



Instruções Gerais:

- Atividade Individual
- Entregar a solução completa em formato ZIP pela tarefa do SIGAA.
- **Horário de Entrega definido na Tarefa do SIGAA**
- Formato de Entrega: **Apague a pasta node_modules**. Compacte o projeto em formato **".zip"** e faça a entrega no SIGAA.
- Para essa tarefa não é necessário criar um backend.

Objetivos: Familiarizar com os conceitos iniciais do React/NextJS.

- Inicializar um projeto React e/ou NextJS.
- Criar Componentes
- Composição de Componentes
- Mapear dados de uma API externa para um Componente React
- Conectar o CSS com Componentes React
- Usar estados (useState)
- Desafios que podem ser encontrados:
 - Ler valores do <input />
 - Tratar erros de requisição quando o recurso não existir
 - Atualizar estado de objetos
 - Impedir imagens inexistentes (src vazio) serem renderizadas

Tarefa Única: Sistema de Busca do Universo Star Wars

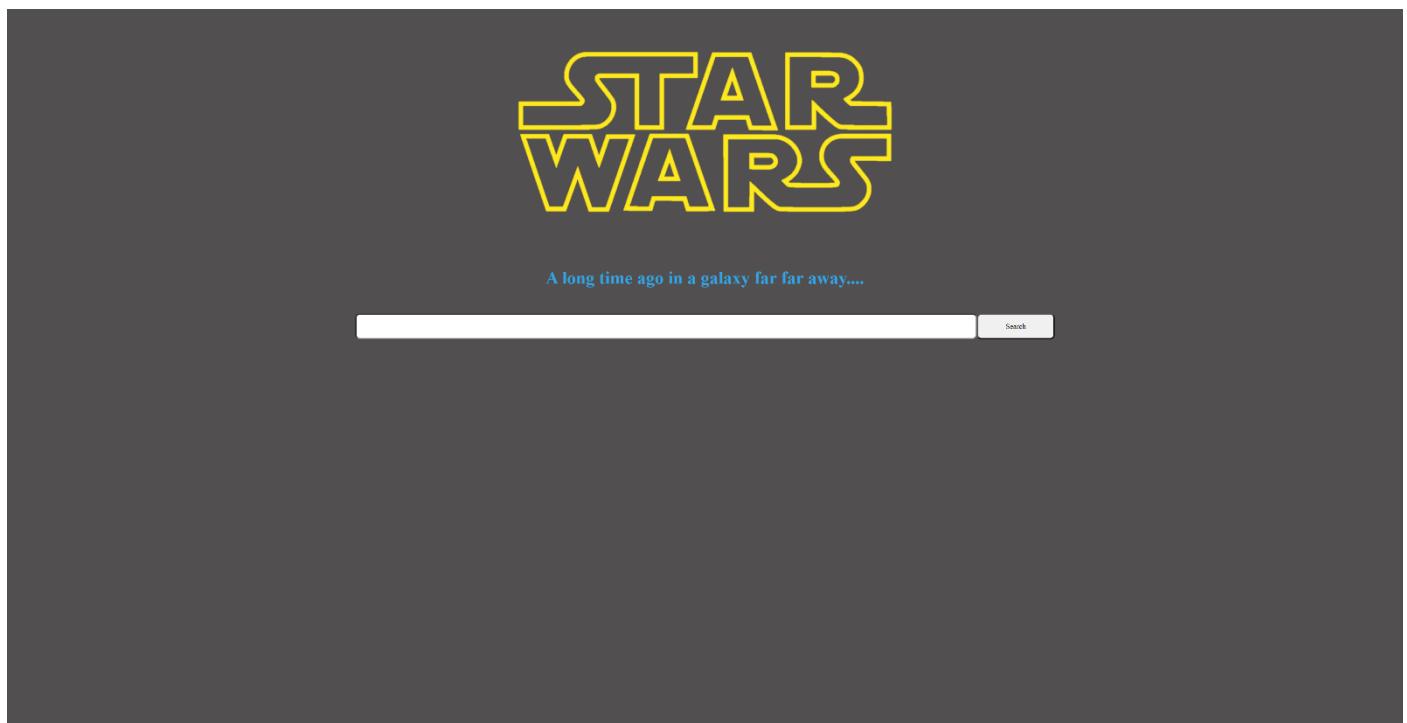


Figura 1

Nessa tarefa você deverá criar uma aplicação web, usando React ou NextJS, que envia requisições para a Star Wars Databank (<https://starwars-databank.vercel.app/>) e o nome de um personagem do universo Star Wars. A API irá retornar dados com: Nome, descrição e imagem do personagem. A Figura 1 apresenta a tela quando a aplicação é carregada.

Utilize o *endpoint*: [https://starwars-databank-server.vercel.app/api/v1/characters/name/\\${personagem}](https://starwars-databank-server.vercel.app/api/v1/characters/name/${personagem})

Requisito 1: Busca do personagem pelo nome

Exemplo buscando o Darth Vader:

https://starwars-databank-server.vercel.app/api/v1/characters/name/Darth_Vader (na prática o navegador irá trocar o espaço em branco por %20)

Observe a letra maiúscula no primeiro caractere de cada palavra e o espaço em branco (%20).

As Figuras 2-4 apresentam, respectivamente, a busca por Darth Vader, Leia Organa e Luke Skywalker

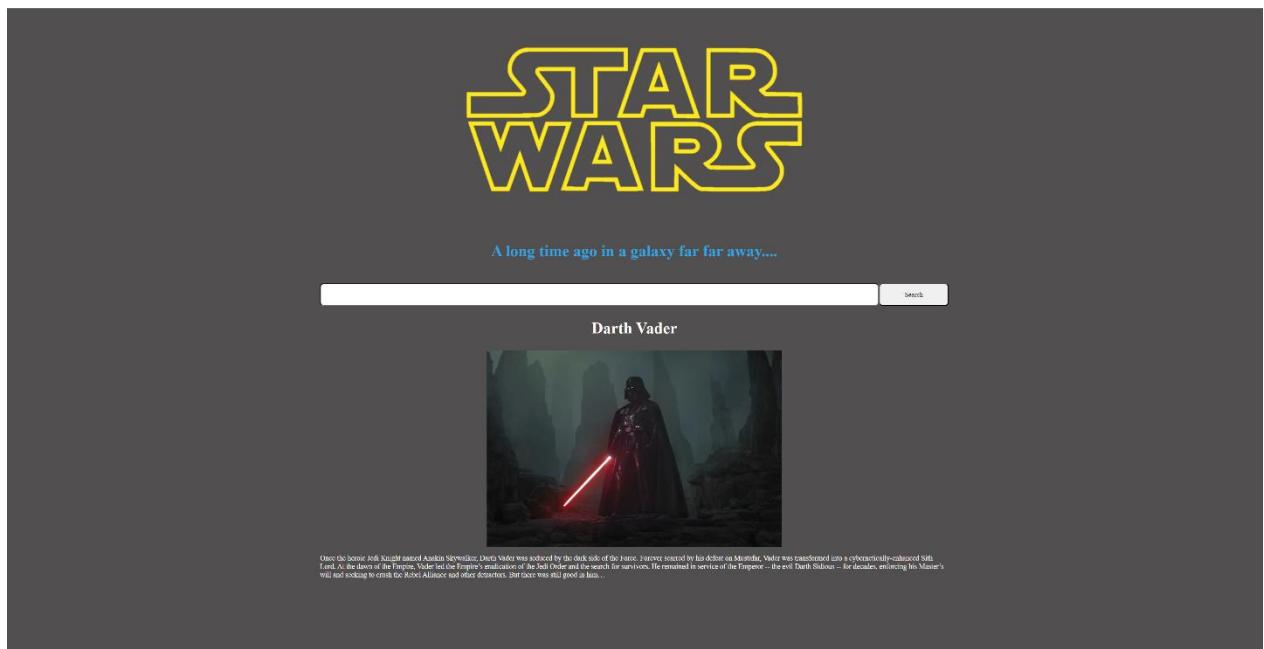


Figura 2

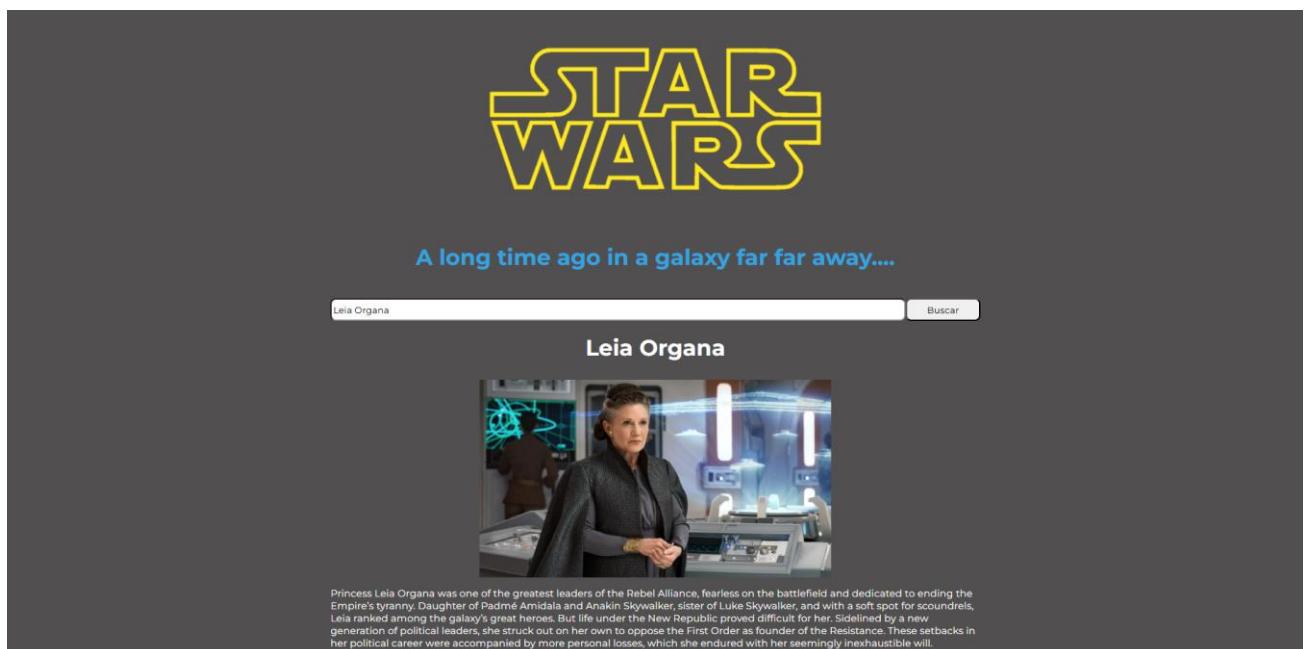


Figura 3

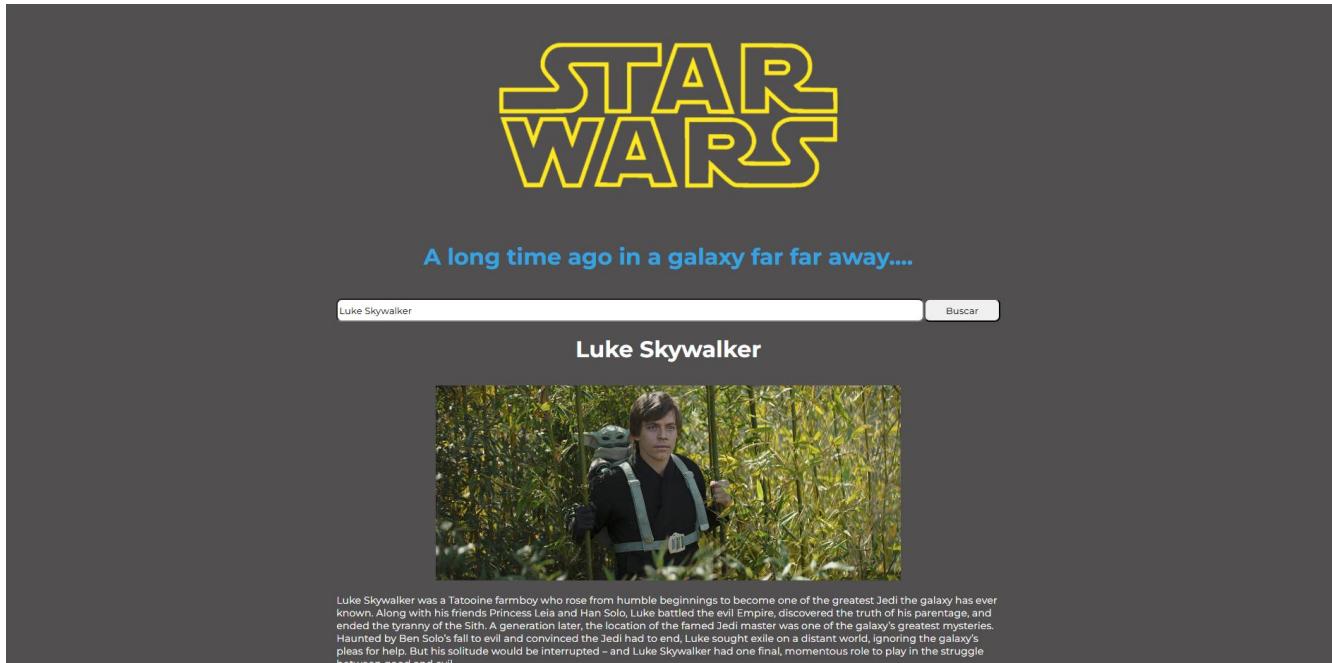


Figura 4

Requisito 2: Busca com nome incorreto ou campo vazio.

Ao deixar o campo em branco ou passar um nome incorreto, API não consegue encontrar o personagem. Sua aplicação deverá tratar isso. Existem algumas alternativas.

1. A mais simples é apenas informar que o personagem não existe.
2. A aplicação pode, antes de fazer a requisição, formatar o nome passado. Por exemplo, passar o primeiro caractere para maiúsculo de cada palavra. Mesmo assim, pode haver erros de digitação e a API não irá encontrar o personagem.

Quaisquer opções utilizadas atendem ao exercício.

Caso o personagem não seja encontrado, uma mensagem de erro deverá ser mostrada como mostrado na Figura 5.

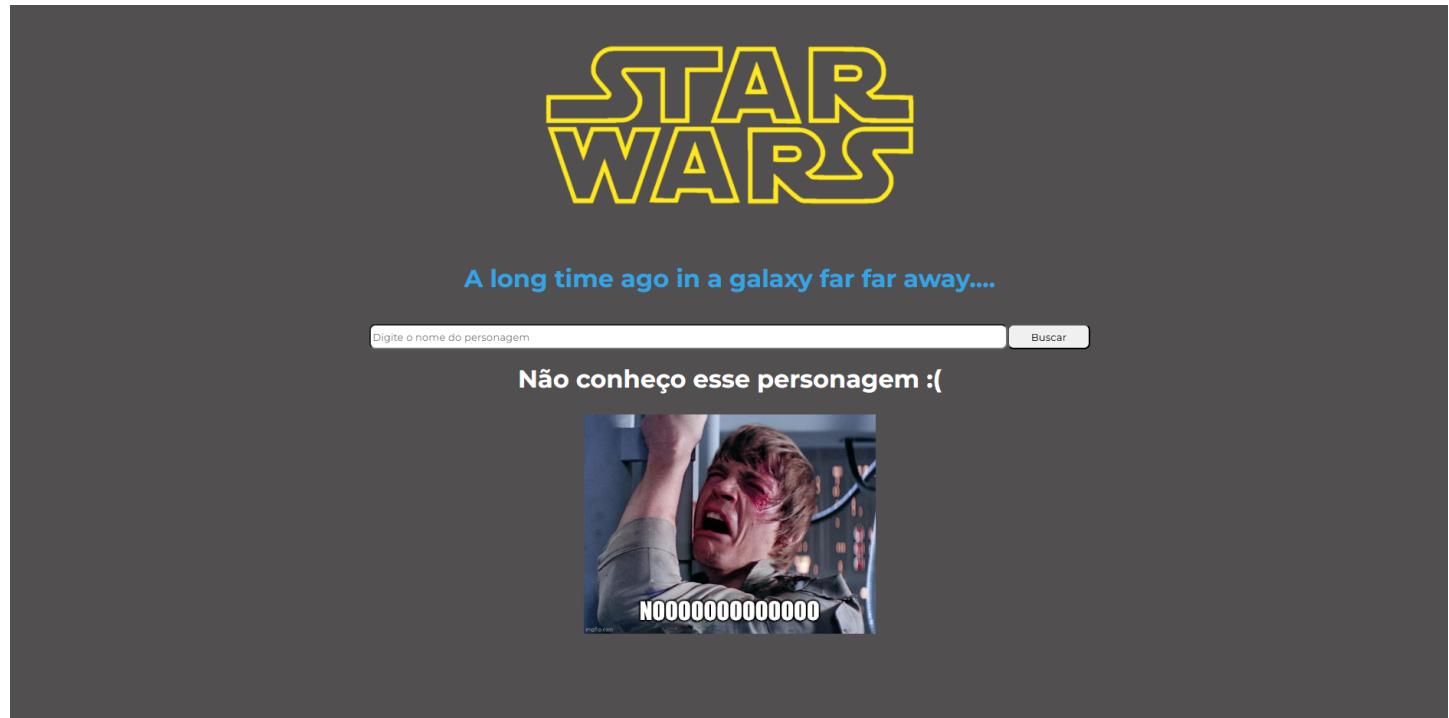


Figura 5

Dicas:

1. Não se preocupe exageradamente com o CSS.
2. Verifique o que a API retorna. Como é a estrutura do JSON? Como extrair os dados de: Nome, descrição e imagem?
3. Verifique o status de retorno:
 - a. Exemplo:

```
personagemData = await axios.get(url);
console.log(personagemData.status);
```
 - b. Faça testes “passando nome incorreto”, tipo “Darth Squirtle”. A API pode retornar status 200 (sucesso), mas os dados em branco. Verifique como tratar isso.
 - c. Faça testes deixando o campo vazio. Qual o status de retorno? Você pode usar Try-Catch.

Divirtam-se 😊

