

Gerenciador de Estoque

Kevin e Luiz

Backend

Dividido em:

Controllers: São os manipuladores de requisições HTTP. Eles recebem uma requisição, processam a lógica de negócios necessária e retornam uma resposta.

Middlewares: São funções que têm acesso ao objeto de requisição (req), ao objeto de resposta (res), e à próxima função middleware no ciclo de solicitação/resposta do aplicativo. Eles podem executar qualquer código, fazer alterações nos objetos de solicitação e resposta, terminar o ciclo de solicitação/resposta, ou chamar a próxima função middleware na pilha.

Repositories: São uma camada entre a lógica de negócios e a persistência de dados. Eles lidam com a leitura e gravação de dados, geralmente de/para um banco de dados.

Routes: Definem os endpoints da API e como eles respondem às solicitações do cliente. Cada rota pode ter uma ou mais funções handler, que são executadas quando a rota é correspondida.

Dependências usadas

bcrypt- criptografar as senhas.

dotenv- define variáveis de ambiente.

eslint- formatação e padronização de código.

prettier- formatação e padronização de código.

express- fazer as rotas.

express-async-errors- tratar erros.

jsonwebtoken- gerar token do login.

nodemon- rodar.

pg-promise- consultas no banco.

Frontend

Dividido em:

Components (Componentes): São blocos de construção independentes e reutilizáveis que compõem a interface do usuário. Um componente controla uma parte da interface do usuário e pode conter outros componentes.

Contexts (Contextos): No React, Context fornece uma maneira de passar dados através da árvore de componentes sem ter que passar props manualmente em cada nível. É comumente usado para armazenar e compartilhar dados de estado global.

Pages (Páginas): Em aplicações de página única (SPA), as "páginas" são componentes que representam uma unidade completa de funcionalidade, como uma tela ou vista específica em sua aplicação. Elas geralmente contêm outros componentes e são associadas a uma rota específica.

Services (Serviços): São usados para separar a lógica de negócios dos componentes, como chamadas de API, validação de formulários, manipulação de eventos, etc. Eles ajudam a manter os componentes limpos e focados na apresentação.

Dependências usadas

MaterialUi- Biblioteca de componentes MUI.

anstack/react-query- Biblioteca para buscar, armazenar em cache e sincronizar dados assíncronos em aplicativos React.

apexcharts- Biblioteca de gráficos interativos em JavaScript.

axios- Cliente HTTP baseado em promessa para o navegador e node.js.

jspdf- Biblioteca para gerar PDFs em JavaScript.

react"- Biblioteca JavaScript para construir interfaces de usuário.

react-dom- Serve como ponto de entrada para a renderização do DOM e a compatibilidade com o servidor para React.

react-hook-form- Biblioteca para validar formulários em React.

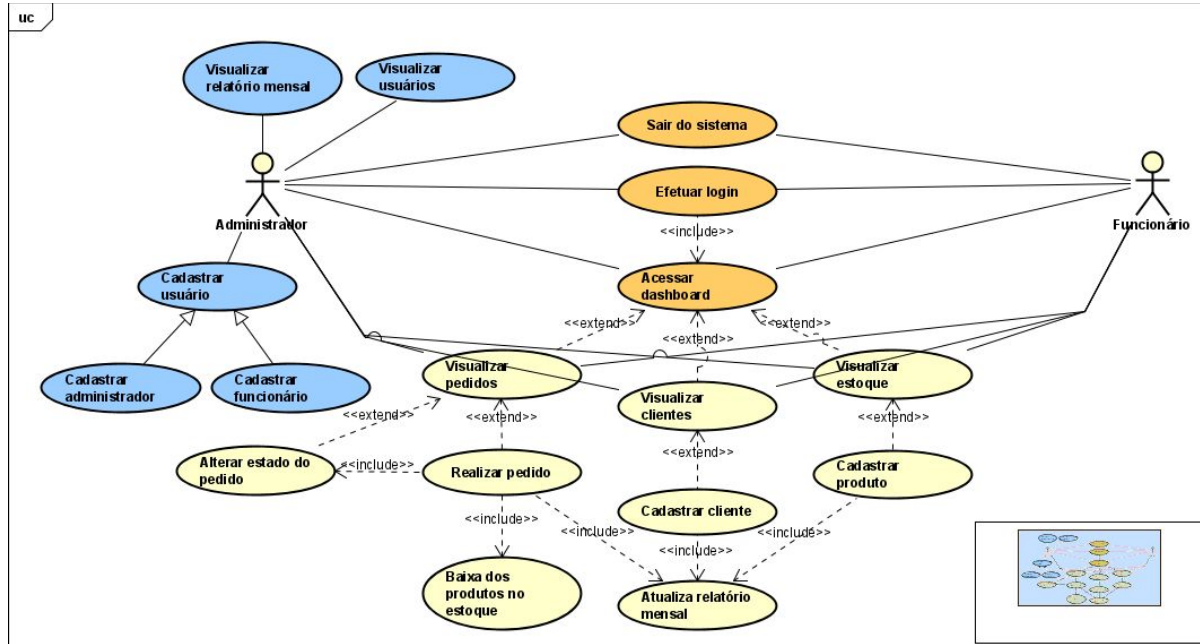
react-query- Biblioteca para buscar, armazenar em cache e atualizar dados assíncronos em aplicativos React.

react-router-dom- Biblioteca para roteamento em aplicativos React.

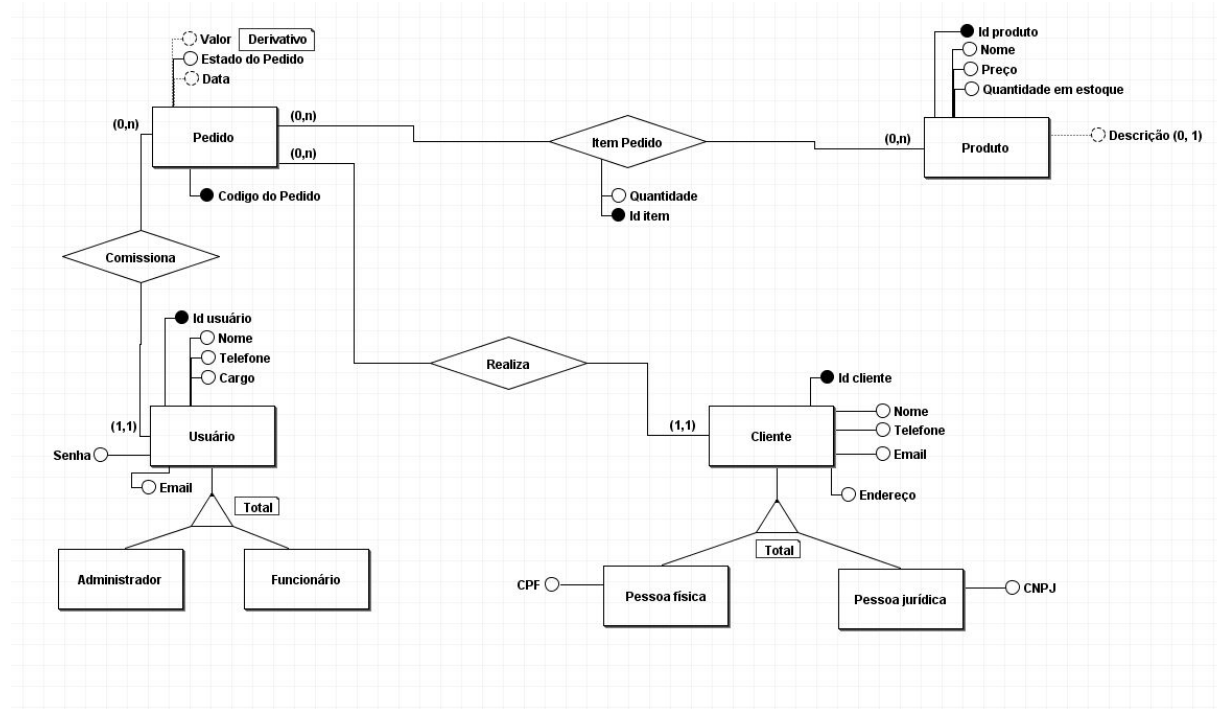
eslint- Ferramenta de análise de código estático para identificar padrões problemáticos encontrados no código JavaScript.

vite- Ferramenta de construção que visa fornecer um desenvolvimento mais rápido e mais eficiente.

Diagrama de casos de uso



Modelo conceitual



Modelo lógico

