Seção 04 - Mão na Massa com React

26 - Meu Primeiro Projeto

Para criar um novo React App:

```
npx create-react-app nome-app
```

Para rodar um novo React App:

```
1° -> cd nome-app
2° -> npm start
```

src/App.js:

```
export default function App() {
    return(
        <div>
            <h1>Bem-vindo ao sistema</h1>
            <h2>@allan_smith_pb tem 726 seguiidoreeessss</h2>
        </div>
    )
}
```

Uma pequena introdução do funcionamento do React, e seu componente principal, o **App.js.**

32 - Entendendo props e Components (Parte 1)

1- Components

Web Components são um conjunto de especificações elaboradas para permitir a criação de elementos web de forma customizada e independente.

2- Props

Quando o React vê um elemento representando um componente definido pelo usuário, ele passa atributos JSX e componentes filhos para esse componente como um único objeto. Nós chamamos esse objeto de “props”.

```
import React from 'react';

const Presentation = (props) => {
       // Criei uma função que retorne as propriedades(props) que estou
interessado
    return(
        <div>
            <h2>Olá! Eu sou {props.nome} e tenho {props.idade}
anos.</h2>
      // Sendo assim, aparece uma saudação com nome e idade que foi
transmitido em App()
        </div>
    )
}

export default function App() {
    // A função App que retornará nosso Aplicativo
    return(
        <div>
            <h1>Olá Mundo</h1>
           // Aqui abaixo, estou chamando o Component Presentation()
            <Presentation nome="Allan" idade="18"/>
           // Aqui onde o Componente Presentation() está recebendo as
props que foram passadas
            <Presentation nome="Anderson" idade="19" />
        </div>
    )
}
```

33 - Entendendo props e Components (Parte 2)

```
// 27/01/2023
import React from 'react';

const Equipe = (props) => {
    Equipe está recebendo as propriedades de App e atribuindo a
variáveis, que serão usadas como propriedades pelo Sobre. O Sobre está
sendo chamado
```

```
    return(
        <div>
            <Sobre username={props.nome} cargo={props.cargo}
idade={props.idade} />
            <Social />
            <hr/>
        </div>
    ) // Social também está sendo chamado, mas devolvendo conteúdo
estático
}

const Sobre = (props) => {
    // O sobre está usando as propriedades que lhes foi atribuidas.
    return(
        <div>
            <h2>Olá! Sou o(a) {props.username}. Sou {props.cargo} e
tenho {props.idade} anos</h2>
        </div>
    )
}

const Social = () => {
    // Social está fazendo um papel apenas de componente estático
    return(
        <div>
            <a href="https://www.google.com/">Facebook </a>
            <a href='https://www.google.com/'>Instagram </a>
            <a href='https://www.google.com/'>LinkedIn </a>
        </div>
    )
}

export default function App() {
    // A função App que retornará nosso Aplicativo, onde estou chamando
o componente Equipe, passando propriedades que serão acessadas por ele
    return(
        <div>
            <h1>Conheça nossa equipe:</h1>
            <Equipe nome="Allan" cargo="programador" idade="18"
redes_sociais="https://www.google.com/"/>
            <Equipe nome="Anderson" cargo="nutricionista" idade="19"
redes_sociais="https://www.google.com/"/>
        </div>
```

```
    )
}
```

34 - Aplicando Class Components

Os componentes são bits de código independentes e reutilizáveis. Eles servem ao mesmo propósito que as funções JavaScript, mas trabalham isoladamente e retornam HTML por meio de uma função render().

Os componentes vêm em dois tipos, componentes de classe e componentes de função. Neste capítulo, você aprenderá sobre os componentes de classe.

```
import React, { Component } from 'react'; // Precisa incluir o
React.Component antes de fazer uma Class Component



class Equipe extends Component {

    // A Class Component "renderiza" com o método render(), que se
torna obrigatório

    render() {

        return(

            <Sobre name={this.props.nome} position={this.props.cargo}
old={this.props.idade}/> // O sobre acessa rapidamente as propriedades
que são atribuídas a ele, sem que tenha explicitamente o parâmetro
"props" no Componente

        )

    }

}



class Sobre extends Component {

    // Também renderiza aqui, acessando rapidamente as proprieades

    render() {
```

```
        return(
            <div>
            <h2>Olá, sou(a) {this.props.name} </h2>
            <h3>Cargo: {this.props.position}</h3>
            <h3>idade: {this.props.old} anos</h3>
            <hr/>
            </div>
        )
    }
}

function App() {
    // Aqui também faço a atribuição de propriedades para Equipe
    return(
        <div>
            <h1>Conheça nossa equipe:</h1>
            <Equipe nome="Allan" cargo="Programador" idade="18" />
            <Equipe nome="Anderson" cargo="Nutricionista" idade="19"/>
        </div>
    )
}

export default App;
```

35 - Trabalhando com States

```
import React, { Component } from 'react';

class App extends Component {
    // Class Component

    constructor(props) {
        // Construtor que possui as propriedades mais importantes
        super(props); // Acessar todas as informações do componente pai
(App)
        this.state = { // Os estados desse Component
            nome: "Allan",
            contador: 0
        };

        // Lembrando que o this serve para acessar cada variável

        this.aumentar = this.aumentar.bind(this) // bind() serve para
que a função consiga ser usada pela aplicação
        this.diminuir = this.diminuir.bind(this)
    }

    aumentar() {
        let state = this.state;
        state.contador += 1
        state.nome = "Anderson"
        this.setState(state);
    }

    diminuir () {
        let stateDiminuir = this.state;
        if(stateDiminuir.contador === 0) {
            alert('Chegamos a zero! Não pode diminuir mais.');
            return;
        }

        stateDiminuir.contador -= 1;
        this.setState(stateDiminuir);
    }
```

```
    render() {
        return (
            <div>
                <h1>Contador</h1>
                <h2>{this.state.nome}</h2>
                <h3>
                    <button onClick={this.diminuir}>-</button>
                    {this.state.contador}
                    <button onClick={this.aumentar}>+</button>
                </h3>
            </div>
        )
    }
}

export default App;
```

Referências:

MEDIUM. Web Components. Disponível em: https://medium.com/@valdeirpsr/web-components-9cdcad1f87b9. Acesso em 23 jan. 2023;

REACT site. Componentes e Props. Disponível em: https://pt-br.reactjs.org/docs/components-and-props.html#:~:text=Quando%20o%20React%20v%C3%AA%20um,esse%20objeto%20de%20%E2%80%9Cprops%E2%80%9D. Acesso em 23 jan. 2023.