

Vitor Amadeu Souza

Introdução ao

React

e

CSS

No ambiente Replit

Parte XX

© 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.



Sumário

Capítulo I – Metodologia de desenvolvimento.....	6
1. Introdução.....	6
Capítulo II – Programação em React e CSS.....	7
1. CSS inline.....	7
2. CSS externo	10
3. Módulo CSS.....	15

Capítulo I

Metodologia de desenvolvimento

1. Introdução

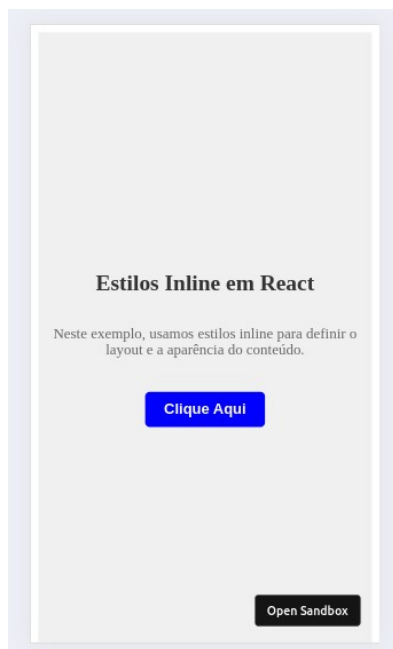
Neste livro, você encontrará uma parte da série sobre o desenvolvimento de projetos utilizando React e CSS. Ao longo desta obra, diversos experimentos serão apresentados segundo o sumário da obra, permitindo o aperfeiçoamento neste framework para a construção de interfaces responsivas juntamente com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

Capítulo II

Programação em React e CSS

1. CSS inline

O objetivo do próximo exemplo é apresentar como usar o estilo CSS inline em React.



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

App.jsx

```
import React from 'react';

function App() {
  // Estilos Inline Aninhados
  const buttonStyle = {
    backgroundColor: 'blue',
    color: 'white',
    padding: '10px 20px',
    border: 'none',
    borderRadius: '5px',
    cursor: 'pointer',
    marginTop: '20px',
    fontSize: '16px',
    fontWeight: 'bold',
  };

  const containerStyle = {
    display: 'flex',
    flexDirection: 'column',
    alignItems: 'center',
    justifyContent: 'center',
    height: '100vh',
    backgroundColor: '#f0f0f0',
    textAlign: 'center',
  };

  return (
    <div style={containerStyle}>
      <h1 style={{ color: '#333', fontSize: '24px' }}>
        Estilos Inline em React</h1>
      <p style={{ color: '#666', maxWidth: '600px' }}>
        Neste exemplo, usamos estilos inline para definir o layout e a
        aparência do conteúdo.
      </p>
    </div>
  );
}
```

```

    </p>
    <button
      style={buttonStyle}
      onClick={() => alert('Você clicou no botão!')}
    >
      Clique Aqui
    </button>
  </div>
);
}

export default App;

```

O código acima utiliza o React para criar um componente funcional chamado `App`, que implementa um layout estilizado utilizando estilos inline. No componente, duas constantes `buttonStyle` e `containerStyle` são definidas para aplicar estilos diretamente aos elementos. A constante `buttonStyle` estiliza um botão com fundo azul, texto branco, padding, borda arredondada, cursor de ponteiro, e margens superiores, além de especificar o tamanho e o peso da fonte. Já a constante `containerStyle` define o estilo do contêiner principal, organizando-o em coluna (`flexDirection: 'column'`), centralizando os itens horizontal e verticalmente, preenchendo a altura total da tela, e aplicando um fundo cinza claro.

Na renderização, o componente App contém um título (h1), um parágrafo (p), e um botão. O título tem uma cor de texto escura e um tamanho de fonte de 24px, enquanto o parágrafo usa uma cor de texto mais clara e uma largura máxima de 600px para limitar o comprimento do texto. Quando o botão é clicado, uma função onClick exibe um alerta com a mensagem "Você clicou no botão!". Essa abordagem usa estilos inline diretamente nos elementos do JSX, sendo uma maneira prática de aplicar estilos sem a necessidade de arquivos CSS externos.

2. CSS externo

Neste exemplo todas as classes CSS estarão incorporadas em um arquivo externo chamado estilos.css