

Desafio de Produto - Frontend

Neste desafio você precisa criar uma aplicação de lista e favoritos.

Este aplicativo tem como objetivo listar itens de duas ou mais fontes de API's distintas ao seu gosto. **Ex:** Uma API de um pet shop, outra para uma loja de doces, etc. A aplicação exibirá para o usuário todas as fontes de produto disponíveis. Quando o usuário selecionar uma, o conteúdo desta fonte deverá ser listado (como Mercado Livre, tipo de e-commerce da Netshoes). O usuário pode adicionar itens como favoritos e acessar em uma aba de favoritos.

Features:

- Fontes, pesquisa selecionadas persistem após o carregamento da página;
- Adicionar e remover itens dos favoritos;
- Classifique os itens por atributos, lembrando que cada API possui seus próprios dados, com seus próprios atributos;
- Pesquise itens por atributos, lembrando que cada API possui seus próprios dados, com seus próprios atributos;
- Posso ver todos os dados de fontes da API;
- Posso mudar a fonte da API;
- Posso visualizar e pesquisar a lista de itens;
- Posso adicionar e remover itens dos favoritos;
- Posso listar todos os meus favoritos independentemente da fonte;
- A página de favoritos deve listar todos os meus itens, independente da fonte, nenhuma categoria pode ser feita aqui, todos os itens devem ser exibidos na mesma lista.

Requirements:

- O Frontend deve ser implementado em ReactJS;
- As APIs implementadas podem estar em REST ou GraphQL. Você pode encontrar alguns exemplos aqui <http://apis.guru/graphql-apis/>
- Apollo GraphQL ou Axios podem ser usados;
- Os testes devem usar Jest. A Enzyme or React Testing Library pode ser usada como biblioteca de suporte;
- Quaisquer usuários e favoritos podem ser armazenados no armazenamento local. O uso de um "GraphQL as a service" ou um "Crud ready service" para armazenar e recuperar os dados é uma vantagem;
- Pode ser usado Bootstrap, Ant Design, Chakra UI para estilos;
- Se qualquer estrutura padrão for usada (templates, boilerplates), espera-se que os "recursos não usados" e os arquivos não usados sejam removidos (nenhum código ou arquivo inútil);
- Todas as boas práticas, estrutura de código e recursos mais recentes do framework escolhido serão avaliados;
- É esperado um bom README;
- A quantidade, hora e tamanho dos commits será avaliada.

Delivery:

- Desenvolver um script que efetue o build, configurações e tudo que for necessário para a aplicação poder ser testada (acessível via Browser);
- O código deve estar em um repositório público ou privado no github. No caso de um repositório privado, compartilhado **@ronnyfv** (usuário github).

Extra:

- O projeto pode ser executando em algum serviço de hospedagem (ex: Heroku, Vercel);