

Desafio

Módulo 4	React III
----------	-----------

Enunciado da avaliação

Você atua como freelancer e acaba de receber uma proposta de trabalho para uma listagem paginada das *exchanges* de cripto. Seu cliente especificou que deseja buscar essas informações da API do Coingecko.

- Campos a serem exibidos: name, image, year_established, country, trust_score, trade_volume_24h_btc
- URL:
`https://api.coingecko.com/api/v3/exchanges/?per_page=<itemsPorPagina>&page=<pageIndex>`

O tamanho da página deve ser de **100 itens** e o índice da página deve começar com 1.

Algumas regras:

- 1) O projeto deve ser construído com a metodologia **TDD**.
- 2) O usuário deve ser capaz de navegar entre as páginas através de dois botões: **próxima página** e **página anterior**.
- 3) O botão de **página anterior** deve ser desabilitado quando o usuário estiver na primeira página.
- 4) O botão de **próxima página** deve ser desabilitado caso o número de itens trazido pela API seja menor que o valor dos **itemsPorPagina**.
- 5) Caso a página atual possua 0 elementos, deve-se ser exibida a mensagem: “**Sem resultados**”.

- 6) A página deve ser estilizada com **styled-components**.
- 7) O usuário deve ser capaz de **filtrar**, por **nome**, o resultado da página atual trazido pela **API**. Ao passo em que o usuário digita, a lista deve ser **filtrada em tempo real**. O filtro precisa ser case-insensitive.
- 8) Caso o usuário navegue para a **próxima página** ou **página anterior**, o filtro deve ser **resetado**.
- 9) Caso exista algum texto no filtro, deve-se exibir uma mensagem: **Exibindo X de Y**, onde X é a quantidade de itens retornados pelo filtro e Y é o total de itens retornado pela API para aquela página.
- 10) As **Imagens** devem ser carregadas com Next/image.

O link abaixo exibe um exemplo de como a página deve se comportar. Não é necessário que a página tenha a mesma aparência, mas é importante que se tenha as mesmas funcionalidades.

- <https://desafioreactiii.vercel.app/>