

Meu Time Pokémon

1. Introdução

Meu time Pokémon é um projeto elaborado para que os usuários possam cadastrar times com seus pokémons favoritos e ver o time de outras pessoas, com as combinações dos pokémons favoritos de cada um.

2. Objetivo

O projeto propõe um desafio visando o estudo, aprendizado e/ou prática de conceitos que vão do básico ao intermediário no desenvolvimento de aplicações web.

Os seguintes temas serão abordados no desenvolvimento do aplicativo proposto:

- Consumo de API externa;
- Desenvolvimento de API Rest;
- Cadastro, Leitura, Alteração e Deleção de informações (CRUD);
- Login social (login com serviços de terceiros, Google, Facebook, Github, entre outros);
- Interação com banco de dados;
- Gerenciamento das interações do usuário com o app;
- Formulário com validação de campo para arquivo e upload de imagem.

3. Motivação

O projeto surgiu numa conversa entre amigos pensando em algo que poderia ser desenvolvido para praticar o desenvolvimento web, porque muitas vezes aprendemos algo e não colocamos muito em prática e consequentemente esquece-se daquilo que aprendeu. Como é algo recorrente, pensamos “que tal desafiar outros devs?” e com isso decidimos escrever essa mini documentação com as especificações e divulgar para quem quiser participar do desafio.

4. Proposta

Consumir dados da pokeAPI* para construir do zero um aplicativo fulltsack onde sejam listados os pokémons existentes e seja possível um usuário se cadastrar e escolher os pokémons favoritos para compor um time.

O aplicativo deve ser construído em dois módulos, um para o frontend e um para o backend seguindo as especificações funcionais e técnicas descritas nas seções seguintes.

* pokeAPI é uma API Rest pública, onde todos os dados sobre pokémons são fornecidos, sua documentação pode ser encontrada em <https://pokeapi.co/docs/v2>.

5. Especificação Funcional

- Deve ser possível um usuário se cadastrar e se autenticar;
- Deve ser possível listar todos os pokémons (todos os 151 da primeira temporada são obrigatório. Além deles fica a critério do desenvolvedor incluir ou não);
- Deve ser possível criar um time com os 6 pokémons favoritos do usuário.
- Não deve ser possível criar times com mais de 6 pokémons;
- Não deve ser possível repetir o mesmo pokémon no time.
- Deve ser possível editar o time, ou seja, remover um pokémon para adicionar outro.
- Deve ser possível visualizar os times dos outros usuários.

6. Especificação Técnica

6.1 - Frontend

O frontend da aplicação deve ter as 5 telas que seguem:

- **Landing page (página de apresentação):** Deve contar uma apresentação do aplicativo e botões para o usuário se cadastrar ou se autenticar utilizando um login social*;
- **Home Page:** Deve mostrar a listagem de todos os pokémons, permitindo que o usuário selecione quais quer para compor seu time.
Deve mostrar o time sendo formado, para o usuário ter o controle de quantos e quais pokemons foram adicionados no time.
(opcional: fazer paginação para otimizar o carregamento da página, já que serão muitos pokémons sendo exibidos em tela).
Paginação é um conceito utilizado em praticamente qualquer site, aplicativo que tenha um carregamento de lista de dados. Desenvolver esse passo opcional deixará o projeto mais rico.
- **Meu Time:** Deve mostrar informações com mais detalhes sobre o time do usuário, dando opção de remover e adicionar novos pokémons;
- **Times:** Deve mostrar uma listagem com o time de todos os usuários;
- **Configurações:** Deve mostrar os detalhes da conta do usuário. (opcional: deve permitir que o usuário faça upload de uma imagem de avatar para seu perfil).

O nome das páginas é apenas para ilustrar o que elas devem conter, sinta-se livre para alterar o nome de cada página como achar mais conveniente.

* Login Social é uma forma de autenticação utilizando serviços de terceiros, como Google, Facebook, Apple, Github.

6.2 - Backend

- O backend do aplicativo deve ser uma API construída tentando ser o mais próximo possível do padrão Rest (<https://restfulapi.net/>).
- Deve ser possível salvar informações do usuário recebidas do provedor de autenticação (login social) em um banco de dados;
- Deve ser possível salvar informações do time do usuário em banco de dados;
- Deve fornecer os dados do usuário autenticado e do seu time;
- Deve fornecer uma listagem do time de todos os usuários;
- (opcional: Deve ser possível fazer upload de uma imagem de avatar para o usuário).

7. Conclusão

Todas as especificações funcionais e técnicas foram escritas de forma a guiar e facilitar o desenvolvimento do aplicativo, mas se você quiser participar e fazer de outra forma, alterando algumas especificações, adicionando ou removendo outras, fique à vontade. A ideia é botar a mão na massa (ou no código) e se divertir.

As especificações foram escritas de forma que possam ser construídas com a biblioteca, framework ou linguagem de programação preferida por cada um, escolha a sua e bora codar.

Apêndice A

"Sou dev frontend e não tenho interesse em aprender backend, como participar?"

Desenvolva apenas o frontend e salve os dados do time favorito do usuário no localStorage dele. Fazendo dessa forma não será possível ter acesso ao time dos outros usuários, então você pode remover a página de times.

Outra opção é utilizar APIs fakes, como MirageJS ou JSON-server para salvar os dados

MirageJS: <https://miragejs.com/>

JSON-server: <https://www.npmjs.com/package/json-server>

Apêndice B

“Sou dev backend e não tenho interesse em aprender frontend, como participar?”

Desenvolva apenas um backend que fique completinho, como se ele fosse ser consumido por um frontend.

Apêndice C

Curiosos pra saber a stack de tecnologias que os elaboradores do desafio vão usar?

Fernando Cardozo

- Typescript no back e no front.
- Frontend:
 - React
 - NextJS
 - Styled Components
- Backend
 - NodeJS
 - Express
 - Postgres
 - PrismaIO

Vinicius Troyack

- Javascript no back e no front.
- Frontend:
 - React
 - NextJS
 - Styled Components
- Backend
 - NodeJS
 - Express
 - Postgres
 - PrismaIO

Apêndice D

Concluiu o desafio?

Manda o github pra gente ver, será um prazer dar uma olhada na solução de cada um.

Fernando: fhpcardozo@gmail.com

Vinicius: troyackvinicius@gmail.com