



Universidade do Minho

Escola de Engenharia

## **Mestrado em Bioinformática**

**Unidade Curricular:**  
Bases de Dados para Bioinformática

# **O AGENDAMENTO E A REALIZAÇÃO DE EXAMES DENTÁRIOS E CONSULTAS NUMA CLÍNICA DENTÁRIA**

### **Trabalho realizado por:**

Miguel Alexandre Oliveira Rocha PG40967

Miguel André Rocha Dias PG40968

Samuel Gustavo Correia Nogueira Alves PG40975

**Docente:** Dr. José Manuel Ferreira Machado



## **INTRODUÇÃO**

Com o crescimento tecnológico acentuado que se tem vindo a verificar ao longo dos anos veio, também, acoplado uma necessidade urgente da catalogação de dados, sendo imperativo que esta fosse tratada de uma forma mais organizada e eficiente. Quando se refere a estabelecimentos saúde, quer hospitais ou clínicas dentárias, está sempre associada uma natureza caótica do próprio ambiente de trabalho, e com o aumentar do número de pacientes/clientes, é importante existir o mais alto nível de organização para evitar infortúnios.

## **Contextualização**

O principal foco deste trabalho foi posto numa clínica dentária hipotética e a sua base de dados para guardar a informação dos clientes, profissionais de saúde e técnicos, e, principalmente, das marcações de exames e consultas, de modo a manter um ambiente organizado na clínica.

## **Objetivos**

Um dos objetivos foi a implementação um sistema de gerenciamento de banco de dados para uma clínica dentária. Com intuito de atingir este mesmo objetivo foi proposto um modelo conceptual, com recurso a uma ferramenta informática TerraER, que permitiu a posterior implementação dum modelo lógico no software MySQL workbench. Para tal, foi imperativo fazer a recolha dos requisitos necessários e organizar os dados obtidos de forma a maximizar a eficiência do sistema e tentar eliminar a redundância dos dados.

## **Recolha de requisitos**

Como o trabalho gira em torno de uma clínica dentária, procurou-se criar uma base de dados que armazena dados referentes à mesma, como os dados relativos a cada funcionário e cliente, nomeadamente os das consultas e exames efetuados, e os seus registos, ou seja, um histórico. Foi, também, necessário que cada cliente tivesse uma ficha clínica associada, documento este que contém todos os dados identificadores do cliente, assim como um historial clínico de tratamentos aos dentes.

A figura 1 dá nos um exemplo de uma típica ficha clínica que se pode encontrar num consultório dentário.

FICHA CLÍNICA		ODONTOGRAMA																
Nome.....																		
Idade..... Profissão.....																		
Residência.....	Telefone.....																	
Serviço Social..... N.º.....																		
Antecedentes clínicos.....																		
História clínica.....																		
Medicação { Anda a tomar..... Medicado (a) com.....																		
Observações:.....																		
Radiografia N.º..... Tartarectomia.....																		
Higiene dentária.....																		
Necessita de.....		Marcar..... consultas, em: <table border="1"> <tr> <td>/</td> <td>às</td> <td>/</td> <td>às</td> </tr> <tr> <td>/</td> <td>às</td> <td>/</td> <td>às</td> </tr> <tr> <td>/</td> <td>às</td> <td>/</td> <td>às</td> </tr> <tr> <td>/</td> <td>às</td> <td>/</td> <td>às</td> </tr> </table>	/	às	/	às	/	às	/	às	/	às	/	às	/	às	/	às
/	às	/	às															
/	às	/	às															
/	às	/	às															
/	às	/	às															

Figura 1: Exemplo de uma ficha clínica dentária.

É possível observar que uma ficha deste género precisa de identificadores básicos de uma pessoa como o nome, idade e contacto. Outras características importantes para a marcação de uma consulta são também o historial clínico da pessoa ou se o indivíduo toma medicação. Devido aos efeitos que certos analgésicos podem ter é importante ter o conhecimento de qualquer medicação que um indivíduo possa estar a ser sujeito para evitar problemas.

Outra informação importante que é exclusiva a fichas clínicas dentárias é o odontograma. Isto é um formulário que está associado a uma ficha que descreve a situação de cada dente. Normalmente representada com o desenho de uma dentadura, cujos dentes estão assinalados com um número, é possível desta forma assinalar os dentes que foram tratados e armazenar a informação. Para cada dente é efetuado tratamentos (quer consulta ou exame) numa data específica com um preço associado (coluna honorários neste caso).

A figura 2 ilustra como é feito este registo num formulário impresso.



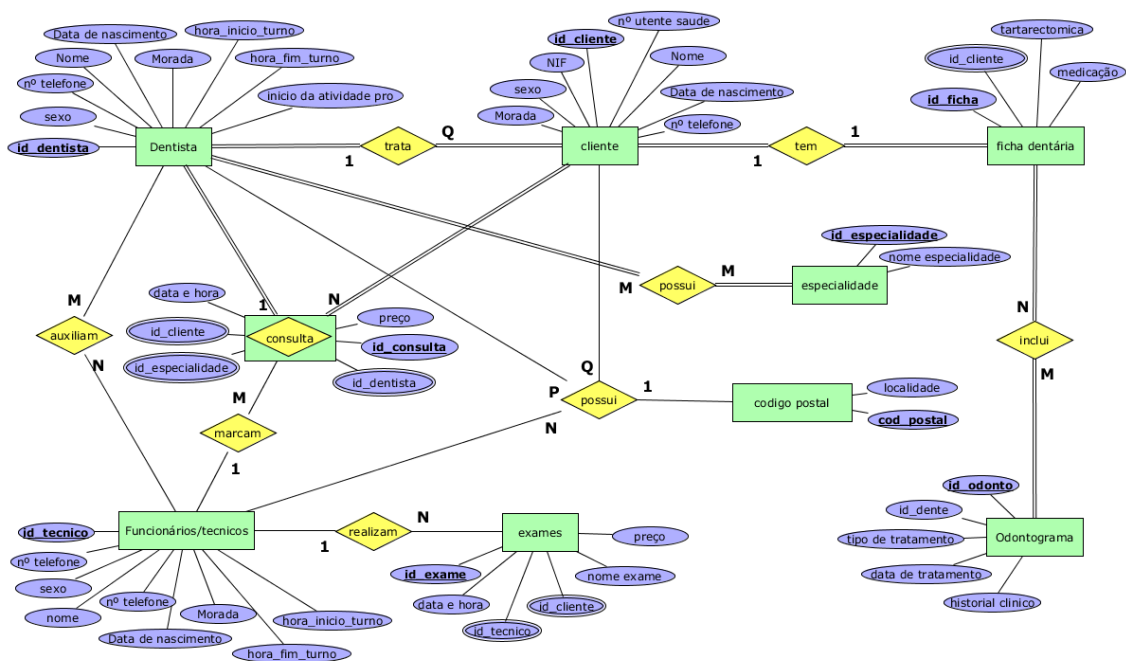


Figura 3: Modelo conceptual esboçado usando programa TerraER para uma base de dados relativa a um estabelecimento dentário.

Foram escolhidas como entidades principais os seguintes parâmetros: dentista, cliente, ficha dentária, odontograma, exames, consultas, especialidade, código postal e funcionários/técnicos.

A identidade clientes vai ter atributos que permitem caracterizar um indivíduo na altura em que se faz um exame ou consulta, como o sexo, nome, nº de telefone assim como 2 atributos únicos, o número de identificação fiscal e o número de utente de saúde. Esta identidade vai ter associada uma única ficha dentaria que nos diz observações acerca do cliente como por exemplo se ele toma medicação que pode interferir no tratamento dos dentes ou se já realizou alguma limpeza aos dentes (tartarectomia) que por sua vez vai ter associada um atributo estrangeiro relativo ao odontograma. Os atributos para a identidade odontograma são o tipo de tratamento realizado, o número do dente a que foi feito o tratamento(id\_dente) e o tipo de tratamento.

Os dentistas vão ter certos atributos de caracterização iguais em comparação com os clientes, no entanto diferem destes no facto de possuírem os atributos de hora de fim de turno e hora de início de turno, dois atributos importantes para se fazer uma marcação, assim como o atributo. Está relacionada com a identidade especialidade assim como a identidade consulta. Técnicos são os que marcam as consultas e tratam dos exames dentários por isso estão associados a esses atributos.

Por fim existe a identidade código postal que se relaciona com 3 outras entidades de forma a guardar o código postal e respetiva localidade associada dos médicos, técnicos e clientes. Esta entidade foi criada separadamente porque um único código postal pode estar associado a vários indivíduos, portanto de modo a evitar redundância de dados foi feita esta decisão.

## Modelo Lógico

Após ser realizado o modelo lógico procedeu-se á tradução numa base de dados real. Com o auxílio do programa mySQL Workbench foi possível obter o diagrama lógico que se pode observar na figura 4.

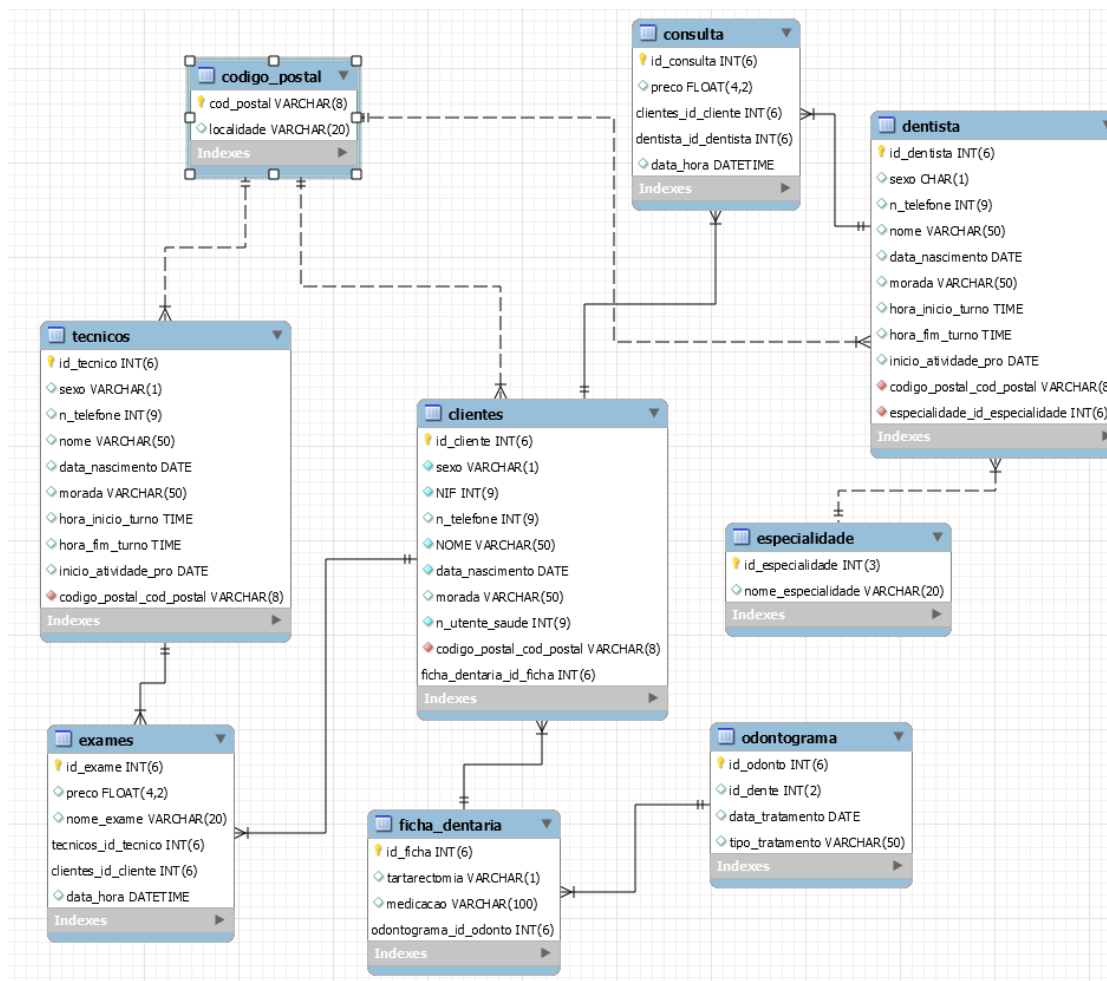


Figura 4: Modelo logico de uma base dados para uma clinica dentaria realizado no mySQL workbench.

Neste modelo logico foi feito em primeiro lugar a criação das entidades utilizando o comando CREATE TABLE, onde foi escrito os atributos assim como o tipo de variável associado. A figura 5 representa um exemplo de uma tabela criada para a entidade clientes.

```

CREATE TABLE IF NOT EXISTS `clinica_dentaria`.`clientes` (
  `id_cliente` INT(6) NOT NULL AUTO_INCREMENT,
  `sexo` VARCHAR(1) NOT NULL,
  `NIF` INT(9) NOT NULL,
  `n_telefone` INT(9) NULL DEFAULT NULL,
  `NOME` VARCHAR(50) NOT NULL,
  `data_nascimento` DATE NOT NULL,
  `morada` VARCHAR(50) NULL DEFAULT NULL,
  `n_utente_saude` INT(9) NOT NULL,
  `codigo_postal_cod_postal` VARCHAR(8) NOT NULL,
  `ficha_dentaria_id_ficha` INT(6) NOT NULL,
  PRIMARY KEY (`id_cliente`, `ficha_dentaria_id_ficha`),
  UNIQUE INDEX `id_cliente` (`id_cliente` ASC) VISIBLE,
  UNIQUE INDEX `NIF` (`NIF` ASC) VISIBLE,
  INDEX `fk_clientes_codigo_postal1_idx` (`codigo_postal_cod_postal` ASC) VISIBLE,
  INDEX `fk_clientes_ficha_dentaria1_idx` (`ficha_dentaria_id_ficha` ASC) VISIBLE,
  CONSTRAINT `fk_clientes_codigo_postal1`
    FOREIGN KEY (`codigo_postal_cod_postal`)
      REFERENCES `clinica_dentaria`.`codigo_postal` (`cod_postal`),
  CONSTRAINT `fk_clientes_ficha_dentaria1`
    FOREIGN KEY (`ficha_dentaria_id_ficha`)
      REFERENCES `clinica_dentaria`.`ficha_dentaria` (`id_ficha`))

```

Figura 5: Código para a criação de uma tabela obtido a partir da funcionalidade forward engineer do MySQL workbench.

Para cada atributo está associado um tipo de variável e constraints caso se apliquem. Atributos que são só números foram colocados com a variável INT e nos parêntesis o número de caracteres necessários enquanto atributos com VARCHAR necessitam quer de letras, números ou outros caracteres para serem caracterizados. Os atributos que necessitam de datas foram caracterizados com a variável DATE ou se fosse necessário saber também a hora, como no caso da entidade dos exames, utiliza-se alternativamente a variável DATETIME. Por fim a variável FLOAT foi usada para caracterizar preços de consultas e exames. Nesta base de dados assumiu-se que os preços não vão acima dos 99 euros, portanto a variável FLOAT tem o valor 4 para indicar o total de casas do número inserido sendo que o 2 significa que duas das casas são decimais.

Na caracterização dos atributos de cada entidade também foi atribuída a constraint de NOT NULL para atributos cujo valor é necessário haver para caracterizar uma entidade, utilizou-se a função AUTOINCREMENT para o sistema atribuir automaticamente números aos atributos de códigos de identificação (ID) para não haver a necessidade do utilizador ter que colocar o input manualmente. Além disso cada entidade tem um atributo principal que foi oferecida a designação de chave primaria e certos atributos que são chaves primarias de outras tabelas foram implementados como chaves estrangeiras.



## Triggers, funções e procedures

Foram implementadas as seguintes funções, procedimentos e triggers na Base de Dados. Para cada uma também está representado o código escrito no mySQL workbench.

**Função Idade:** Faz o cálculo da idade de um cliente, técnico ou médico a partir da sua data de nascimento.

```
DELIMITER //
CREATE FUNCTION idade (dta date) RETURNS INT(3)
DETERMINISTIC
BEGIN
RETURN TIMESTAMPDIFF(YEAR, dta, CURDATE());
END //
DELIMITER ;
```

**Trigger não medicação:** Caso num cliente for inserido um valor NULL no atributo medicação este trigger muda o valor na base de dados para “Não toma medicação”.

```
DELIMITER $$
CREATE TRIGGER nao_Medicao BEFORE INSERT
ON ficha_dentaria
FOR EACH ROW BEGIN
IF NEW.medicacao IS NULL THEN
SET NEW.medicacao = 'Não toma medicação';
END IF;
END$$
DELIMITER;
```

**Trigger não\_tartarectomia:** Insere a letra “N” caso o valor da tartarectomia for NULL, significando que o cliente nunca fez este procedimento.

```
DELIMITER $$
CREATE TRIGGER nao_tartarectomia BEFORE INSERT
ON ficha_dentaria
FOR EACH ROW BEGIN
IF NEW.tartarectomia IS NULL THEN
SET new.tartarectomia = 'N';
END IF;
END$$
\DELIMITER;
```

**Trigger sexo clientes:** Cria uma mensagem numa tabela cada vez que é inserido um novo cliente referindo também o sexo do cliente.

```
CREATE TABLE trigger_dentaria(mensagem VARCHAR(100));
```

```
DELIMITER $$
CREATE TRIGGER sexo_clientes BEFORE INSERT
ON clientes
```

```

FOR EACH ROW BEGIN
IF NEW.sexo = 'M' THEN
INSERT INTO trigger_dentaria VALUES('cliente adicionado do sexo
masculino');
ELSEIF NEW.sexo = 'F' THEN
INSERT INTO trigger_dentaria VALUES('cliente adicionado do sexo
feminino');
END IF;
END$$
DELIMITER ;

```

**Procedure marcações:** Cria uma tabela com todas as marcações, quer de consultas ou de exames, para cada cliente.

```

delimiter //
CREATE DEFINER=`root`@`localhost` PROCEDURE `marcacoes`()
BEGIN
SELECT c.clientes_id_cliente, cl.NOME, c.data_hora as data_hora,
'Consulta' as Tipo, concat('Médico', ' ', c.dentista_id_dentista)
as
Funcionario FROM consulta c, clientes cl WHERE
c.clientes_id_cliente =
cl.id_cliente UNION SELECT e.clientes_id_cliente, cl.NOME,
e.data_hora as data_hora, 'Exame' as Tipo, concat('Técnico', ' ',
e.tecnicos_id_tecnico) as Funcionario
FROM exames e, clientes cl where e.clientes_id_cliente =
cl.id_cliente ORDER BY data_hora desc;
end //
delimiter ;

```

## Povoamento das tabelas:

O povoamento das tabelas foi feito criando um query que utiliza o comando INSERT seguindo a seguinte notação:

```

INSERT INTO nome_Tabela (coluna 1, coluna 2, coluna 3...)
VALUES (valor 1, valor 2, valor 3, ...);

```

Também utilizou-se o comando ALTER TABLE para fazer com que o números de identificação comecem num certo numero seguindo a notação:

```

ALTER table tabela_coluna AUTO_INCREMENT = numero;

```

Em baixo encontra-se representado um exemplo do tipo de dados a inserir para cada tabela.

### Codigo Postal

```
INSERT INTO clinica_dentaria.codigo_postal (cod_postal, localidade)
VALUES ('4710-001', 'Barcelos');
```

### Especialidade

```
INSERT INTO clinica_dentaria.especialidade (id_especialidade,
nome_especialidade) VALUES (301, 'Implantologia');
```

### Dentista

```
ALTER table clinica_dentaria.dentista AUTO INCREMENT = 100000;

INSERT INTO clinica_dentaria.dentista (sexo, n_telefone, nome,
data_nascimento, morada, hora_inicio_turno, hora_fim_turno,
inicio_atividade_pro, codigo_postal_cod_postal,
especialidade_id_especialidade) VALUES ('M', 967895642, 'Jose Maria',
'1970-01-23', 'Rua de Cima n.º 223, 5.º DTO', '14:00:00', '19:00:00',
'1990-02-23', '4710-001', 301);
```

### Odontograma

```
ALTER table clinica_dentaria.odontograma AUTO_INCREMENT = 255000;

INSERT INTO clinica_dentaria.odontograma (id_dente, data_tratamento,
tipo_tratamento) VALUES (28, '2018-03-14', 'Restauração dentária');
```

### Ficha dentaria

```
ALTER table clinica_dentaria.ficha_dentaria AUTO_INCREMENT = 250000;

INSERT INTO clinica_dentaria.ficha_dentaria (tartarectomia, medicacao,
odontograma_id_odonto) VALUES (null, null, 255000);
```

### Clientes

```
ALTER table clinica_dentaria.clientes AUTO_INCREMENT = 200000;

INSERT INTO clinica_dentaria.clientes (sexo, NIF, n_telefone, NOME,
data_nascimento, morada, n_utente_saude, codigo_postal_cod_postal,
ficha_dentaria_id_ficha) VALUES ('M', 500001001, 919999999, 'Manuel
Marques', '1970-01-23', 'Rua de Cima n.º 753, 4.º DTO', 123456789,
'4700-001', 250000);
```

## Técnicos

```
ALTER table clinica_dentaria.tecnicos AUTO_INCREMENT = 300000;  
  
INSERT INTO clinica_dentaria.tecnicos (sexo, n_telefone, nome,  
data_nascimento, morada, hora_inicio_turno, hora_fim_turno,  
inicio_atividade_pro, codigo_postal_cod_postal) VALUES ('M', 960000012,  
'Rui José', '1989-02-03', 'Rua de Canto n.º 223, 1.º DTO', '08:00:00',  
'19:00:00', '2010-02-23', '4710-001');
```

## Consulta

```
ALTER table clinica_dentaria.consulta AUTO_INCREMENT = 400000;  
  
INSERT INTO clinica_dentaria.consulta (preco, clientes_id_cliente,  
dentista_id_dentista, data_hora) VALUES (45, 200007, 100005, '2017-01-  
23 14:30:00');
```

## Exames

```
ALTER table clinica_dentaria.exames AUTO_INCREMENT = 500000;  
  
INSERT INTO clinica_dentaria.exames (preco, nome_exame,  
tecnicos_id_tecnico, clientes_id_cliente, data_hora) VALUES(15,  
'Ortopantomografia', 300000, 200001, '2019-08-17 17:00:00');
```

É de notar que devido às constraints de chaves estrangeiras implementadas o MySQL Workbench não aceita a colocação de certos dados sem a implementação da tabela associada ao atributo estrangeiro. Dessa forma deve-se colocar primeiro dado de tabelas sem chaves estrangeiras e só depois os dados com atributos associados a outras entidades. No caso desta base de dados realizou-se o povoamento em primeiro lugar dos dados relativos aos códigos postais seguido das especialidades e só após estes estarem implementados inseriu-se os dados relativos aos dentistas. A restante ordem de colocação foi do odontograma, ficha dentaria, clientes, técnicos, consultas e finalmente os exames.

## Implementação num programa informático:

Com o objetivo de tornar esta base de dados funcional e acessível aos utilizadores (e.g. médicos, técnicos e funcionários de secretaria), seria necessário a criação de uma interface gráfica (GUI) capacitando os utilizadores a introduzirem e alterarem dados conforme a sua necessidade. Para a criação de um esboço desta GUI, foi usado Microsoft PowerPoint.

Na janela representada na figura 6, é possível consultar e alterar informações do paciente, sendo importante os esquemas dos dentes através de um odontograma para o médico poder sinalizar os dentes tratados. Outra informação de elevada importância é a indicação de uma possível medicação e problemas de saúde, visto poder haver reação com os tratamentos prestados pelo médico.

**Cliente**

Nome: Manuel Marques

Nº de utente: 123456789

Sexo: Masculino

NIF: 500001001

Data de nascimento: 1970-01-23

Cód. postal: 4700-001

Idade: 49

Morada: Rua de Cima nº753 4ºDto

ID: 200000

Contacto: 919999999

Medicação: Não toma medicação

Próxima consulta: Ainda não marcada

Próximo exame: Ainda não marcado

Nova consulta Novo exame

SMILE AWAY THE PAIN™ 2019

Figura 6: Interface gráfica de ficha cliente.

Na figura 7 está representada a janela com informações do médico, necessárias para o bom trabalho do mesmo.

**Médico**

Nome: Manuel Maria Neves

Cód. postal: 4700-005

Sexo: Masculino

Morada: Rua de Sul, nº789 9ºDto

Data de nascimento: 1964-03-24

Contacto: 960085632

ID: 100002

Início do turno: 14 horas

Fim do turno: 19 horas

Idade: 55

Especialidade: Periodontologia

Início atividade profissional: 1979-07-23

FRESH SMILE

SMILE AWAY THE PAIN™ 2019

Figura 7: Interface gráfica de ficha dentista

A implementação de uma interface gráfica de bases de dados poderia ser feita de inúmeras formas. Existem linguagens de programação como o visual basic ou o PHP que foram feitas para esse propósito, no entanto também poderia ser utilizado uma linguagem mais abrangente como o Python. Caso fosse utilizado o Python teria que se fazer o download de dois pacotes, o MySQL Driver que conecta o Python a uma base de dados no MySQL permitindo que a criação de tabelas e inserção de dados possa ser feita no próprio Python, e um pacote de criação de GUI que por exemplo podia ser o tkinter.

## **Análise crítica**

Na realização deste trabalho foram encontrados certos obstáculos que afetaram a fiabilidade da base de dados. A implementação de certos triggers provou ser mais difícil que o esperado devido à complexidade da escrita do código no SQL. Alguns exemplos de triggers que podiam ser implementados são um trigger que alerta se o cliente é demasiado velho para realizar uma odontopediatria ou um trigger que verifica se uma nova marcação tem uma hora que sobrepõe outra marcação prévia. Uma função que calcula as despesas totais de um cliente também foi inicialmente planeada, mas a sua implementação provou ser mais complicado do que se esperava.

Tendo em conta a importância de uma base de dados, em especial quando ligada à saúde, é importante garantir o seu correto funcionamento.

Assim, um grande número de testes é necessário para garantir que funciona sem qualquer erro, garantindo também a escalabilidade da mesma.

Seria também de elevada importância, a implementação de mecanismos capazes de proteger os dados nas mais diferentes situações (e.g. falha de energia). Um mecanismo seria a possibilidade de aceder de modo offline à base de dados, e quando o sistema ficasse online outra vez, as alterações seriam implementadas. Visto ser uma clínica dentária que não realiza internamentos, estando assim fechada após o horário normal de funcionamento (09 horas às 19 horas), poderia ser realizado um backup dos dados durante a noite (não afetando assim o correto acesso à base de dados), tendo cópias desses mesmo dados em locais físicos diferentes, para caso aconteça algum problema no local principal, se poder recuperar os dados.

## **Referencias:**

[https://www.w3schools.com/Python/Python\\_mysql\\_create\\_db.asp](https://www.w3schools.com/Python/Python_mysql_create_db.asp)

## Anexos

### Query das tabelas criadas no mySQL workbench

```
CREATE SCHEMA IF NOT EXISTS `clinica_dentaria` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci ;
USE `clinica_dentaria` ;

-----
-- Table `clinica_dentaria`.`codigo_postal`
-----
CREATE TABLE IF NOT EXISTS `clinica_dentaria`.`codigo_postal` (
  `cod_postal` VARCHAR(8) NOT NULL,
  `localidade` VARCHAR(20) NULL DEFAULT NULL,
  PRIMARY KEY (`cod_postal`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

-----
-- Table `clinica_dentaria`.`odontograma`
-----
CREATE TABLE IF NOT EXISTS `clinica_dentaria`.`odontograma` (
  `id_odonto` INT(6) NOT NULL AUTO_INCREMENT,
  `id_dente` INT(2) NULL DEFAULT NULL,
  `data_tratamento` DATE NULL DEFAULT NULL,
  `tipo_tratamento` VARCHAR(50) NULL DEFAULT NULL,
  PRIMARY KEY (`id_odonto`),
  UNIQUE INDEX `id_odonto` (`id_odonto` ASC) VISIBLE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

-----
-- Table `clinica_dentaria`.`ficha_dentaria`
-----
CREATE TABLE IF NOT EXISTS `clinica_dentaria`.`ficha_dentaria` (
  `id_ficha` INT(6) NOT NULL AUTO_INCREMENT,
  `tartarectomia` VARCHAR(1) NULL DEFAULT NULL,
  `medicacao` VARCHAR(100) NULL DEFAULT NULL,
  `odontograma_id_odonto` INT(6) NOT NULL,
  PRIMARY KEY (`id_ficha`, `odontograma_id_odonto`),
  UNIQUE INDEX `id_ficha` (`id_ficha` ASC) VISIBLE,
  INDEX `fk_ficha_dentaria_odontograma1_idx` (`odontograma_id_odonto` ASC) VISIBLE,
  CONSTRAINT `fk_ficha_dentaria_odontograma1`
    FOREIGN KEY (`odontograma_id_odonto`)
    REFERENCES `clinica_dentaria`.`odontograma` (`id_odonto`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

```

-----
-- Table `clinica_dentaria`.`clientes`
-----

CREATE TABLE IF NOT EXISTS `clinica_dentaria`.`clientes` (
  `id_cliente` INT(6) NOT NULL AUTO_INCREMENT,
  `sexo` VARCHAR(1) NOT NULL,
  `NIF` INT(9) NOT NULL,
  `n_telefone` INT(9) NULL DEFAULT NULL,
  `NOME` VARCHAR(50) NOT NULL,
  `data_nascimento` DATE NOT NULL,
  `morada` VARCHAR(50) NULL DEFAULT NULL,
  `n_utente_saude` INT(9) NOT NULL,
  `codigo_postal_cod_postal` VARCHAR(8) NOT NULL,
  `ficha_dentaria_id_ficha` INT(6) NOT NULL,
  PRIMARY KEY (`id_cliente`, `ficha_dentaria_id_ficha`),
  UNIQUE INDEX `id_cliente` (`id_cliente` ASC) VISIBLE,
  UNIQUE INDEX `NIF` (`NIF` ASC) VISIBLE,
  INDEX `fk_clientes_codigo_postal1_idx` (`codigo_postal_cod_postal` ASC) VISIBLE,
  INDEX `fk_clientes_ficha_dentaria1_idx` (`ficha_dentaria_id_ficha` ASC) VISIBLE,
  CONSTRAINT `fk_clientes_codigo_postal1`
    FOREIGN KEY (`codigo_postal_cod_postal`)
      REFERENCES `clinica_dentaria`.`codigo_postal` (`cod_postal`),
  CONSTRAINT `fk_clientes_ficha_dentaria1`
    FOREIGN KEY (`ficha_dentaria_id_ficha`)
      REFERENCES `clinica_dentaria`.`ficha_dentaria` (`id_ficha`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
-----

-- Table `clinica_dentaria`.`especialidade`
-----

CREATE TABLE IF NOT EXISTS `clinica_dentaria`.`especialidade` (
  `id_especialidade` INT(3) NOT NULL,
  `nome_especialidade` VARCHAR(20) NULL DEFAULT NULL,
  PRIMARY KEY (`id_especialidade`),
  UNIQUE INDEX `id_especialidade` (`id_especialidade` ASC) VISIBLE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

```



```

-----
-- Table `clinica_dentaria`.`dentista`
-----
CREATE TABLE IF NOT EXISTS `clinica_dentaria`.`dentista` (
  `id_dentista` INT(6) NOT NULL AUTO_INCREMENT,
  `sexo` CHAR(1) NULL DEFAULT NULL,
  `n_telefone` INT(9) NULL DEFAULT NULL,
  `nome` VARCHAR(50) NULL DEFAULT NULL,
  `data_nascimento` DATE NULL DEFAULT NULL,
  `morada` VARCHAR(50) NULL DEFAULT NULL,
  `hora_inicio_turno` TIME NULL DEFAULT NULL,
  `hora_fim_turno` TIME NULL DEFAULT NULL,
  `inicio_atividade_pro` DATE NULL DEFAULT NULL,
  `codigo_postal_cod_postal` VARCHAR(8) NOT NULL,
  `especialidade_id_especialidade` INT(6) NOT NULL,
  PRIMARY KEY (`id_dentista`),
  UNIQUE INDEX `id_dentista` (`id_dentista` ASC) VISIBLE,
  INDEX `fk_dentista_codigo_postal1_idx` (`codigo_postal_cod_postal` ASC) VISIBLE,
  INDEX `fk_dentista_especialidade1_idx` (`especialidade_id_especialidade` ASC) VISIBLE,
  CONSTRAINT `fk_dentista_codigo_postal1`
    FOREIGN KEY (`codigo_postal_cod_postal`)
    REFERENCES `clinica_dentaria`.`codigo_postal` (`cod_postal`),
  CONSTRAINT `fk_dentista_especialidade1`
    FOREIGN KEY (`especialidade_id_especialidade`)
    REFERENCES `clinica_dentaria`.`especialidade` (`id_especialidade`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

```

```

-----
-- Table `clinica_dentaria`.`consulta`
-----
CREATE TABLE IF NOT EXISTS `clinica_dentaria`.`consulta` (
  `id_consulta` INT(6) NOT NULL AUTO_INCREMENT,
  `preco` FLOAT(4,2) NULL DEFAULT NULL,
  `clientes_id_cliente` INT(6) NOT NULL,
  `dentista_id_dentista` INT(6) NOT NULL,
  `data_hora` DATETIME NULL DEFAULT NULL,
  PRIMARY KEY (`id_consulta`, `clientes_id_cliente`, `dentista_id_dentista`),
  UNIQUE INDEX `id_consulta` (`id_consulta` ASC) VISIBLE,
  INDEX `fk_consulta_clientes1_idx` (`clientes_id_cliente` ASC) VISIBLE,
  INDEX `fk_consulta_dentista1_idx` (`dentista_id_dentista` ASC) VISIBLE,
  CONSTRAINT `fk_consulta_clientes1`
    FOREIGN KEY (`clientes_id_cliente`)
      REFERENCES `clinica_dentaria`.`clientes` (`id_cliente`),
  CONSTRAINT `fk_consulta_dentista1`
    FOREIGN KEY (`dentista_id_dentista`)
      REFERENCES `clinica_dentaria`.`dentista` (`id_dentista`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

-----
-- Table `clinica_dentaria`.`tecnicos`
-----
CREATE TABLE IF NOT EXISTS `clinica_dentaria`.`tecnicos` (
  `id_tecnico` INT(6) NOT NULL AUTO_INCREMENT,
  `sexo` VARCHAR(1) NULL DEFAULT NULL,
  `n_telefone` INT(9) NULL DEFAULT NULL,
  `nome` VARCHAR(50) NULL DEFAULT NULL,
  `data_nascimento` DATE NULL DEFAULT NULL,
  `morada` VARCHAR(50) NULL DEFAULT NULL,
  `hora_inicio_turno` TIME NULL DEFAULT NULL,
  `hora_fim_turno` TIME NULL DEFAULT NULL,
  `inicio_atividade_pro` DATE NULL DEFAULT NULL,
  `codigo_postal_cod_postal` VARCHAR(8) NOT NULL,
  PRIMARY KEY (`id_tecnico`),
  UNIQUE INDEX `id_tecnico` (`id_tecnico` ASC) VISIBLE,
  INDEX `fk_tecnicos_codigo_postal1_idx` (`codigo_postal_cod_postal` ASC) VISIBLE,
  CONSTRAINT `fk_tecnicos_codigo_postal1`
    FOREIGN KEY (`codigo_postal_cod_postal`)
      REFERENCES `clinica_dentaria`.`codigo_postal` (`cod_postal`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

```

```

-----
-- Table `clinica_dentaria`.`exames`
-----

CREATE TABLE IF NOT EXISTS `clinica_dentaria`.`exames` (
  `id_exame` INT(6) NOT NULL AUTO_INCREMENT,
  `preco` FLOAT(4,2) NULL DEFAULT NULL,
  `nome_exame` VARCHAR(20) NULL DEFAULT NULL,
  `tecnicos_id_tecnico` INT(6) NOT NULL,
  `clientes_id_cliente` INT(6) NOT NULL,
  `data_hora` DATETIME NULL DEFAULT NULL,
  PRIMARY KEY (`id_exame`, `tecnicos_id_tecnico`, `clientes_id_cliente`),
  UNIQUE INDEX `id_exame` (`id_exame` ASC) VISIBLE,
  INDEX `fk_exames_tecnicos1_idx` (`tecnicos_id_tecnico` ASC) VISIBLE,
  INDEX `fk_exames_clientes1_idx` (`clientes_id_cliente` ASC) VISIBLE,
  CONSTRAINT `fk_exames_clientes1`
    FOREIGN KEY (`clientes_id_cliente`)
      REFERENCES `clinica_dentaria`.`clientes` (`id_cliente`),
  CONSTRAINT `fk_exames_tecnicos1`
    FOREIGN KEY (`tecnicos_id_tecnico`)
      REFERENCES `clinica_dentaria`.`tecnicos` (`id_tecnico`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

```