



Ancorados



Photo by Vishal Jadhav on Unsplash

Módulo 3 - Aula 5 - HARD



Todos os direitos reservados
©2022 Resilia Educação

Hooks



Hooks

Hooks são **funções** que permitem a conexão dos recursos de estado e ciclo de vida do React, a partir de componentes funcionais.

Eles não substituem conceitos básicos no React

São **retrocompatíveis**, ou seja, não possuem nenhuma quebra de compatibilidade.

Hooks - useState

Estado é **todo dado que varia com o tempo**, sendo que essa alteração pode ser causada ou não pela interação do usuário.

Os estados são utilizados para valores dinâmicos, enquanto as props são utilizadas para passar valores entre componentes (somente leitura).

Hooks - useState (2)

Com o **useState** você pode criar estados e setar valores iniciais para eles.

Essa função retorna um array com o **estado** e uma **função que altera o estado**, nessa sequência.

Ao invocá-la, você pode definir um valor inicial para aquele estado, a sintaxe é assim:

```
const [estado, setEstado] = React.useState(valorInicial)
```

Hooks - useState (3)

Também podemos desestruturar o `useState` junto ao import do React.

Assim não precisamos chamar nenhum dos métodos vindos de React como propriedades.

```
import React, { useState } from 'react'
```




DONE

TODO
DOING

To Do

TODO: Criação de estado

- ⇒ Crie o componente NovaTarefa.jsx
- ⇒ Dentro dele iremos:
 - Criar um estado para receber o valor do input de título
 - Usar o evento de change para capturar esse valor
 - A cada alteração do valor do input, ele deve ser salvo no estado

Dica: usar o atributo onChange



Axios



Axios

É uma biblioteca para **requisições HTTP** usando node.js.

Ele torna o processo de realizar uma requisição simples e dinâmica ,nos fornecendo algumas funcionalidades já pré-inseridas, como por exemplo a **conversão automática da resposta da API em Json**.

Axios - Configuração

Para **instalar** a biblioteca iremos rodar:

```
npm install axios
```





GET



Axios - GET

```
import axios from "axios";

function App() {
  function getAPI() {
    axios
      .get("LINK")
      .then((resposta) => console.log(resposta))
      .catch((erro) => console.log(erro));
  }
}
```

POST



Axios - POST

```
import axios from "axios";

function app() {
  function postAPI() {
    axios
      .post("LINK", {body})
      .then((resposta) => console.log(resposta))
      .catch((erro) => console.log(erro));
  }
}
```


TODO: GET tarefas

- ⇒ Crie um componente Tarefas.jsx
- ⇒ Dentro dele:
 - Execute um GET ao nosso Json-server (criado anteriormente);
 - Salve as tarefas em um estado;
 - Mostre os títulos das tarefas na tela.

Dica: Para mostrar na tela utilize o **map()**



Atividade em pares

