$GIF\ do\ funcionamento:\ {\tt https://github.com/jeffersoncarvalho/WEB\_2020-1/blob/master/TRABALHOS/pokemon.gif}$ 

# **Pokedéx**

## Parte 1

(Gonna catch'em all!)

O objetivo desta parte do trabalho, é acessar a **pokemon api**, listar os pokemons disponíveis em formato de paginação.

#### 1 - O acesso a API

A **pokemon api** é um site que disponibiliza diversas informações sobre os mais variados Pokemons. Os dados são apresentados em formato JSON. Para acessar a **raiz** da API, basta usar o endereço:

#### https://pokeapi.co/api/v2/

pal-park-area:	"https://pokeapi.co/api/v2/pal-park-area/"	
pokeathlon-stat:	"https://pokeapi.co/api/v2/pokeathlon-stat/"	
pokedex:	"https://pokeapi.co/api/v2/pokedex/"	
pokemon:	"https://pokeapi.co/api/v2/pokemon/"	
	"https://pokeapi.co/api/v2/pokemon-color/"	
pokemon-color:	"https://pokeapi.co/api/v2/pokemon-color/"	

Uma vez nesta página, note que é possível ter acesso a diversas informações sobre o universo Pokemon. No momento, o endereço que nos interessa é:

#### https://pokeapi.co/api/v2/pokemon/

Nesta página, você terá acesso a todos os 964 (até o momento) pokemons da API. Note as seguintes propriedades:

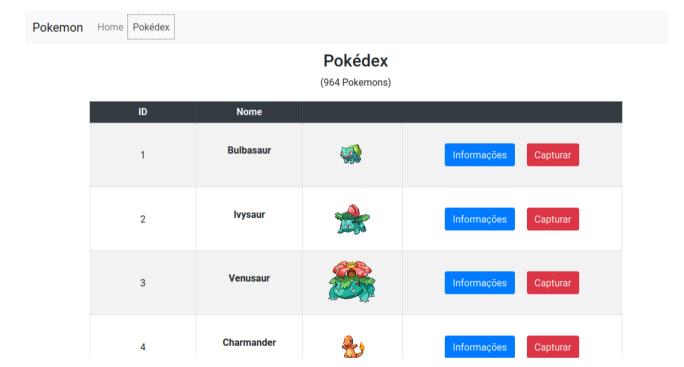
- **count**: a quantidade de pokemons cadastrados até o momento;
- **next**: a próxima página com mais pokemons, de acordo com os paâmetro da URL offset e **limit** (no caso, fixado pra 20);
- previous: a página anterior de pokemons;

- **results**: uma lista (vetor) com uma quantidade fixa de pokemons. Cada elemento da lista é um objeto JSON formado por:
  - o **name**: o nome do pokemon
  - **url**: mais informações sobre o pokemon.

Bem, agora que você entendeu mais ou menos o que é a API, eu sugiro ir clicando nos links e descobrindo por si próprio o que cada link mostra. Clique no next, previous e navegue pelos Pokemons.

#### 2 – A primeira atividade: listar os pokemons.

A primeira tarefa é criar uma página chamada Pokedex.jsx onde você irá listar os pokemons disponíveis ao clicar em <a href="https://pokeapi.co/api/v2/">https://pokeapi.co/api/v2/</a>. Minha sugestão é que você REUSE o mesmo código do projeto CRUD, visto em aulas anteriores. Veja como ficará a interface:

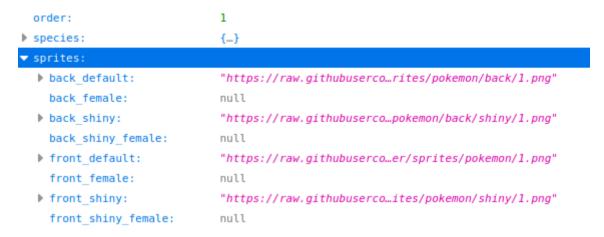


#### Sendo assim, você deve:

- 1. Criar uma tabela com quatro colunas (ID, Nome, Imagem e Ações).
- 2. Note que o ID de cada pokemon está no final de sua URL, dentro da propriedade results

3. O nome do pokemon deve estar **Capitalizado**, ou seja, a primeira letra deve estar em maiúsculo.

4. Você deve mostrar a imagem de frente do pokemon. Clicando na url de um pokemon, por exemplo, o Bulbasaur:

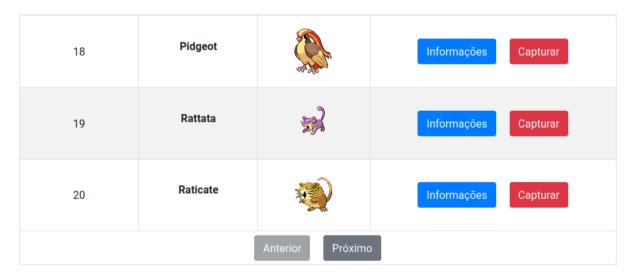


Você irá notar a propriedade **sprites** e dentro dela a propriedade **front\_default.** É essa a imagem que eu quero que você mostre na tabela.

5. Os botões **Informações** e **Capturar** ainda não terão NENHUMA funcionalidade. Apenas os coloque lá.

#### 3 – A segunda atividade: paginação

Ao rolar a página do Pokedéx, no rodapé da tabela teremos o seguinte:



Os botões **Anterior** e **Próximo** navegam entre as páginas de pokemons, monstrando de 20 em 20 pokemons (ou mais ou menos, caso queira). Sendo assim:

- 1. Implemente o botão Anterior fazendo uso da propriedade **previous**, ou seja, ao clicar em **Anterior**, a sua aplicação deve recarregar a página com os novos pokemons da página anterior.
- 2. O mesmo raciocínio para o botão **Próximo**.
- 3. Note que se não tem mais páginas anteriores e nem próximas, o botão deve ficar desabilitado.

## Parte 2

(I choose you!)

O objetivo desta parte é mostrar informações sobre o Pokemon selecionado, em outra tela.

#### 1 – Acessando mais detalhes de um Pokemon, atráves da api.

Cada Pokemon tem uma url associada. Se você clicar na url, verá o seguinte (para o Bulbasaur):

```
abilities:
                            [...]
 base experience:
                            64
forms:
                            [...]
pame indices:
                            [...]
 height:
                            7
 held items:
                            []
 id:
                            1
 is default:
                           true
 location area encounters: "https://pokeapi.co/api/v2/pokemon/l/encounters"
moves:
                            [...]
 name:
                            "bulbasaur"
 order:
                            1
> species:
                            {...}
sprites:
                            {...}
> stats:
                            [...]
types:
                            [...]
 weight:
                            69
```

Note que são MUITAS informações para cada Pokemon. Caso você realmente esteja inspirado, poderia criar um jogo realmente GRANDE, com todas as informações, em especial, as de movimentos (propriedade **moves**).

#### 2 – A terceira atividade: mostrar as informações de um Pokemon, em outra página.

Você deverá criar uma página chamada **PokemonInfo.jsx**, que exibirá as informações de um Pokemon selecionado na Pokédex. Veja:

#### **Pokemon Info**



As informações deverão ser exibidas em uma tabela (ou noutra forma que você achar melhor esticamente), da seguinte forma:

- Use um axios.get para acessar o obejto JSON que representa um Pokemon, assim como você fez para acessar a lista de Pokemons na Pokédex.
- Mostre as imagens de frente e costas do Pokemon (propriedade images)
- Mostre também: ID, Height, Order e Weight.

#### 3 – A quarta atividade: mostrar movimentos (golpes) de um Pokemon.

Logo abaixo das informações principais, você também deve mostrar os movimentos de um Pokemon selecionado, veja:

Venusaur				
	<ul><li>ID:3</li><li>Height:20</li><li>Order:3</li><li>Weight:1000</li></ul>			
Movimentos				
swords-dance	https://pokeapi.co/api/v2/move/14/			
cut	https://pokeapi.co/api/v2/move/15/			
bind	https://pokeapi.co/api/v2/move/20/			
vine-whip	https://pokeapi.co/api/v2/move/22/			
headbutt	https://pokeapi.co/api/v2/move/29/			

Você deve mostrar o nome e o link para a página com MUITOS mais detalhes do movimento em questão. Não se preocupe que não iremos renderizar essas informações. Para acessar esses dados, você terá que usar a propriedade **moves** que é um vetor de objetos. Veja:

No caso do movimento <b>razor-wind,</b> a propriedade <b>move.name</b> , diz seu nome. E a propriedade <b>move.url</b> , a url com mais informações.			

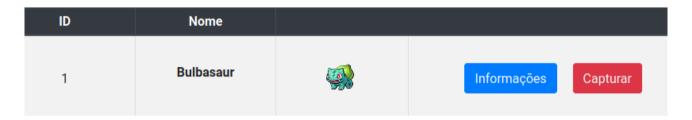
### Parte 3

(POKEBALL, GO!)

O objetivo desta parte é capturar os Pokemons listados na Pokédex (máximo de 6).

#### 1 – A quinta atividade: capturar um Pokemon.

Para capturar um Pokemon, você deve clicar no botão "Capturar".



Ao capturar um Pokemon, armazene informações como ID e nome dentro de um objeto. Este objeto deverá ficar dentro de um Array (a pokebola ou **pokeball**), o qual será armazenado no SessionStorage. Veja um exemplo:

capturar(id,nome){

```
let pokeball = JSON.parse(sessionStorage.getItem('pokeball'))
       if(!pokeball){
              pokeball = []
       }
       if(pokeball.length===6){
              alert('Capacidade máxima da Pokebola atingida.')
       }
       for (let index = 0; index < pokeball.length; index++) {
              const element = pokeball[index];
              if(id===element.id) {
                      alert('Pokemon já capturado! Escolha outro.')
              }
       }
       pokeball.push({id:id,nome:nome})
       sessionStorage.setItem('pokeball',JSON.stringify(pokeball))
       alert('Pokemon capturado com sucesso!')
}
```

Note que este código também testa se a pokebola (pokeball) já está cheia (6 Pokemons) ou se já tem o Pokemon que você quer caputar.

Veja mais sobre SessionStorage e LocalStorage em: <a href="https://www.treinaweb.com.br/blog/quando-usar-sessionstorage-e-localstorage/">https://www.treinaweb.com.br/blog/quando-usar-sessionstorage-e-localstorage/</a>

O próximo passo é EXIBIR os Pokemons na Pokebola. Para isso, você irá criar uma nova página chamada **Pokeball.jsx**. Veja como ficará seu layout:

Pokemon Home Pokédex Poké Ball

### Poké Ball

ID	Nome			
1	Bulbasaur			
3	Venusaur			
6	Charizard			
Pokédex				

Crie um link no Menu, para acessar a Pokebola. Mostre todos os Pokemons dentro da Pokebola e um botão para voltar para a Pokedéx.

#### Observações:

- 1. Este trabalho não deve ser entregue e também não vale nenhuma nota ou presença. Tente fazê-lo com o intuito de aprender e treinar.
- 2. Eventualmente eu irei disponibilizar a resolução como videoaula. Sugiro que você TENTE fazer antes de partir para a videoaula.
- 3. Este trabalho, pelo menos a Parte 01, não usará o MongoDB, apenas o AXIOS, para acessar a API dos pokemons.

© 2020 Pokémon. TM, ® Nintendo.