

DENOMINAÇÃO DO DOCUMENTO:

Projeto Banco de Dados - Quarantine

ABRANGÊNCIA DE USO: Ciência da Computação	RA 27838	PRÓXIMA REVISÃO: -	DATA INÍCIO VIGÊNCIA: 22/05/2025
ÁREA RESPONSÁVEL: TECNOLOGIA DA INFORMAÇÃO	CLASSIFICAÇÃO: RESTRITO		
ELABORADO POR: João Batista			

Sumário

1.	OBJETIVO	3
2.	MODELO CONCEITUAL (MER)	4
3.	MODELO LÓGICO (DER)	5
4.	MODELO FÍSICO (SQL)	6
5.	SCRIPTS DE TESTE	7
6.	CONCLUSÃO	8

1. Objetivo

Este projeto tem como objetivo o desenvolvimento da modelagem de banco de dados para um sistema web utilizado por atendentes de bancos de sangue. O sistema permite registrar doadores, aplicar regras de quarentena, controlar agendamentos e emitir notificações de liberação automática.

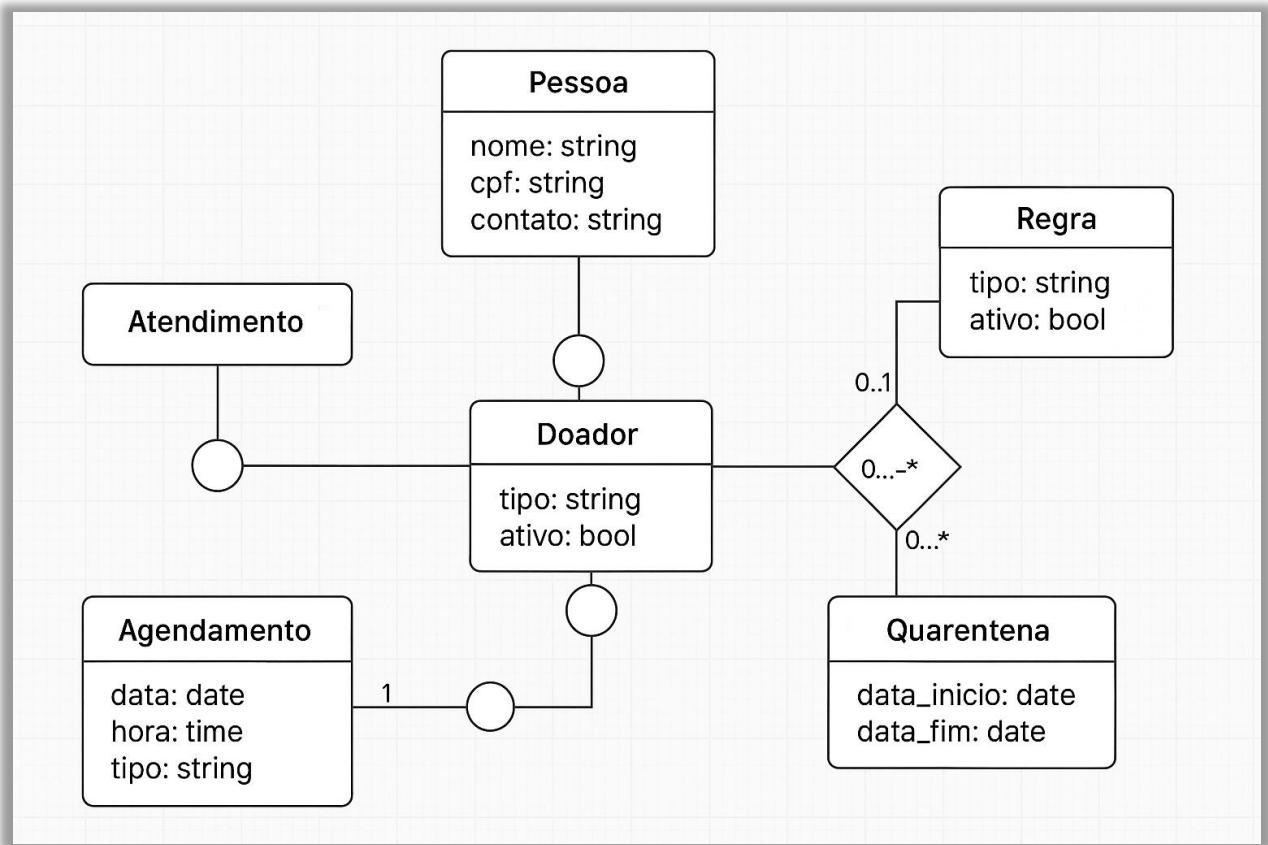
2. Modelo Conceitual (MER)

Entidades Identificadas:

- **Pessoa**: entidade genérica contendo nome, CPF e contato
- **Doador**: especialização de Pessoa com tipo sanguíneo e status ativo
- **Atendimento**: especialização de Pessoa (por ora sem atributos)
- **Regra**: define as condições de quarentena (ex: tatuagem, cirurgia) com quantidade de dias
- **Quarentena**: registra uma quarentena aplicada a um doador
- **Agendamento**: registra interações ou visitas programadas com os doadores

Relacionamentos:

- Pessoa → Doador e Atendimento (**herança**)
- Doador **1:N** Quarentena
- Quarentena **N:1** Regra
- Doador **1:N** Agendamento



3. Modelo Lógico (DER)

Entidades Identificadas:

Entidade	Atributo	Tipo	Chave
Pessoa	id_pessoa	INT	PK
Pessoa	nome	VARCHAR(100)	
Pessoa	cpf	VARCHAR(14)	
Pessoa	contato	VARCHAR(100)	
Doador	id_doador	INT	PK, FK (Pessoa)
Doador	tipo	VARCHAR(50)	
Doador	ativo	BOOLEAN	
Atendimento	id_atendimento	INT	PK, FK (Pessoa)
Regra	id_regra	INT	PK
Regra	condicao	VARCHAR(100)	
Regra	dias_quarentena	INT	
Quarentena	id_quarentena	INT	PK
Quarentena	data_inicio	DATE	
Quarentena	data_fim	DATE	
Quarentena	id_doador	INT	FK (Doador)
Quarentena	id_regra	INT	FK (Regra)
Agendamento	id_agendamento	INT	PK
Agendamento	data	DATE	
Agendamento	hora	TIME	
Agendamento	tipo	VARCHAR(50)	
Agendamento	id_doador	INT	FK (Doador)

4. Modelo Físico (SQL)

```
CREATE TABLE Pessoa (
    id_pessoa INT PRIMARY KEY,
    nome VARCHAR(100),
    cpf VARCHAR(14),
    contato VARCHAR(100)
);

CREATE TABLE Doador (
    id_doador INT PRIMARY KEY,
    tipo VARCHAR(50),
    ativo BOOLEAN,
    FOREIGN KEY (id_doador) REFERENCES Pessoa(id_pessoa)
);

CREATE TABLE Atendimento (
    id_atendimento INT PRIMARY KEY,
    FOREIGN KEY (id_atendimento) REFERENCES Pessoa(id_pessoa)
);

CREATE TABLE Regra (
    id_regra INT PRIMARY KEY,
    condicao VARCHAR(100),
    dias_quarentena INT
);

CREATE TABLE Quarentena (
    id_quarentena INT PRIMARY KEY,
    data_inicio DATE,
    data_fim DATE,
    id_doador INT,
    id_regra INT,
    FOREIGN KEY (id_doador) REFERENCES Doador(id_doador),
    FOREIGN KEY (id_regra) REFERENCES Regra(id_regra)
);

CREATE TABLE Agendamento (
    id_agendamento INT PRIMARY KEY,
    data DATE,
    hora TIME,
    tipo VARCHAR(50),
    id_doador INT,
    FOREIGN KEY (id_doador) REFERENCES Doador(id_doador)
);
```

5. Scripts de Teste

-- Inserção de dados de exemplo

```
INSERT INTO Pessoa (id_pessoa, nome, cpf, contato) VALUES
(1, 'João Silva', '123.456.789-00', 'joao@email.com'),
(2, 'Maria Souza', '987.654.321-00', 'maria@email.com');
```

```
INSERT INTO Doador (id_doador, tipo, ativo) VALUES
(1, 'O+', TRUE),
(2, 'A-', FALSE);
```

```
INSERT INTO Regra (id_regra, condicao, dias_quarentena) VALUES
(1, 'Tatuagem', 30),
(2, 'Cirurgia Dentária', 7);
```

```
INSERT INTO Quarentena (id_quarentena, data_inicio, data_fim, id_doador, id_regra) VALUES
(1, '2025-05-01', '2025-05-31', 1, 1),
(2, '2025-05-15', '2025-05-22', 2, 2);
```

```
INSERT INTO Agendamento (id_agendamento, data, hora, tipo, id_doador) VALUES
(1, '2025-06-01', '09:00:00', 'Doação', 1),
(2, '2025-06-05', '10:30:00', 'Revisão', 2);
```

Consultas úteis:

-- Dadores com quarentena e suas regras

```
SELECT d.id_doador, p.nome, q.data_inicio, q.data_fim, r.condicao
FROM Doador d
JOIN Pessoa p ON d.id_doador = p.id_pessoa
LEFT JOIN Quarentena q ON d.id_doador = q.id_doador
LEFT JOIN Regra r ON q.id_regra = r.id_regra;
```

-- Dadores liberados hoje (simulação)

```
SELECT p.nome, q.data_fim
FROM Quarentena q
JOIN Doador d ON q.id_doador = d.id_doador
JOIN Pessoa p ON d.id_doador = p.id_pessoa
WHERE q.data_fim = CURDATE();
```

-- Histórico de quarentenas por dador

```
SELECT p.nome, r.condicao, q.data_inicio, q.data_fim
FROM Quarentena q
JOIN Regra r ON q.id_regra = r.id_regra
JOIN Doador d ON q.id_doador = d.id_doador
JOIN Pessoa p ON d.id_doador = p.id_pessoa
ORDER BY p.nome, q.data_inicio;
```

6. Conclusão

O modelo proposto atende aos critérios de:

- *Normalização*
- *Separação por responsabilidade*
- *Supporte a históricos*
- *Regra de negócio clara: controle de quarentena por condição com data de liberação automatizada*

O banco está pronto para ser implementado em sistemas reais com persistência em MariaDB ou PostgreSQL.