Bootcamp React 3.0 - Cap. 08

Integração com API

# Competências

* Props
* Mais sobre layout
  + Tela ProductDetails
  + Tela Catalog
  + Tela Admin
* Mais sobre Rotas
  + Parâmetros de URL
  + Hierarquias (nesting)
  + Redirecionamentos
* Integração com back end
  + Axios
* React Hooks
  + useState
  + useEffect
* Efeitos: “loaders”
* Formulários “raiz”
  + Tratamento de eventos
  + Manipulação do estado do formulário
  + Submissão de formulário

# Figma do DSCatalog

<https://www.figma.com/file/cNa2l3TqZXxbU6NBDPruNw/BDS-DSCatalog>

## Props

<https://pt-br.reactjs.org/docs/components-and-props.html>

## Estado e ciclo de vida

<https://pt-br.reactjs.org/docs/state-and-lifecycle.html>

## Formatação de números apenas com JavaScript

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Intl/NumberFormat>

## Planejamento inicial de rotas

| **ROTA** | **Página (componente)** |
| --- | --- |
| / | Home |
| /products | Catalog |
| /products/:productId | ProductDetails |
| /admin | Admin/Products (redirecionamento) |
| **/admin**/auth/login | **Admin/**Auth/Login |
| **/admin**/auth/signup | **Admin/**Auth/Signup |
| **/admin**/auth/recover | **Admin/**Auth/Recover |
| /admin/products | Admin/Products/List |
| /admin/products/create | Admin/Products/Form |
| /admin/products/:productId | Admin/Products/Form |
| /admin/categories | Admin/Categories/List |
| /admin/categories/create | Admin/Categories/Form |
| /admin/categories/:categoryId | Admin/Categories/Form |
| /admin/users | Admin/Users/List |
| /admin/users/create | Admin/Users/Form |
| /admin/users/:categoryId | Admin/Users/Form |

## Instalar Axios

yarn add axios

## React hooks

<https://pt-br.reactjs.org/docs/hooks-overview.html>

Hooks são funções que permitem a você “ligar-se” aos recursos de state e ciclo de vida do React a partir de componentes funcionais. Hooks não funcionam dentro de classes.

## Trabalhando com parâmetros de URL: useParams

Estudo de caso com página ProductDetails

## Parâmetros da requisição do Axios

export type AxiosParams = {

method?: Method;

url: string;

data?: object;

params?: object;

};

## Página Spring

export type SpringPage<T> = {

content: T[];

last: boolean;

totalElements: number;

totalPages: number;

size: number;

number: number;

first: boolean;

numberOfElements?: number;

empty: boolean;

};

## Loaders

<https://skeletonreact.com/>

<https://github.com/danilowoz/react-content-loader>

## Dicas para inspecionar página

Abrir inspecionar -> Segurar o botão de reload do Chrome -> Empty Cache and Hard Reload

Network -> Online -> Slow 3G

# Estudo de caso: formulário “raiz”

## Código fonte:

<https://github.com/devsuperior/bds07>

## Design Figma:

<https://www.figma.com/file/3ChbzdkeZAsFP6IJXfUudT/estudo-de-caso-viacep>

## Formulários no ReactJS

<https://pt-br.reactjs.org/docs/forms.html>

## End point da API ViaCEP

<https://viacep.com.br/ws/01001000/json/>

## Checklist de preparação do projeto:

* Criar projeto com create-react-app
* Limpar o projeto
* Adicionar bootstrap
* Adicionar sass
* Pegar fonte indicada no design Figma
* Customizar tema do bootstrap (custom.scss)

Arquivo App.css:

@import url('https://fonts.googleapis.com/css2?family=Ubuntu:wght@400;700&display=swap');

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: 'Ubuntu', sans-serif;

}

# 

# Desafio para entregar

**TAREFA: Github API**

Construa um pequeno app conforme design abaixo, que disponibilize um formulário para buscar os dados de um usuário do Github, utilizando a API pública do Github.

**Opcional**: mostre loaders enquanto a requisição é realizada.

## Design Figma

<https://www.figma.com/file/YhRCC5Ly0tWOcxwxjheTJ3/desafio-github-api>

## End point da API Github

<https://api.github.com/users/acenelio>