

1. Banco de Dados

Estratégia: Uso do SQLite

- **Motivo:** SQLite é um banco de dados leve e fácil de configurar, ideal para projetos de pequeno a médio porte. Ele não requer um servidor de banco de dados separado, o que simplifica a instalação e manutenção.
- **Arquivo:** `db.sqlite` é o banco de dados utilizado na aplicação.
- **Inicialização:** O arquivo `init_db.py` contém o script para inicializar o banco de dados, garantindo que todas as tabelas necessárias estejam criadas antes do uso da aplicação.

2. Estrutura de Código

Estratégia: Organização em Scripts Separados

- **Motivo:** Separar a lógica de inicialização do banco de dados (`init_db.py`), a aplicação principal (`app.py`) e outros scripts auxiliares (`python.py`) ajuda a manter o código organizado e modular, facilitando a manutenção e a expansão futura.
- **Detalhes:** O `app.py` é o ponto de entrada principal da aplicação, onde a lógica central da aplicação é gerenciada.

3. Interface do Usuário

Estratégia: Diretório Separado para Arquivos de Interface

- **Motivo:** Manter os arquivos relacionados à interface do usuário (HTML, CSS, JavaScript) em um diretório separado (`estoque-app`) ajuda a manter a estrutura do projeto organizada e facilita o desenvolvimento front-end independente do back-end.
- **Detalhes:** O diretório `estoque-app` contém todos os arquivos relacionados à interface do usuário.

4. Ferramentas Utilizadas

Estratégia: Uso de Ferramentas Simples e Eficientes

- **Motivo:** Ferramentas como o SQLite para banco de dados e scripts Python para a lógica de aplicação são escolhas eficazes para um projeto de médio porte, proporcionando um equilíbrio entre simplicidade e funcionalidade.
- **Detalhes:** Python é usado pela sua simplicidade e extensa biblioteca padrão, facilitando tarefas como a manipulação de bancos de dados e a criação de servidores web.

5. Controle de Versão

Estratégia: Uso do Git para Controle de Versão

- **Motivo:** Git é uma ferramenta de controle de versão amplamente utilizada que facilita o rastreamento de alterações no código, colaborações e o gerenciamento de diferentes versões do projeto.
- **Detalhes:** O diretório `.git` indica que o Git está sendo usado para controlar as versões do projeto.

Conclusão

As estratégias para a escolha da solução desenvolvida neste projeto foram guiadas por princípios de simplicidade, modularidade e eficiência. O uso de SQLite e Python permite um desenvolvimento rápido e fácil de manter, enquanto a organização em scripts separados e a utilização de um diretório específico para a interface do usuário ajudam a manter o código limpo e organizado. O uso do Git para controle de versão garante que todas as mudanças no projeto sejam rastreadas adequadamente.