

Relatório do Projeto: Sistema de Gerenciamento de Vendas

Nome: Adrian Lima, Vinicius Santos e Yuri Santos

Data: 24/09/25

1. Introdução

O objetivo deste projeto foi desenvolver uma aplicação de desktop completa para o gerenciamento de um sistema de vendas. Utilizando a linguagem Python e a biblioteca Tkinter para a interface gráfica, a aplicação visa fornecer funcionalidades essenciais como cadastro de produtos, clientes e registro de vendas. O sistema se conecta a um banco de dados MySQL para garantir a persistência e a manipulação segura dos dados, implementando todas as operações de CRUD (Create, Read, Update, Delete). Além disso, o projeto inclui a visualização de dados através de gráficos gerados com a biblioteca Matplotlib e um sistema de log para registro de erros, seguindo os princípios da Programação Orientada a Objetos.

2. Diagrama de Classes (UML Simplificado)

[Aqui, você deve inserir uma imagem do seu diagrama de classes. Você pode usar uma ferramenta online como o "draw.io" ou "Lucidchart" para desenhar. O diagrama deve mostrar as classes principais e suas relações.]

Descrição das Classes:

- **DatabaseManager**: Responsável por toda a comunicação com o banco de dados MySQL. Encapsula os métodos para conectar, desconectar, e executar as operações de CRUD (INSERT, SELECT, UPDATE, DELETE).
- **LoginWindow**: Representa a tela inicial de login. Ela interage com o **DatabaseManager** para validar as credenciais do usuário. Se o login for bem-sucedido, esta janela é destruída e a **MainWindow** é instanciada.
- **MainWindow**: É a janela principal da aplicação, exibida após o login. Ela contém um **Notebook** (sistema de abas) que organiza as diferentes funcionalidades:
 - **Aba de Produtos**: Contém os widgets e a lógica para o CRUD de produtos.
 - **Aba de Clientes**: Contém os widgets e a lógica para o CRUD de clientes.
 - **Aba de Vendas**: Permite o registro de novas vendas, utilizando um ComboBox para selecionar produtos e atualizando o estoque.
 - **Aba de Gráfico**: Exibe um gráfico de barras com o total de vendas por produto, utilizando Matplotlib.

3. Explicação do Código

O projeto foi estruturado em módulos para separar as responsabilidades, seguindo as boas práticas de desenvolvimento de software.

- **main.py**: É o ponto de entrada da aplicação. Suas principais responsabilidades são:
 - Configurar o logger de erros.
 - Instanciar o **DatabaseManager** com as credenciais de conexão.
 - Estabelecer a conexão com o banco de dados.
 - Instanciar e iniciar a **LoginWindow**, que dá início ao loop da interface gráfica.
- **database.py**: Contém a classe **DatabaseManager**. Este módulo abstrai toda a complexidade da interação com o banco de dados. Seus métodos genéricos **execute_query**, **fetch_all** e **fetch_one** permitem que a camada de interface (**gui.py**) realize operações no banco sem conhecer os detalhes da sintaxe SQL.
- **gui.py**: Módulo mais extenso, responsável por toda a interface gráfica e pela lógica de interação com o usuário.
 - A classe **LoginWindow** cria os campos de usuário e senha e o botão de login.
 - A classe **MainWindow** cria e gerencia as abas. Cada funcionalidade de CRUD (Produtos, Clientes) possui métodos específicos para adicionar, atualizar, deletar e carregar os dados na sua respectiva **Treeview** (tabela).
 - A lógica de registro de vendas inclui a validação de estoque e a atualização da tabela de produtos após uma venda ser concluída.
- **logger.py**: Um módulo simples que configura o sistema de logging do Python para registrar quaisquer exceções que ocorram durante a execução em um arquivo **error.log**. Isso facilita a depuração e o monitoramento de problemas em produção.

4. Instruções para Execução

Para executar o projeto, siga os passos abaixo:

1. **Pré-requisitos:**
 - Python 3.x instalado.
 - Servidor de banco de dados MySQL instalado e em execução.
2. **Configuração do Banco de Dados:**
 - Crie um banco de dados no MySQL com o nome **sales_system**.
 - Execute o script **database_setup.sql** fornecido para criar todas as tabelas e inserir o usuário administrador padrão (**admin/admin123**).
3. **Configuração do Projeto:**
 - Clone ou baixe o repositório do projeto.
 - Abra o arquivo **main.py** e altere as credenciais (**host**, **user**, **password**) no dicionário **DB_CONFIG** para corresponder à sua configuração do MySQL.

4. Instalação das Dependências:

- Navegue até a pasta do projeto pelo terminal.
- Recomenda-se criar um ambiente virtual: `python -m venv .venv` e ativá-lo.

Instale todas as bibliotecas necessárias com o comando:

```
pip install -r requirements.txt
```

○

5. Execução da Aplicação:

Com o ambiente virtual ativo, execute o seguinte comando no terminal:

```
python main.py
```

○

- A tela de login será exibida.

5. Exemplos de Uso (Prints de Tela)

Abaixo estão algumas telas que demonstram a aplicação em funcionamento.

Figura 1: Tela de Login

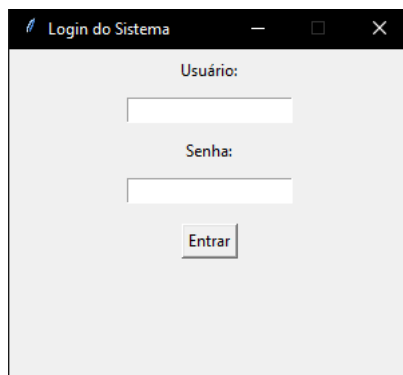


Figura 2: Gerenciamento de Produtos

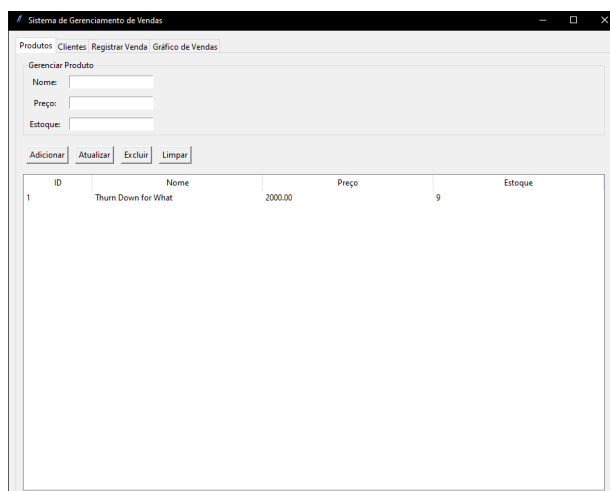


Figura 3: Gerenciamento de Clientes

Sistema de Gerenciamento de Vendas

Produtos Clientes Registrar Venda Gráfico de Vendas

Gerenciar Cliente

Nome:

Email:

Telefone:

Adicionar Atualizar Excluir Limpar

ID	Nome	Email	Telefone
1	Junior Cabeção	Kbeção@gmail.com	54997062651

Figura 4: Registro de uma Nova Venda

Sistema de Gerenciamento de Vendas

Produtos Clientes Registrar Venda Gráfico de Vendas

Registrar Nova Venda

Produto:

Quantidade:

Registrar Venda

Histórico de Vendas

ID Venda	Produto	Quantidade	Data da Venda
1	Thurn Down for What	1	2025-09-30 22:03:04

Figura 5: Gráfico de Vendas por Produto

