiniMercado`.`Funcionario` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_fatura_Cliente1`
    FOREIGN KEY (`Cliente_id`)
    REFERENCES `MiniMercado`.`Cliente` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `MiniMercado`.`fatura_produto`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `MiniMercado`.`fatura_produto` (
  `fatura_nr` INT NOT NULL,
  `produto_id` INT NOT NULL,
  `quantidade` INT NOT NULL,
  PRIMARY KEY (`fatura_nr`, `produto_id`),
  INDEX `fk_fatura_has_produto_produto1_idx` (`produto_id` ASC) VISIBLE,
  INDEX `fk_fatura_has_produto_fatura1_idx` (`fatura_nr` ASC) VISIBLE,
  CONSTRAINT `fk_fatura_has_produto_fatura1`
    FOREIGN KEY (`fatura_nr`)
    REFERENCES `MiniMercado`.`fatura` (`nr`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_fatura_has_produto_produto1`
    FOREIGN KEY (`produto_id`)
```

```
  REFERENCES `MiniMercado`.`produto` (`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. Anexo 2: Sript de Povoamento

```
use minimercado;

insert into listacontactos values (1);
insert into listacontactos values (2);
insert into listacontactos values (3);
insert into listacontactos values (4);
insert into listacontactos values (5);
insert into listacontactos values (6);
insert into listacontactos values (7);
insert into listacontactos values (8);
insert into listacontactos values (9);
insert into listacontactos values (10);
insert into listacontactos values (11);
insert into listacontactos values (12);
insert into listacontactos values (13);
insert into listacontactos values (14);
insert into listacontactos values (15);
insert into listacontactos values (16);
insert into listacontactos values (17);

insert into emails values ('taniafelix@gmail.com',1);
insert into emails values ('guilima@gmail.com',2);
insert into emails values ('filipeteixeira@gmail.com',3);
insert into emails values ('brunosoares@gmail.com',4);
insert into emails values ('ritamendes@gmail.com',5);
insert into emails values ('julianamiranda@gmail.com',6);
```

```
insert into emails values ('biancamoura@gmail.com',7);
insert into emails values ('joaocruz@gmail.com',8);
insert into emails values ('pedrovalente@gmail.com',9);
insert into emails values ('elisacampos@gmail.com',10);
insert into emails values ('luizalvim@gmail.com',11);
insert into emails values ('filomenateixeira@gmail.com',12);
insert into emails values ('gonçalosaraiva@gmail.com',13);
insert into emails values ('gonçalosaraiva@gmail.com',14);
insert into emails values ('gonçalosaraiva@gmail.com',15);
insert into emails values ('superpriceporto@gmail.com',16);
insert into emails values ('superpriceporto@gmail.com',17);

insert into nr_telemovel values ('+351960000000',1);
insert into nr_telemovel values ('+351960000001',2);
insert into nr_telemovel values ('+351960000002',3);
insert into nr_telemovel values ('+351960000003',4);
insert into nr_telemovel values ('+351960000004',5);
insert into nr_telemovel values ('+351960000005',6);
insert into nr_telemovel values ('+351960000006',7);
insert into nr_telemovel values ('+351960000007',8);
insert into nr_telemovel values ('+351960000008',9);
insert into nr_telemovel values ('+351960000009',10);
insert into nr_telemovel values ('+351960000010',11);
insert into nr_telemovel values ('+351960000011',12);
insert into nr_telemovel values ('+351960000012',13);
insert into nr_telemovel values ('+351960000013',14);
insert into nr_telemovel values ('+351253000000',15);
insert into nr_telemovel values ('+351253000000',16);
insert into nr_telemovel values ('+351253000000',17);

insert into rua values (1,'Rua 1');
insert into rua values (2,'Rua 2');
insert into rua values (3,'Rua 3');
insert into rua values (4,'Rua 4');
insert into rua values (5,'Rua 5');
insert into rua values (6,'Rua 6');
insert into rua values (7,'Rua 7');
insert into rua values (8,'Rua 8');
```

```
insert into rua values (9,'Rua 9');
insert into rua values (10,'Rua 10');

insert into cidade values (1,'Braga');
insert into cidade values (2,'Porto');
insert into cidade values (3,'Lisboa');

insert into pais values (1,'Portugal');

insert into codpostal values (1,'0000-000','Distrito 1');
insert into codpostal values (2,'0000-001','Distrito 2');
insert into codpostal values (3,'0000-002','Distrito 3');
insert into codpostal values (4,'0000-003','Distrito 4');
insert into codpostal values (5,'0000-004','Distrito 5');
insert into codpostal values (6,'0000-005','Distrito 6');
insert into codpostal values (7,'0000-006','Distrito 7');
insert into codpostal values (8,'0000-007','Distrito 8');
insert into codpostal values (9,'0000-008','Distrito 9');
insert into codpostal values (10,'0000-009','Distrito 10');

insert into localizacao values (1,1,1,1,1);
insert into localizacao values (2,2,1,2,1);
insert into localizacao values (3,3,1,3,1);
insert into localizacao values (4,4,1,3,1);
insert into localizacao values (5,5,1,4,2);
insert into localizacao values (6,6,1,7,3);
insert into localizacao values (7,7,1,5,2);
insert into localizacao values (8,8,1,6,3);
insert into localizacao values (9,9,1,9,2);
insert into localizacao values (10,10,1,10,1);

insert into cliente values (1,111111111,'Tânia Félix',1,4);
insert into cliente values (2,222222222,'Guilherme Lima',2,2);
insert into cliente values (3,333333333,'Filipe Teixeira',3,4);
insert into cliente values (4,444444444,'Bruno Soares',4,3);
insert into cliente values (5,555555555,'Rita Mendes',5,5);
insert into cliente values (6,666666666,'Juliana Miranda',6,5);
insert into cliente values (7,777777777,'Bianca Moura',7,5);
```

```
insert into cliente values (8,888888888,'João Cruz',8,1);
insert into cliente values (9,999999999,'Pedro Valente',9,2);
insert into cliente values (10,101010101,'Elisa Campos',10,1);

insert into minimercado values (1,'SuperPrice',15,10);
insert into minimercado values (2,'SuperPrice',16,9);
insert into minimercado values (3,'SuperPrice',17,8);

insert into minimercado_cliente values (1,1);
insert into minimercado_cliente values (1,2);
insert into minimercado_cliente values (1,3);
insert into minimercado_cliente values (1,4);
insert into minimercado_cliente values (2,5);
insert into minimercado_cliente values (2,6);
insert into minimercado_cliente values (2,7);
insert into minimercado_cliente values (1,8);
insert into minimercado_cliente values (1,9);
insert into minimercado_cliente values (1,10);

insert into funcionario values (1,'Luiza Alvim','Caixa','1990-09-26',1,11,3);
insert into funcionario values (2,'Filomena Teixeira','Limpeza','1969-01-01',1,12,3);
insert into funcionario values (3,'Gonçalo Saraiva','Caixa e Limpeza','1982-08-14',2,13,7);
insert into funcionario values (4,'Nelson Nunes','Caixa e Limpeza','1985-09-10',3,14,6);

insert into fatura values (1,'2019-02-03',14.95,1,1);
insert into fatura values (2,'2019-02-10',12.58,1,1);
insert into fatura values (3,'2019-03-21',4.89,1,2);
insert into fatura values (4,'2019-03-21',5.98,1,3);
insert into fatura values (5,'2019-04-01',3.00,1,3);
insert into fatura values (6,'2019-04-15',2.82,1,3);
insert into fatura values (7,'2019-04-08',3.70,1,4);
insert into fatura values (8,'2019-04-12',8.97,3,5);
insert into fatura values (9,'2019-04-22',4.20,3,5);
insert into fatura values (10,'2019-05-09',3.74,3,5);
insert into fatura values (11,'2019-05-09',3.70,3,6);
insert into fatura values (12,'2019-05-21',5.18,3,7);
insert into fatura values (13,'2019-05-23',13.90,1,8);
insert into fatura values (14,'2019-06-01',2.58,3,9);
```

```
insert into fatura values (15,'2019-06-22',2.45,1,10);
insert into fatura values (16,'2019-06-22',7.48,1,10);


insert into produto values (1,2.99,'leite');
insert into produto values (2,6.29,'Queijo');
insert into produto values (3,2.39,'Chouriço');
insert into produto values (4,1.85,'Bacon');
insert into produto values (5,4.89,'Bolo Rei');
insert into produto values (6,2.45,'Pão de Ló');
insert into produto values (7,1.50,'Garrafão de Água 5L');
insert into produto values (8,2.99,'Azeite Virgem');
insert into produto values (9,2.19,'Óleo de Cozinha');
insert into produto values (10,0.94,'Arroz 1KG');
insert into produto values (11,0.74,'Açucar 1KG');
insert into produto values (12,2.99,'Cereais');
insert into produto values (13,1.39,'Iogurtes');
insert into produto values (14,4.20,'Champô');
insert into produto values (15,3.74,'Gel de Banho');
insert into produto values (16,2.59,'Pasta dos Dentes');
insert into produto values (17,1.41,'Papel de Cozinha');
insert into produto values (18,1.29,'Detergente Líquido');
insert into produto values (19,0.99,'Esponja');
insert into produto values (20,0.74,'Gomas');

insert into fatura_produto values (1,1,5);
insert into fatura_produto values (2,2,2);
insert into fatura_produto values (3,5,1);
insert into fatura_produto values (4,8,2);
insert into fatura_produto values (5,7,2);
insert into fatura_produto values (6,10,3);
insert into fatura_produto values (7,11,5);
insert into fatura_produto values (8,12,3);
insert into fatura_produto values (9,14,1);
insert into fatura_produto values (10,15,1);
insert into fatura_produto values (11,20,5);
insert into fatura_produto values (12,16,2);
insert into fatura_produto values (13,13,10);
```

```
insert into fatura_produto values (14,18,2);
insert into fatura_produto values (15,6,1);
insert into fatura_produto values (16,7,1);
insert into fatura_produto values (16,8,2);
```

III. Anexo 3: Script da tradução das interrogações do utilizador em SQL

```
use minimercado;
-- Procurar Clientes
select *from Cliente;

-- Procurar Funcionários
select *from funcionario;

-- Procurar Contactos de Cliente
DELIMITER //
CREATE PROCEDURE lista_nrtelmovel_cliente (IN id_cliente INT)
BEGIN
        SELECT cliente.id, cliente.Nome, nr_telemovel.Numero
  FROM nr_telemovel, listacontactos, cliente
        WHERE cliente.id = id_cliente
  AND cliente.ListaContactos_id = listacontactos.id
  AND listacontactos.id = nr_telemovel.ListaContactos_id;
END; //
DELIMITER ;

DELIMITER //
CREATE PROCEDURE lista_email_cliente (IN id_client INT)
BEGIN
        SELECT cliente.id, cliente.Nome, emails.Email
  FROM emails, listacontactos, cliente
        WHERE cliente.id = id_client
  AND cliente.ListaContactos_id = listacontactos.id
  AND listacontactos.id = emails.ListaContactos_id;
END; //
DELIMITER ;
```

```
-- Procurar Contactos de Funcionário
DELIMITER //
CREATE PROCEDURE lista_nrtelmovel_func (IN id_funcionario INT)
BEGIN
        SELECT funcionario.id, funcionario.nome,nr_telemovel.Numero
  FROM nr_telemovel, listacontactos, funcionario
        WHERE funcionario.id = id_funcionario
  AND funcionario.ListaContactos_id = listacontactos.id
  AND listacontactos.id = nr_telemovel.ListaContactos_id;
END; //
DELIMITER ;

DELIMITER //
CREATE PROCEDURE lista_email_func (IN id_func INT)
BEGIN
        SELECT funcionario.id, funcionario.Nome, emails.Email
  FROM emails, listacontactos, funcionario
        WHERE funcionario.id = id_func
  AND funcionario.ListaContactos_id = listacontactos.id
  AND listacontactos.id = emails.ListaContactos_id;
END; //
DELIMITER ;


-- Procurar Contactos Minimercado
DELIMITER //
CREATE PROCEDURE lista_nrtelmovel_merc (IN id_merc INT)
BEGIN
        SELECT minimercado.id, minimercado.nome,nr_telemovel.Numero
  FROM nr_telemovel, listacontactos, minimercado
        WHERE minimercado.id = id_merc
  AND minimercado.ListaContactos_id = listacontactos.id
  AND listacontactos.id = nr_telemovel.ListaContactos_id;
END; //
DELIMITER ;

DELIMITER //
CREATE PROCEDURE lista_email_merc (IN id_merc INT)
```

```
BEGIN
        SELECT minimercado.id, minimercado.Nome, emails.Email
  FROM emails, listacontactos, minimercado
        WHERE minimercado.id = id_merc
  AND minimercado.ListaContactos_id = listacontactos.id
  AND listacontactos.id = emails.ListaContactos_id;
END; //
DELIMITER ;

-- Procurar Residência de um Cliente
DELIMITER //
CREATE PROCEDURE residencia_cliente (IN id_cliente INT)
BEGIN
        SELECT cliente.id, cliente.Nome, rua.designacao, cidade.designacao, codpostal.Codigo,
codpostal.Designacao, pais.designacao
  FROM cliente, localizacao, rua, cidade, codpostal, pais
        WHERE cliente.id=id_cliente
  AND cliente.localizacao_id = localizacao.id
  AND localizacao.rua_id = rua.id
  AND localizacao.cidade_id1 = cidade.id
  AND localizacao.CodPostal_id = codpostal.id
  AND localizacao.Pais_is = pais.id;
END; //
DELIMITER ;

-- Procurar Residência de Funcionário
DELIMITER //
CREATE PROCEDURE residencia_func (IN id_func INT)
BEGIN
        SELECT funcionario.id, funcionario.Nome, rua.designacao, cidade.designacao,
codpostal.Codigo, codpostal.Designacao, pais.designacao
  FROM funcionario, localizacao, rua, cidade, codpostal, pais
        WHERE funcionario.id=id_func
  AND funcionario.localizacao_id = localizacao.id
  AND localizacao.rua_id = rua.id
  AND localizacao.cidade_id1 = cidade.id
  AND localizacao.CodPostal_id = codpostal.id
  AND localizacao.Pais_is = pais.id;
```

```
END; //
DELIMITER ;

-- Procurar Localização de Minimercado
DELIMITER //
CREATE PROCEDURE residencia_merc (IN id_merc INT)
BEGIN
	SELECT minimercado.id, minimercado.Nome, rua.designacao, cidade.designacao,
codpostal.Codigo, codpostal.Designacao, pais.designacao
  FROM minimercado, localizacao, rua, cidade, codpostal, pais
	WHERE minimercado.id=id_merc
  AND minimercado.localizacao_id = localizacao.id
  AND localizacao.rua_id = rua.id
  AND localizacao.cidade_id1 = cidade.id
  AND localizacao.CodPostal_id = codpostal.id
  AND localizacao.Pais_is = pais.id;
END; //
DELIMITER ;

-- Procurar Faturas
select *from fatura;

-- Procurar Preços de Produtos
DELIMITER //
CREATE PROCEDURE preco_prod (IN id_prod INT)
BEGIN
	SELECT produto.preco
  FROM produto
	WHERE produto.id=id_prod;
END; //
DELIMITER ;

-- Total gasto por Cliente
DELIMITER //
create procedure gasto_por_cliente (in id_total INT)
begin
  select cliente.id, cliente.nome, sum(fatura.valor)
  from fatura,cliente
```

```
   where id_total = cliente.id
   and fatura.cliente_id = cliente.id;
end; //
DELIMITER ;

-- Procurar Compras feitas por Clientes
DELIMITER //
create procedure compras_por_cliente (IN id_cliente INT)
begin
            select cliente.nome, fatura.nr, produto.descricao, fatura_produto.quantidade
   from fatura, cliente, produto, fatura_produto
   where id_cliente = cliente.id
   and cliente.id = fatura.Cliente_id
   and fatura.nr = fatura_produto.fatura_nr
   and fatura_produto.produto_id = produto.id;
end; //
DELIMITER ;

-- Procurar faturas emitidas por funcionario
DELIMITER //
create procedure fatura_por_func (in id_func INT)
begin
   select fatura.Funcionario_id, fatura.nr, funcionario.nome
   from fatura,funcionario
   where id_func = funcionario.id
   and fatura.funcionario_id = funcionario.id;
end; //
DELIMITER ;
```