

**Farmácia do Prometeu**

**Universidade do Minho**

Mestrado Integrado em Engenharia Informática

**Unidade Curricular de**

**Bases de Dados**

Ano Letivo de 2018/2019

**A80760 Alexandre Pacheco**

**A81728 Mariana Fernandes**

**A80785 Pedro Lima**

**A80741 Pedro Pinto**

Novembro, 2018

**BD**

|  |  |
| --- | --- |
| Data de Receção |  |
| Responsável |  |
| Avaliação |  |
| Observações |  |

**A80760 Alexandre Pacheco**

**A81728 Mariana Fernandes**

**A80785 Pedro Lima**

**A80741 Pedro Pinto**

Novembro, 2018

**Farmácia do Prometeu**

Resumo

Aqui fica o resumo

**Áreas de Aplicaçao:** Identificação da área de trabalho, por exemplo, desenho e arquitetura de SBD’s

**Palavras-Chave:** Bases de Dados, Bases de Dados Relacionais, Modelo Conceptual, Modelo Lógico, Modelo Físico, MySQL, MySQL Workbench, brModelo, Entidades, Relacionamentos, SHA-1.

Índice

[1 Introdução 8](#_Toc530848817)

[1.1 Contextualização 8](#_Toc530848818)

[1.2 Apresentação do Caso de Estudo 9](#_Toc530848819)

[1.3 Motivação e Objetivos 10](#_Toc530848820)

[2 Levantamento e Análise de Requisitos 11](#_Toc530848821)

[2.1 Requisitos Levantados 11](#_Toc530848822)

[2.1.1 Requisitos de Descrição 12](#_Toc530848823)

[2.1.2 Requisitos de exploração 13](#_Toc530848824)

[2.1.3 Requisitos de controlo 13](#_Toc530848825)

[3 Modelação Conceptual 14](#_Toc530848826)

[3.1 Apresentação da Abordagem de modelação realizada 14](#_Toc530848827)

[3.2 Identificação e caracterização das entidades 14](#_Toc530848828)

[3.3 Identificação e caracterização dos relacionamentos 14](#_Toc530848829)

[3.4 Identificação e caracterização das associações dos Atributos com as Entidades e Relacionamentos 14](#_Toc530848830)

[3.5 Detalhe ou generalização de entidades 14](#_Toc530848831)

[3.6 Apresentação e explicação do diagrama ER 14](#_Toc530848832)

[3.7 Validação do modelo de dados com o utilizador 15](#_Toc530848833)

[4 Modelação Lógica 16](#_Toc530848834)

[4.1 Construção e validação do modelo de dados lógico 16](#_Toc530848835)

[4.2 Desenho do modelo lógico 16](#_Toc530848836)

[4.3 Validação do modelo através da normalização 16](#_Toc530848837)

[4.4 Validação do modelo com interrogações do utilizador 16](#_Toc530848838)

[4.5 Validação do modelo com as transações estabelecidas 16](#_Toc530848839)

[4.6 Revisão do modelo lógico com o utilizador 16](#_Toc530848840)

[5 Implementação Física 17](#_Toc530848841)

[6 Conclusões e Trabalho Futuro 18](#_Toc530848842)

**Não foi encontrada nenhuma entrada do índice de ilustrações.**

**Não foi encontrada nenhuma entrada do índice de ilustrações.**

# Introdução

## Contextualização

Nos dias de hoje, as tecnologias de informação estão cada vez mais presentes em contexto pessoal ou empresarial. Descurando o primeiro caso, é fácil entender a importância das TI nas organizações. Por exemplo, o armazenamento digital de dados permite acesso rápido e concorrente à informação e também possibilita a descoberta de padrões recorrentes. Com isto, podemos concluir que seja do interesse de qualquer novo estabelecimento implementar as devidas tecnologias que forneçam ao negócio organização e eficiência. Foi com estes aspetos em mente que a Farmácias Olimpo, S.A. decidiu requisitar os nossos excelentes serviços de consultoria informática.

A Farmácia do Prometeu é a mais recente ideia de negócio da empresa Farmácias Olimpo, S.A., que já conta com sucessos como a Farmácia da Hera no Porto e a Farmácia de Dioniso em Lisboa. A qualidade e rapidez do serviço prestado aos seus utentes impulsionou Farmácias Olimpo, S.A. que está rapidamente a tornar-se a maior vendedora de medicamentos em território nacional.

Após uma análise de mercado foi decidido que a próxima farmácia será aberta na cidade de Braga, com aproximadamente 181.400 residentes, na sua maior freguesia, São Vítor, com aproximadamente 30500 habitantes presentes (dados censos 2011). A localização é privilegiada também por ter nas suas imediações um dos polos da Universidade do Minho.

Refletindo na importância de se manterem a par com os desenvolvimentos tecnológicos, a gerência da Farmácia do Prometeu pretende não só ter implementado um sistema informático que lhes sirva de suporte ao bom funcionamento do seu negócio, como também disponibilizarem uma aplicação mobile que permita aos clientes consultarem histórico de compras e pontos acumulados.

## Apresentação do Caso de Estudo

A Farmácia do Prometeu pretende estabelecer-se num ambiente com alguma competição pelo que deve ser assegurada a alta fiabilidade do sistema em conjunto com a sua eficiência e utilidade. Gerida pelo Sr. Prometeu, e com uma equipa profissional de farmacêuticos, a farmácia irá disponibilizar aos alunos da Universidade do Minho e demais residentes de Braga acesso a diversos medicamentos, sujeitos, ou não, a receita médica. Os mesmos chegam à farmácia através de diversos fornecedores para que seja possível manter os stocks em níveis adequados.

Um farmacêutico, depois de se autenticar no POS (“Point of Service”, ou Ponto de Venda), com o seu ID e palavra-passe, pode começar a efetuar pesquisas de medicamentos e vendas. Cada embalagem dispensada está obrigatoriamente associada a uma fatura e esta pode ou não estar associada a uma receita. No entanto, caso sejam vendidos medicamentos sujeitos a receita médica, que por lei necessitam de receita para poderem ser dispensados, o código da receita tem de estar associado à fatura. A farmácia dispõe acesso controlado à base de dados do Sistema Nacional de Saúde (SNS) onde constam todos os detalhes das receitas lançadas, para consultarem uma receita específica necessitam apenas de ler o seu código com o leitor de código de barras.

Para cada cliente que se dirija à farmácia pela primeira vez é criada uma ficha com o seu nome, NIF, número de telemóvel e email. Assim que o cliente associar o seu NIF pode dar login na aplicação mobile, sendo que na primeira vez que acede à sua área tem de escolher uma palavra-passe. A partir deste momento pode consultar as suas compras anteriores, pontos acumulados e gastos, alterar dados de contacto e a palavra-passe.

## Motivação e Objetivos

A realização deste projeto advém da proposta de implementação de uma BD com tema à escolha. Não foi fácil selecionar um tema pois este deveria ser complexo suficiente para ser possível desenvolver um trabalho sólido em seu torno, mas não demasiado trabalhoso de tal forma que a sua realização em tempo útil fosse impossível. Após algum debate resolvemos desenvolver o projeto em torno de uma farmácia, dado que nos permitia ter um nível de complexidade adequado e porque é algo com que estávamos familiarizados.

O nosso objetivo foi implementar um SBD que sirva de suporte ao sistema informático da farmácia e à aplicação mobile dos clientes. O esperado é que o sistema consiga manter a integridade perante todas as transações que possam ocorrer simultaneamente, como também permitir aos farmacêuticos a localização rápida das embalagens que pretendem dispensar. Dada a natureza do problema também é espectável que a BD possa revelar alguma informação como quais os medicamentos mais vendidos, quem realizou mais vendas, quais os medicamentos que os pacientes crónicos adquirem periodicamente.

Como tal, construímos uma base de dados simples e funcional tendo sempre em conta o funcionamento da farmácia, bem como as interações do utilizador com a aplicação, tentado facilitar o processo de venda e a consulta de informação por parte dos clientes.

# Levantamento e Análise de Requisitos

## Método de Levantamento de Requisitos Adotado

Para o levantamento de requisitos, foi convocada uma reunião entre a equipa de trabalho e o Sr. Prometeu por forma a termos em consideração todos os aspetos gerais e particulares do funcionamento de uma farmácia bem como esclarecer quaisquer dúvidas.

## Requisitos Levantados

A farmácia está aberta das 8h às 24h, sem pausas, 365 dias por ano. Assim que o cliente chega à farmácia, tira uma senha com um número e marcação da data e hora de chegada, que no final da compra é associada à fatura por forma a ser possível fazer uma análise posterior sobre os tempos de atendimento.

Cada cliente tem uma ficha que contém o seu nome, Número de Identificação Fiscal (NIF), email, número de telemóvel. A ficha é criada sempre que um novo cliente se dirige à farmácia. O farmacêutico pode confirmar a existência, ou não, de um cliente procurando pelo seu NIF, nº de telemóvel ou até do nome.

Ao efetuar compras superiores as 10€ o cliente ganha 10 pontos. Por cada 10€ acima, obtém mais 15 pontos. Em cada compra o cliente pode gastar apenas os pontos adquiridos anteriormente, podendo assim optar por trocar 35 pontos por um vale de 3€ (em compras iguais ou superiores a 3 euros) ou 50 pontos por um desconto de 5€ (em compras iguais ou superiores a 5€).

Os clientes podem adquirir medicamentos que estejam, ou não, associados a uma ou mais receitas. Ao fornecerem o código da receita (que pode ser lido do papel ou ter sido enviado para o telemóvel do cliente) ao farmacêutico, é feita uma pesquisa na Base de Dados do SNS e extraídos quais os medicamentos receitados. Caso queiram medicamentos sujeitos a receita médica, a respetiva receita terá de estar obrigatoriamente associada à fatura.

Sobre o medicamento em si, é imperativo saber a sua designação, o tamanho da embalagem (quantidade e em que unidades está representada a quantidade), a que categoria pertence (analgésico, estatina, antiemético, antibiótico), se é ou não sujeito a receita médica e em que formato se apresenta (comprimido, xarope e granulado) e qual o laboratório que o produz bem como o preço de venda atual e a quantidade em stock. Os medicamentos estão organizados por armário (A-Z), gaveta (1-9) e separador (1-9).

Sobre cada funcionário, a gerência pretende saber o nome, nº de telemóvel, NISS e nº da cédula profissional. De momento, o Sr. Prometeu apenas pretende saber qual o ordenado atual e para que IBAN terá de o transferir. Caso a farmácia tenha o sucesso previsto, o sistema terá de registar as horas trabalhadas por cada funcionário e calcular o montante a transferir. Como não é algo relevante para o funcionamento inicial da farmácia, foi decidido que esta funcionalidade não será implementada, nesta versão, sendo implementada apenas a simplificação previamente descrita.

A venda dos medicamentos pressupõe o lançamento duma fatura no sistema, onde consta o funcionário responsável, o cliente a quem pertence e os medicamentos que foram vendidos (e as respetivas quantidades e preços). O cliente pode então escolher gastar pontos para obter um desconto no valor total da fatura. Todos estes valores devem aparecer descriminados na fatura.

### Requisitos de Descrição

**Cliente**

* Id
* Nome
* NIF
* Email
* Número de Telemóvel
* Pontos
* Palavra-chave

**Funcionário**

* Id
* Nome
* Número de Telemóvel
* Número de Identificação da Segurança Social (NISS)
* Identificador da conta bancária (IBAN)
* Número da cédula profissional
* Ordenado
* Palavra-chave

**Medicamento**

* Id
* Designação
* É ou não sujeito a receita médica
* Posição em que está guardado (armário, gaveta, separador)
* Formato em que se apresenta
* Quantidade presente na embalagem
* Unidades em que a quantidade é medida
* Categoria
* Laboratório
* Preço
* Stock

**Fatura**

* Id
* Pontos (Recebidos e Utilizados)
* Desconto
* Data/hora da fatura
* Data/hora da senha
* Códigos das receitas
* Total
* Que medicamentos foram vendidos e a que preço

### Requisitos de exploração

**Cliente**

* Consultar faturas
* Consultar pontos

**Funcionário**

* Procurar clientes
* Consultar medicamentos

**Administrador**

* Consultar dados dos funcionários
  + Consultar os dados de pagamento de vencimentos aos funcionários
  + Consultar funcionário do mês
* Consultar faturas
  + Solicitar o tempo médio de atendimento
  + Solicitar o tempo médio de atendimento por hora
* Consultar medicamentos
* Consultar dados dos clientes
  + Procurar o medicamento mais vendido
  + Procurar quais os pacientes crónicos e que medicamentos adquirem (pacientes que adquiriram o mesmo medicamento mais do que 5 vezes)[[1]](#footnote-1)

### Requisitos de controlo

**Cliente**

* Alterar contactos pessoais

**Funcionário**

* Inserir clientes
* Adicionar novos dados de clientes já existentes
* Alterar contactos pessoais do cliente
* Inserir faturas

**Administrador**

* Inserir funcionários
* Alterar dados dos funcionários
* Apagar faturas
* Inserir medicamentos
* Alterar dados dos medicamentos

# Modelação Conceptual

## Apresentação da Abordagem de modelação realizada

O modelo conceptual representa a base de dados de uma forma independente de detalhes físicos ou respeitantes à implementação. Após a análise de requisitos elabora-se este modelo tendo em conta as necessidades da base de dados, de maneira que as entidades que representam os dados a armazenar, bem como as suas relações sejam coerentes e coesas.

A abordagem conceptual que tomámos foi centralizada na venda dos medicamentos na farmácia, pelo que as entidades e os relacionamentos modelados visam criar um sistema que suporte e facilite a representação virtual dessa atividade.

## Identificação e caracterização das entidades

De modo a desenvolver um modelo conceptual há que, inicialmente, identificar as entidades necessárias. Abaixo encontra-se essa identificação relativa às entidades que decidimos modelar.

**Funcionário**

Entidade responsável pelo bom funcionamento da farmácia, sendo assim quem executa todas as ações relativas à venda de medicamentos. Este atende os clientes, criando uma ficha de clientes, para os que irão obter medicamentos pela primeira vez.

**Cliente**

Entidade responsável pela aquisição de medicamentos. Este pode obter medicamentos com ou sem receita médica, dependendo do medicamento em questão. No momento de finalizar a compra, pode utilizar uma pequena quantia dos seus pontos de modo a obter um desconto.

**Medicamento**

Entidade central, uma vez que é o motivo da criação da base de dados. Esta refere toda a informação necessária sobre um medicamento, acrescentando também a sua localização na farmácia, para uma maior eficiência no momento da venda, reduzindo assim o tempo de espera por parte do cliente.

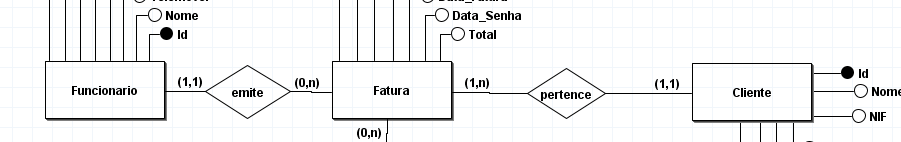
**Fatura**

Entidade responsável pelo armazenamento de toda a informação necessária ao sistema. Esta guarda a informação permitindo controlar o stock de medicamentos, averiguar os tempos de espera para ser atendido, calcular o funcionário que efetua mais vendas e que doentes têm problemas crónicos e necessitam comprar o mesmo medicamento periodicamente.

## Identificação e caracterização dos relacionamentos

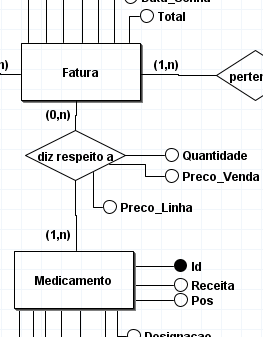
Identificadas as entidades há que estabelecer entre elas relacionamentos que representem as próprias interações entre os “atores” que elas modelam.

**Fatura, Funcionário e Cliente:**



Apesar de as relações entre funcionário, fatura e cliente serem duas relações binárias, como podemos observar nesta porção do modelo, são duas relações que não conseguimos ver isoladas. O funcionário é responsável pela emissão de faturas pertencentes a um cliente, o que faz com que uma fatura seja também a forma de relacionar funcionários com clientes. Em termos de cardinalidade, uma fatura é emitida por um e apenas um funcionário e um funcionário pode emitir várias faturas. Um cliente pode estar associado a várias faturas (uma no mínimo, dado que este é registado aquando da sua primeira compra na farmácia), e uma fatura diz respeito a apenas um cliente.

**Fatura e Medicamento:**

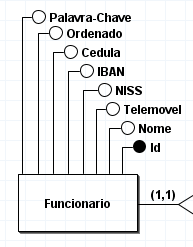


Podem ser passadas muitas faturas respeitantes à venda de um determinado medicamento, bem como podem ser vendidos vários medicamentos numa venda que dê origem a uma dada fatura. Como sabemos, relacionamentos com cardinalidade de N para N exigem cuidados especiais, de que falaremos mais à frente neste relatório.

## Identificação e caracterização das associações dos Atributos com as Entidades e Relacionamentos

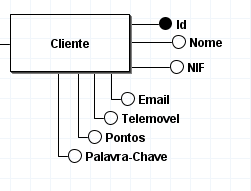
Da mesma maneira que decidimos as entidades a serem representadas na base de dados, há também a necessidade de definir os atributos que estarão presentes em cada uma delas, ou nas relações.

**Funcionário:**



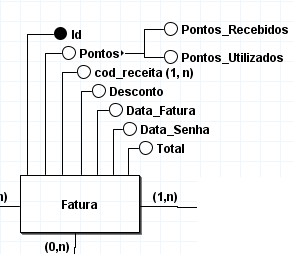
No funcionário há vários atributos que são únicos, como por exemplo o telemóvel, o NISS, a cédula e o IBAN. Decidimos não utilizar nenhum destes atributos como chave primária porque preferimos ter um maior controlo sobre a sua criação e valores que toma, o que não se verificaria se utilizássemos por exemplo o IBAN, que é algo atribuído pelo estado.

**Cliente:**



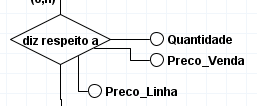
Em relação ao cliente, a abordagem que tivemos foi semelhante à do caso do funcionário.

**Fatura:**

****

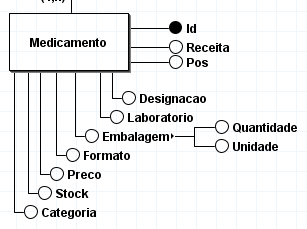
Na entidade fatura decidimos incluir um atributo relativo ao valor total. Na verdade, este valor pode ser calculado recorrendo a outros atributos, no entanto, seria necessário efetuar alguns cálculos aritméticos e recorrer a várias tabelas, o que acaba por justificar a presença deste atributo, tendo em conta que o valor de uma fatura é das suas características principais.

**Relacionamento entre Fatura e Medicamento:**



Neste relacionamento, achámos por bem que existisse um atributo com o preço a que o medicamento foi vendido aquando da criação da fatura, uma vez que o preço de um medicamento pode mudar ao longo do tempo.

**Medicamento:**



Inicialmente idealizámos um modelo em que existiria uma entidade Stock, uma entidade relativa a Princípio Ativo e outra para representar o Laboratório. De modo a simplificar o trabalho, e concentrando-nos em atingir os objetivos do projeto, decidimos simplesmente criar atributos correspondentes ao stock e ao laboratório, e esquecer por completo o princípio ativo. É importante mencionar, no entanto, que num contexto real, essas entidades teriam de existir. Por exemplo, no caso do laboratório, há uma grande repetição na base de dados se muitos medicamentos vierem do mesmo laboratório (e.g. repetição da string “BAYER”). O mesmo vale para o princípio ativo, por exemplo. E se estas entidades existissem, incluíamos na entidade medicamento chaves estrangeiras, o que depois também facilitaria a procura de medicamentos pelo laboratório ou pelo princípio ativo, de forma eficiente.

## 

## Detalhe ou generalização de entidades

Abaixo segue uma síntese das entidades a representar na nossa base de dado, bem como os seus atributos e o domínio de valores em que ocorrem.

**Funcionário:**

* *Id*   
  Números naturais.
* *Nome*  
  String
* *Telemóvel*  
  String
* *NISS*  
  String
* *IBAN*  
  String
* *Cedula*String
* *Ordenado*Número positivo com vírgula flutuante
* *Palavra-chave*String

**Cliente:**

* *Id*Números naturais
* *Nome*String
* *NIF*String
* *Email*String
* *Telemovel*String
* *Pontos*Número natural ou zero
* *Palavra-chave*String

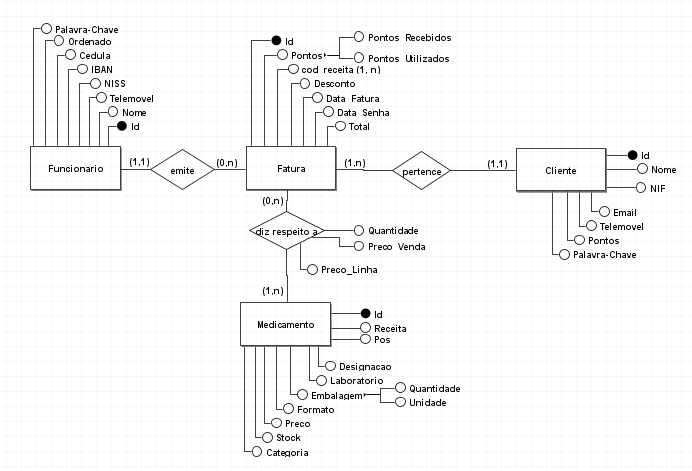
**Fatura:**

* *Id*Número natural
* *Pontos  
  -Pontos\_recebidos*  
  Número natural ou zero  
  *-Pontos\_utilizados*Número natural ou zero
* *Cod\_receita*String
* *Desconto*
* *Data\_fatura*Data e hora
* *Data\_senha*Data e hora
* *Total*Número positivo com vírgula flutuante

**Medicamento:**

* *Id*Número natural
* *Receita*Caracter ‘S’ ou ‘N’
* *Pos*String
* *Designação*String
* *Laboratório*String
* *Embalagem***-***Quantidade*Número positivo com vírgula flutuante  
  -*Unidade*String
* *Formato*  
  String
* *Preço*Número positivo com vírgula flutuante
* *Stock*Número natural ou zero
* *Categoria*String

## Apresentação e explicação do diagrama ER



## Validação do modelo de dados com o utilizador

Terminada a conceptualização da base de dados, é necessária a validação do modelo de modo a confirmar junto do cliente ou utilizador que o modelo representa o pretendido.

Posto isto, foi realizada uma reunião com o Sr. Prometeu a fim de efetuar esta validação. O modelo foi validado e o projeto pôde passar à fase seguinte, a modelação lógica.

# Modelação Lógica

## Construção e validação do modelo de dados lógico

## Desenho do modelo lógico

## Validação do modelo através da normalização

## Validação do modelo com interrogações do utilizador

## Validação do modelo com as transações estabelecidas

## Revisão do modelo lógico com o utilizador

# Implementação Física

## Seleção do sistema de gestão de bases de dados

## Tradução do esquema lógico para o SGBD escolhido em SQL ???

Nesta fase, o modelo lógico foi facilmente convertido para o físico atráves da ferramenta de “foward engineer” do MySQL Workbench. Após uma revisão cuidada ao script gerado, o processo foi concluído.

1. **SET** @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;
2. **SET** @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;
3. **SET** @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';
5. -- -----------------------------------------------------
6. -- Schema farmacia
7. -- -----------------------------------------------------
9. -- -----------------------------------------------------
10. -- Schema farmacia
11. -- -----------------------------------------------------
12. **CREATE** **SCHEMA** IF NOT EXISTS `farmacia` **DEFAULT** **CHARACTER** **SET** utf8 ;
13. USE `farmacia` ;
15. -- -----------------------------------------------------
16. -- Table `farmacia`.`Funcionario`
17. -- -----------------------------------------------------
18. **CREATE** **TABLE** IF NOT EXISTS `farmacia`.`Funcionario` (
19. `id` **INT** NOT NULL,
20. `nome` **VARCHAR**(75) NOT NULL,
21. `tlmv` **VARCHAR**(20) NOT NULL,
22. `niss` **VARCHAR**(11) NOT NULL,
23. `iban` **VARCHAR**(32) NOT NULL,
24. `ordenado` **FLOAT** NOT NULL,
25. `cedula` **VARCHAR**(6) NOT NULL,
26. **PRIMARY** **KEY** (`id`))
27. ENGINE = InnoDB;

30. -- -----------------------------------------------------
31. -- Table `farmacia`.`Cliente`
32. -- -----------------------------------------------------
33. **CREATE** **TABLE** IF NOT EXISTS `farmacia`.`Cliente` (
34. `id` **INT** NOT NULL,
35. `nome` **VARCHAR**(75) NOT NULL,
36. `nif` **VARCHAR**(9) NULL,
37. `email` **VARCHAR**(50) NULL,
38. `tlmv` **VARCHAR**(20) NULL,
39. `pontos` **INT** NOT NULL,
40. **PRIMARY** **KEY** (`id`))
41. ENGINE = InnoDB;

44. -- -----------------------------------------------------
45. -- Table `farmacia`.`Fatura`
46. -- -----------------------------------------------------
47. **CREATE** **TABLE** IF NOT EXISTS `farmacia`.`Fatura` (
48. `id` **INT** NOT NULL,
49. `data\_f` DATETIME NOT NULL,
50. `data\_s` DATETIME NOT NULL,
51. `pontos\_r` **INT** NOT NULL,
52. `pontos\_u` **INT** NOT NULL,
53. `desconto` **FLOAT** NOT NULL,
54. `total` **FLOAT** NOT NULL,
55. `id\_func` **INT** NOT NULL,
56. `id\_c` **INT** NOT NULL,
57. **PRIMARY** **KEY** (`id`),
58. **INDEX** `id\_idx` (`id\_func` **ASC**) VISIBLE,
59. **INDEX** `id\_cliente\_idx` (`id\_c` **ASC**) VISIBLE,
60. **CONSTRAINT** `id\_func`
61. **FOREIGN** **KEY** (`id\_func`)
62. **REFERENCES** `farmacia`.`Funcionario` (`id`)
63. **ON** **DELETE** **RESTRICT**
64. **ON** **UPDATE** **CASCADE**,
65. **CONSTRAINT** `id\_cliente`
66. **FOREIGN** **KEY** (`id\_c`)
67. **REFERENCES** `farmacia`.`Cliente` (`id`)
68. **ON** **DELETE** **RESTRICT**
69. **ON** **UPDATE** **CASCADE**)
70. ENGINE = InnoDB;

73. -- -----------------------------------------------------
74. -- Table `farmacia`.`Fatura\_Receita`
75. -- -----------------------------------------------------
76. **CREATE** **TABLE** IF NOT EXISTS `farmacia`.`Fatura\_Receita` (
77. `id\_fatura` **INT** NOT NULL,
78. `cod\_receita` **INT** NOT NULL,
79. **PRIMARY** **KEY** (`id\_fatura`, `cod\_receita`),
80. **CONSTRAINT** `id\_fatura`
81. **FOREIGN** **KEY** (`id\_fatura`)
82. **REFERENCES** `farmacia`.`Fatura` (`id`)
83. **ON** **DELETE** **CASCADE**
84. **ON** **UPDATE** **CASCADE**)
85. ENGINE = InnoDB;

88. -- -----------------------------------------------------
89. -- Table `farmacia`.`Medicamento`
90. -- -----------------------------------------------------
91. **CREATE** **TABLE** IF NOT EXISTS `farmacia`.`Medicamento` (
92. `id` **INT** NOT NULL,
93. `des` **VARCHAR**(45) NOT NULL,
94. `qt` **FLOAT** NOT NULL,
95. `un` **VARCHAR**(5) NOT NULL,
96. `formato` **VARCHAR**(45) NOT NULL,
97. `categoria` **VARCHAR**(45) NOT NULL,
98. `preco` **FLOAT** NOT NULL,
99. `stock` **INT** NOT NULL,
100. `lab` **VARCHAR**(45) NOT NULL,
101. `receita` **CHAR**(1) NOT NULL,
102. `pos` **CHAR**(3) NOT NULL,
103. **PRIMARY** **KEY** (`id`))
104. ENGINE = InnoDB;

107. -- -----------------------------------------------------
108. -- Table `farmacia`.`Fatura\_Med`
109. -- -----------------------------------------------------
110. **CREATE** **TABLE** IF NOT EXISTS `farmacia`.`Fatura\_Med` (
111. `id\_f` **INT** NOT NULL,
112. `id\_m` **INT** NOT NULL,
113. `qt\_v` **INT** NOT NULL,
114. `preco\_v` **FLOAT** NOT NULL,
115. `preco\_l` **FLOAT** NOT NULL,
116. **PRIMARY** **KEY** (`id\_f`, `id\_m`),
117. **INDEX** `id\_m\_idx` (`id\_m` **ASC**) VISIBLE,
118. **CONSTRAINT** `id\_f`
119. **FOREIGN** **KEY** (`id\_f`)
120. **REFERENCES** `farmacia`.`Fatura` (`id`)
121. **ON** **DELETE** **CASCADE**
122. **ON** **UPDATE** **CASCADE**,
123. **CONSTRAINT** `id\_m`
124. **FOREIGN** **KEY** (`id\_m`)
125. **REFERENCES** `farmacia`.`Medicamento` (`id`)
126. **ON** **DELETE** **RESTRICT**
127. **ON** **UPDATE** **CASCADE**)
128. ENGINE = InnoDB;

131. **SET** SQL\_MODE=@OLD\_SQL\_MODE;
132. **SET** FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;
133. **SET** UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

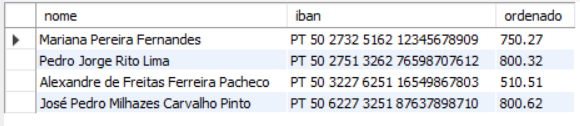
## Tradução das interrogações do utilizador para SQL (alguns exemplos)

### Dados dos vencimentos dos funcionários

**Query:**

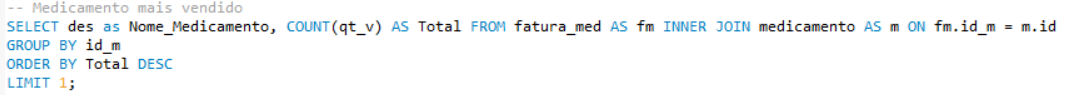


**Resultado:**



### Medicamento mais vendido

**Query:**

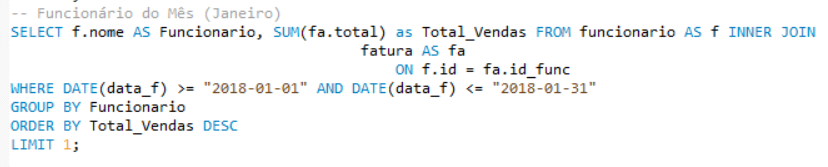


**Resultado:**



### Funcionário do mês (Janeiro)

**Query:**

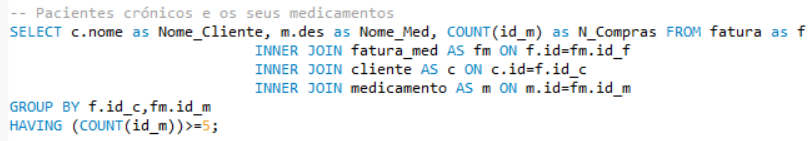


**Resultado:**



### Medicamentos de pacientes crónicos

**Query:**



**Resultado:**



### Tempo médio de atendimento

**Query:**

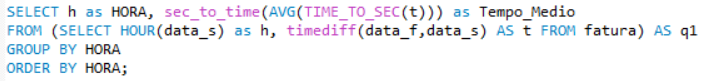


**Resultado:**

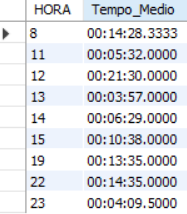


### Tempo médio de atendimento por hora

**Query:**

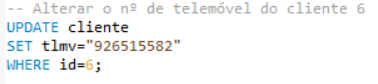


**Resultado:**

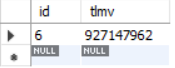
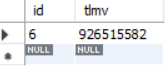


### Alterar o nº de telemóvel do cliente 6

**Query:**

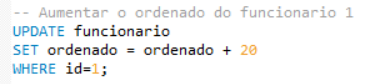


**Resultado:**

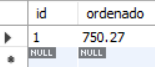
 

### Aumentar o ordenado do funcionário 1

**Query:**



**Resultado**

## Tradução das transações estabelecidas para SQL (alguns exemplos)

## Escolha, definição e caracterização de índices em SQL

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 telefónica com milhões de contactos, será fácil encontrar um contacto pelo número (admitindo que este é a chave primária) mas será difícil, ou pouco eficiente, procurar o contacto segundo o nome a ele associado, por exemplo, dado que a tabela não está organizada consoante esse atributo.

Ora, 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.

No nosso trabalho, utilizámos esta funcionalidade nas tabelas referentes aos clientes e aos funcionários, indexando as entradas pelo número de telefone, para podermos procurar um cliente ou funcionário pelo seu contacto.

Não fizemos uso de índices nos nomes dos medicamentos, por exemplo, porque pretendemos que a procura de medicamentos seja feita de modo a não encontrar apenas encontrar resultados que comecem pela palavra de pesquisa (pretendemos, por exemplo, procurar "fen" e encontrar não só "Fenistil" como também "Brufen"). O uso de índices, nestas situações, não tornaria as queries mais eficientes.

Outro aspeto relativo aos índices que despertou a nossa atenção é o facto de serem automaticamente gerados para chaves estrangeiras, o que é algo bastante útil visto que, por exemplo, não se procura uma fatura pelo seu identificador (chave primária) mas sim pelo cliente e/ou o medicamento que está a ela associado (chaves estrangeiras).

## Estimativa do espaço de disco da base de dados e taxa de crescimento anual

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

## Definição e caracterização dos mecanismos de segurança em SQL

## Revisão do sistema implementado com o utilizador

# Conclusões e Trabalho Futuro

Refletindo no trabalho feito e na implementação atual deste trabalho, considerou-se ainda haverem muitas funcionalidades úteis a acrescentar no futuro. Se nos debruçarmos ainda sobre a venda dos medicamentos facilmente é identificada a necessidade de detalhar melhor os seus preços para ter em conta que um medicamento pode ser comparticipado. Poderia também haver a distinção dos preços de compra e venda para que fosse possível obter os lucros da farmácia. Claro que se considerarmos preços de compra deveríamos também considerar que um fornecedor poderia fornecer o mesmo medicamento a preços diferentes e o preço não mudaria para todas as embalagens no stock, ficando o preço associado a cada entrega.

Refletindo agora nos medicamentos, tendo em conta que cada medicamento pode ter vários princípios ativos em diversas quantidades, era interessante distinguir esta informação até para que fosse possível um farmacêutico obter a lista dos medicamentos que continham um dado princípio ativo. Seria pertinente manter outros atributos do medicamento como o lote e o prazo de validade.

Analisando o funcionamento da farmácia, haveria utilidade em implementar um sistema de horários e calculo de pagamentos que a base de dados teria de suportar.

No mundo real, não é permitido a um farmacêutico dispensar medicamentos que tenham interações medicamentosas conhecidas sem que exista uma justificação bem argumentada, de um médico. Esta é uma funcionalidade muito importante que a base de dados deveria ter implementada futuramente

Anexos

1. Para o povoamento em questão, que tem um intervalo de 7 meses. [↑](#footnote-ref-1)