

DOCUMENTAÇÃO DO SISTEMA DE GESTÃO DE FARMÁCIA

Dayane Pontes

Erick Silva

Rafael Sabino

1. INTRODUÇÃO

O sistema de gestão de farmácia foi projetado para gerenciar as vendas de medicamentos, registrar informações sobre colaboradores, clientes, estoque de medicamentos e realizar relatórios de vendas. A estrutura do banco de dados garante a integridade das informações, atendendo às regras de negócio.

❖ MINIMUNDO

A farmácia utiliza um sistema para gerenciar a venda de medicamentos, permitindo que colaboradores (farmacêuticos e atendentes) registrem vendas, acompanhem o estoque e gerem relatórios. O sistema deve garantir que as operações atendam às regras de negócio e requisitos legais.

❖ REGRAS DE NEGÓCIO

Autenticação de Colaboradores: Todos os colaboradores devem se autenticar no sistema para realizar vendas. As credenciais devem ser verificadas, e a senha deve ser armazenada de forma segura.

Permissões de Acesso: O sistema deve permitir a realização de vendas apenas para colaboradores autenticados.

Registro de Vendas: Cada venda deve incluir informações como data, horário, cliente, colaborador responsável, medicamentos vendidos e total da venda. O sistema deve calcular automaticamente o total com base no preço unitário e na quantidade de medicamentos vendidos.

Controle de Estoque: Antes de confirmar uma venda, o sistema deve verificar se há estoque suficiente dos medicamentos. Se o estoque for insuficiente, a venda não pode ser realizada. Após a venda, o estoque dos medicamentos deve ser atualizado automaticamente, subtraindo a quantidade vendida.

Cadastro de Clientes: Os clientes podem ser cadastrados com informações como nome, CPF e telefone. O sistema deve permitir a consulta do histórico de compras de cada cliente, que pode ser acessado apenas por colaboradores.

Relatórios de Vendas: O sistema deve permitir que colaboradores acessem relatórios detalhados sobre vendas, incluindo total arrecadado, medicamentos mais vendidos e análises de desempenho em períodos específicos.

2- ESTRUTURA DO BANCO DO DADOS

-- CREATES

```
CREATE TABLE Colaboradores (  
ID_colaborador SERIAL PRIMARY KEY,  
nome VARCHAR(100) NOT NULL,  
cargo VARCHAR(50),  
senha Bytea NOT NULL,  
CPF_colaborador VARCHAR(14) UNIQUE NOT NULL  
);
```

```
CREATE TABLE Clientes (  
ID_cliente SERIAL PRIMARY KEY,  
nome VARCHAR(100) NOT NULL,  
CPF VARCHAR(14) UNIQUE NOT NULL,  
telefone VARCHAR(15)  
);
```

```
CREATE TABLE Medicamentos (  
ID_medicamento SERIAL PRIMARY KEY,  
nome VARCHAR(100) NOT NULL,  
categoria VARCHAR(50),  
preco DECIMAL(10, 2) NOT NULL,  
estoque INT NOT NULL  
);
```

```
CREATE TABLE Vendas (  
ID_venda SERIAL PRIMARY KEY,
```

```
data_venda DATE NOT NULL,  
total DECIMAL(10, 2) NOT NULL,  
ID_colaborador INT REFERENCES Colaboradores(ID_colaborador),  
cpf VARCHAR(11) REFERENCES Clientes(cpf)  
);
```

```
CREATE TABLE Item_Venda (  
ID_item_venda SERIAL PRIMARY KEY,  
quantidade INT NOT NULL,  
preco_unitario DECIMAL(10, 2) NOT NULL,  
ID_venda INT REFERENCES Vendas(ID_venda),  
ID_medicamento INT REFERENCES Medicamentos(ID_medicamento)  
);
```

-- INSERTS

```
INSERT INTO Clientes (nome, CPF, telefone) VALUES  
( 'Luiz Souza', '123.456.789-00', '(82) 99999-1111'),  
( 'Carla Lima', '234.567.890-01', '(82) 99999-2222'),  
( 'Pedro Almeida', '345.678.901-02', '(82) 99999-3333'),  
( 'Sofia Ribeiro', '456.789.012-03', '(82) 99999-4444'),  
( 'Lucas Martins', '567.890.123-04', '(82) 99999-5555'),  
( 'Juliana Rocha', '678.901.234-05', '(82) 99999-6666'),  
( 'Marcos Ferreira', '789.012.345-06', '(82) 99999-7777'),  
( 'Fernanda Costa', '890.123.456-07', '(82) 99999-8888'),  
( 'Diego Barbosa', '901.234.567-08', '(82) 99999-9999'),  
( 'Patrícia Moraes', '012.345.678-09', '(82) 98888-0000'),  
( 'Leonardo Mendes', '123.456.789-10', '(82) 98888-1111'),
```

('Jéssica Santos', '234.567.890-11', '(82) 98888-2222'),
('Renata Martins', '345.678.901-12', '(82) 98888-3333'),
('Thiago Nunes', '456.789.012-13', '(82) 98888-4444'),
('Gabriela Alves', '567.890.123-14', '(82) 98888-5555'),
('Rodrigo Silva', '678.901.234-15', '(82) 98888-6666'),
('Bianca Lopes', '789.012.345-16', '(82) 98888-7777'),
('Lucas Pereira', '890.123.456-17', '(82) 98888-8888'),
('Ana Cardoso', '901.234.567-18', '(82) 98888-9999');

INSERT INTO Medicamentos (nome, categoria, preco, estoque) VALUES

('Paracetamol', 'Analgésico', 4.99, 100),
('Ibuprofeno', 'Antiinflamatório', 7.49, 200),
('Amoxicilina', 'Antibiótico', 14.99, 150),
('Omeprazol', 'Antiácido', 9.99, 80),
('Losartana', 'Antihipertensivo', 15.99, 60),
('Simeticona', 'Antigases', 5.49, 120),
('Captopril', 'Antihipertensivo', 8.99, 50),
('Cetirizina', 'Antialérgico', 10.49, 90),
('Diclofenaco', 'Antiinflamatório', 6.99, 100),
('AAS', 'Antipirético', 3.99, 150),
('Clonazepam', 'Ansiolítico', 18.49, 70),
('Lorazepam', 'Ansiolítico', 22.99, 60),
('Metformina', 'Antidiabético', 12.99, 200),
('Insulina', 'Hormônio', 40.99, 30),
('Ranitidina', 'Antiácido', 11.99, 90),

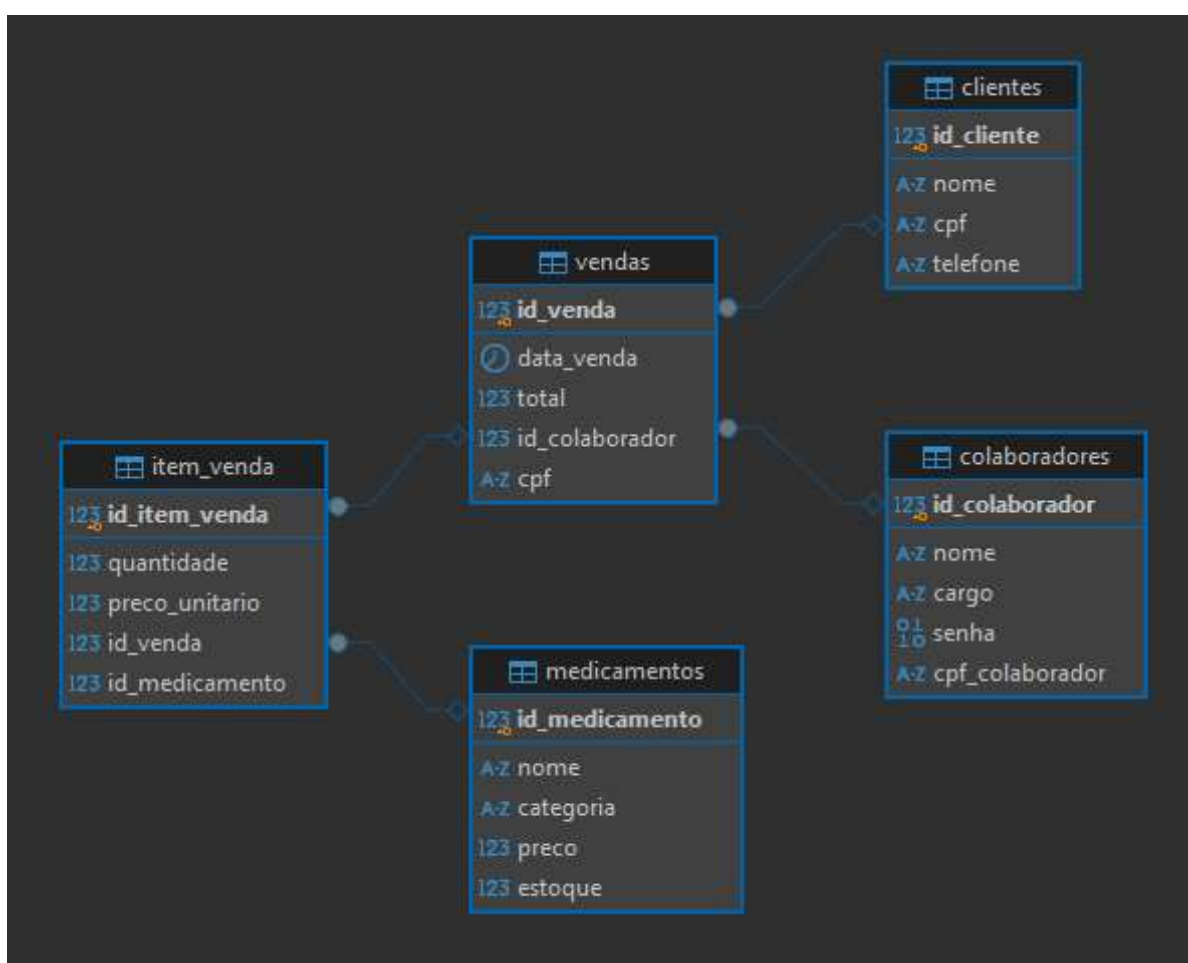
❖ RELACIONAMENTOS:

Colaboradores ↔ Vendas: Um colaborador pode realizar várias vendas, mas uma venda é registrada por apenas um colaborador.

Clientes ↔ Vendas: Um cliente pode realizar várias compras, mas cada venda é associada a um cliente específico.

Vendas ↔ Item_Venda: Uma venda pode conter vários itens, enquanto cada item pertence a uma venda específica.

Medicamentos ↔ Item_Venda: Um medicamento pode ser incluído em várias vendas, mas cada item de venda refere-se a um único medicamento.



(Figura 1: Diagrama de Entidade-Relacionamento – DER)

1) CRUD – CLIENTES

FUNÇÃO	SQL
Cadastrar Cliente	INSERT INTO Clientes (nome, CPF, telefone) VALUES ('João da Silva', '123.456.789-00', '(82) 99999-1111');
Listar Todos Clientes	SELECT * FROM Clientes;

Buscar Cliente por Nome	SELECT * FROM Clientes WHERE ID_cliente = 10;
Buscar Cliente por CPF	SELECT * FROM Clientes WHERE nome = 'Maria Souza';
Atualizar Telefone de um Cliente	UPDATE Clientes SET telefone = '(82) 99999-1221' WHERE ID_cliente = 2;
Deletar Cliente	DELETE FROM Clientes WHERE ID_cliente = 5;

2) CRUD – MEDICAMENTO

FUNÇÃO	SQL
Cadastrar Medicamento	INSERT INTO Medicamentos (nome, categoria, preco, estoque) VALUES (' Glifage XR ', 'Antidiabetico', 10.99, 100);
Listar Todos Medicamento	SELECT * FROM Medicamento;
Buscar Medicamento por Nome	SELECT * FROM Medicamento WHERE nome = 'Ibuprofeno'
Atualizar preço de um medicamento	UPDATE Medicamentos SET preco = 3,99 WHERE nome = 'Dorflex';
Atualizar estoque de um medicamento	UPDATE Medicamentos SET estoque = 20 WHERE nome = 'Dipirona';
Deletar Medicamento	DELETE FROM Medicamentos WHERE id_medicamento = 5;

3) CRUD – VENDAS

FUNÇÃO	SQL
Cadastrar nova Venda	INSERT INTO Vendas (data_venda, total, ID_colaborador, cpf) VALUES (CURRENT_DATE, 0, 1, '12345678901') RETURNING ID_venda; Inserir itens na venda INSERT INTO Item_Venda (quantidade, preco_unitario, ID_venda, ID_medicamento) VALUES (2, 15.50, 1, 101),(1, 10.00, 1, 102); UPDATE Vendas SET total = (SELECT SUM(quantidade * preco_unitario) FROM Item_Venda WHERE ID_venda = 1) WHERE ID_venda = 1;
Listar todas as vendas com informações	SELECT Vendas.ID_venda, Vendas.data_venda, Vendas.total AS total_venda, Clientes.nome AS cliente, Colaboradores.ID_colaborador AS colaborador FROM Vendas

	INNER JOIN Clientes ON Vendas.cpf = Clientes.CPF INNER JOIN Colaboradores ON Vendas.ID_colaborador = Colaboradores.ID_colaborador ORDER BY Vendas.ID_venda;
Listar Medicamentos Mais Vendidos	SELECT M.nome AS medicamento, SUM(IV.quantidade) AS total_vendido FROM Item_Venda IV INNER JOIN Medicamentos M ON IV.ID_medicamento = M.ID_medicamento GROUP BY M.nome ORDER BY total_vendido DESC
Buscar por uma venda especifica	SELECT * FROM Clientes WHERE id_venda = 3;
Histórico de compras de um cliente	SELECT Vendas.ID_venda, Vendas.data_venda, Vendas.total, Clientes.nome AS cliente, Colaboradores.nome AS colaborador FROM Vendas INNER JOIN Clientes ON Vendas.ID_cliente = Clientes.ID_cliente INNER JOIN Colaboradores ON Vendas.ID_colaborador = Colaboradores.ID_colaborador WHERE Clientes.CPF LIKE %s OR Colaboradores.CPF LIKE %s ORDER BY Vendas.ID_venda
Consultar todas as vendas de um medicamento específico	SELECT v.id_venda, v.data_venda, v.total FROM vendas v JOIN medicamentos m ON v.id_venda = m.id_medicamento WHERE m.nome = 'NOME-DO-MEDICAMENTO';
Consultar colaborador que mais realizou vendas	SELECT c.nome , COUNT(v.id_venda) AS total_vendas FROM vendas v JOIN colaboradores c ON v.id_colaborador = c.id_colaborador GROUP BY c.id_colaborador ORDER BY total_vendas DESC LIMIT 1;
Atualizar o Total da Venda	UPDATE Vendas SET total = (SELECT SUM(IV.quantidade * IV.preco_unitario) FROM Item_Venda IV WHERE IV.ID_venda = Vendas.ID_venda) WHERE Vendas.ID_venda = %s;
Atualizar quantidade de medicamento numa venda	UPDATE item_venda SET quantidade = 'NOVA_QUANTIDADE' WHERE id_venda = 'ID_VENDA' AND id_medicamento = 'ID_MEDICAMENTO';
Atualizar quantidade de um item dentro da venda	UPDATE vendas SET total = 'NOVA_QUANTIDADE' WHERE id_venda = 'ID_VENDA' and id_colaborador = 'ID_COLABORADOR'
Remover item de uma venda e atualizar total	DELETE FROM item_venda

	WHERE id_venda = 'ID_VENDA' AND id_medicamento = 'Id_MEDICAMENTO'; SELECT SUM(quantidade * preco_unitario) AS total_venda FROM item_venda iv WHERE id_venda = 'ID_VENDA'; UPDATE vendas SET total = 'TOTAL' WHERE id_venda = 'ID_VENDA';
Deletar venda	DELETE FROM Vendas WHERE id_venda = 'ID_VENDA';