

NÍVEL BÁSICO



PROJETO 02

```
(CONTEÚDO DISPONÍVEL) {  
WIKI;  
PARA;  
FANS;  
(end);  
})();
```

#PORTFÓLIOBOOSTPROGRAM

CONHECIMENTOS REQUIRIDOS:



FRONT-END

WIREFRAME

Nome do seriado/filme/desenho



Personagem



Personagem



Personagem



Personagem



Personagem



Personagem



Personagem



Personagem



Personagem



Personagem



Personagem



Personagem

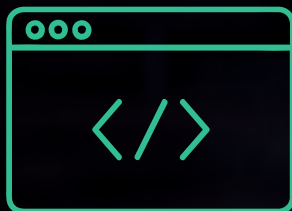
WIKI PARA FANS

Crie um **wiki de fãs** que inclua todos os personagens de seu programa de TV ou filme favorito.



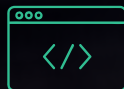
TECH STACK

- React
- NodeJS
- GraphQL (opcional)



LIBRARIES

- React-Router
- Framer Motion (animação)
- React Spring (animação)



BRIEFING

Há uma enorme comunidade de fãs de sua série de TV favorita que deseja uma maneira fácil de acessar informações de personagens individuais. **Já imaginou ter no seu portfolio uma wiki de todos os personagens do seu desenho/filme favorito?**

Provavelmente essa API já existe, se não existir essa é a **oportunidade** para você criar uma.

NÍVEL 1

Podemos usar uma **API** para criar dinamicamente um wiki de todos os personagens do show. Crie um wiki de personagem usando uma API que lista todos os personagens de sua série de TV favorita.

NÍVEL 2

Ver uma lista de personagens é útil, mas as pessoas querem saber mais detalhes sobre cada personagem. O que os torna únicos na série?

Adicione **páginas dinâmicas** ao wiki do personagem, permitindo que as pessoas naveguem até os personagens para ver mais informações.

NÍVEL 3

Procurar um personagem de sua série de TV favorita não precisa ser chato, existem qualidades sobre esse programa que o tornam único.

Use algumas das características do programa de TV para fazer com que o wiki do personagem pareça pertencer ao universo da TV com estilos e animações personalizadas.



REQUISITOS DETALHADOS

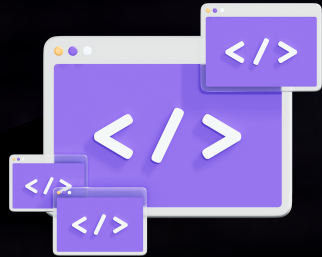
BACK-END:

- ➔ Crie um novo aplicativo **Node.js**.
- ➔ Crie uma **API** que lista todos os personagens de sua série de TV favorita.
- ➔ Buscar dados sobre os personagens.
- ➔ Armazene os dados em um **banco de dados**.
- ➔ Crie uma **API REST** para expor os dados ao **front-end**.

Quer Percorrer a milha extra nesse projeto? Tente desenvolvê-lo com **GraphQL**. As variáveis quase todas vão girar em torno do **type string**, isso facilita bastante a criação de **schemas** caso você não tenha experiência com GraphQL

FRONT-END:

- ➔ Crie um novo aplicativo **React**.
- ➔ Instale o **React Router**.
- ➔ Consuma os dados da **API REST**.
- ➔ Crie um wiki de personagem com **páginas dinâmicas**.
- ➔ Adicione estilos e animações **personalizados**.



Vale lembrar que:

O **back-end** deve ser responsável por **buscar dados**, **armazenar dados** e **fornecer uma API REST**.

O **front-end** deve ser responsável por **exibir dados**, **interagir com o usuário** e **fornecer uma interface de usuário**.

As duas partes devem ser dissociadas para que possam ser desenvolvidas de forma independente.

O que isso quer dizer?

Separe a lógica em **folders** que tenham nomes bem semânticos e distintos de acordo com a sua responsabilidade.
Exemplo: Web para Front e API para back.

