

Vitor Amadeu Souza

Introdução ao

React

e

CSS

No ambiente Replit

Parte XXI

© 2024 by Cerne Tecnologia e Treinamento Ltda.

© 2024 by Vitor Amadeu Souza

Nenhuma parte desta publicação poderá ser reproduzida sem autorização prévia e escrita de **Cerne Tecnologia e Treinamento Ltda.** Este livro publica nomes comerciais e marcas registradas de produtos pertencentes a diversas companhias. O editor utiliza as marcas somente para fins editoriais e em benefício dos proprietários das marcas, sem nenhuma intenção de atingir seus direitos.

Setembro de 2024

Direitos reservados por:

Cerne Tecnologia e Treinamento Ltda

Produção: Cerne Tecnologia e Treinamento

E-mail da Empresa: cerne@cerne-tec.com.br

Home Page: www.cerne-tec.com.br.com.br

Atendimento ao Consumidor: sac@cerne-tec.com.br

Contato com o Autor: vitor@cerne-tec.com.br



FEITO NO BRASIL

***“Venha também sobre mim a tua benignidade, ó Senhor, e a tua
salvação, segundo a tua palavra.”***

Sl 119:41

Cerne Tecnologia

A Cerne Tecnologia tem uma equipe preparada para desenvolvimento de projetos eletrônicos em diversas áreas: Médica, Entretenimento, Industrial, Robótica, Científica, Automobilística, Aeronáutica, etc. Trabalhamos com tecnologia microcontrolada usando o PIC, ARM, AVR, 8051, dsPIC, PIC24, PIC32 além do Arduino, Raspberry, Beaglebone etc. Desenvolvemos o projeto desde sua concepção até a entrega do produto final, passando pelas etapas de esquema elétrico, protótipo e desenvolvimento de circuito impresso.

Desenvolvemos aplicativos para smartphones/tablets Android, iOS, Blackberry, Windows Phone e no desenvolvimento de softwares a nível PC para plataforma Windows, usando ferramentas como o Visual Basic, C# e C++.

Atuamos na parte de montagem de placas, onde podemos fornecer ambos os serviços de desenvolvimento de projetos e produção ou apenas um destes. Desenvolvemos esquemas elétricos e layout de PCI, tanto em tecnologia convencional como SMD. Temos a flexibilidade de customizar um de nossos produtos, de modo a atender a uma necessidade específica do cliente, tornando o custo de desenvolvimento menor se comparado a construção de um projeto desde a sua fase inicial.

Desenvolvemos e fornecemos kits didáticos para diversos microcontroladores além de apostilas, livros e e-books.

Na hora de desenvolver um projeto ou equipar seu laboratório não hesite em nos contatar. Entre em contato conosco através do endereço cerne-tec.com.br para obter mais informações.



cerne-tec.com.br

Sumário

Capítulo I – Metodologia de desenvolvimento.....	6
1. Introdução.....	6
Capítulo II – Programação em React e CSS.....	7
1. CSS in JS.....	7
2. Componente Card.....	10
3. Componente Button.....	16'

Capítulo I

Metodologia de desenvolvimento

1. Introdução

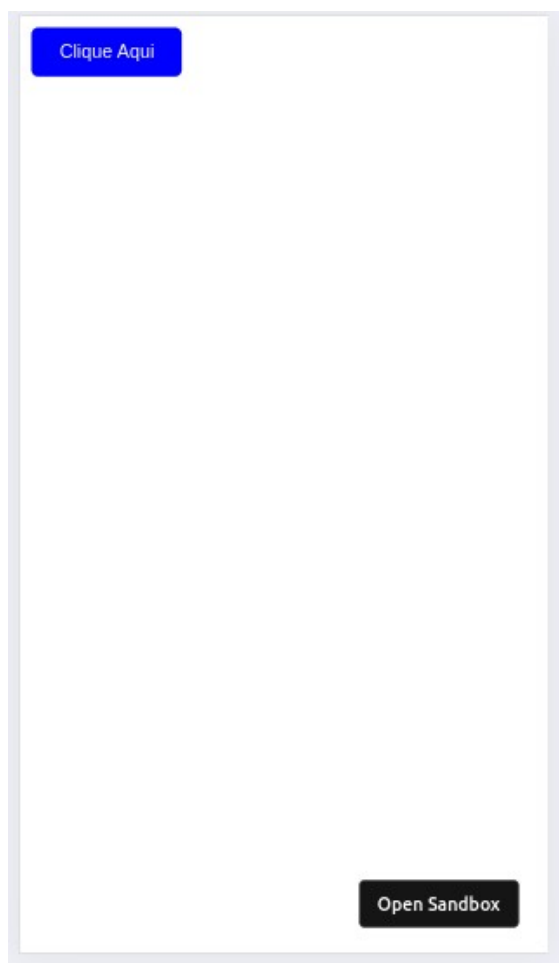
Neste livro, você encontrará uma parte da série sobre o desenvolvimento de projetos utilizando React e CSS. Ao longo desta obra, diversos experimentos serão apresentados segundo o sumário da obra, permitindo o aperfeiçoamento neste framework para a construção de interfaces responsivas juntamente com o CSS.

Capítulo II

Programação em React e CSS

1. CSS in JS

CSS-in-JS é uma técnica que permite escrever estilos CSS diretamente dentro de arquivos JavaScript, integrando os estilos ao código de componentes. Em vez de utilizar arquivos CSS separados, o CSS-in-JS aplica os estilos como parte do próprio componente, o que garante que o escopo dos estilos seja limitado ao componente, evitando conflitos globais. Além disso, essa abordagem permite que os estilos sejam dinâmicos, podendo mudar com base no estado ou nas propriedades do componente. Isso facilita a manutenção, pois o código de estilo e comportamento fica concentrado no mesmo local. Bibliotecas como Styled-Components, Emotion e JSS são amplamente usadas em frameworks como React para implementar CSS-in-JS. O objetivo do próximo exemplo é apresentar como usar o estilo CSS in JS em React.



Acompanhe como ficam os arquivos App.jsx para este exercício.

App.jsx

```
import { css } from "@emotion/react";
import styled from "@emotion/styled";

const buttonStyle = css`
  background-color: blue;
  color: white;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;

  &:hover {
    background-color: darkblue;
  }
`;

const Button = styled.button(buttonStyle);

function App() {
  return (
    <div>
      <Button>Clique Aqui</Button>
    </div>
  );
}
```

Esse código utiliza a biblioteca Emotion para aplicar estilos CSS diretamente dentro de um componente React usando a abordagem

CSS-in-JS. O estilo do botão é definido usando a função `css` importada de `@emotion/react`. A constante `buttonStyle` contém o estilo do botão, especificando a cor de fundo, cor do texto, espaçamento, bordas arredondadas e um efeito de `hover` que altera a cor de fundo para azul escuro quando o botão é sobrevoado com o mouse. O `Button` é um componente criado com `styled.button` a partir do estilo definido em `buttonStyle`. Esse componente encapsula os estilos e os aplica diretamente ao botão. Na função `App`, o componente `Button` é renderizado dentro de uma `div`, exibindo o texto "Clique Aqui". Esse código demonstra como aplicar estilos de forma modular e reutilizável usando `Emotion` no `React`.

2. Componente Card

Este exemplo cria um componente `Card` em `React` para apresentar os `Cards` apresentados a seguir.