

Documentação do Projeto

Introdução

Este documento descreve o projeto de desenvolvimento de um sistema de gerenciamento de estoque para uma faculdade. O objetivo deste projeto é criar um sistema que permita à faculdade gerenciar seu estoque de forma eficiente e eficaz.

Escopo

O sistema de gerenciamento de estoque será desenvolvido para a faculdade de XYZ. O sistema permitirá que a faculdade gerencie seu estoque de produtos, incluindo a adição de novos produtos, a atualização de informações de produtos existentes e a remoção de produtos do estoque. O sistema também permitirá que a instituição gere relatórios sobre o estoque de produtos.

Requisitos Funcionais

- O sistema deve permitir que a faculdade adicione novos produtos ao estoque.
- O sistema deve permitir que a faculdade atualize informações de produtos existentes no estoque.
- O sistema deve permitir que a faculdade remova produtos do estoque.
- O sistema deve permitir que a faculdade gere relatórios sobre o estoque de produtos.

Requisitos Não Funcionais

- O sistema deve ser fácil de usar e intuitivo.
- O sistema deve ser seguro e protegido contra acesso não autorizado.
- O sistema deve ser escalável e capaz de lidar com grandes volumes de dados

Arquitetura do Sistema

O sistema será baseado em uma arquitetura MVC(Model-View-Controller). O objetivo do servidor terá responsabilidade por separar a lógica de negócio da interface do usuário, permitindo que cada componente seja desenvolvido e testado independentemente, como assim:

- **Model:** O modelo é responsável por gerenciar os dados e a lógica de negócios. Ele representa a camada de acesso aos dados e é responsável por armazenar e recuperar informações do banco de dados. O modelo também é responsável por validar as informações inseridas pelo usuário e garantir que elas estejam corretas.
- **View:** A visualização é responsável por exibir os dados ao usuário. Ela representa a camada de apresentação e é responsável por apresentar os dados recuperados pelo modelo ao usuário. A visualização é responsável por fornecer uma interface de usuário para que o usuário possa interagir com o aplicativo.
- **Controller:** O controlador é responsável por gerenciar a interação do usuário com o aplicativo. Ele representa a camada de controle e é responsável por receber as solicitações do usuário, processá-las e enviar as informações relevantes para o modelo e a visualização. O controlador é responsável por garantir que as informações inseridas pelo usuário sejam processadas corretamente e que as informações relevantes sejam exibidas na visualização.

Tecnologias Utilizadas

- Banco de dados SQLite
- Linguagem de programação JavaScript
- Framework Node.js, Express, Sequelize e React