



Atividade Obrigatória 1

Desafio Pokémon - Pokédex e Análises



Nesta atividade, utilizaremos a API aberta PokéAPI, que lista todos os pokémons das diversas gerações, contendo dados acerca de suas habilidades, cidades, nome, tipo, etc.

1 - A API:

Os dados que utilizaremos para essa atividade estão contidos numa API REST desenvolvida por Paul Hallett como um projeto de final de semana. Todas as informações acerca dos endpoints e dados disponíveis estão contidas em:

<https://pokeapi.co/>

A estrutura da API é bastante completa, o que a torna também bastante complexa. Você precisará filtrar as informações que são mais relevantes. Para fins da nossa atividade, analisaremos os dados que estão nos seguintes endpoints:

POKÉMON

GET /api/v2/pokemon/{id or name}/

ABILITIES

GET /api/v2/ability/{id or name}/

TYPES

GET /api/v2/type/{id or name}/

O primeiro endpoint retorna o JSON de um Pokémon, obtido a partir de seu ID ou Nome. O segundo, uma Habilidade, identificada também por seu ID ou nome. Cada Pokémon pode possuir múltiplas habilidades. O terceiro endpoint retorna dados acerca dos Tipos de Pokémons. Normalmente os Pokémons têm um ou dois Tipos distintos.

2 - A tarefa:

O objetivo do desafio é conseguir consumir essas APIs diretamente para nossa aplicação, permitindo consultar dados básicos sobre algum Pokémon à escolha do usuário. Há três dificuldades distintas nas entregas propostas. Você deveria resolver primeiro a tarefa básica, depois a intermediária, e só posteriormente a avançada.

2.1 - Tarefa básica:

Seu programa deve:

2.1.1 - Pedir ao usuário que defina um pokémon que ele gostaria de saber mais informações sobre. Essa escolha pode ser feita através do input do ID do Pokémon ou pelo nome.

2.1.2 - Fazer o display do nome do Pokémon, seu tipo e também sua lista de habilidades (na tela do terminal)

2.1.3 - Se o usuário quiser salvar esse pokémon em sua pokedex local para consultar mais tarde, ele pode fazê-lo. Aqui, os dados devem ficar armazenados na máquina, em um arquivo .json. Você escolherá a estrutura mais adequada.

2.2 - Tarefa Intermediária:

Seu programa deve:

2.2.1 - Além de mostrar uma lista com os nomes das habilidades, o programa deve fornecer a descrição dessas habilidades, conforme é possível obter no endpoint de ABILITIES.

2.2.2 - Mostar para quais tipos de pokémon esse pokémon não causa dano, para quais ele causa metade do dano e para quais ele causa o dobro do dano. Essas informações podem ser obtidas no endpoint de TYPES, dentro da chave 'damage_relations', da seguinte maneira: 'no_damage_to', 'half_damage_to', 'double_damage_to', respectivamente.

Exemplo:

```
id: 5
name: "ground"
▼ damage_relations: {} 6 keys
  ▼ no_damage_to: [] 1 item
    ▼ 0: {} 2 keys
      name: "flying"
      url: "https://pokeapi.co/api/v2/type/3/"
  ▼ half_damage_to: [] 1 item
    ▼ 0: {} 2 keys
      name: "bug"
      url: "https://pokeapi.co/api/v2/type/7/"
  ▼ double_damage_to: [] 1 item
    ▼ 0: {} 2 keys
      name: "poison"
      url: "https://pokeapi.co/api/v2/type/4/"
```

Relações de dano para o tipo 5, 'ground'

2.2.3 - Além de mostrar relações de dano, o programa deveria mostrar exemplos de outros pokémons que têm o mesmo tipo que o pokémon mostrado. Esse dado pode ser extraído da chave 'pokémon', também no JSON retornado pelo endpoint de TYPES.

2.3 - Tarefa Avançada:

2.3.1 - Todos os dados do seu programa devem ser exibidos no Browser ou em aplicativo mobile (usando REACT.JS, REACT NATIVE ou puro JS com JQUERY).

2.3.2 - O pokémon deveria exibir também sua imagem principal para que o usuário possa vê-lo. Esse dado encontra-se na chave 'sprites'.

2.3.3 - A pokédex deveria ter um LOGIN, e quando um usuário salva seus pokémons, eles deveriam ir para sua pokédex na nuvem, todos os dados armazenados no firebase. Essa é a primeira etapa para, no futuro, podermos fazer um joguinho de batalha pokémon entre nós hehehe

3 - Ferramentas que você provavelmente vai precisar usar:

3.1 - readline-sync

3.2 - Axios

3.3 - Manipulação de DOM, REACT.JS ou REACT NATIVE.