



FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA

Tecnologia Análise e Desenvolvimento de Sistemas

Front-end design Engineering

Prof. Vinícius Marcili

CHECKPOINT 06

INSTRUÇÕES PARA A ATIVIDADE

1. Esta atividade é em **trio**.
2. Não serão aceitos trabalhos exatamente iguais. Lembrem-se somos pessoas diferentes, pensando de modo diferente para a solução do mesmo problema.
3. Não utilize imagens para criar o layout.
4. Evite o uso de CSS interno ou *inline*;
5. Envie a pasta compactada pelo sistema da FIAP sem os arquivos listados no gitignore
6. No rodapé da página, coloque o nome completo e RA do trio (na ausência -1 ponto por nome faltante)
7. Somente um integrante do trio precisa efetuar a entrega

Boa atividade!

CASE: Minhas cartas favoritas

Você é responsável por criar um sistema onde um usuário pode favoritar cartas de Pokémon TCG, exibindo informações uteis sobre as cartas. Apenas usuários logados devem ter acesso ao sistema. **A stack escolhida foi Next.js. A estilização pode ser feita com CSS puro ou Tailwind**

O Sistema deve fazer a integração com a API do pokémon TCG.

URL	https://api.pokemontcg.io/v2/cards
Metodo	GET
Exemplo	https://api.pokemontcg.io/v2/cards?q=name:charizard&orderBy=number,name&perPage=15 onde <u>q=name:charizard</u> estamos fazendo a busca por qualquer carta que tenha o nome charizard, e <u>orderBy</u> estamos ordenando pelo número e nome do pokémon A rota também aceita <u>page</u> e <u>pageSize</u> , onde page é o número da página atual e pageSize a quantidade de itens que devem retornar

Critérios de avaliação:

A avaliação consistirá dos seguintes pontos:

1. Organização do projeto (1,0 pontos)

- A estrutura das pastas e dos arquivos será levada em consideração (0,5 ponto)
- Typagens – Lembre de criar os tipos e usar eles de forma correta (0,5 ponto)
- **Os arquivos listados no arquivo .gitignore não devem ser entregues (o arquivo .gitignore em si deve ser entregue!) – penalidade de 50% dos pontos**
- **O sistema DEVE ser feito usando a estrutura de app, no caso do uso da estrutura de pages penalidade de 50% dos pontos**

2. Login (3,0 pontos)

- Criar uma página de login com campos de login e senha (1 ponto)
- Criar uma rota de autenticação no API Handler do Next, validar apenas a senha fixa em uma variável de ambiente Sensível (1 ponto)
- Tratamento de erros e redirecionamento para área logada devem ser levados em consideração, o usuário deve permanecer logado ao atualizar a página (1,0 pontos)

3. Tela de Busca de cartas (3,0 pontos)

- Após o usuário logar, o sistema deve exibir uma tela com um campo de busca por texto e uma lista pré carregada de cartas (GET na API) deve exibir nome (propriedade name), imagem (propriedade images.large), número (propriedade nationalPokedexNumbers?.[0]), tipos (propriedade array types) e hp (propriedade hp). (1,0 ponto)
- Deve retornar NO MÁXIMO 15 Cartas (1,0 ponto)
- Quando o usuário digitar algo, a busca deve ser feita na API e retornar os resultados também limitado a 15 cartas (1,0 ponto)

4. Favoritando cartas (2,0 pontos)

- O sistema deve possuir uma funcionalidade de favoritos, possuindo um botão que ao usuário clicar ele possa salvar esse favorito (1 ponto)
- Os favoritos devem ficar no localStorage, ao atualizar a página devo continuar com os mesmos favoritos (0,5 ponto)
- Em algum lugar no topo da tela deve ter um botão de logout . Ao deslogar o usuário, os favoritos devem ser limpos. (0,5 pontos)

5. Deploy na Vercel (1,0 ponto)

- Subir o projeto no GIT e fazer deploy da Vercel, colocar o link no README.md (Deixar o link PÚBLICO, é o segundo link e não o primeiro gerado)
- Lembrar de configurar as variáveis de ambiente para o projeto funcionar (como a senha solicitada no item 2)

Extra (1,0 pontos)

- Fazer uma página ao clicar em uma carta sem favoritar e exibir informações adicionais sobre a carta (campos livres) (0,5 ponto)
- Favoritar e remover o favorito por essa página de informações adicionais (0,5 ponto)

Dicas:

Fetch – Para fazer requisições e parametros na URL

- https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch
- https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch#sending_data_in_a_get_request

Arquivos auxiliares:

- Repositório geral: <https://github.com/vinimarcili/fiap-ads-cp6-tips>
- Tipagem do retorno da API (interface): <https://github.com/vinimarcili/fiap-ads-cp6-tips/blob/master/src/interfaces/response-api-pokemon.interface.ts>
- Componente de Input (Estilo feito com TAILWIND, se for usar e estiver usando CSS puro cuidado):
<https://github.com/vinimarcili/fiap-ads-cp6-tips/blob/master/src/components/Input/Input.tsx>
- Componente de Button (Estilo feito com TAILWIND, se for usar e estiver usando CSS puro cuidado): <https://github.com/vinimarcili/fiap-ads-cp6-tips/blob/master/src/components/Button/Button.tsx>
- Layout.tsx principal: <https://github.com/vinimarcili/fiap-ads-cp6-tips/blob/master/src/app/layout.tsx>

ARQUIVO OBRIGATÓRIO PARA CARREGAR AS IMAGENS DAS CARTAS: SEU ARQUIVO `next.config.ts` DEVE TER O CONTEUDO DO ARQUIVO ABAIXO:

<https://github.com/vinimarcili/fiap-ads-cp6-tips/blob/master/next.config.ts>

Exemplo

Pokémon TCG Explorer

Descubra cartas e detalhes de Pokémon

Lista de Pokémon

Busque um Pokémon...



#460 - Abomasnow

HP: 140

Grass



#359 - Absol

HP: 70

Darkness



#359 - Absol G

HP: 70

Darkness



#142 - Aerodactyl



#142 - Aerodactyl



#142 - Aerodactyl