

# Alunos

Helio de Jesus dos Santos Junior

Lais Costa Santos Teixeira

Lucas Balbino Santos da Silva

## Criação das Tabelas

```
-- Table: doença
CREATE TABLE doença (
  id int NOT NULL AUTO_INCREMENT,
  nome varchar(100) NOT NULL,
  descricao Text NOT NULL,
  CONSTRAINT doença_pk PRIMARY KEY (id)
);

-- Table: equipe_saude
CREATE TABLE equipe_saude (
  id int NOT NULL AUTO_INCREMENT,
  nome varchar(100) NOT NULL,
  cargo varchar(100) NOT NULL,
  especialidade varchar(100) NOT NULL,
  CONSTRAINT equipe_saude_pk PRIMARY KEY (id)
);

-- Table: paciente
CREATE TABLE paciente (
  id int NOT NULL AUTO_INCREMENT,
  nome varchar(100) NOT NULL,
  cpf varchar(14) NOT NULL UNIQUE,
  idade tinyint NOT NULL,
  genero ENUM("MASCULINO", "FEMININO", "OUTRO") NOT
NULL,
  endereco varchar(255) NOT NULL,
  CONSTRAINT paciente_pk PRIMARY KEY (id)
);
```

```
-- Table: tratamento
CREATE TABLE tratamento (
    id int NOT NULL AUTO_INCREMENT,
    nome varchar(100) NOT NULL,
    descricao Text NOT NULL,
    CONSTRAINT tratamento_pk PRIMARY KEY (id)
);

-- Table: registro_medico
CREATE TABLE registro_medico (
    id int NOT NULL AUTO_INCREMENT,
    data_registro datetime NOT NULL,
    paciente_id int NOT NULL,
    tratamento_id int NOT NULL,
    doenca_id int NOT NULL,
    equipe_saude_id int NOT NULL,
    CONSTRAINT registro_medico_pk PRIMARY KEY (id)
);

-- foreign keys:
ALTER TABLE registro_medico ADD CONSTRAINT
registro_medico_doenca FOREIGN KEY (doenca_id)
REFERENCES doenca (id);

ALTER TABLE registro_medico ADD CONSTRAINT
registro_medico_equipe_saude FOREIGN KEY
(equipe_saude_id)
REFERENCES equipe_saude (id);

ALTER TABLE registro_medico ADD CONSTRAINT
registro_medico_paciente FOREIGN KEY (paciente_id)
REFERENCES paciente (id);

ALTER TABLE registro_medico ADD CONSTRAINT
registro_medico_tratamento FOREIGN KEY (tratamento_id)
REFERENCES tratamento (id);

-- Alter Table: restrição de unicidade
```

```
ALTER TABLE registro_medico  
ADD CONSTRAINT registro_unico UNIQUE (paciente_id,  
data_registro);
```

## Criação de Triggers

```
CREATE TRIGGER formatar_cpf_before_insert
BEFORE INSERT ON paciente FOR EACH ROW
BEGIN
    DECLARE cpf_temp VARCHAR(14);
    SET cpf_temp = NEW.cpf;

    -- Verifica se o CPF já está no formato desejado
    IF CHAR_LENGTH(cpf_temp) = 14 THEN
        SET NEW.cpf = cpf_temp;
    ELSE
        SET NEW.cpf =
            CONCAT(
                SUBSTRING(cpf_temp, 1, 3), '.',
                SUBSTRING(cpf_temp, 4, 3), '.',
                SUBSTRING(cpf_temp, 7, 3), '-',
                SUBSTRING(cpf_temp, 10, 2)
            );
    END IF;
END; //
```

## Inserts

```
-- Inserções na tabela doença
INSERT INTO doença (nome, descricao) VALUES
    ('Gripe', 'Infecção viral comum causando febre e dor de garganta.'),
    ('Fratura', 'Osso quebrado devido a um acidente.'),
    ('Hipertensão', 'Pressão arterial elevada, muitas vezes relacionada ao estresse.');
```

```
-- Inserções na tabela equipe_saude
INSERT INTO equipe_saude (nome, cargo, especialidade)
VALUES
    ('Dr. Smith', 'Médico', 'Cardiologista'),
    ('Enfermeira Johnson', 'Enfermeira', 'Cuidados Intensivos'),
```

```

('Dr. Garcia', 'Cirurgião', 'Ortopedia');

-- Inserções na tabela paciente
-- Inserção de paciente com CPF não formatado
INSERT INTO paciente (nome, idade, cpf, genero, endereco)
VALUES
('Fulano', 30, '12345678900', 'MASCULINO', 'Rua A, 123'),
-- Inserção de paciente com CPF formatado corretamente
('Ciclano', 25, '123.456.789-00', 'FEMININO', 'Rua B,
456'),
('Beltrano', 40, '789.987.654-32', 'OUTRO', 'Rua C,
789');

-- Inserções na tabela tratamento
INSERT INTO tratamento (nome, descricao) VALUES
('Fisioterapia', 'Reabilitação física para recuperação
de lesões'),
('Medicação para gripe', 'Antiviral para tratar
sintomas gripais'),
('Cirurgia de Fratura', 'Procedimento cirúrgico para
reparar ossos quebrados');

-- Inserções na tabela registro_medico
INSERT INTO registro_medico (data_registro, paciente_id,
tratamento_id, doença_id, equipe_saude_id) VALUES
('2023-11-01 10:10:00', 1, 1, 1, 1),
('2023-11-02 11:30:00', 2, 2, 2, 2),
('2023-11-03 09:45:00', 3, 3, 3, 3);

```

## Selects

```

-- Busca Simples
SELECT * FROM doença;
SELECT * FROM equipe_saude;
SELECT * FROM paciente;
SELECT * FROM tratamento;
-- Buscar Tratamento Específico:

```

```
SELECT *
FROM tratamento
WHERE nome = 'Fisioterapia';

-- Buscar Membro da Equipe de Saúde Específico:
SET @nome_equipe_saude := 'Smith';

SELECT *
FROM equipe_saude
WHERE nome LIKE CONCAT('%', @nome_equipe_saude, '%');

-- Buscar Paciente pelo CPF:
SELECT *
FROM paciente
WHERE cpf = '123.456.789-00';

-- Buscar Registro Médico Específico:
SELECT *
FROM registro_medico
WHERE id = 1;

-- Consulta para Ver Consultas Agendadas para um
determinado Dia:
SELECT *
FROM registro_medico
WHERE DATE(data_registro) = '2023-12-01';

-- Consulta para Histórico de Tratamentos por Paciente:
SELECT p.nome AS paciente, d.nome AS doenca, t.nome AS
tratamento, rm.data_registro
FROM registro_medico rm
JOIN paciente p ON rm.paciente_id = p.id
JOIN doenca d ON rm.doenca_id = d.id
JOIN tratamento t ON rm.tratamento_id = t.id
WHERE p.id = 1;

-- Consultar Pacientes com Tratamento de uma Doença
Específica:
```

```
SELECT p.nome AS nome_paciente, d.nome AS doenca,  
r.data_registro  
FROM registro_medico r  
JOIN paciente p ON r.paciente_id = p.id  
JOIN doenca d ON r.doenca_id = d.id  
WHERE d.nome = 'Gripe';
```

# Requisitos Funcionais e Não Funcionais

## Requisitos Funcionais (RF):

1. **RF1: Cadastro de Doenças, Equipe de Saúde, Pacientes e Tratamentos**
  - Permitir a inserção, atualização e remoção de informações nas tabelas.
2. **RF2: Registro Médico de Consultas**
  - Possibilitar o registro de consultas médicas, relacionando paciente, doença, equipe de saúde envolvida e tratamento aplicado.
3. **RF3: Restrição de Unicidade nos Registros Médicos**
  - Garantir que não existam registros duplicados para um mesmo paciente na mesma data/hora.
4. **RF4: Consultas e Relatórios**
  - Permitir a busca e geração de relatórios, por exemplo, consultas agendadas para um determinado dia ou histórico de tratamentos por paciente.

## Requisitos Não Funcionais (RNF):

1. **RNF1: Segurança dos Dados**
  - Garantir a segurança dos dados sensíveis, como informações médicas dos pacientes, por meio de criptografia e controles de acesso adequados.
2. **RNF2: Desempenho e Escalabilidade**
  - O sistema deve ser capaz de lidar com um grande volume de dados e consultas de maneira eficiente, mantendo um desempenho aceitável mesmo com o aumento da carga.
3. **RNF3: Usabilidade e Interface Intuitiva**
  - A interface do sistema deve ser intuitiva e de fácil utilização, permitindo que os profissionais da saúde realizem suas operações de maneira simples e eficaz.
4. **RNF4: Disponibilidade e Tolerância a Falhas**



- O sistema deve ser resiliente a falhas, garantindo uma alta disponibilidade para que as informações estejam sempre acessíveis quando necessário.

#### **5. RNF5: Conformidade com Padrões e Regulamentações**

- O sistema deve estar em conformidade com regulamentações e padrões de segurança, privacidade e armazenamento de dados na área da saúde.