

Webdev - Módulo 3

Roteiro de aula

Aula 15 - Hard: REACTions



Tópicos da aula:

- Vite;
- Componentes;
- Props;
- Hooks;
- useState;
- useEffect;
- react-router-dom;



Objetivos da aula:

1. Recordar o conceito de Vite
2. Relembrar a sintaxe de componentes;
3. Recordar a utilização de props;
4. Relembrar hooks;
5. Recordar o conceito do useState;
6. Recordar o conceito do useEffect;
7. Recordar o conceito e a sintaxe do react router dom.



Atividades da Aula

→ Mão na Massa: Criar componente de botão

- ◆ Crie um novo projeto usando o vite
 - Dentro da pasta components criar um arquivo Button.jsx
 - Criar um objeto com 10 cores escolhidas pelo grupo
 - O componente Button terá uma prop chamada color
 - A cor de fundo deve ser alterada de acordo com a propriedade escrita em color
- ◆ Insira o componente Button no app.jsx



Momento Aprendizagem em Pares

Esse momento é dedicado para vocês desenvolverem suas demandas e entregas para o curso.

- Utilize esse tempo da maneira que preferir, mas atente-se às aulas que você deve realizar as entregas.
 - ◆ **Dica:** Nas Propostas dos projetos, vocês encontram uma sugestão de organização para a realização das atividades.
 - ◆ **Lembre-se:** O momento de Aprendizagem em Pares é justamente para fazer trocas e aprender em comunidade, aproveite seus colegas e se desenvolvam juntos!
- **Entregas:**
 - ◆ Projeto individual: aula 5 - HARD
 - ◆ Projeto em grupo: aula 10 - HARD
 - ◆ Apresentação do projeto: aula 10 - HARD



Revisão da aula

- **Vite** é a palavra francesa para rápido. É uma ferramenta de construção moderna para desenvolvimento web front-end que utiliza módulos ES, ou ESM. O Vite apresenta um empacotador mais enxuto e rápido que vem com um servidor de desenvolvimento pré-configurado.
- Um **componente** é uma mistura de HTML e JavaScript que captura toda a lógica necessária para exibir uma pequena seção de uma interface do usuário maior. **Cada componente do projeto** pode ser construído em partes sucessivamente complexas de um aplicativo. O React oferece várias extensões para todo o suporte arquitetônico do aplicativo, como Flux e React Native, além da mera interface do usuário.
- **Props** significa simplesmente propriedades. São eles que tornam os componentes reutilizáveis, porque executam uma função essencial – eles passam dados de um componente para outro.
- Os **adereços** atuam como um canal para comunicação de componentes. Os adereços são passados de pai para filho e ajudam seu filho a acessar as propriedades que chegaram à árvore do pai.
- O **Hook useState()** no React é um hook que nos permite declarar variáveis em nossos componentes funcionais.

- React tem dois tipos de componentes. O primeiro são **componentes de classe** que são classes ES6 que se estendem do React, e o segundo são **componentes funcionais**. Class-to-components terá métodos de estado e ciclo de vida: a classe de mensagem estende React. UseState Hook é uma função especial que recebe o estado inicial como argumento e o retorna como uma matriz de entrada.
- O **useEffect** é um hook do React. Ele permite que você execute efeitos colaterais no seu código. Mas o que seriam esses efeitos colaterais? Buscar dados de uma API, mudar a DOM, ou configurar uma subscription, por exemplo, são algumas opções de efeitos colaterais no seu código.



Para ajudar



Links interessantes:

- [Creating a React.js App using Vite | Engineering Education \(EngEd\) Program | Section](#)
- [Getting Started | Vite](#)
- [Features | Vite](#)
- [Componentes e Props – React](#)
- [Introduzindo JSX – React](#)
- [Estado e Ciclo de Vida – React](#)
- [SPA \(Single-page application\) - MDN Web Docs Glossary: Definitions of Web-related terms](#)