**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ**

PROJETO DE BANCO DE DADOS

1ª PARTE

FARMÁCIA

Equipe:

Lucas Stopinski da Silva

Richard Mickaell Santos Nascimento

Curitiba

2024

Sumário

[1 Domínio de Aplicação para o Banco de Dados 3](#_Toc174135297)

[1.1 Identificação do Projeto 3](#_Toc174135298)

[1.2 Tema do Projeto 3](#_Toc174135299)

[1.3 Usuários do sistema 3](#_Toc174135300)

[1.4 Funcionalidade 1 do Projeto 3](#_Toc174135301)

[2 Modelo Conceitual 4](#_Toc174135302)

[3 Modelo Lógico (2FN) 5](#_Toc174135303)

[4 Modelo Lógico (2FN) 6](#_Toc174135304)

[5 Modelo Físico (2FN) 7](#_Toc174135305)

[5.1 SQL para criação de tabelas e restrições 7](#_Toc174135306)

[5.2 SQL para inserção de pelo menos 10 registros para cada tabela 7](#_Toc174135307)

[6 Consultas 8](#_Toc174135308)

[6.1 SQL para 2 consultas simples 8](#_Toc174135309)

[6.2 SQL para 2 consultas com IR (PK + FK) de recuperação de dados 9](#_Toc174135310)

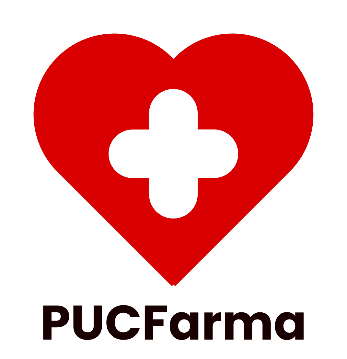
# Domínio de Aplicação para o Banco de Dados

## Identificação do Projeto

Nome Projeto

PUCFarma

Logomarca



## Tema do Projeto

O objetivo deste sistema de farmácia é automatizar o processo de gestão de uma farmácia, facilitando o controle de medicamentos, vendas, estoque e prescrições médicas. O sistema permitirá que funcionários da farmácia realizem o cadastro e controle de produtos, gerenciem o estoque de medicamentos, consultem informações detalhadas sobre cada medicamento (como data de validade, lote, fabricante), além de registrar e controlar as vendas realizadas para os clientes.

Este sistema visa melhorar a eficiência das operações da farmácia, diminuindo o risco de erros humanos, garantindo que o estoque esteja sempre atualizado, e assegurando que medicamentos vencidos sejam removidos das prateleiras. Também será possível gerar relatórios de vendas e estoque, além de facilitar o controle sobre medicamentos que exigem receita médica.

## Usuários do sistema

Os principais usuários do sistema de farmácia são:

* Farmacêuticos: Responsáveis pelo cadastro de medicamentos, controle de estoque e validação das prescrições médicas. Eles devem ter acesso a funcionalidades que permitam visualizar e editar os dados de medicamentos, gerenciar o estoque e verificar as receitas médicas.
* Atendentes: Realizam as vendas de medicamentos e auxiliam os clientes. Eles precisam de acesso ao sistema para registrar as vendas e verificar a disponibilidade de medicamentos no estoque.
* Gerente: Responsável por supervisionar as operações da farmácia, incluindo o controle de estoque e o acompanhamento das vendas. O gerente terá acesso a relatórios de desempenho e poderá visualizar o histórico de vendas e movimentação de estoque.
* Clientes (opcional): Caso o sistema tenha um módulo de autoatendimento ou aplicativo para os clientes da farmácia, eles poderão acessar o sistema para visualizar o catálogo de medicamentos e fazer pedidos, com ou sem receita médica (dependendo do tipo de medicamento).

Todos os usuários precisarão realizar login no sistema para acessar suas funcionalidades específicas, garantindo a segurança e a integridade dos dados.

## Funcionalidade 1 do Projeto

**Cadastro e Controle de Medicamentos**

Uma funcionalidade fundamental do sistema é o Cadastro de Medicamentos, que permite ao farmacêutico registrar novos medicamentos no banco de dados da farmácia. Ao cadastrar um novo medicamento, o farmacêutico deve inserir informações detalhadas como:

* Nome do medicamento
* Código do produto
* Descrição e composição
* Lote e data de validade
* Fabricante
* Preço unitário
* Quantidade disponível em estoque
* Necessidade de receita médica (sim ou não)

A partir dessa funcionalidade, o sistema também deverá verificar a validade dos medicamentos periodicamente e emitir alertas para produtos que estão próximos da data de vencimento. Além disso, o sistema deve controlar o estoque, atualizando a quantidade disponível sempre que uma venda for realizada ou um novo lote de medicamentos for recebido.

# Modelo Conceitual

Imagem de boa resolução do Modelo Conceitual (**Modelo Entidade-Relacionamento - MER**) na **2FN**

Diagrama

Descrição gerada automaticamente

# Modelo Lógico (2FN)

Imagem de boa resolução do Modelo Lógico (**Modelo Relacional - MR**) na **3FN**, feito com o **brModelo**.

Diagrama

Descrição gerada automaticamente

# Modelo Lógico (2FN)

Imagem de boa resolução do Modelo Lógico (**Modelo Relacional - MR**) na **3FN**, feito com o **ENGENHARIA REVERSA do MySQL Workbench**.

Diagrama

Descrição gerada automaticamente

# Modelo Físico (2FN)

## SQL para criação de tabelas e restrições

CREATE DATABASE Farmacia;

USE Farmacia;

CREATE TABLE Usuario (

ID\_Usuario INT PRIMARY KEY AUTO\_INCREMENT,

Nome VARCHAR(50) NOT NULL,

Email VARCHAR(50) NOT NULL,

Senha VARCHAR(20) NOT NULL,

Endereco VARCHAR(100),

Celular VARCHAR(20),

Tipo ENUM('Administrador', 'Cliente') NOT NULL

);

CREATE TABLE Categoria (

ID\_Categoria INT PRIMARY KEY AUTO\_INCREMENT,

Nome\_Categoria VARCHAR(20) NOT NULL

);

CREATE TABLE Produto (

ID\_Produto INT PRIMARY KEY AUTO\_INCREMENT,

Nome VARCHAR(50) NOT NULL,

Descricao TEXT,

Preco DECIMAL(10,2) NOT NULL,

Estoque INT NOT NULL,

Codigo\_Barras VARCHAR(50),

fk\_Categoria\_ID\_Categoria INT,

CONSTRAINT FK\_Produto\_Categoria

FOREIGN KEY (fk\_Categoria\_ID\_Categoria)

REFERENCES Categoria (ID\_Categoria)

ON DELETE RESTRICT

);

CREATE TABLE Pedido (

ID\_Pedido INT PRIMARY KEY AUTO\_INCREMENT,

ID\_Usuario INT NOT NULL,

Data\_Pedido DATETIME NOT NULL,

Status ENUM('Pendente', 'Enviado', 'Entregue', 'Cancelado') NOT NULL,

Total DECIMAL(10,2) NOT NULL,

fk\_Usuario\_ID\_Usuario INT,

CONSTRAINT FK\_Pedido\_Usuario

FOREIGN KEY (fk\_Usuario\_ID\_Usuario)

REFERENCES Usuario (ID\_Usuario)

ON DELETE RESTRICT

);

CREATE TABLE Contem (

fk\_Pedido\_ID\_Pedido INT,

fk\_Produto\_ID\_Produto INT,

PRIMARY KEY (fk\_Pedido\_ID\_Pedido, fk\_Produto\_ID\_Produto),

CONSTRAINT FK\_Contem\_Pedido

FOREIGN KEY (fk\_Pedido\_ID\_Pedido)

REFERENCES Pedido (ID\_Pedido)

ON DELETE RESTRICT,

CONSTRAINT FK\_Contem\_Produto

FOREIGN KEY (fk\_Produto\_ID\_Produto)

REFERENCES Produto (ID\_Produto)

ON DELETE RESTRICT

);

## 

## SQL para inserção de pelo menos 10 registros para cada tabela

Importante: Indicar em imagens de boa resolução, o SELECT das tabelas já criadas e povoadas.

INSERT INTO Usuario (Nome, Email, Senha, Endereco, Celular, Tipo)

VALUES

('João Silva', 'joao@gmail.com', 'senha123', 'Rua A, 123', '11999999999', 'Cliente'),

('Maria Oliveira', 'maria@gmail.com', 'senha456', 'Rua B, 456', '11988888888', 'Cliente'),

('Pedro Souza', 'pedro@gmail.com', 'senha789', 'Rua C, 789', '11977777777', 'Cliente'),

('Ana Santos', 'ana@gmail.com', 'senhaabc', 'Rua D, 101', '11966666666', 'Cliente'),

('Carlos Lima', 'carlos@gmail.com', 'senhaxyz', 'Rua E, 102', '11955555555', 'Cliente'),

('Paula Ribeiro', 'paula@gmail.com', 'senha098', 'Rua F, 103', '11944444444', 'Cliente'),

('Lucas Alves', 'lucas@gmail.com', 'senha345', 'Rua G, 104', '11933333333', 'Cliente'),

('Carla Pereira', 'carla@gmail.com', 'senha567', 'Rua H, 105', '11922222222', 'Cliente'),

('Ricardo Costa', 'ricardo@gmail.com', 'senha1234', 'Rua I, 106', '11911111111', 'Cliente'),

('Administrador', 'admin@farmacia.com', 'adminsenha', 'Rua Admin, 100', '11900000000', 'Administrador');

Tabela

Descrição gerada automaticamente

INSERT INTO Categoria (Nome\_Categoria)

VALUES

('Medicamentos'),

('Suplementos'),

('Cosméticos'),

('Equipamentos'),

('Higiene'),

('Fraldas'),

('Produtos Naturais'),

('Dermocosméticos'),

('Ortopedia'),

('Nutrição');

Tabela

Descrição gerada automaticamente

INSERT INTO Produto (Nome, Descricao, Preco, Estoque, Codigo\_Barras, fk\_Categoria\_ID\_Categoria)

VALUES

('Paracetamol 500mg', 'Analgésico e antitérmico', 9.99, 100, '7891000100101', 1),

('Vitamina C 1g', 'Suplemento vitamínico', 19.90, 50, '7891000200202', 2),

('Protetor Solar FPS 50', 'Protetor solar de alta proteção', 29.90, 200, '7891000300303', 3),

('Termômetro Digital', 'Termômetro de medição rápida', 49.90, 30, '7891000400404', 4),

('Sabonete Antibacteriano', 'Sabonete líquido antibacteriano', 5.90, 300, '7891000500505', 5),

('Fralda Geriátrica', 'Fralda descartável tamanho G', 39.90, 80, '7891000600606', 6),

('Óleo de Coco', 'Suplemento natural com óleo de coco', 24.90, 60, '7891000700707', 7),

('Creme Anti-idade', 'Creme para redução de rugas', 89.90, 120, '7891000800808', 8),

('Muleta Ajustável', 'Muleta com altura ajustável', 99.90, 15, '7891000900909', 9),

('Shake de Proteína', 'Suplemento nutricional com proteínas', 49.90, 40, '7891001001010', 10);

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

INSERT INTO Pedido (ID\_Usuario, Data\_Pedido, Status, Total, fk\_Usuario\_ID\_Usuario)

VALUES

(1, NOW(), 'Pendente', 59.80, 1),

(2, NOW(), 'Enviado', 29.90, 2),

(3, NOW(), 'Entregue', 19.90, 3),

(4, NOW(), 'Cancelado', 49.90, 4),

(5, NOW(), 'Pendente', 89.90, 5),

(6, NOW(), 'Enviado', 39.90, 6),

(7, NOW(), 'Entregue', 24.90, 7),

(8, NOW(), 'Cancelado', 49.90, 8),

(9, NOW(), 'Pendente', 99.90, 9),

(10, NOW(), 'Enviado', 149.70, 10);

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

INSERT INTO Contem (fk\_Pedido\_ID\_Pedido, fk\_Produto\_ID\_Produto)

VALUES

(1, 1), -- Pedido 1 contém Produto 1 (Paracetamol)

(1, 2), -- Pedido 1 contém Produto 2 (Vitamina C)

(2, 3), -- Pedido 2 contém Produto 3 (Protetor Solar)

(3, 4), -- Pedido 3 contém Produto 4 (Termômetro Digital)

(4, 5), -- Pedido 4 contém Produto 5 (Sabonete Antibacteriano)

(5, 6), -- Pedido 5 contém Produto 6 (Fralda Geriátrica)

(6, 7), -- Pedido 6 contém Produto 7 (Óleo de Coco)

(7, 8), -- Pedido 7 contém Produto 8 (Creme Anti-idade)

(8, 9), -- Pedido 8 contém Produto 9 (Muleta Ajustável)

(9, 10); -- Pedido 9 contém Produto 10 (Shake de Proteína)

Tabela

Descrição gerada automaticamente

# Consultas

Não usar **SELECT \***.

## SQL para 2 consultas simples

**Consulta 1**:

1. Esta consulta ajuda o usuário a identificar produtos com um estoque considerável (acima de 50), permitindo decisões de promoção ou reposição.
2. Código SQL:

SELECT Nome, Preco, Estoque

FROM Produto

WHERE Estoque > 50;

1. Imagem com o resultado da consulta.

Interface gráfica do usuário, Aplicativo, Tabela

Descrição gerada automaticamente

**Consulta 2**:

1. Esta consulta permite visualizar os pedidos que estão no status "Pendente", ou seja, que ainda precisam ser processados, ajudando no gerenciamento de pedidos.
2. Código SQL:

SELECT ID\_Pedido, Data\_Pedido, Total

FROM Pedido

WHERE Status = 'Pendente';

1. Imagem com o resultado da consulta.

Interface gráfica do usuário, Texto, Aplicativo, Tabela

Descrição gerada automaticamente

## SQL para 2 consultas com IR (PK + FK) de recuperação de dados

Para consultas com referências entre tabelas, usar apenas **JOIN**.

**Consulta 1**:

1. Esta consulta permite ver todos os pedidos de um usuário específico, o que é útil para acompanhamento do histórico de compras de clientes.
2. Código SQL:

SELECT Pedido.ID\_Pedido, Pedido.Data\_Pedido, Pedido.Status, Pedido.Total

FROM Pedido

INNER JOIN Usuario ON Pedido.fk\_Usuario\_ID\_Usuario = Usuario.ID\_Usuario

WHERE Usuario.ID\_Usuario = 1;

1. Imagem com o resultado da consulta.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**Consulta 2**:

1. A consulta traz informações sobre os produtos que estão contidos em um pedido específico, facilitando o controle de quais itens foram solicitados.
2. Código SQL:

SELECT Produto.Nome, Produto.Preco, Produto.Estoque

FROM Contem

INNER JOIN Produto ON Contem.fk\_Produto\_ID\_Produto = Produto.ID\_Produto

WHERE Contem.fk\_Pedido\_ID\_Pedido = 1;

1. Imagem com o resultado da consulta.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente