

Documentação – Sistema de Estoque

Requisitos funcionais

Eu como estoquista desejo cadastrar/alterar e excluir produtos no meu sistema de estoque.

Desejo as seguintes informações:

- Nome do produto
- Quantidade de itens do produto
- Valor unitário do produto

Desejo que seja disponível uma tela de listagem onde eu possa selecionar um produto para exclusão ou edição (exibir uma nova tela para edição).

Nesta tela de listagem também deverá ser possível chamar uma tela para cadastrar um novo produto.

Requisitos não funcionais

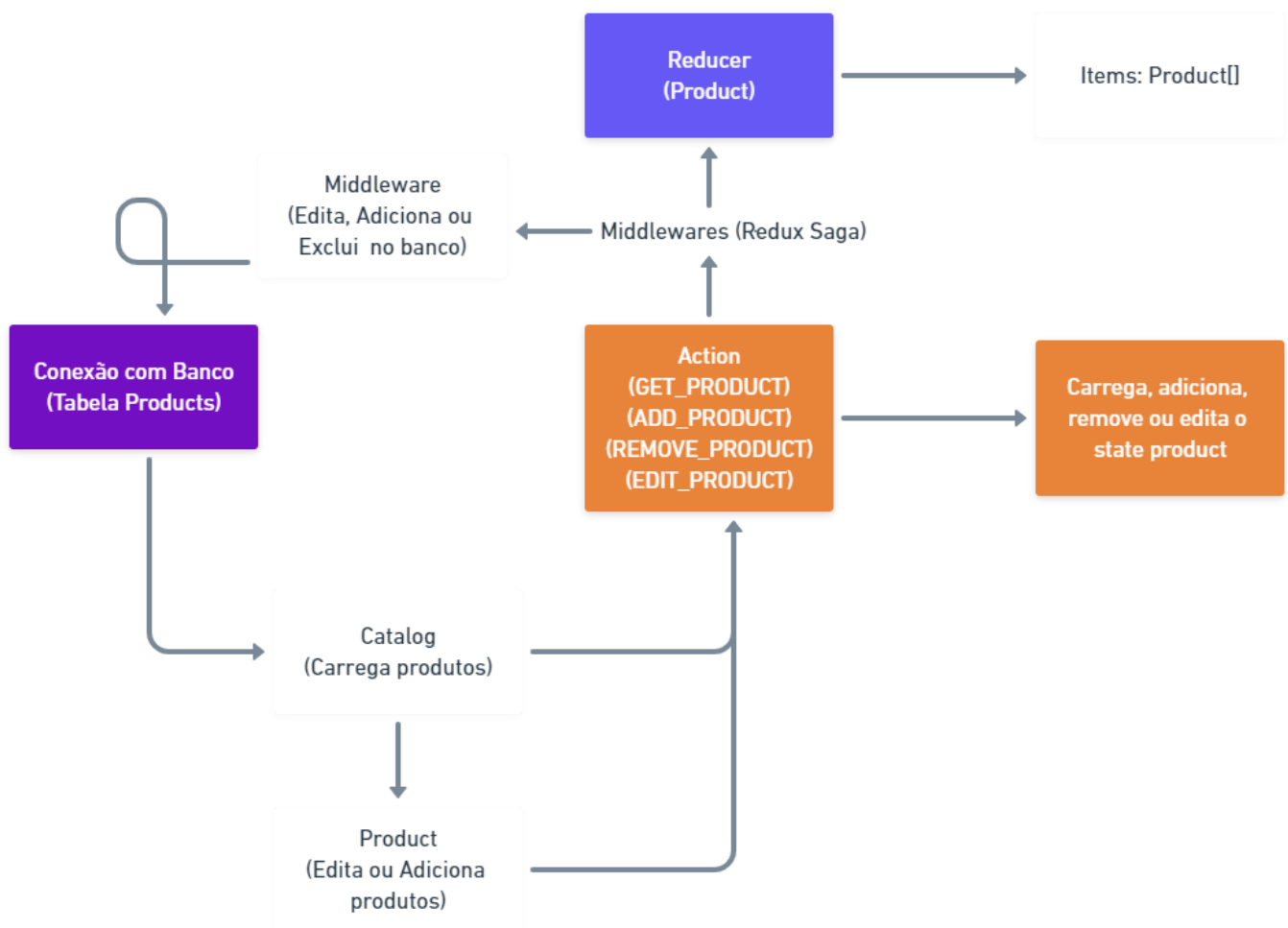
A história deverá ser desenvolvida utilizando os seguintes requisitos não funcionais:

1. O app deverá ser desenvolvido em React.
2. Backend: REST WebApi. Para o backend: linguagem e frameworks a sua escolha.
3. Banco de dados: Sqlite ou qualquer outro banco relacional de sua preferência.
4. Não é necessário fazer nenhum tipo de autenticação/autorização.
5. Deve ser utilizado Redux para gerenciamento do estado.

Implementação - Frontend

Foi utilizado as tecnologias de React e Redux para construir o sistema de gerenciamento de estado e as telas da aplicação.

As informações seguem o seguinte esquema:



Pensando na organização e na evolução do projeto ele está no seguinte formato:

- Pasta *pages*: Contém os componentes que gera as páginas do sistema
 - *catalog*: Carrega a lista de produtos e chama o componente *catalogitem* para cada produto e sua remoção
 - *products*: Página responsável por adicionar ou remover um produto de acordo com o parâmetro passado na página.
- Pasta *components*: contém componentes que podem ser reutilizados em todas as páginas.
 - *catalogitem*: Carrega a informação do produto e aciona a funcionalidade de editar e remover.
 - *header*: carrega a estilização do header

Explicando o funcionamento do estado no Redux:

- Contém *product* no store que gerencia o estado dos produtos na aplicação
- Possui as *actions*:
 - Remover Produto
 - Adicionar Produto
 - Criar Produto

- Para a criação dos estilos dos componentes foi utilizado a biblioteca de SASS.

Observação: Devido a problemas de tipagem no Redux não consegui implementar ativamente o gerenciamento de estados a tempo para a entrevista. Porém deixei todo o código implementado do Redux, porém utilizando o conceito de state do React para realizar as manipulações.

Implementação - Backend

Foi construído utilizando Nodejs e Express e para manipulação do Banco de dados foi utilizado Knexjs.

Basicamente o sistema faz a conexão com o banco de dados e cria as seguintes rotas:

- Get Products: Retorna todos os produtos da aplicação
- Get Product: Retorna um produto com base no id
- Delete: Remove o produto
- Create: Cria um novo produto
- Edit: Edita um novo produto

O formato dos dados utilizados foi:

Uma tabela de dados chamada *product* que contém:

- id,
- title: Nome do produto.
- quantity: Quantidade do produto
- price: Preço ou valor unitário do produto.