

Vitor Amadeu Souza

Introdução ao

React

e

CSS

No ambiente Replit

Parte XXVI

© 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. Componente ThemeToggle	7
2. Componente Form.....	13
3. Componente ToolTip	26

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.

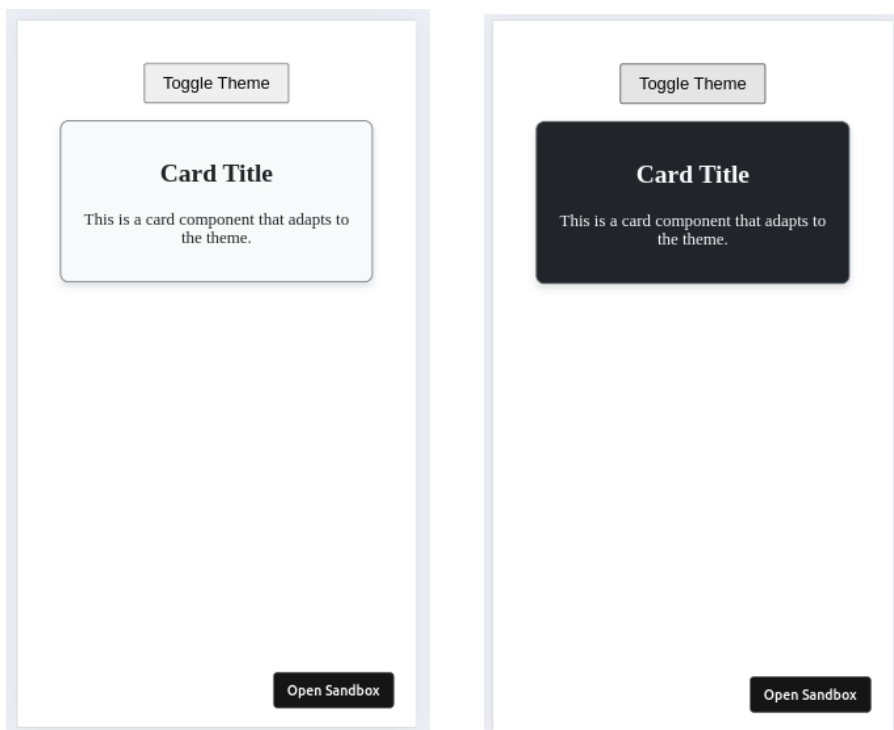
Capítulo II

Programação em React e CSS

1. Componente ThemeToggle

O componente ThemeToggle é responsável por permitir a alternância entre temas de uma aplicação, como o modo claro e escuro. Ele utiliza o estado interno do React para armazenar o tema atual e, ao ser acionado pelo usuário, através de um botão ou ícone, alterna entre os temas. O estado pode ser representado por valores como "light" e "dark", indicando o tema claro ou escuro. Quando o tema é alternado, o componente altera dinamicamente as classes CSS aplicadas à interface, mudando as cores de fundo, do texto e outros elementos visuais, de acordo com o tema escolhido. Essa mudança de classes pode ser aplicada ao corpo do documento ou à raiz da aplicação, garantindo que toda a interface adote o novo esquema de cores de forma consistente. A transição entre os temas é suave, proporcionando uma melhor experiência para o usuário, que pode escolher o estilo visual que mais lhe agrada.

Assim, o ThemeToggle oferece uma funcionalidade essencial para personalizar a aparência da aplicação, adaptando-a às preferências de cada usuário.



Acompanhe como ficam os arquivos para este exercício.

styles.css

```
.App {  
  font-family: sans-serif;  
  text-align: center;  
}
```

App.jsx

```
import React from "react";  
import ThemeToggle from "./ThemeToggle";  
  
const App = () => {  
  return (  
    <div>  
      <ThemeToggle />  
    </div>  
  );  
};  
  
export default App;
```

Card.jsx

```
/** @jsxImportSource @emotion/react */  
import { css } from "@emotion/react";  
import React from "react";  
  
const cardStyle = (theme) => css`  
  background-color: ${theme.background};
```

```

color: ${theme.text};
border: 1px solid ${theme.secondary};
border-radius: 8px;
padding: 16px;
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
transition: background-color 0.3s ease, color 0.3s ease;
`;

```

```

const Card = ({ theme, children }) => {
  return <div css={cardStyle(theme)}>{children}</div>;
};

```

```

export default Card;

```

ThemesToggle.jsx

```

import React, { useState } from "react";
import { ThemeProvider } from "@emotion/react";
import { lightTheme, darkTheme } from "../themes";
import Card from "../Card";

```

```

const ThemeToggle = () => {
  const [isDarkTheme, setIsDarkTheme] = useState(false);
  const currentTheme = isDarkTheme ? darkTheme : lightTheme;

```

```

  return (
    <ThemeProvider theme={currentTheme}>
      <div style={{ padding: "2rem", textAlign: "center" }}>
        <button
          onClick={() => setIsDarkTheme(!isDarkTheme)}
          style={{

```