

UENF

Universidade Estadual do Norte Fluminense Darcy Ribeiro

RELATÓRIO TÉCNICO SISTEMA DE GERENCIAMENTO DE ESTOQUE DE FARMÁCIA

Disciplina: Paradigma Orientado a Objetos para Desenvolvimento de Software

AUTORES

Carlos Alexandre de Azevedo Titoneli

Mateus do Nascimento Viana

Campos dos Goytacazes

Novembro de 2025

1 Relatório Técnico

O sistema desenvolvido tem como objetivo automatizar o controle de medicamentos, estoques, vendas, receitas e operações administrativas de uma farmácia. Ele oferece funcionalidades completas para registro, consulta, movimentação e análise de dados, garantindo organização, segurança e rastreabilidade das operações.

Além disso, o sistema também fornece uma rede neural simples que faz o planejamento da demanda futura considerando: histórico de vendas, época do ano (sazonalidade) e eventos locais (como surtos, epidemias, pandemias entre outros).

1.1 Link Repositório GitHub

<https://github.com/matviana/POODEV-Sistema-Gest-o-de-Estoque-de-Farm-cia>

2 Funcionalidades (Requisitos Funcionais)

- Permite cadastrar medicamentos com informações completas: nome, lote, validade, quantidade mínima, código de barras e estoque inicial.
- Suporta medicamentos controlados, incluindo a exigência de receita.
- Pode realizar a consulta por: código de barras e nome do medicamento, além de visualizar todos os medicamentos já cadastrados.
- Permite excluir medicamentos já cadastrados.
- Cadastro de farmácias com nome, endereço, telefone e CNPJ.
- Consulta de todas as farmácias cadastradas.
- Exclusão de farmácias por CNPJ.
- Permite adicionar novas unidades ao estoque, registrando automaticamente a movimentação.
- Permite retirar unidades (vendas).
- Alertas para medicamentos vencidos ou próximos.
- Alertas para medicamentos com estoque abaixo do mínimo.
- Permite repor automaticamente todos os itens com estoque baixo até atingir o mínimo exigido.
- Cada movimentação: entrada, saída ou reposição é devidamente registrada no histórico.
- Relatório de medicamentos mais vendidos do mês.
- Análise do histórico e identifica quais medicamentos tiveram mais vendas, informação que será útil para previsão da demanda futura.

3 Estrutura Geral do Sistema

O sistema é organizado em vários arquivos que conversam entre si, cada um responsável por uma parte do processo:

- **main.py**: Gerencia o menu e a interação com o usuário (CRUD).
- **medicamentos.py**: Lógica de medicamentos.
- **farmacias.py**: Controle de farmácias.
- **estoque.py**: Entradas, saídas e reposições.
- **historico.py**: Relatórios e histórico salvo.
- **database.py**: Conexão e criação de tabelas no PostgreSQL.
- **redeneural.py**: Parte da IA para previsão de demanda futura levando em conta os 3 critérios.

4 Modelagem

4.1 Modelagem UML

O diagrama UML (Unified Modeling Language) ilustra as classes, atributos, métodos e os relacionamentos entre as entidades do sistema.

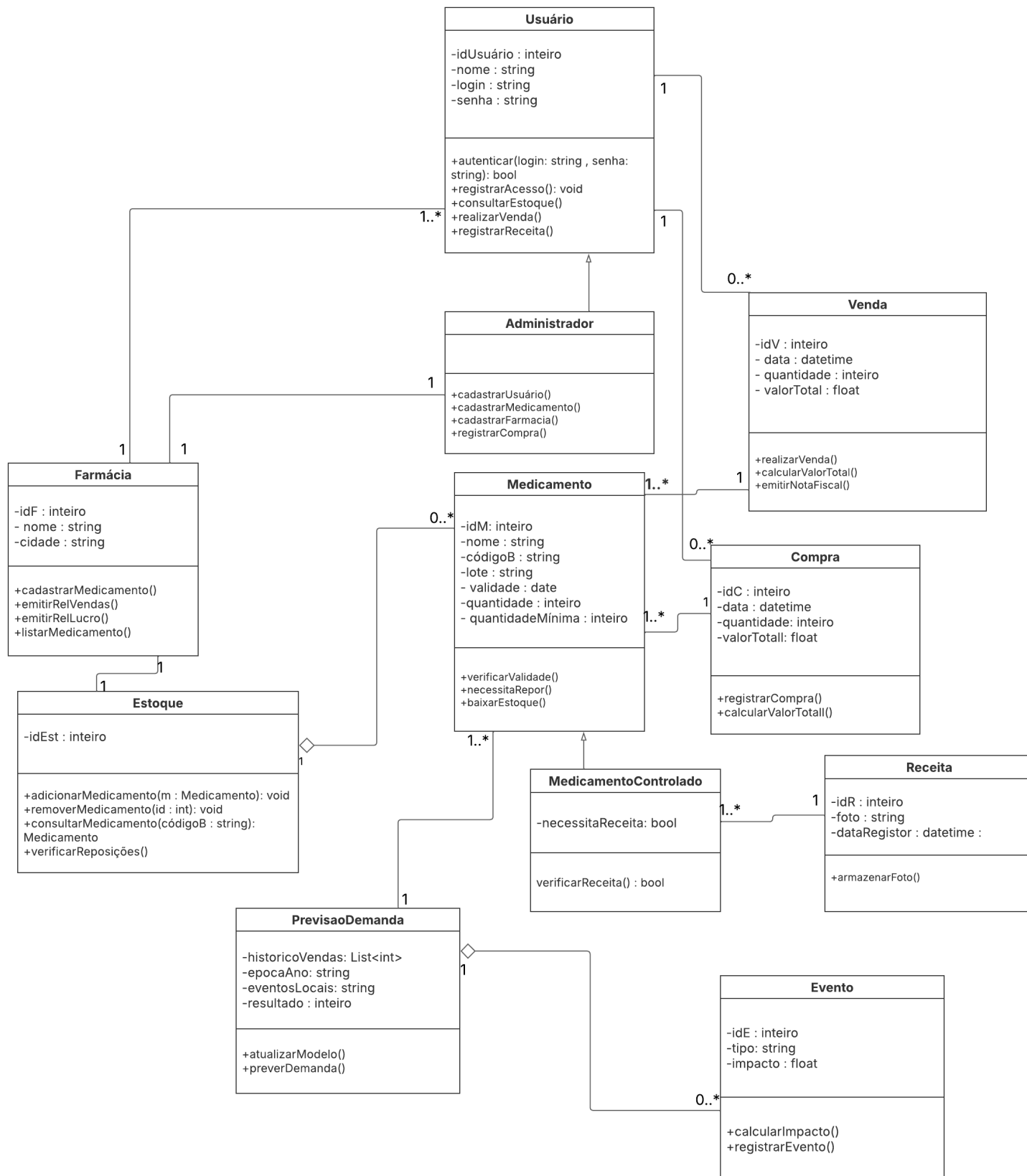


Figura 1: Modelagem UML

4.2 Modelo Entidade Relacionamento (MER)

O MER, atualizado após consulta com o cliente Luis Mariano, representa as entidades e seus relacionamentos no contexto do banco de dados.

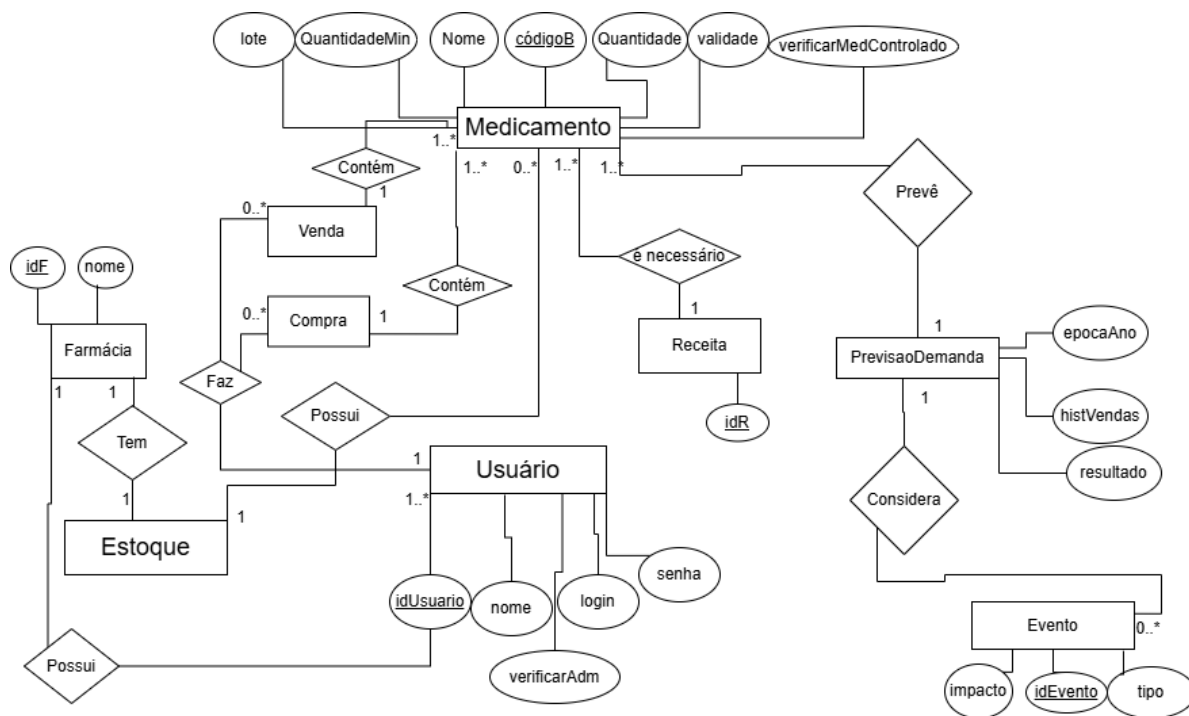


Figura 2: Modelo Entidade Relacionamento (MER)

5 Conceitos Técnicos Utilizados no Sistema

5.1 Orientação a Objetos (OO)

O sistema foi desenvolvido seguindo os princípios da Programação Orientada a Objetos (POO), que facilita a organização, reutilização e manutenção do código. Estão entre os principais:

- **Classe:** Estrutura que modela entidades do mundo real (Exemplos: `Medicamento`, `Farmacia`, `Estoque` e `Histórico`).
- **Objeto:** Instância de uma classe. Cada medicamento cadastrado é um objeto da classe `Medicamento`.
- **Encapsulamento:** Organização do código de modo que cada classe controle seus próprios dados e métodos, evitando erros e garantindo maior segurança.
- **Métodos:** Funções internas às classes que definem comportamentos (Exemplos: `cadastrar()`, `consultar_por_codigo()`, `saida()`, entre outras).
- **Modularização:** Divisão do sistema em múltiplos arquivos (`medicamentos.py`, `estoque.py`, etc.), cada um responsável por uma parte específica da lógica, facilitando manutenção e escalabilidade.

5.2 Inteligência Artificial (Rede Neural)

Os principais conceitos de IA que serão utilizados incluem:

- **Rede Neural Artificial:** Modelo computacional inspirado no cérebro humano, composto por "neurônios" que processam informações. Ideal para identificar padrões em grandes volumes de dados.
- **Treinamento de Modelo:** Processo no qual a rede aprende a partir de exemplos históricos (como vendas, reposições e sazonalidade).
- **Previsão de Demanda:** Objetivo principal da IA no sistema de gerenciamento de estoque.

5.3 Banco de Dados (PostgreSQL)

O sistema utiliza um banco de dados relacional PostgreSQL, garantindo segurança, integridade e persistência das informações.

- **Tabelas:** Estruturas onde os dados são armazenados (Exemplos: `medicamentos`, `farmacias`, `historico`).
- **Chaves Primárias (Primary Keys):** Identificadores únicos para cada registro, evitando duplicidade.
- **Chaves Estrangeiras (Foreign Keys):** Mantêm o relacionamento entre medicamentos e o histórico de movimentações.
- **SQL:** Linguagem usada para consultar, inserir, atualizar e excluir dados. Exemplos utilizados no sistema:
 - `SELECT` para consultar medicamentos.
 - `INSERT` para registrar novas entradas.
 - `UPDATE` para atualizações de estoque.
 - `DELETE` para exclusão de farmácias e medicamentos.

6 Cronograma seguido durante semestre:

Semana	Período (Data)	Atividades Realizadas
Semana 1	15/9 até 21/9	Estrutura inicial do projeto - Backend e Banco de dados
Semana 2	22/9 até 28/9	Desenvolver, implementar classes, atributos e alguns métodos + banco de dados
Semana 3	29/9 até 5/10	Implementar cadastro e consulta de medicamentos
Semana 4	6/10 até 12/10	Integração com banco de dados e cadastrar rede de farmácias
Semana 5	13/10 até 19/10	Implementar controle de validade e alertas de vencimento
Semana 6	20/10 até 26/10	Implementar controle de estoque (entrada, saída e mínimo) com alertas
Semana 7	27/10 até 2/11	Criar reposição automática (para estoque baixo)
Semana 8	3/11 até 9/11	Começar rede neural e preparar registro e histórico de compras e vendas
Semana 9	10/11 até 16/11	Implementar rede neural (previsão demanda)
Semana 10	17/11 até 23/11	Criar avisos e guardar fotos de receita + relatório dos mais vendidos (mês)
Semana 11	24/11 até 30/11	Desenvolvimento da interface (login, registro, controle, avisos)
Semana 12	1/12 até 7/12	Testes finais e finalização do projeto

Possíveis atrasos no projeto em decorrência de semanas provas foram sanados nas semanas subsequentes para que não acumulasse. Todas modificações foram adicionadas incrementalmente no repositório do Github.

7 Instruções para Execução do Sistema

7.1 Back-End do Sistema

O back-end foi implementado em Python, utilizando uma abordagem orientada a objetos. Cada funcionalidade principal foi isolada em um arquivo específico:

- **medicamentos.py**: Gerencia cadastros, atributos e classificação de medicamentos.
- **estoque.py**: Responsável pelo controle de entrada, saída e atualização de estoque.
- **farmacias.py**: Reúne operações relacionadas às unidades da rede de farmácias.
- **database.py**: Implementa a comunicação e integração com o banco de dados.
- **historico.py**: Mostra o histórico das movimentações, seja entrada ou venda.
- **main.py**: Ponto de entrada do programa, oferecendo menu interativo e conectando todos os módulos.

7.2 Integração com Banco de Dados e Uso do Arquivo .env

O sistema utiliza um banco de dados para armazenar: informações de estoque atual, vendas, e cadastros de medicamentos e farmácias (rede). Os dados sensíveis (host, usuário, senha e nome do banco) ficam em um arquivo `.env` para segurança.

Exemplo do Arquivo `.env`:

```
PG_HOST = localhost
PG_PORTA = número da porta escolhido
PG_BD = farmácia_bd
PG_SENHA = senha escolhida
```

O módulo `database.py` realiza a leitura dessas variáveis e estabelece a conexão com o banco automaticamente.

7.3 Rede Neural para Planejamento de Demanda Futura

O projeto prevê um módulo de inteligência artificial baseado em rede neural com objetivo de prever a demanda futura de medicamentos. A previsão utilizará três critérios principais: Histórico de vendas, Sazonalidade e Possíveis eventos locais. A IA irá servir também para apoiar decisões automatizadas de compra e reposição.

7.4 Front-End em React

O projeto inclui o planejamento de uma interface visual desenvolvida em React, que se comunicará com o back-end e permitirá:

- Visualização de estoque em tempo real.
- Realizar ações do CRUD, conversando com `main.py`.
- Carregamento dos relatórios gerados pelo back-end.
- Interface para consulta de vendas, consumo e previsão da IA.

7.5 Pré-requisitos

- Python 3.10 ou superior.
- Banco de dados PostgreSQL (foi usado no projeto o Pg adm 4).
- Biblioteca `python-dotenv` instalada.
- Uso da biblioteca `psycopg2` (principal driver Python para comunicação com bancos de dados PostgreSQL) para conectar, criar tabelas, inserir, atualizar e consultar dados.

