

# Simulador de E-commerce com Tkinter e SQLite3

## Descrição Geral

Este projeto simula um sistema de e-commerce básico, permitindo o cadastro de produtos, clientes e pedidos. Ele possui uma interface gráfica criada com Tkinter e utiliza SQLite3 para armazenar os dados de forma persistente.

## Funcionalidades

### 1. Tela Inicial

- Exibe um menu principal com botões para acessar:
  - Cadastro de Produtos
  - Cadastro de Clientes
  - Registro de Pedidos
  - Relatório de Pedidos

### 2. Cadastro de Produtos

- Tela com os seguintes campos:
  - Nome do Produto
  - Preço
  - Quantidade em Estoque
- Funções:
  - Adicionar novo produto
  - Editar produto existente
  - Remover produto
  - Listar todos os produtos em um Treeview.

### 3. Cadastro de Clientes

- Tela com os seguintes campos:

- Nome do Cliente
- E-mail
- Endereço

- Funções:

- Adicionar novo cliente
- Editar cliente existente
- Remover cliente
- Listar todos os clientes em um Treeview.

### 4. Registro de Pedidos

- Tela com os seguintes passos:

- Seleção de Cliente (dropdown ou pesquisa)
- Seleção de Produtos (lista com checkbox para escolher múltiplos produtos)
- Campo para definir a quantidade de cada produto
- Botão para finalizar o pedido

- Funções:

- Atualizar o estoque após o registro do pedido
- Calcular o total do pedido automaticamente

### 5. Relatório de Pedidos

- Tela que exibe:

- Lista de pedidos realizados com cliente, produtos, quantidade e total
- Opções de filtro por data ou cliente
- Resumo total de vendas em um período.

## Requisitos

### Funcionais

1. O sistema deve permitir o cadastro de produtos com nome, preço e quantidade.
2. O sistema deve permitir o cadastro de clientes com nome, e-mail e endereço.
3. Deve ser possível registrar pedidos vinculando clientes e produtos, atualizando o estoque automaticamente.
4. O sistema deve exibir relatórios de pedidos realizados com opções de filtro.

### Não Funcionais

1. Utilizar SQLite3 como banco de dados para armazenamento.
2. Interface gráfica amigável usando Tkinter.
3. O sistema deve ser modularizado, com classes para Produtos, Clientes e Pedidos.
4. As operações de CRUD devem ser realizadas por meio de métodos que interagem com o banco de dados.

## Estrutura do Projeto

### Classes

#### 1. Produto

- Atributos: id, nome, preco, quantidade.
- Métodos:
  - adicionar\_produto()
  - editar\_produto()
  - remover\_produto()
  - listar\_produtos()

## 2. Cliente

- Atributos: id, nome, email, endereco.
- Métodos:
  - adicionar\_cliente()
  - editar\_cliente()
  - remover\_cliente()
  - listar\_clientes()

## 3. Pedido

- Atributos: id, cliente\_id, produtos, quantidades, total.
- Métodos:
  - registrar\_pedido()
  - listar\_pedidos()
  - filtrar\_pedidos()

## Banco de Dados (SQLite3)

### Tabelas

#### 1. Produtos

- id (INTEGER, PRIMARY KEY)
- nome (TEXT)
- preco (REAL)
- quantidade (INTEGER)

#### 2. Clientes

- id (INTEGER, PRIMARY KEY)
- nome (TEXT)

- email (TEXT)
- endereço (TEXT)

### 3. Pedidos

- id (INTEGER, PRIMARY KEY)
- cliente\_id (INTEGER, FOREIGN KEY para Clientes)
- produtos (TEXT, lista de produtos e quantidades)
- total (REAL)

### Divisão de Tarefas

- Dev 1:
  - Implementar as classes e operações do banco de dados (CRUD).
  - Gerenciar o banco SQLite3.
- Dev 2:
  - Desenvolver a interface gráfica com Tkinter.
  - Integrar as funções das classes às telas.

Com essas informações, é possível iniciar o desenvolvimento de um projeto bem estruturado e organizado para os dois desenvolvedores. Caso precise de ajuda com algum detalhe específico, é só pedir!