

REACT

**Desvendando os Segredos do Mar do
front-end**

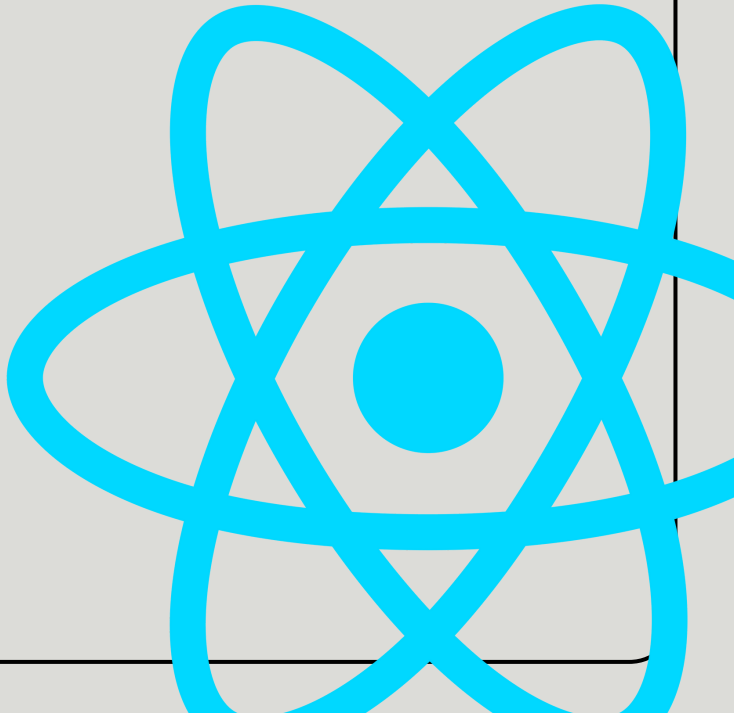
A Grande Saga dos Hooks

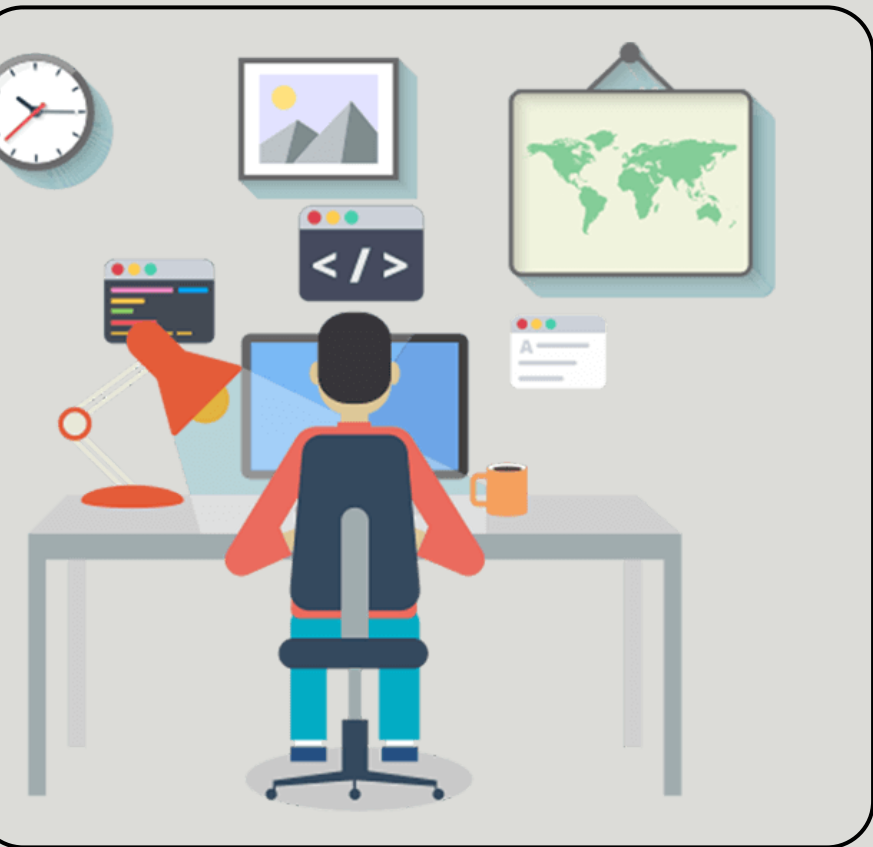


José Anderson da Silva Costa

01 – Capítulo

Introdução ao React





O React revolucionou o desenvolvimento front-end, permitindo a criação de interfaces dinâmicas e responsivas de maneira eficiente. Neste capítulo, vamos dar os primeiros passos com React e entender seus conceitos fundamentais.

1.1: Componentes e JSX

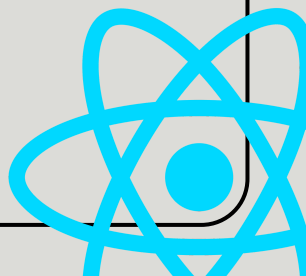
React se baseia na criação de componentes reutilizáveis para construir interfaces. Utilizamos JSX (JavaScript XML) para declarar a estrutura do componente de forma semelhante ao HTML, mas com a potência do JavaScript.

Exemplo de código:

```
import React from 'react';

const App = () => {
  return <div>Hello, React!
</div>;
};

export default App;
```



1.2: Props e State

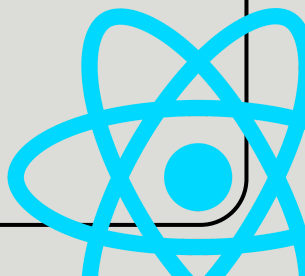
Props (propriedades) e State (estado) são a base da comunicação e manipulação de dados em componentes React. Exemplo de código:

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

```
const Counter = ({ initialCount }) => {  
  const [count, setCount] =  
    useState(initialCount);
```

```
  const increment = () => {  
    setCount(count + 1);  
  };  
  return (  
    <div>  
      <p>Count: {count}</p>  
      <button onClick=  
{increment}>Increment</button>  
    </div>  
  );  
};
```

```
export default Counter;
```



02 - Capítulo

Manipulando Eventos e Renderização Condicional





Neste capítulo, aprenderemos como lidar com eventos do usuário e como renderizar condicionalmente elementos no React.

2.1: Manipulando Eventos

React nos permite criar interatividade através da manipulação de eventos.

Exemplo de código:

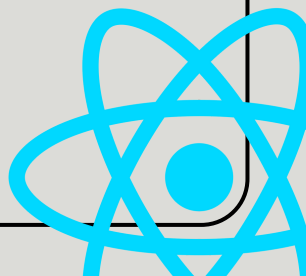
```
import React, { useState } from  
'react';
```

```
const Toggle = () => {  
  const [isOn, setIsOn] =  
  useState(false);
```

```
  const toggle = () => {  
    setIsOn(!isOn);  
  };
```

```
  return (  
    <div>  
      <button onClick={toggle}>  
{isOn ? 'ON' : 'OFF'}</button>  
    </div>  
  );  
};
```

```
export default Toggle;
```



2.2: Renderização Condicional

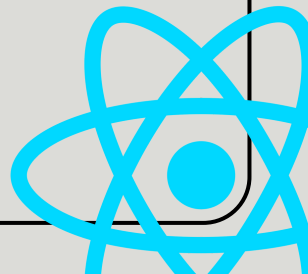
Renderizar componentes condicionalmente é essencial para exibir diferentes conteúdos com base em estados ou props.

Exemplo de código:

```
import React from 'react';

const Greeting = ({ isLoggedIn })
=> {
  return (
    <div>
      {isLoggedIn ? <p>Welcome
back!</p> : <p>Please log in.
</p>}
    </div>
  );
};

export default Greeting;
```



03 – Capítulo

Estilização e Gerenciamento de Estado





Neste capítulo, exploraremos técnicas de estilização e como gerenciar o estado global da aplicação.

3.1: Estilização

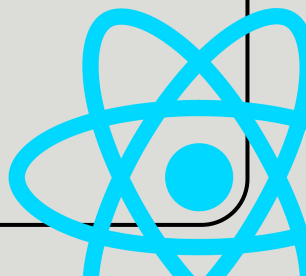
React oferece diversas abordagens para estilização, incluindo CSS regular, módulos CSS e bibliotecas de estilos.

Exemplo de código:

```
import React from 'react';
import './Button.css';

const Button = () => {
  return <button className="my-button">Click me</button>;
};

export default Button;
```



3.2: Gerenciamento de Estado com Redux

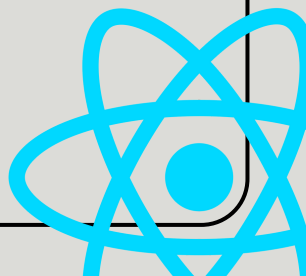
Redux é uma biblioteca popular para gerenciamento de estado global em aplicações React.

Exemplo de código (configuração básica):

```
// Em store.js
import { createStore } from
'redux';
import rootReducer from
'./reducers';

const store =
createStore(rootReducer);

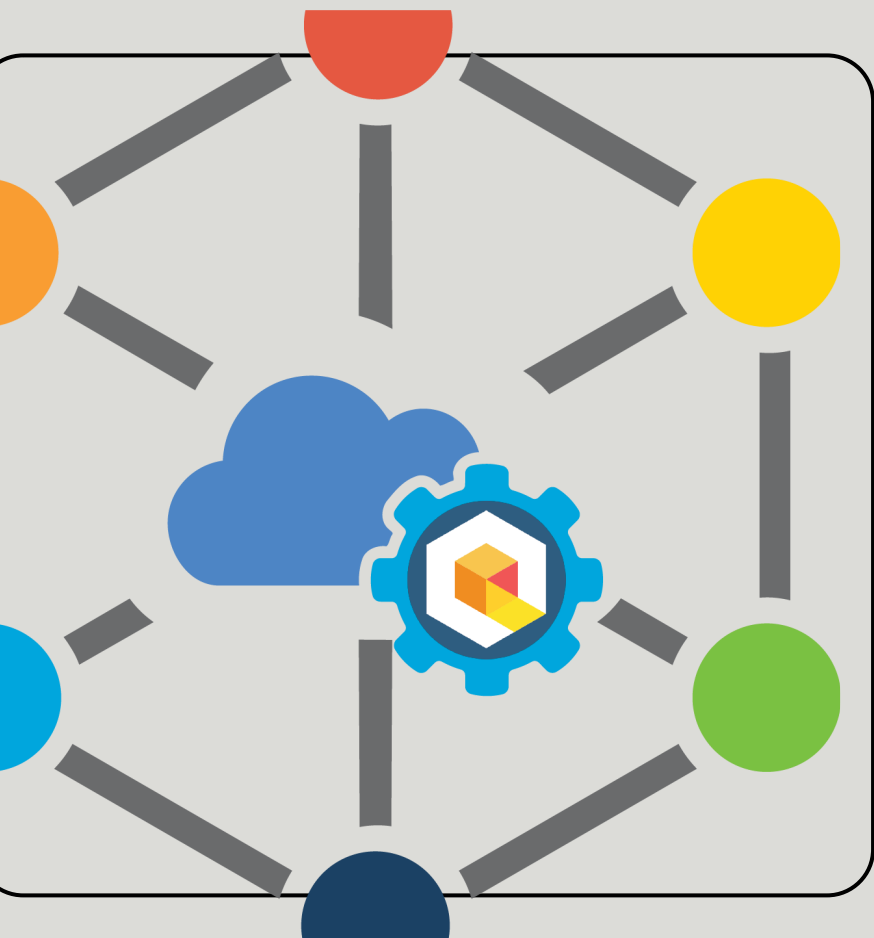
export default store;
```



04 - Capítulo

Roteamento e Integração com APIs





Neste capítulo, vamos aprender sobre roteamento de páginas e integração com APIs externas.

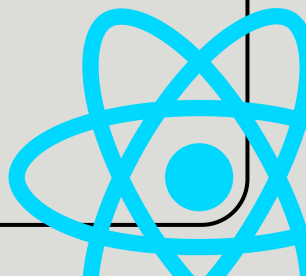
4.1: Roteamento com React Router

React Router permite a navegação entre diferentes páginas em aplicações React. Exemplo de código (configuração básica):

```
import { BrowserRouter, Route,
Switch } from 'react-router-dom';
import Home from './Home';
import About from './About';

const App = () => {
  return (
    <BrowserRouter>
      <Switch>
        <Route path="/" exact
component={Home} />
        <Route path="/about"
component={About} />
      </Switch>
    </BrowserRouter>
  );
};

export default App;
```

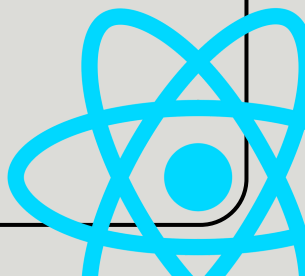


4.2: Integração com APIs

Consumir dados de APIs externas é uma parte crucial de muitas aplicações front-end.

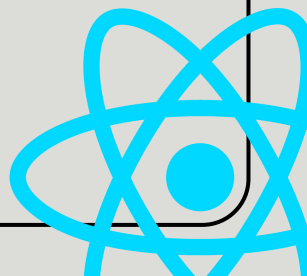
Exemplo de código (usando fetch):

```
import React, { useState,
useEffect } from 'react';
const UserList = () => {
  const [users, setUsers] =
useState([]);
  useEffect(() => {
    fetch('https://jsonplaceholder.t
ypicode.com/users')
      .then(response =>
response.json())
      .then(data =>
setUsers(data));
  }, []);
  return (
    <ul>
      {users.map(user => (
        <li key={user.id}>
{user.name}</li>
      ))}
    </ul>
  );
};
export default UserList;
```



CONCLUSÃO

Dominar as técnicas essenciais de front-end com React é um passo fundamental para criar interfaces modernas e interativas. Neste ebook, exploramos os conceitos fundamentais do React e apresentamos exemplos de código em contextos reais. Esperamos que você tenha adquirido uma base sólida para aprimorar suas habilidades de desenvolvimento front-end com React.



Créditos

Este ebook foi produzido por uma colaboração entre as seguintes tecnologias e recursos:

- Canvas: Plataforma de criação de conteúdo multimídia interativo que possibilitou a formatação e design deste ebook de maneira envolvente e visualmente atraente.
- Midjourne: Ferramenta avançada de geração e organização de textos, que auxiliou na estruturação do conteúdo, na elaboração dos tópicos abordados e na coesão geral do material.
- ChatGPT: Desenvolvido pela OpenAI, o ChatGPT é um modelo de linguagem baseado em IA que proporcionou a redação inicial deste ebook, incluindo os tópicos, exemplos de código e explicações de conceitos.
- Revisão Humana: José Anderson da Silva Costa

Agradecemos a contribuição dessas tecnologias, que trabalharam em conjunto para criar um recurso educacional valioso e acessível sobre as técnicas essenciais de front-end com React. Seu comprometimento com a excelência é o que torna este ebook possível e benéfico para os leitores.

