



Universidade do Minho

Mestrado Integrado em Engenharia Informática
Licenciatura em Ciências da Computação

Unidade Curricular de Bases de Dados

Ano Lectivo de 2020/2021

Quartel's Grill

Paulo Silva Sousa – A89465

João Figueiredo Martins Peixe dos Santos – A89520

Luís Filipe Cruz Sobral – A89474

Francisco Alves Andrade – A89513

Dezembro, 2020

BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

Quartel's Grill

Paulo Silva Sousa – A89465

João Figueiredo Martins Peixe dos Santos – A89520

Luís Filipe Cruz Sobral – A89474

Francisco Alves Andrade – A89513

Dezembro, 2020

Resumo

Este documento retrata de forma detalhada a organização e execução de uma arquitetura de bases de dados de uma cadeia de restaurantes de fast food, começando pela criação do seu modelo concetual e, posteriormente, passando à sua implementação modelar física.

Partindo da proposta da formulação de uma base de dados com um tema à nossa escolha surgiu a ideia de implementar o sistema de base de dados de uma cadeia de restaurantes de fast food, uma vez que se trata de um tema com o qual estamos familiarizados e com um nível de complexidade que consideramos adequado para o projeto em questão.

Numa parte inicial serão retratadas a contextualização e fundamentação da implementação da base de dados e a sua viabilidade.

Em seguida, apresentaremos os requisitos de controlo, descrição e exploração por nós levantados. Tendo em conta estes requisitos, apresentaremos o modelo concetual para a base de dados pretendida, onde identificaremos as entidades e os relacionamentos entre elas.

A parte seguinte diz respeito à modelação do modelo lógico da base de dados para o Quartel's Grill.

Por último, será apresentada a elaboração do modelo físico da base de dados. E traduzimos as interrogações do utilizador e as transações em código sql.

Área de Aplicação: Desenho e arquitetura de Sistemas de Base de Dados, no contexto de uma cadeia de restaurantes de fast food

Palavras-Chave: Bases de Dados, Bases de Dados Relacionais, Modelo Concetual, Modelo Lógico, Modelo Físico, MySQL, MySQL Workbench, brModelo, Entidades, Relacionamentos, Atributos, Requisitos, Índices SQL, Vistas SQL.

Índice

Resumo	1
Índice	1
Índice de Figuras	3
Índice de Tabelas	5
Introdução	1
Contexto de aplicação do sistema	1
Fundamentação da Implementação da Base de Dados	1
Análise da Viabilidade do Processo	2
Levantamento de Requisitos	3
Método de levantamento dos requisitos	3
Requisitos levantados	3
2.2.1. Requisitos de Descrição	3
2.2.2. Requisitos de Exploração	4
2.2.3. Requisitos de Controlo	4
Análise Geral de Requisitos	5
Modelo Concetual	6
Apresentação da abordagem de modelação realizada	6
Identificação e caracterização das entidades	7
Identificação e caracterização dos relacionamentos	8
Identificação e caracterização da associação dos atributos com as entidades e relacionamentos	11
Detalhe ou generalização de entidades	13
Apresentação e explicação do diagrama ER	15
Validação do modelo de dados com o utilizador	16
Modelo Lógico	17
Construção do Modelo Lógico	17
Desenho do Modelo	17
Validação do Modelo com interrogações do utilizador	18
Revisão do modelo lógico produzido	19
Modelo Físico	20
Seleção do sistema de gestão de bases de dados	20
Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL	20

Tradução das Interrogações do Utilizador	25
Escolha, definição e caracterização de índices em SQL (alguns exemplos)	28
Estimativa do espaço em disco da base de dados e taxa de crescimento anual	29
Definição e caracterização das vistas de utilização em SQL	31
Revisão do sistema implementado	32
Conclusões e Trabalho Futuro	33
Anexos	34
Anexo 1 - Povoamento da Base de Dados	34

Índice de Figuras

Figura 1 - Relacionamento Restaurante-Sede	8
Figura 2 - Relacionamento Relacionamento Restaurante-Funcionario	9
Figura 3 - Relacionamento Funcionario-Fatura	9
Figura 4 - Relacionamento Fatura-ItemMenu	10
Figura 5 - Relacionamento Restaurante-ItemMenu	10
Figura 6 - Relacionamento Restaurante-Fornecedor	11
Figura 7 - Diagrama ER	15
Figura 8 - Modelo Lógico	17
Figura 9 - Implementação Física da Tabela Sede	20
Figura 10 - Implementação Física da Tabela Restaurante	21
Figura 11 - Implementação Física da Tabela ItemMenu	21
Figura 12 - Implementação Física da Tabela Fornecedor	22
Figura 13 - Implementação Física da Tabela Funcionario	22
Figura 14 - Implementação Física da Tabela Fatura	23
Figura 15 - Implementação Física da Tabela Fatura_has_ItemMenu	23
Figura 16 - Implementação Física da Tabela Restaurante_has_Fornecedor	24
Figura 17 - Implementação Física da Tabela Restaurante_has_ItemMenu	24
Figura 18 - Procedimento referente à obtenção dos Funcionários que registaram mais faturas	25
Figura 19 - Procedimento referente à obtenção das Faturas de um Cliente	25
Figura 20 - Procedimento referente à obtenção dos Fornecedores que trabalham com um Restaurante	25
Figura 21 - Procedimento referente à obtenção da localização do Restaurante de um Funcionário	26
Figura 22 - Procedimento referente à obtenção dos Restaurantes que mais faturaram	26
Figura 23 - Procedimento referente à obtenção dos ItensMenu mais vendidos	26
Figura 24 - Procedimento referente à obtenção da receita total da cadeia de Restaurantes	27
Figura 25 - Procedimento referente à obtenção de quantas faturas foram emitidas até uma certa data	27

Figura 26 - Índice da Fatura	28
Figura 27 - Vista dos melhores 5 Funcionários por receita de Faturas	31
Figura 28 - Vista da idade média dos Funcionários	31
Figura 29 - Povoamento da Base de Dados (Parte 1)	34
Figura 30 - Povoamento da Base de Dados (Parte 2)	35
Figura 31 - Povoamento da Base de Dados (Parte 3)	36
Figura 32 - Povoamento da Base de Dados (Parte 4)	37

Índice de Tabelas

Tabela 1 - Caracterização das entidades	7
Tabela 2 - Relacionamentos entre Entidades	12
Tabela 3 - Estimativa do espaço em disco da base de dados	29
Tabela 4 - Estimativa do aumento do espaço anual da base de dados	30
Tabela 5 - Estimativa do espaço ocupado do ano 1 ao ano 5	30

1. Introdução

1.1. Contexto de aplicação do sistema

O desenvolvimento urbano das passadas décadas está intimamente ligado ao conceito de fast food, guarnições simples de cozinhar e, acima de tudo, rápidas. A informática tem um papel importantíssimo na organização e gestão de todo o processo de funcionamento de todos os estabelecimentos constituintes de uma cadeia de fast food. Tendo isto em mente, a prestigiada cadeia de restaurantes “Quartel’s Grill” decidiu recorrer aos nossos serviços de modo a elevar exponencialmente o nível do seu desempenho.

Esta cadeia é um dos mais recentes sucessos no contexto das cadeias de fast food e conta já com 5 estabelecimentos espalhados pela região norte e centro de Portugal que têm demonstrado um enorme sucesso e reconhecimento por parte dos seus consumidores.

Com a crescente evolução tecnológica a cadeia “Quartel’s Grill” pretende implementar um sistema que suporte todo o funcionamento do seu negócio e a consulta do histórico de vendas de cada restaurante.

1.2. Fundamentação da Implementação da Base de Dados

Com a humanidade a viver cada vez mais num ambiente stressante e de pressão surgiu a necessidade de restaurantes capazes de servir comida de forma extremamente rápida e barata de modo a satisfazer a necessidade de velocidade dos clientes. Os serviços disponibilizados pelo Quartel’s Grill registaram que a percentagem de portugueses que referem ter consumido refeições fast food num restaurante nos últimos 12 meses é cerca de 55.2% dos residentes no Continente.

Com o objetivo de possibilitar fast food “saudável”, e assim expandir o negócio, a empresa Quartel’s Grill pretende abrir vários restaurantes em várias localidades. Este passo representa um grande progresso no que toca à não nocividade da fast food, uma vez que com um maior alcance, qualidade e preços acessíveis, maior será o número de clientes a não consumir fast food nociva.

No entanto, com este aumento de restaurantes, o custo de manutenção da empresa será muito maior, pois terá mais funcionários, terá de adquirir mais produtos e as rendas serão maiores.

Com todo este aumento de espaço, stock e funcionários revela-se necessária a implementação de uma base de dados que seja capaz de lidar com toda esta informação da cadeia de restaurantes. Ou seja, uma base de dados capaz de controlar os vários restaurantes, o que implica a gestão dos funcionários, produtos adquiridos e as múltiplas interações com os clientes (venda de comida), e guardar informação sobre todos estes pontos. A sede da cadeia poderá, então, por exemplo, consultar a receita total, consultar faturas de um cliente, saber o restaurante que mais faturou e o item do menu mais vendido.

1.3. Análise da Viabilidade do Processo

O Quartel's Grill assume-se como uma entidade responsável e socialmente ativa. Como tal, não pode deixar que a sua obrigação para com os clientes e para com o mundo desapareça. No entanto, é, acima de tudo, uma empresa, e, por isso, todos os seus projetos têm de ser devidamente analisados e justificados, de modo a assegurar o seu futuro.

O nosso propósito foi implementar um Sistema de Base de Dados que suporte o funcionamento de uma cadeia de restaurantes, tendo em conta os recursos necessários e as vantagens financeiras que este sistema poderá trazer no futuro. Pretendemos que este mantenha a integridade face a todas as operações que possam ocorrer em simultâneo, bem como permitir a todo o staff e administração o acesso rápido e des congestionado a qualquer secção da base de dados e outros. Dada a natureza do problema também é expectável que a base dados possa revelar alguma informação como quais os itens do menu mais vendidos, o funcionário com mais vendas realizadas (através da faturação individual de cada um), qual a receita de cada restaurante da cadeia e qual faturou mais num determinado intervalo de tempo.

Analisando mais detalhadamente a viabilidade deste processo, podemos concluir que a implementação deste sistema irá trazer inegáveis vantagens no que toca à organização e gestão da cadeia Quartel's Grill, algo que irá certamente trazer vantagens económicas para esta empresa. Consideramos assim a construção deste sistema altamente viável.

2. Levantamento de Requisitos

2.1. Método de levantamento dos requisitos

Para levantar os requisitos, foi convocada uma reunião com a direção da cadeia de restaurantes Quartel's Grill. O objetivo principal desta reunião foi obtermos informações sobre o funcionamento dos restaurantes e esclarecer qualquer tipo de dúvidas.

2.2. Requisitos levantados

2.2.1. Requisitos de Descrição

Sede

A sede dos restaurantes é responsável pela organização da cadeia. Os dados guardados sobre a sede serão os contactos (número de telefone e email), a morada (código postal, rua e localidade) e o NIF. Ainda, à sede está associado o nome do diretor da cadeia de restaurantes, responsável pela gestão desta.

Restaurante

O restaurante é o constituinte da cadeia, sendo assim, é o elemento central desta. Como tal, este elemento está associado a vários fornecedores, funcionários e itens de menu. Os dados associados a cada restaurante, para além dos já referidos, são os contactos (número de telefone e email do restaurante), o nome do gerente do restaurante, a capacidade (quantidade de pessoas que o restaurante suporta), o NIF e a morada (código postal, rua e localidade).

Fornecedor

O fornecedor é o responsável pelo fornecimento dos materiais necessários à produção da comida, estando associado, como foi dito anteriormente, ao restaurante e, ainda, associado aos itens do menu. Os dados guardados para o fornecedor são o NIF, o tipo (peixe, carne, fruta, etc.), os contactos (número de telefone e email) e o nome do fornecedor.

Funcionário

O funcionário é o responsável por manter o funcionamento do restaurante, estando associado a este mesmo e à fatura. A informação guardada para cada funcionário passa pelo seu número, nome, contactos (número de telefone e email), data de nascimento, função, género e nacionalidade.

ItemMenu

O itemMenu representa os pratos presentes no menu de um restaurante e é obtido através do fornecedor. Os dados guardados para o itemMenu são o nome, categoria (carne, peixe, vegetariano), descrição (acompanhamento ou constituição), preço e tipo (entrada, prato principal, sobremesa).

Fatura

A fatura é a representação das transações efetuadas entre o cliente e o restaurante e é emitida por um funcionário. Assim, os dados guardados para a fatura são o número da fatura, o preço total do consumido sem IVA, o IVA, a data de emissão e a forma de pagamento. Ainda, é opcional ao cliente facultar o seu nome e NIF, podendo estar estes presentes ou não na fatura.

2.2.2. Requisitos de Exploração

- Saber os funcionários que registaram mais faturas
- Dado o NIF de um Cliente, saber quais as suas faturas
- Dado o NIF de um Restaurante, saber com que fornecedores é que trabalham
- Dado o Número de um Funcionário, saber a localidade e o NIF do restaurante em que trabalham
- Saber os Restaurantes que faturaram mais
- Saber quais os Itens do Menu mais vendidos por quantidade
- Saber qual a receita total da cadeia de restaurantes
- Saber quantas faturas foram emitidas até uma certa data

2.2.3. Requisitos de Controlo

- O utilizador pode adicionar um novo funcionário
- O utilizador pode adicionar um novo fornecedor
- O utilizador pode adicionar um novo restaurante
- O utilizador pode adicionar um novo item no menu
- O utilizador pode adicionar uma nova fatura
- O utilizador pode alterar as informações de um funcionário
- O utilizador pode alterar as informações de um item no menu
- O utilizador pode remover um funcionário
- O utilizador pode remover um item do menu

2.3. Análise Geral de Requisitos

Em conclusão, a nossa aplicação apenas é usada por parte do utilizador (Sede do Restaurante Quatel's Grill). Com esta ele consegue:

- Aceder a várias informações sobre os Restaurantes, Fornecedores, Faturas e Itens do Menu.
- Acrescentar novos Restaurantes, Fornecedores, Faturas e Itens de Menu.

3. Modelo Concetual

3.1. Apresentação da abordagem de modelação realizada

O modelo concetual representa a base de dados de uma forma independente de detalhes físicos ou associadas à implementação. Após a análise de requisitos é formulado este modelo tendo em conta critérios não estabelecidos.

A nossa abordagem concetual teve como foco as vendas e faturação dos vários restaurantes que constituem a cadeia, de forma a que as entidades e os relacionamentos delineados contribuam para a criação de um sistema que auxilie e viabilize a reprodução virtual dessa atividade.

3.2. Identificação e caracterização das entidades

De modo a desenvolver um modelo concetual há que, inicialmente, identificar as entidades necessárias.

Abaixo encontra-se essa identificação relativa às entidades que decidimos modelar.

Entidade	Descrição	Ocorrência
Sede	Termo geral responsável pela organização da cadeia.	Consulta e gere as receitas dos restaurantes constituintes.
Restaurante	Termo geral central da cadeia associado a todo o tipo de dados (clientes, menu, fornecedores, contactos).	Está associado a vários fornecedores, funcionários e itens de menu.
Fornecedor	Termo geral responsável pelo fornecimento dos materiais necessários à produção.	Fornece os itens do menu aos restaurantes.
Funcionário	Termo geral responsável por manter o funcionamento do restaurante.	Trabalha num restaurante e regista as faturas.
Item menu	Termo geral que representa os pratos presentes no menu de um restaurante.	Indica as características de cada elemento do cardápio: categoria (vegetariano, peixe ou carne), tipo (sobremesa, prato principal...).
Fatura	Termo geral que representa as transações efetuadas entre os clientes e o restaurante.	Engloba o preço, forma de pagamento e, opcionalmente, o nome e NIF do cliente.

Tabela 1 - Caracterização das entidades

3.3. Identificação e caracterização dos relacionamentos

Após a modelação conceptual, realizou-se a identificação dos relacionamentos existentes entre as entidades anteriormente referidas. Tendo por base os requisitos registados para os estabelecer, listamos os seguintes relacionamentos:

Sede - Restaurante:

O relacionamento “Tem” entre as entidades Restaurante e Funcionário representa a possibilidade de uma sede ter N restaurantes.

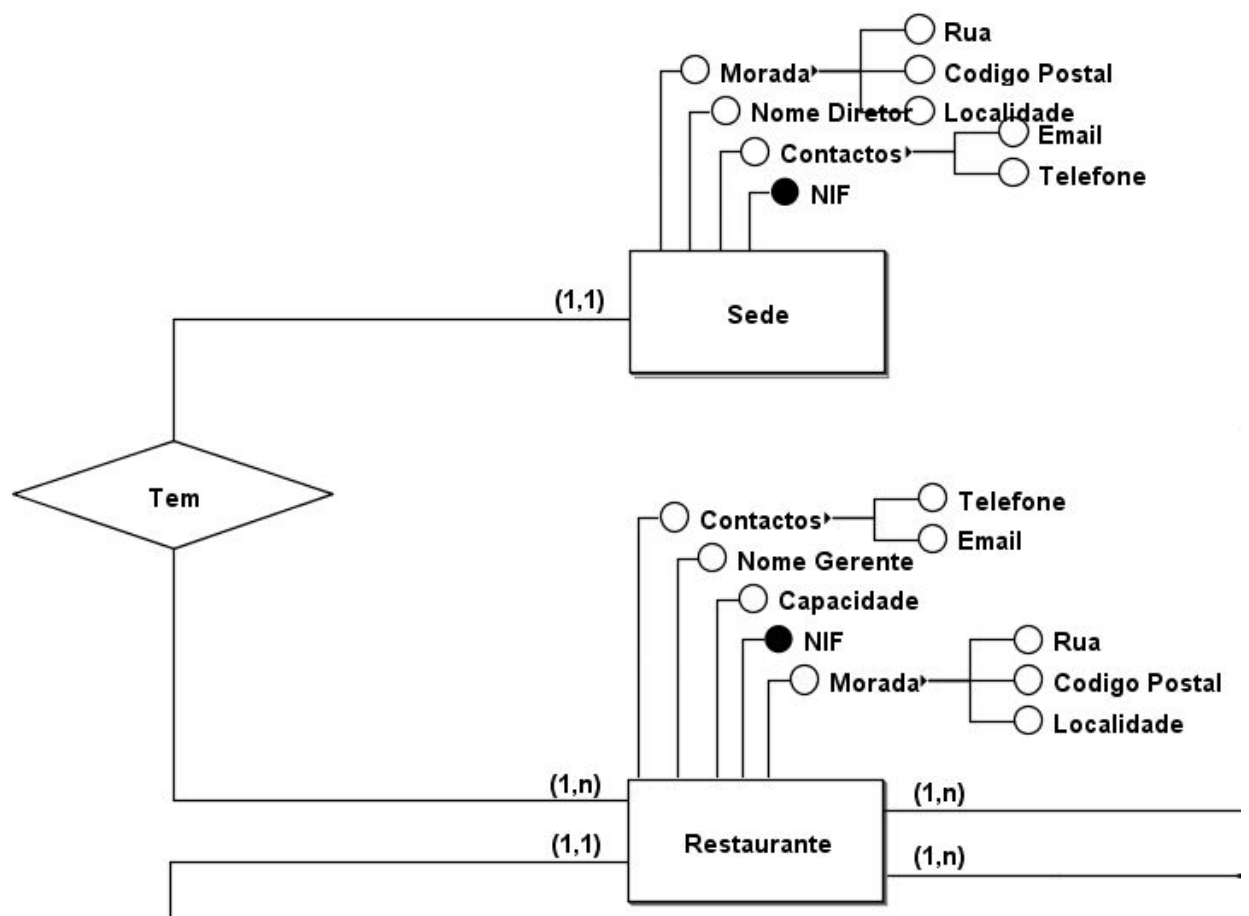


Figura 1 - Relacionamento Sede-Restaurante

Restaurante-Funcionário:

O relacionamento “Emprega” entre as entidades Restaurante e Funcionário representa a possibilidade de um restaurante empregar N funcionários.

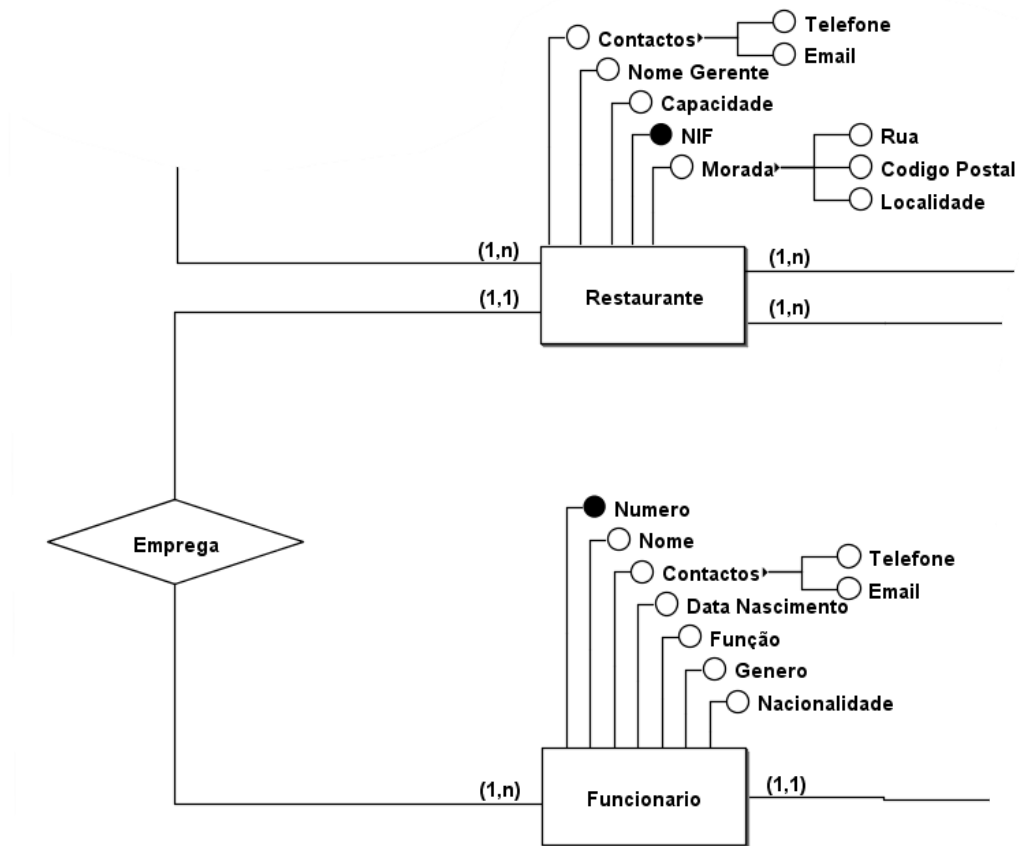


Figura 2 - Relacionamento Restaurante-Funcionario

Funcionário-Fatura:

O relacionamento “Emite” entre as entidades Funcionário e Fatura representa a possibilidade de um funcionário emitir N faturas.

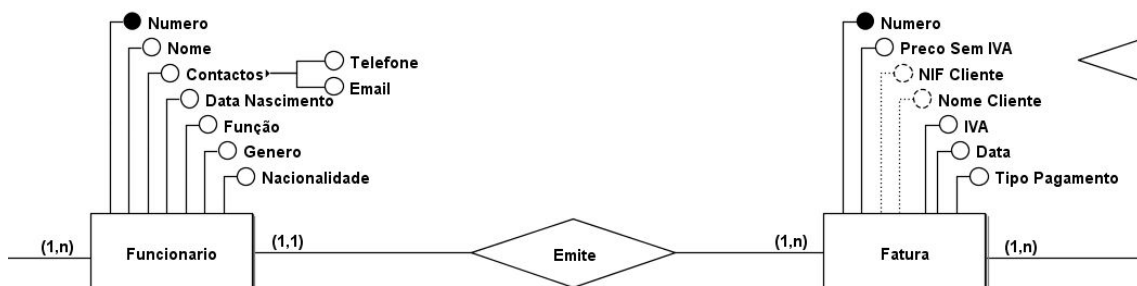


Figura 3 - Relacionamento Funcionario-Fatura

Fatura-ItemMenu:

O relacionamento “Tem” entre as entidades Fatura e ItemMenu representa a possibilidade de um ItemMenu ter N faturas e vice-versa (relação n para n).

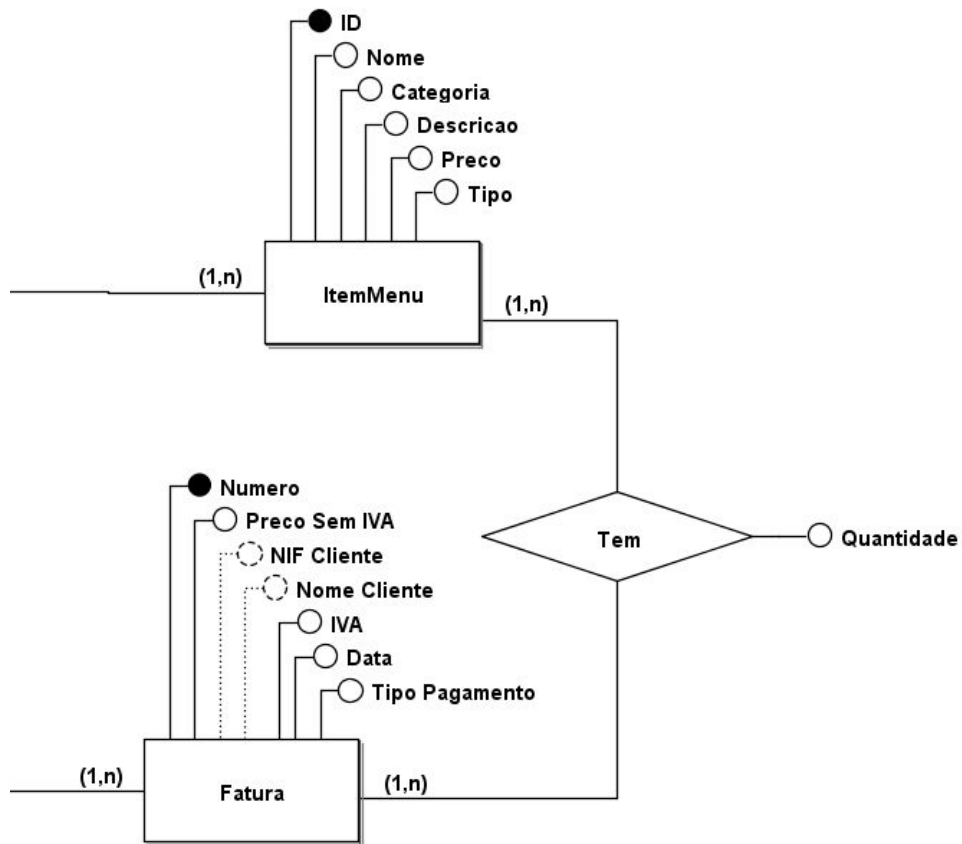


Figura 4 - Relacionamento Fatura-ItemMenu

Restaurante-ItemMenu:

O relacionamento “Tem” entre as entidades Restaurante e ItemMenu representa a possibilidade de um Restaurante ter N ItemMenu e vice-versa (relação n para n).

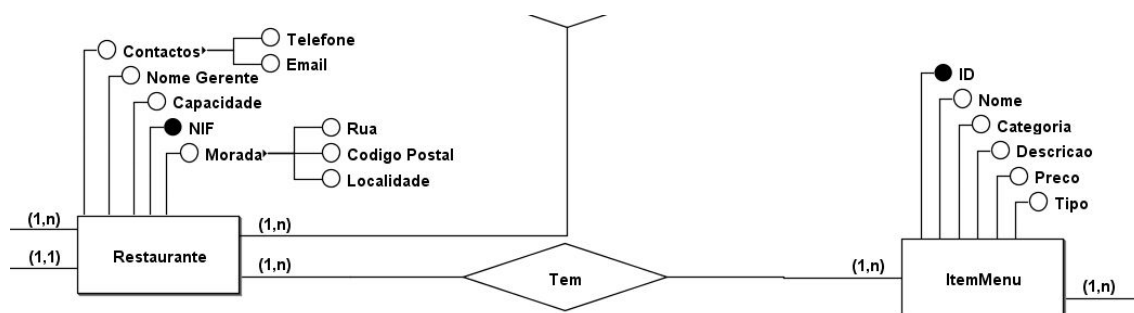


Figura 5 - Relacionamento Restaurante-ItemMenu

Restaurante-Fornecedor:

O relacionamento “Tem” entre as entidades Restaurante e Fornecedor representa a possibilidade de um restaurante ter N fornecedores e vice-versa (relação n para n).

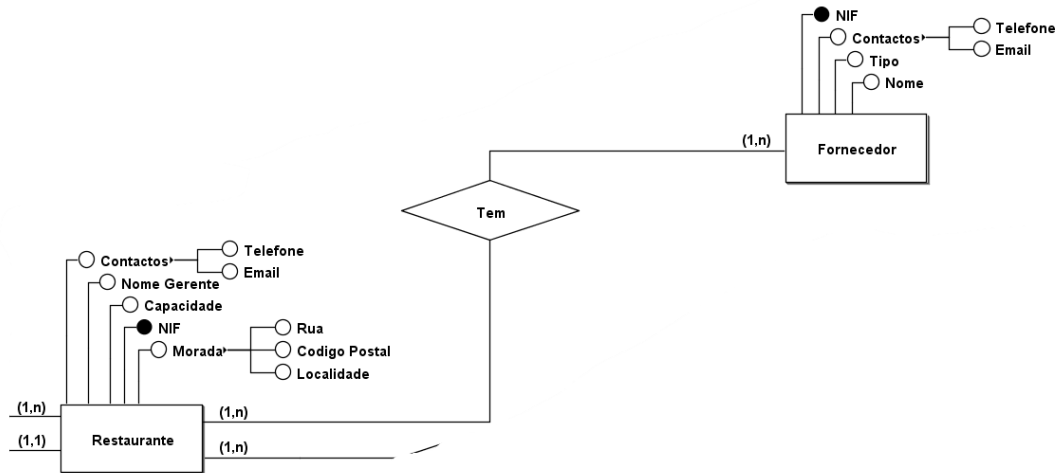


Figura 6 - Relacionamento Restaurante-Fornecedor

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

Tal como aconteceu com a representação das entidades na base de dados, é também indispensável a definição dos atributos que estarão presentes em cada uma delas, ou nas relações.

- Uma **Sede** tem como chave primária o seu NIF (número de identificação fiscal). Tem como atributo simples o nome do diretor. Como atributos compostos tem: a morada, composta por rua, localidade e código postal; e os contactos que é composto por número de telefone e email.
- O **Restaurante** tem como chave primária o seu NIF. É caracterizado por capacidade (número de clientes que consegue suportar) e nome do gerente como atributos simples. Ainda, como atributos compostos tem a morada (rua, localidade e código postal) e os contactos (número de telefone e email).
- O **Funcionário** tem como chave principal o seu número de funcionário. Os atributos simples que compõem esta entidade são o nome, a data de nascimento, a função que desempenha no restaurante, o género e a nacionalidade. Tem como atributo composto os contactos, composto por número de telefone e email.

- A **Fatura** tem como chave principal o número de emissão da fatura. É constituída por vários atributos simples, sendo estes: preço sem IVA, data e tipo de pagamento. É também caracterizada por dois atributos opcionais: NIF do cliente e nome do cliente, podendo este aquando da emissão da fatura escolher se o seu nome e NIF irão constar na mesma.
- O **ItemMenu** tem como chave principal o seu ID, que é um identificador. É caracterizado apenas por atributos simples, sendo estes: nome, categoria(carne, peixe, vegetariano, ...), descrição do item, preço e tipo (prato principal e sobremesa).
- O **Fornecedor** tem como chave primária o seu NIF. Tem como atributos simples o tipo (fornecedor de carne, fornecedor de peixe, pastelaria, ...) e o nome. Ainda, tem como atributo composto os seus contactos (número de telefone e email).

Nome da Entidade	Multiplicidade	Relação	Atributos	Descrição	Multiplicidade	Nome da Entidade
Sede	(1,1)	Tem	-	-	(1,n)	Restaurant e
Restaurante	(1,1)	Emprega	-	-	(1, n)	Funcionário
Restaurante	(1,n)	Tem	-	-	(1, n)	ItemMenu
Restaurante	(1,n)	Tem	-	-	(1,n)	Fornecedor
Funcionário	(1,1)	Emite	-	-	(1,n)	Fatura
Fatura	(1,n)	Tem	Quantidade	Quantidade do item do menu presente na fatura	(1,n)	ItemMenu

Tabela 2 - Relacionamentos entre Entidades

3.5. Detalhe ou generalização de entidades

Abaixo apresentaremos uma representação sucinta das entidades presentes na nossa base de dados, juntamente com o seus atributos e o domínio de valores em que ocorrem.

Sede:

- *Morada (Rua, Código Postal, Localidade)*
Varchar(75), Varchar(20), Varchar(50)
- *Nome Diretor*
Varchar(50)
- *Contactos (Email, Telefone)*
Varchar(100), Varchar(20)
- *NIF*
Int

Restaurante:

- *Contactos (Email, Telefone)*
Varchar(100), Varchar(20)
- *NIF*
Int
- *Morada (Rua, Código Postal, Localidade)*
Varchar(75), Varchar(20), Varchar(50)
- *Nome Gerente*
Varchar(50)
- *Capacidade*
Int

Fornecedor:

- *NIF*
Int
- *Contactos (Email, Telefone)*
Varchar(100), Varchar(20)
- *Tipo*
Varchar(20)
- *Nome*
Varchar(50)

Funcionário:

- *Contactos (Email, Telefone)*
Varchar(100), Varchar(20)
- *Número*
Int
- *Nome*
Varchar(50)
- *Data de Nascimento*
Date
- *Função*
Varchar(20)
- *Género*
Varchar(45)
- *Nacionalidade*
Varchar(45)

ItemMenu:

- *Número*
Int
- *Nome*
Varchar(50)
- *Categoria*
Varchar(20)
- *Descrição*
Varchar(200)
- *Preço*
Decimal(6,2)
- *Tipo*
Varchar(30)

Fatura:

- *Número*
Int
- *Preço sem IVA*
Decimal(6,2)
- *NIF cliente (opcional)*
Int
- *Nome Cliente (opcional)*
Varchar(50)

- *IVA*
Decimal(3,2)
- *Tipo Pagamento*
Varchar(45)
- *Data*
DateTime

3.6. Apresentação e explicação do diagrama ER

Após a definição das entidades, relacionamentos e atributos, formulamos o seguinte diagrama ER:

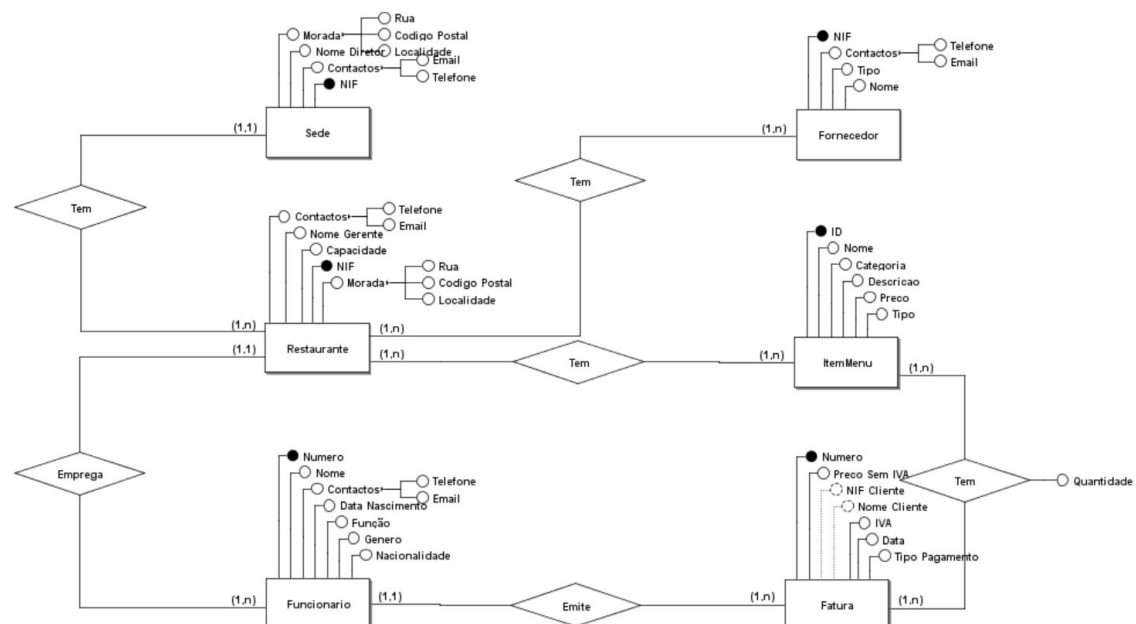


Figura 7 - Diagrama ER

3.7. Validação do modelo de dados com o utilizador

Após o término da modelação concetual da base de dados foi necessária a validação do modelo de dados com o utilizador.

Para isso, foi realizada uma reunião com o restaurante da cadeia Quartel's Grill onde procuramos explicar minuciosamente, de forma que seja possível satisfazer tudo aquilo que tinha sido previamente requisitado pelo cliente.

No fim de todo este processo o modelo foi validado e ficou acordado por ambas as partes passar à sua fase seguinte, a Modelação Lógica.

4. Modelo Lógico

4.1. Construção do Modelo Lógico

Para levantar os requisitos, foi convocada uma reunião com a direção da cadeia de restaurantes Quartel's Grill. O objetivo principal desta reunião foi obtermos informações sobre o funcionamento dos restaurantes e esclarecer qualquer tipo de dúvidas.

4.2. Desenho do Modelo

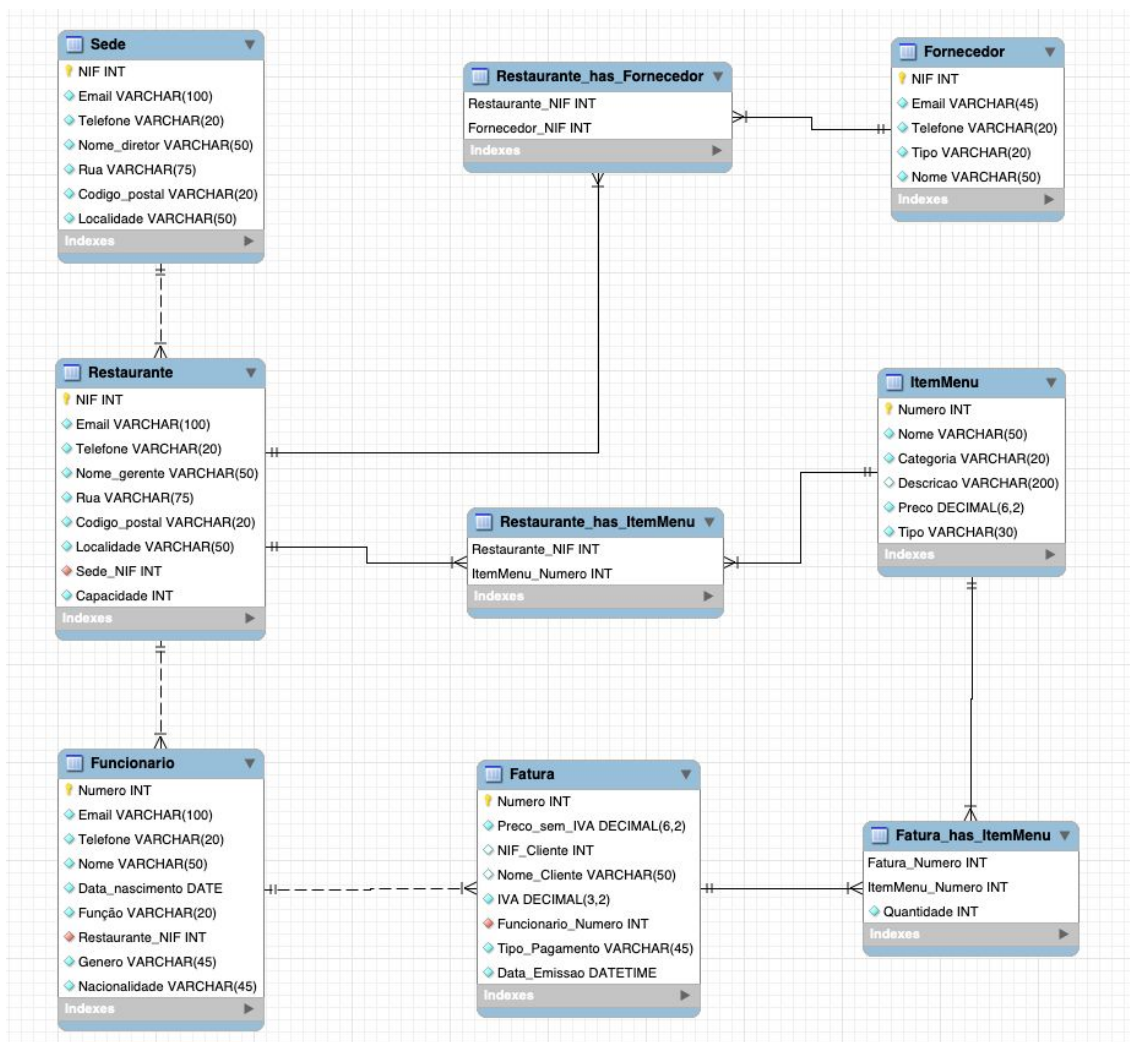


Figura 8 - Modelo Lógico

4.3. Validação do Modelo com interrogações do utilizador

Consultar lista de funcionários que registaram mais facturas

É feita uma junção entre a tabela do funcionário e a da fatura. Através do número de funcionário associado a cada fatura é calculado o número total de faturas registadas por cada funcionário.

Consultar faturas de um cliente

É necessário seleccionar na tabela da fatura aquelas cujo NIF corresponde ao do cliente pretendido.

Consultar fornecedores de um restaurante

É necessário consultar a tabela dos fornecedores e de seguida seleccionar os que têm uma ligação com o restaurante, utilizando o NIF fornecido.

Consultar localidade e nif do restaurante de um certo funcionário

É feita uma junção entre a tabela do funcionário e a do restaurante. Assim, é possível consultar o NIF e a localidade do restaurante utilizando, para isso, o NIF associado ao funcionário pretendido.

Consultar restaurante que mais faturou

É feita uma junção da tabela da fatura e do funcionário (correspondência entre funcionários e faturas registadas por eles).

É feita uma junção da tabela do funcionário e do restaurante (correspondência entre funcionário e restaurante onde trabalha). Com estas ligações é possível associar um preço de uma fatura a um funcionário e, de seguida, a um restaurante. Assim, conseguimos calcular o restaurante que mais faturou.

Consultar itens do menu mais vendidos

É necessário seleccionar a tabela do ItemMenu e calcular a quantidade de cada um, através da ligação à tabela da fatura.

Consultar receita total da cadeia de restaurantes

É necessário seleccionar a tabela das faturas e adicionar o preço de cada fatura, calculando, desta forma, o preço com IVA.

Consultar quantas faturas foram emitidas até uma data

Inicialmente seleciona-se a tabela da fatura e, de seguida, é comparada a data da fatura com a data dada. Por fim é necessário selecionar as faturas que correspondem à restrição pedida.

4.4. Revisão do modelo lógico produzido

Após a finalização do modelo lógico este necessita de uma revisão, tendo esta o objetivo de garantir que a estrutura criada satisfaz todos os requisitos pretendidos e desejados. Foi efetuada uma nova reunião, onde foi apresentada à administração da cadeia Quartel's Grill a estrutura desenvolvida. Tendo em conta o feedback positivo, o modelo lógico foi aprovado, uma vez que cumpria todos os requisitos propostos.

5. Modelo Físico

5.1. Seleção do sistema de gestão de bases de dados

Nesta parte do relatório falamos da conversão do Modelo Lógico para o Modelo Físico.

O Sistema de Bases de Dados foi escolhido pelo grupo, uma vez que o utilizador não tinha conhecimentos suficientes para realizar essa seleção. Assim, o grupo optou por utilizar o sistema com que estava mais confortável de ter trabalhado nas aulas, o MySQL Workbench.

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

De modo a traduzir o Modelo Lógico para o modelo físico, decidimos utilizar uma ferramenta bastante simples disponibilizada pelo MySQL Workbench chamada “Forward Engineer...”

- Criação da Tabela Sede

```
CREATE TABLE IF NOT EXISTS `Restaurante`.`Sede` (  
  `NIF` INT NOT NULL,  
  `Email` VARCHAR(100) NOT NULL,  
  `Telefone` VARCHAR(20) NOT NULL,  
  `Nome_diretor` VARCHAR(50) NOT NULL,  
  `Rua` VARCHAR(75) NOT NULL,  
  `Codigo_postal` VARCHAR(20) NOT NULL,  
  `Localidade` VARCHAR(50) NOT NULL,  
  PRIMARY KEY (`NIF`),  
  UNIQUE INDEX `Email_UNIQUE` (`Email` ASC) VISIBLE,  
  UNIQUE INDEX `Telefone_UNIQUE` (`Telefone` ASC) VISIBLE)  
ENGINE = InnoDB;
```

Figura 9 - Implementação Física da Tabela Sede

- Criação da Tabela Restaurante

```
CREATE TABLE IF NOT EXISTS `Restaurante`.`Restaurante` (  
  `NIF` INT NOT NULL,  
  `Email` VARCHAR(100) NOT NULL,  
  `Telefone` VARCHAR(20) NOT NULL,  
  `Nome_gerente` VARCHAR(50) NOT NULL,  
  `Rua` VARCHAR(75) NOT NULL,  
  `Codigo_postal` VARCHAR(20) NOT NULL,  
  `Localidade` VARCHAR(50) NOT NULL,  
  `Sede_NIF` INT NOT NULL,  
  `Capacidade` INT NOT NULL,  
  PRIMARY KEY (`NIF`),  
  INDEX `fk_Restaurante_Sede_idx` (`Sede_NIF` ASC) VISIBLE,  
  CONSTRAINT `fk_Restaurante_Sede`  
    FOREIGN KEY (`Sede_NIF`)  
      REFERENCES `Restaurante`.`Sede` (`NIF`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

Figura 10 - Implementação Física da Tabela Restaurante

- Criação da Tabela ItemMenu

```
CREATE TABLE IF NOT EXISTS `Restaurante`.`ItemMenu` (  
  `Numero` INT NOT NULL,  
  `Nome` VARCHAR(50) NOT NULL,  
  `Categoria` VARCHAR(20) NOT NULL,  
  `Descricao` VARCHAR(200) NULL,  
  `Preco` DECIMAL(6,2) NOT NULL,  
  `Tipo` VARCHAR(30) NOT NULL,  
  PRIMARY KEY (`Numero`))  
ENGINE = InnoDB;
```

Figura 11 - Implementação Física da Tabela ItemMenu

- Criação da Tabela Fornecedor

```
CREATE TABLE IF NOT EXISTS `Restaurante`.`Fornecedor` (  
  `NIF` INT NOT NULL,  
  `Email` VARCHAR(45) NOT NULL,  
  `Telefone` VARCHAR(20) NOT NULL,  
  `Tipo` VARCHAR(20) NOT NULL,  
  `Nome` VARCHAR(50) NOT NULL,  
  PRIMARY KEY (`NIF`))  
ENGINE = InnoDB;
```

Figura 12 - Implementação Física da Tabela Fornecedor

- Criação da Tabela Funcionário

```
CREATE TABLE IF NOT EXISTS `Restaurante`.`Funcionario` (  
  `Numero` INT NOT NULL,  
  `Email` VARCHAR(100) NOT NULL,  
  `Telefone` VARCHAR(20) NOT NULL,  
  `Nome` VARCHAR(50) NOT NULL,  
  `Data_nascimento` DATE NOT NULL,  
  `Função` VARCHAR(20) NOT NULL,  
  `Restaurante_NIF` INT NOT NULL,  
  `Genero` VARCHAR(45) NOT NULL,  
  `Nacionalidade` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`Numero`),  
  INDEX `fk_Funcionario_Restaurante1_idx` (`Restaurante_NIF` ASC) VISIBLE,  
  CONSTRAINT `fk_Funcionario_Restaurante1`  
    FOREIGN KEY (`Restaurante_NIF`)  
    REFERENCES `Restaurante`.`Restaurante` (`NIF`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

Figura 13 - Implementação Física da Tabela Funcionario

- Criação da Tabela Fatura

```
CREATE TABLE IF NOT EXISTS `Restaurante`.`Fatura` (
  `Numero` INT NOT NULL,
  `Preco_sem_IVA` DECIMAL(6,2) NOT NULL,
  `NIF_Cliente` INT NULL,
  `Nome_Cliente` VARCHAR(50) NULL,
  `IVA` DECIMAL(3,2) NOT NULL,
  `Funcionario_Numero` INT NOT NULL,
  `Tipo_Pagamento` VARCHAR(45) NOT NULL,
  `Data_Emissao` DATETIME NOT NULL,
  PRIMARY KEY (`Numero`),
  INDEX `fk_Fatura_Funcionario1_idx` (`Funcionario_Numero` ASC) VISIBLE,
  UNIQUE INDEX `NIF_Cliente_UNIQUE` (`NIF_Cliente` ASC) VISIBLE,
  CONSTRAINT `fk_Fatura_Funcionario1`
    FOREIGN KEY (`Funcionario_Numero`)
    REFERENCES `Restaurante`.`Funcionario` (`Numero`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

Figura 14 - Implementação Física da Tabela Fatura

- Criação da Tabela Fatura_has_ItemMenu

```
CREATE TABLE IF NOT EXISTS `Restaurante`.`Fatura_has_ItemMenu` (
  `Fatura_Numero` INT NOT NULL,
  `ItemMenu_Numero` INT NOT NULL,
  `Quantidade` INT NOT NULL,
  PRIMARY KEY (`Fatura_Numero`, `ItemMenu_Numero`),
  INDEX `fk_Fatura_has_ItemMenu_ItemMenu1_idx` (`ItemMenu_Numero` ASC) VISIBLE,
  INDEX `fk_Fatura_has_ItemMenu_Fatura1_idx` (`Fatura_Numero` ASC) VISIBLE,
  CONSTRAINT `fk_Fatura_has_ItemMenu_Fatura1`
    FOREIGN KEY (`Fatura_Numero`)
    REFERENCES `Restaurante`.`Fatura` (`Numero`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Fatura_has_ItemMenu_ItemMenu1`
    FOREIGN KEY (`ItemMenu_Numero`)
    REFERENCES `Restaurante`.`ItemMenu` (`Numero`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

Figura 15 - Implementação Física da Tabela Fatura_has_ItemMenu

- Criação da Tabela Restaurante_has_Fornecedor

```
CREATE TABLE IF NOT EXISTS `Restaurante`.`Restaurante_has_Fornecedor` (
  `Restaurante_NIF` INT NOT NULL,
  `Fornecedor_NIF` INT NOT NULL,
  PRIMARY KEY (`Restaurante_NIF`, `Fornecedor_NIF`),
  INDEX `fk_Restaurante_has_Fornecedor_Fornecedor1_idx` (`Fornecedor_NIF` ASC) VISIBLE,
  INDEX `fk_Restaurante_has_Fornecedor_Restaurante1_idx` (`Restaurante_NIF` ASC) VISIBLE,
  CONSTRAINT `fk_Restaurante_has_Fornecedor_Restaurante1`
    FOREIGN KEY (`Restaurante_NIF`)
      REFERENCES `Restaurante`.`Restaurante` (`NIF`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_Restaurante_has_Fornecedor_Fornecedor1`
    FOREIGN KEY (`Fornecedor_NIF`)
      REFERENCES `Restaurante`.`Fornecedor` (`NIF`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

Figura 16 - Implementação Física da Tabela Restaurante_has_Fornecedor

- Criação da Tabela Restaurante_has_ItemMenu

```
CREATE TABLE IF NOT EXISTS `Restaurante`.`Restaurante_has_ItemMenu` (
  `Restaurante_NIF` INT NOT NULL,
  `ItemMenu_Numero` INT NOT NULL,
  PRIMARY KEY (`Restaurante_NIF`, `ItemMenu_Numero`),
  INDEX `fk_Restaurante_has_ItemMenu_ItemMenu1_idx` (`ItemMenu_Numero` ASC) VISIBLE,
  INDEX `fk_Restaurante_has_ItemMenu_Restaurante1_idx` (`Restaurante_NIF` ASC) VISIBLE,
  CONSTRAINT `fk_Restaurante_has_ItemMenu_Restaurante1`
    FOREIGN KEY (`Restaurante_NIF`)
      REFERENCES `Restaurante`.`Restaurante` (`NIF`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_Restaurante_has_ItemMenu_ItemMenu1`
    FOREIGN KEY (`ItemMenu_Numero`)
      REFERENCES `Restaurante`.`ItemMenu` (`Numero`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

Figura 17 - Implementação Física da Tabela Restaurante_has_ItemMenu

5.3. Tradução das Interrogações do Utilizador

Do Ponto de Vista do Utilizador (Sede Quartel's Grill)

- Saber os funcionários que registaram mais faturas

```
DELIMITER //
```

```
CREATE PROCEDURE funcionario_mais_faturas ()
```

```
  BEGIN
```

```
    SELECT f.Nome, COUNT(fat.Funcionario_Numero) AS Faturas
```

```
    FROM Funcionario f, Fatura fat
```

```
    WHERE fat.Funcionario_Numero = f.Numero
```

```
    GROUP BY fat.Funcionario_Numero
```

```
    ORDER BY Faturas DESC;
```

```
  END //
```

```
DELIMITER //
```

Figura 18 - Procedimento referente à obtenção dos Funcionários que registaram mais faturas

- Dado o NIF de um Cliente, saber quais as suas faturas

```
DELIMITER //
```

```
CREATE PROCEDURE faturas_Cliente_NIF (IN NIF INT)
```

```
  BEGIN
```

```
    SELECT * FROM fatura
```

```
    WHERE fatura.NIF_Cliente = NIF;
```

```
  END //
```

```
DELIMITER //
```

Figura 19 - Procedimento referente à obtenção das Faturas de um Cliente

- Dado o NIF de um Restaurante, saber com que fornecedores é que trabalham

```
DELIMITER //
```

```
CREATE PROCEDURE fornecedores_restaurante (IN Restaurante_NIF INT)
```

```
  BEGIN
```

```
    SELECT f.Nome from Fornecedor f, Restaurante_has_Fornecedor rf
```

```
    WHERE rf.Fornecedor_NIF = f.NIF
```

```
    AND rf.Restaurante_NIF = Restaurante_NIF;
```

```
  END //
```

```
DELIMITER //
```

Figura 20 - Procedimento referente à obtenção dos Fornecedoros que trabalham com um Restaurante

- Dado o Número de um Funcionário, saber a localidade e o NIF do restaurante em que trabalham

```
DELIMITER //
CREATE PROCEDURE restaurante_funcionario (IN Funcionario_Numero INT)
BEGIN
    SELECT r.Localidade, r.NIF from Restaurante r, Funcionario f
    WHERE f.Numero = Funcionario_Numero
    AND f.Restaurante_NIF = r.NIF;
END //
DELIMITER //
```

Figura 21 - Procedimento referente à obtenção da localização do Restaurante de um Funcionário

- Saber os Restaurantes que mais faturaram

```
DELIMITER //
CREATE PROCEDURE restauranteMaisFaturou ()
BEGIN
    SELECT r.Localidade, SUM(fat.Preco_sem_IVA) AS receita
    FROM restaurante r, funcionario f, fatura fat
    WHERE fat.Funcionario_Numero = f.Numero AND f.Restaurante_NIF = r.NIF
    GROUP BY r.Localidade
    ORDER BY receita DESC;
END //
DELIMITER //
```

Figura 22 - Procedimento referente à obtenção dos Restaurantes que mais faturaram

- Saber quais os Itens do Menu mais vendidos por quantidade

```
DELIMITER //
CREATE PROCEDURE itens_mais_vendidos ()
BEGIN
    SELECT i.Nome, SUM(fi.quantidade) AS Quantidade
    FROM ItemMenu i, Fatura_has_ItemMenu fi
    WHERE i.Numero = fi.ItemMenu_Numero
    GROUP BY fi.ItemMenu_Numero
    ORDER BY Quantidade DESC;
END //
DELIMITER //
```

Figura 23 - Procedimento referente à obtenção dos ItensMenu mais vendidos

- Saber qual a receita total da cadeia de restaurantes

```
DELIMITER //
```

```
CREATE PROCEDURE receitaTotalCadeia ()
```

```
  BEGIN
```

```
    SELECT SUM(Preco_sem_Iva * (1 + IVA))
```

```
    FROM fatura;
```

```
  END //
```

```
DELIMITER //
```

Figura 24 - Procedimento referente à obtenção da receita total da cadeia de Restaurantes

- Saber quantas faturas foram emitidas até uma certa data

```
DELIMITER //
```

```
CREATE PROCEDURE venda_periodo (IN datamax DATE)
```

```
  BEGIN
```

```
    SELECT * FROM Fatura
```

```
    WHERE data_emissao < datamax;
```

```
  END //
```

```
DELIMITER //
```

Figura 25 - Procedimento referente à obtenção de quantas faturas foram emitidas até uma certa data

5.4. Escolha, definição e caracterização de índices em SQL (alguns exemplos)

Os índices são uma funcionalidade do MySQL que permite a ordenação dos dados de uma tabela segundo atributos que não a sua chave primária. Por exemplo, numa lista de correios eletrónicos com milhões de exemplares, será fácil encontrar um deles pelo endereço (admitindo que este é a chave primária). No entanto, se procurarmos pelo nome associado ao mesmo, deparar-nos-emos com um desafio maior, para além de possivelmente colocar em causa a sua eficiência visto que a tabela em causa não está organizada segundo esse atributo.

Os índices implementam estruturas de dados, como Árvores Binárias, que organizam as entradas da tabela segundo um determinado atributo. Essas estruturas são depois utilizadas por um sistema de optimização nativo do MySQL, tornando as queries mais eficientes.

Neste trabalho, usamos esta funcionalidade na tabela da fatura, indexando as entradas pelo nif de cada cliente tornando mais eficiente procurar as faturas referentes a cada cliente.

Relativamente aos índices é importante referir que o facto de estes serem gerados automaticamente para chaves estrangeiras é bastante útil, uma vez que, procuramos várias vezes as faturas pelo número de funcionário.

```
CREATE TABLE IF NOT EXISTS `Restaurante`.`Fatura` (  
  `Numero` INT NOT NULL,  
  `Preco_sem_IVA` DECIMAL(6,2) NOT NULL,  
  `NIF_Cliente` INT NULL,  
  `Nome_Cliente` VARCHAR(50) NULL,  
  `IVA` DECIMAL(3,2) NOT NULL,  
  `Funcionario_Numero` INT NOT NULL,  
  `Tipo_Pagamento` VARCHAR(45) NOT NULL,  
  `Data_Emissao` DATETIME NOT NULL,  
  PRIMARY KEY (`Numero`),  
  INDEX `fk_Fatura_Funcionario1_idx` (`Funcionario_Numero` ASC) VISIBLE,  
  UNIQUE INDEX `NIF_Cliente_UNIQUE` (`NIF_Cliente` ASC) VISIBLE,  
  CONSTRAINT `fk_Fatura_Funcionario1`  
    FOREIGN KEY (`Funcionario_Numero`)  
    REFERENCES `Restaurante`.`Funcionario` (`Numero`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

Figura 26 - Índice da Fatura

5.5. Estimativa do espaço em disco da base de dados e taxa de crescimento anual

Entidade	Atributo	Tipo de Dados	Tamanho	Ocorrencias	Subtotal
Sede	NIF	INT	4	1	4
	Email	VARCHAR(100)	101		101
	Telefone	VARCHAR(20)	21		21
	Nome_Diretor	VARCHAR(50)	51		51
	Rua	VARCHAR(75)	76		76
	Codigo_Postal	VARCHAR(20)	21		21
	Localidade	VARCHAR(50)	51		51
Restaurante	NIF	INT	4	5	20
	Email	VARCHAR(100)	101		505
	Telefone	VARCHAR(20)	21		105
	Nome_Gerente	VARCHAR(50)	51		255
	Rua	VARCHAR(75)	76		380
	Codigo_Postal	VARCHAR(20)	21		105
	Localidade	VARCHAR(50)	51		255
	Sede_NIF	INT	4		20
	Capacidade	INT	4		20
Funcionario	Numero	INT	4	25	100
	Email	VARCHAR(100)	101		2525
	Telefone	VARCHAR(20)	21		525
	Nome	VARCHAR(50)	51		1275
	Data_Nascimento	DATE	3		75
	Funcao	VARCHAR(20)	21		525
	Restaurante_NIF	INT	4		100
	Genero	VARCHAR(45)	46		1150
	Nacionalidade	VARCHAR(45)	46		1150
	NIF	INT	4		40
Fornecedor	Email	VARCHAR(100)	101	10	1010
	Telefone	VARCHAR(20)	21		210
	Tipo	VARCHAR(20)	21		210
	Nome	VARCHAR(50)	51		510
ItemMenu	Numero	INT	4	25	100
	Nome	VARCHAR(50)	51		1275
	Categoria	VARCHAR(20)	21		525
	Descricao	VARCHAR(200)	201		5025
	Preco	DECIMAL(6,2)	5		125
	Tipo	VARCHAR(30)	31		775
Fatura	Numero	INT	4	10000	40000
	Preco_sem_IVA	DECIMAL(6,2)	5		50000
	NIF_Cliente	INT	4		40000
	Nome_Cliente	VARCHAR(50)	51		510000
	IVA	DECIMAL(3,2)	5		50000
	Funcionario_Numero	INT	4		40000
	Tipo_Pagamento	VARCHAR(45)	46		460000
	Data_Emissao	DATETIME	8		80000
Restaurante_has_Fornecedor	Restaurante_NIF	INT	4	25	100
Fornecedor_NIF	INT	4	100		
Restaurante_has_ItemMenu	Restaurante_NIF	INT	4	40	160
	ItemMenu_Numero	INT	4		160
Fatura_has_ItemMenu	Fatura_Numero	INT	4	25000	100000
	ItemMenu_Numero	INT	4		100000
	Quantidade	INT	4		100000
				Total	1589740

Tabela 3 - Estimativa do espaço em disco da base de dados

Para a estimativa do espaço em disco da base de dados decidimos, em reunião com Diretor da Quartel's Grill, os valores para o cálculo.

Assim, criámos uma tabela com o tamanho em bytes de cada tipo de dados, pelo qual multiplicamos o seu número de ocorrências, somando tudo no fim, onde obtemos o valor para o primeiro ano de aproximadamente 1.59 Megabytes.

É de notar que este valor será o pior caso porque, como estamos a usar varchar, o tamanho que este ocupa em bytes é $N + 1$ (sendo N o número de caracteres usados).

Para calcular a estimativa de crescimento anual da base de dados voltamos a ter uma reunião com o Diretor do Quartel's Grill.

O que foi retirado desta reunião foi que o tamanho da maioria das tabelas da base de dados iria permanecer constante, exceto para a Fatura e a Fatura_has_ItemMenu. Através de alguns dados estatísticos recolhidos nos restaurantes, verificou-se que o número de faturas anuais é cerca de 10000 e que o número de itens por fatura é, em média, 2.5.

Fatura	Numero	INT	4	10000	40000
	Preco sem IVA	DECIMAL(6,2)	5		50000
	NIF Cliente	INT	4		40000
	Nome Cliente	VARCHAR(50)	51		510000
	IVA	DECIMAL(3,2)	5		50000
	Funcionario Numero	INT	4		40000
	Tipo Pagamento	VARCHAR(45)	46		460000
	Data Emissao	DATETIME	8		80000
Fatura_has_ItemMenu	Fatura Numero	INT	4	25000	100000
	ItemMenu Numero	INT	4		100000
	Quantidade	INT	4		100000
				Total	1570000

Tabela 4 - Estimativa do aumento do espaço anual da base de dados

Sendo assim, o número de linhas da tabela Fatura será 10000 por ano e o número de linhas da tabela Fatura_has_ItemMenu será 25000. Através dos valores da tabela anterior, concluímos que o aumento anual será de cerca de 1.57 Megabytes.

Assim, nos próximos 5 anos, o tamanho em disco estimado da base de dados será (em bytes):

Ano 1	1589740
Ano 2	3159740
Ano 3	4729740
Ano 4	6299740
Ano 5	7869740

Tabela 5 - Estimativa do espaço ocupado do ano 1 ao ano 5

5.6. Definição e caracterização das vistas de utilização em SQL

Uma view é uma tabela virtual cujos conteúdos são definidos por uma query e, tal como uma tabela, uma view também tem linhas e colunas. Por outro lado, a view não existe como um conjunto de dados armazenado na base de dados, mas sim como referências a tabelas.

Algumas das vantagens da utilização das views são:

- Segurança
- Simplicidade da estrutura
- Integridade dos dados

Por outro lado, as views também apresentam algumas desvantagens, como por exemplo:

- Performance
- Restrições ao atualizar a view

Seguem agora algumas implementações de views na nossa aplicação:

- Melhores 5 funcionários por receita das faturas

```
CREATE VIEW Top5Funcionarios AS
    SELECT f.Nome, SUM(fat.preco_sem_IVA) AS Total
    FROM Funcionario f, Fatura fat
    WHERE f.numero = fat.Funcionario_numero
    GROUP BY fat.Funcionario_numero
    ORDER BY Total DESC
    LIMIT 5;
```

Figura 27 - Vista dos melhores 5 Funcionários por receita das Faturas

- Idade média dos funcionários

```
CREATE VIEW mediaIdadesFuncionarios AS
    Select AVG(TIMESTAMPDIFF(YEAR, Data_nascimento, CURDATE()))
    FROM Funcionario;
```

Figura 28 - Vista da idade média dos Funcionários

5.7. Revisão do sistema implementado

Findada a implementação do sistema da base de dados, é necessária uma revisão desta com o utilizador de modo a poder viabilizar e aprovar o projeto desenvolvido. Sendo assim possível pôr a base em funcionamento.

Após rever a implementação dos requisitos especificados no modelo físico, concluímos que o produto final correspondeu às expectativas do utilizador, levando, desta forma, à aprovação da nossa base de dados.

6. Conclusões e Trabalho Futuro

Partindo do ponto anterior, depois de toda a implementação, estruturação e desenvolvimento da base de dados ter sido concluída, foi feita uma apresentação ao utilizador que se mostrou agradado com o resultado final da nossa base de dados, o que, consequentemente, nos leva a acreditar que a nossa base de dados foi uma missão bem sucedida.

Geralmente, todas as etapas necessárias à realização deste trabalho foram efetuadas com o devido cuidado, procurando sempre por uma alternativa viável e simples para a resolução dos desafios propostos, ou seja, de fácil utilização para o utilizador. Nunca partindo do princípio que todos os possíveis utilizadores tivessem a mesma facilidade em interagir com as novas tecnologias.

Abordando agora o trabalho futuro, considerou-se a possibilidade de acrescentar várias funcionalidades úteis. Observando os itens do menu facilmente denotamos que poderia haver uma melhor especificação dos seus preços, isto é, distinção entre preço de compra e preço de venda, de modo a ser possível calcular o lucro/despesa da cadeia de restaurantes. Da mesma forma, ao atribuir um salário aos funcionários seria possível aproximar ao custo de manutenção da cadeia à realidade.

Quanto ao fornecedor também seria interessante este poder fornecer diferentes tipos de mercadoria, ao invés de apenas estar limitado a um tipo (carne, peixe, vegetariano, ...), podendo até vender os mesmos produtos a diferentes preços, não mudando este preço nos produtos em stock, ficando, assim, o preço associado a cada entrega.

Ainda, considerando que a cadeia de restaurantes irá crescer, existe o aumento do número de clientes, aumentando assim a diversidade de gostos e a exigência. De forma a minimizar os problemas que advêm deste aumento de diversidade e exigência e manter o crescimento da cadeia deverão ser criadas novas categorias dos itens do menu, de forma a poder satisfazer os clientes.

I. Anexos

Anexo 1 - Povoamento da Base de Dados

```
1 use restaurante;
2
3 insert into Sede values (245654378, 'sede@quartelgrill.com', '224567653', 'André Martins', 'Praceta do Vilar', '4710-453', 'Braga');
4
5 insert into Restaurante values
6 (478364835, 'braga@quartelgrill.com', '225765926', 'António Rodrigues', 'Rua da Liberdade', '2345-156', 'Braga', 245654378, 4),
7 (873234798, 'porto@quartelgrill.com', '228567341', 'Matilde Santos', 'Travessa da Liberdade', '7363-789', 'Porto', 245654378, 109),
8 (123567432, 'pacos@quartelgrill.com', '255735100', 'Paulo Costela', 'Rotunda da Liberdade', '6738-243', 'Pacos de Ferreiro', 245654378, 29),
9 (123456789, 'viseu@quartelgrill.com', '232232232', 'Oscar Dias', 'Praceta da Liberdade', '1234-123', 'Viseu', 245654378, 61),
10 (343278934, 'vilaconde@quartelgrill.com', '229876543', 'João Pinto', 'Avenida da Liberdade', '7363-789', 'Vila do Conde', 245654378, 14);
11
12 insert into ItemMenu values
13 (1, 'Costela de Porco', 'Carne', 'Acompanhamento: Arroz e batata frita', 6, 'Prato Principal'),
14 (2, 'Robalo ao Sal', 'Peixe', 'Acompanhamento: Arroz e batata cozida', 10, 'Prato Principal'),
15 (3, 'Alga na Grelha', 'Vegetariano', 'Acompanhamento: Arroz e batata cozida', 1, 'Prato Principal'),
16 (4, 'Picanha Brasileira', 'Carne', 'Acompanhamento: Arroz, batata frita e feijão preto', 13, 'Prato Principal'),
17 (5, 'Tubarão Grelhado', 'Peixe', 'Acompanhamento: Arroz e milho estufado', 20, 'Prato Principal'),
18 (6, 'Beringela recheada de Soja na Brasa', 'Vegetariano', 'Acompanhamento: Bacon Vegetariano', 1.5, 'Prato Principal'),
19 (7, 'Ribs com Molho Barbecue', 'Carne', 'Acompanhamento: Batatas fritas Rústicas', 8.5, 'Prato Principal'),
20 (8, 'Bacalhau na Brasa', 'Peixe', 'Acompanhamento: Batata à murro', 5.5, 'Prato Principal'),
21 (9, 'Espargo Grelhado', 'Vegetariano', 'Acompanhamento: Milho', 3.5, 'Prato Principal'),
22 (10, 'Cheesecake de Limão', 'Bolo', 'Molho de limão, Queijo Mascarpone e Bolacha Torrada', 3, 'Sobremesa'),
23 (11, 'Salada de Fruta', 'Fruta', 'Uva, Manga, Morango, Pessego, Laranja, Romã e Amendoim', 4, 'Sobremesa'),
24 (12, 'Tarte de Nata', 'Bolo', 'Massa Folhada recheada de creme de ovo', 2.5, 'Sobremesa');
25
26 insert into Funcionario values
27 (1, 'toro@quartelgrill.com', '929292929', 'António Rodrigues', '2000-11-21', 'Gerente', 478364835, 'Non-Binary', 'Espanhol'),
28 (2, 'masa@quartelgrill.com', '922222229', 'Matilde Santos', '2000-6-2', 'Gerente', 873234798, 'Feminino', 'Português'),
29 (3, 'paco@quartelgrill.com', '929999999', 'Paulo Costela', '2000-2-13', 'Gerente', 123567432, 'Masculino', 'Português'),
30 (4, 'osdi@quartelgrill.com', '921234567', 'Oscar Dias', '2000-4-2', 'Gerente', 123456789, 'Masculino', 'Português'),
31 (5, 'jopi@quartelgrill.com', '929876543', 'João Pinto', '2000-8-9', 'Gerente', 343278934, 'Masculino', 'Português'),
32 (6, 'ripe@quartelgrill.com', '922323232', 'Ricardo Pereira', '2000-1-19', 'Cozinheiro', 478364835, 'Masculino', 'Luxemburguês'),
33 (7, 'roni@quartelgrill.com', '921111111', 'Ronaldo Nilson', '2000-3-9', 'Cozinheiro', 873234798, 'Masculino', 'Brasileiro'),
34 (8, 'pelo@quartelgrill.com', '926666666', 'Pedro Lopes', '2000-12-24', 'Cozinheiro', 123567432, 'Masculino', 'Português'),
35 (9, 'jaga@quartelgrill.com', '927777777', 'Jacqueline Gáúcho', '2000-5-31', 'Cozinheiro', 123456789, 'Feminino', 'Brasileiro'),
36 (10, 'judi@quartelgrill.com', '928888888', 'Juan Diaz', '2000-7-22', 'Cozinheiro', 343278934, 'Masculino', 'Venezuelano'),
37 (11, 'pepi@quartelgrill.com', '929876543', 'Pedro Pinto', '1998-1-9', 'Empregado de Mesa', 478364835, 'Masculino', 'Português'),
38 (12, 'anmo@quartelgrill.com', '923333333', 'Anita Moraes', '1999-10-23', 'Empregado de Mesa', 873234798, 'Feminino', 'Brasileira'),
39 (13, 'rima@quartelgrill.com', '924444444', 'Rita Marques', '1993-11-3', 'Empregado de Mesa', 123567432, 'Feminino', 'Português'),
40 (14, 'medj@quartelgrill.com', '925555555', 'Mésaq Dju', '1994-2-22', 'Empregado de Mesa', 123456789, 'Masculino', 'Angolano'),
41 (15, 'beco@quartelgrill.com', '920000000', 'Bernardo Costa', '1997-4-14', 'Empregado de Mesa', 343278934, 'Masculino', 'Português');
42
43 insert into Fornecedor values
44 (111111111, 'geral@sendvegies.com', '213441234', 'Vegetais', 'SendVegies'),
45 (111111112, 'comercial@ludite.com', '217541214', 'Vegetais', 'Ludite'),
46 (111111113, 'geral@carnesmeireles.com', '213447934', 'Carne', 'Carnes Meireles'),
47 (111111114, 'geral@talhooliveira.com', '225242567', 'Carne', 'Talho Oliveira'),
48 (111111115, 'geral@lotadeangeiras.com', '238464936', 'Peixe', 'Lota de Angeiras'),
49 (111111116, 'geral@peixedateresa.com', '225242567', 'Peixe', 'Peixe da Teresa'),
50 (111111117, 'entregas@mariabolacha.com', '225242567', 'Bolos', 'Maria Bolacha');
```

Figura 29 - Povoamento da Base de Dados (Parte 1)

```

51
52  insert into Fatura values
53      (1, 19, null, null, 0.23, 14, 'Dinheiro', '2020-8-15'),
54      (2, 20, 125432357, 'Luís Sousa', 0.23, 12, 'Cartao', '2020-9-3'),
55      (3, 45, null, null, 0.23, 10, 'Dinheiro', '2020-9-11'),
56      (4, 86, null, null, 0.23, 10, 'Cartao', '2020-9-20'),
57      (5, 35, 193747463, 'Rui Vieira', 0.23, 11, 'Cartao', '2020-10-5'),
58      (6, 82, null, null, 0.23, 15, 'Dinheiro', '2020-10-14'),
59      (7, 180, 594738379, 'Novo Banco', 0.23, 14, 'Dinheiro', '2020-10-17'),
60      (8, 83, null, null, 0.23, 14, 'Dinheiro', '2020-10-20'),
61      (9, 82, null, null, 0.23, 12, 'Cartao', '2020-10-26'),
62      (10, 64, 736464738, 'Pedro Faria', 0.23, 11, 'Cheque', '2020-11-4'),
63      (11, 10, null, null, 0.23, 10, 'Dinheiro', '2020-11-9'),
64      (12, 13, null, null, 0.23, 13, 'Dinheiro', '2020-11-14'),
65      (13, 100, null, null, 0.23, 13, 'Dinheiro', '2020-11-17'),
66      (14, 50, null, null, 0.23, 12, 'Cartao', '2020-11-23'),
67      (15, 20, null, null, 0.23, 15, 'Cheque', '2020-11-29');
68
69  insert into Restaurante_has_Fornecedor values
70      (478364835, 111111111),
71      (478364835, 111111113),
72      (478364835, 111111115),
73      (478364835, 111111117),
74      (873234798, 111111111),
75      (873234798, 111111114),
76      (873234798, 111111116),
77      (873234798, 111111117),
78      (123567432, 111111112),
79      (123567432, 111111114),
80      (123567432, 111111115),
81      (123567432, 111111117),
82      (123456789, 111111112),
83      (123456789, 111111113),
84      (123456789, 111111116),
85      (123456789, 111111117),
86      (343278934, 111111112),
87      (343278934, 111111113),
88      (343278934, 111111115),
89      (343278934, 111111117);
90
91  insert into Restaurante_has_ItemMenu values
92      (478364835, 1),
93      (478364835, 2),
94      (478364835, 3),
95      (478364835, 4),
96      (478364835, 5),
97      (478364835, 6),
98      (478364835, 10),
99      (478364835, 11),
100

```

Figura 30 - Povoamento da Base de Dados (Parte 2)

```

101      (873234798, 4),
102      (873234798, 5),
103      (873234798, 6),
104      (873234798, 7),
105      (873234798, 8),
106      (873234798, 9),
107      (873234798, 10),
108      (873234798, 12),
109
110      (123567432, 1),
111      (123567432, 2),
112      (123567432, 3),
113      (123567432, 7),
114      (123567432, 8),
115      (123567432, 9),
116      (123567432, 11),
117      (123567432, 12),
118
119      (123456789, 1),
120      (123456789, 2),
121      (123456789, 6),
122      (123456789, 7),
123      (123456789, 5),
124      (123456789, 9),
125      (123456789, 11),
126      (123456789, 12),
127
128      (343278934, 4),
129      (343278934, 5),
130      (343278934, 3),
131      (343278934, 7),
132      (343278934, 2),
133      (343278934, 9),
134      (343278934, 10),
135      (343278934, 11);
136
137      insert into Fatura_has_ItemMenu values
138      (1, 1, 2),
139      (1, 12, 2),
140
141      (2, 2, 1),
142      (2, 3, 1),
143      (2, 11, 2),
144
145      (3, 7, 1),
146      (3, 8, 2),
147      (3, 9, 1),
148      (3, 10, 2),
149      (3, 11, 1),
150      (3, 12, 1),

```

Figura 31 - Povoamento da Base de Dados (Parte 3)

151	
152	(4, 5, 1),
153	(4, 11, 1),
154	
155	(5, 4, 4),
156	
157	(6, 6, 1),
158	(6, 7, 3),
159	(6, 1, 2),
160	(6, 12, 6),
161	
162	(7, 2, 4),
163	(7, 10, 2),
164	
165	(8, 7, 3),
166	(8, 8, 1),
167	(8, 10, 2),
168	(8, 11, 1),
169	(8, 12, 2),
170	
171	(9, 4, 1),
172	
173	(10, 9, 2),
174	(10, 8, 1),
175	(10, 3, 1),
176	(10, 12, 3),
177	(10, 10, 1),
178	
179	(11, 1, 1),
180	(11, 3, 1),
181	(11, 10, 1),
182	
183	(12, 10, 3),
184	
185	(13, 4, 2),
186	(13, 5, 3),
187	(13, 11, 4),
188	(13, 12, 1),
189	
190	(14, 8, 5),
191	(14, 10, 5),
192	
193	(15, 3, 4);

Figura 32 - Povoamento da Base de Dados (Parte 4)