

Requisitos de Software

Sistema de Controle de Estoque para Revendedora de Peças Automotivas

“Este documento tem como objetivo descrever os requisitos funcionais e não funcionais do sistema de cadastro e consulta de preços de um estoque para uma revendedora de peças automotivas.”

Aluno: Lucas Junio

Disciplina: Desenvolvimento rápido em python

1. *framework*

O ambiente de desenvolvimento utilizado foi o Visual Studio Code, com a biblioteca Tkinter, utilizando a linguagem Python.

2. *Objetivo do sistema*

O sistema foi desenvolvido com o objetivo de cadastrar e consultar produtos automotivos, facilitando o controle de estoque da loja.

3. *Funcionalidade do sistema*

3.1 - *Cadastro de produto*

No sistema será possível cadastrar produtos informando marca, modelo, preço, categoria e quantidade em estoque. Ao final do cadastro, o sistema gerará automaticamente um código único para cada produto e armazenará as informações em um banco de dados.

3.2 - *Consulta*

Será possível consultar produtos pelo código ou nome, por meio de uma barra de pesquisa.

3.3 - *Informações dos produtos*

Quando pesquisado (produto) será mostrado todas as informações disponíveis cadastradas no sistema e seu código, será possível visualizar o código, marca, modelo, categoria, preços e estoque de produtos

3.4 - *Atualização dos produtos*

O sistema permitirá a edição ilimitada dos dados de um produto, incluindo nome, marca, modelo, categoria, preço e quantidade em estoque. O campo "código" será utilizado como identificador único e não poderá ser alterado.

3.5 - *Limpeza da Treeview*

O sistema permitirá que o usuário limpe os dados exibidos na Treeview por meio do botão "Limpar", removendo todas as entradas temporariamente carregadas a partir de uma pesquisa, sem alterar os dados do banco de dados.

4. Requisitos Funcionais

- RF01 – O sistema deve permitir o cadastro de novos produtos.
- RF02 – O sistema deve permitir a consulta de produtos pelo nome ou código.
- RF03 – O sistema deve gerar um código único para cada produto cadastrado.
- RF04 - O sistema deve permitir que o usuário atualize os dados de um produto existente informando seu código.
- RF05 - O sistema deve permitir limpar o campo de entrada e a lista de produtos exibida.
- RF06 - O sistema deve fornecer um menu com opções como "Sair" e "Sobre o Projeto".
- RF07 - O sistema deve apresentar uma interface gráfica com janelas para cadastro, atualização e sobre o projeto.

5. Requisitos Não funcionais

- RNFs 01 - O sistema deve apresentar uma interface visual clara, intuitiva e de fácil navegação, desenvolvida com a biblioteca Tkinter.
- RNFs 02 - Os dados dos produtos devem ser armazenados de forma permanente utilizando o banco de dados SQLite3, mesmo após o encerramento da aplicação.
- RNFs 03 - O sistema deve ser compatível com sistemas operacionais Windows (mínimo Windows 7), onde o Python esteja instalado.
- RNFs 04 - As operações de cadastro, busca, atualização e exibição devem ocorrer em tempo razoável (sem travamentos ou atrasos perceptíveis).
- RNFs 05 - O código deve ser organizado em funções e classes, favorecendo a manutenção, a leitura e futuras expansões do sistema.

- RNFs 06 - O sistema deve ter um uso eficiente de memória e processamento, sendo capaz de rodar em máquinas com configurações modestas.
- RNFs 07 - Não é necessária a instalação de servidores ou dependências externas complexas. O sistema deve funcionar apenas com Python e bibliotecas padrão.

Recife – PE 06.06.2025