

Vitor Amadeu Souza

Introdução ao

React

e

CSS

No ambiente Replit

Parte XXII

© 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 PCB, 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 ToggleSwitch.....	7
2. Componente DynamicBox	12
3. Componente HoverButton.....	17

Capítulo I

Metodologia de desenvolvimento

1. Introdução

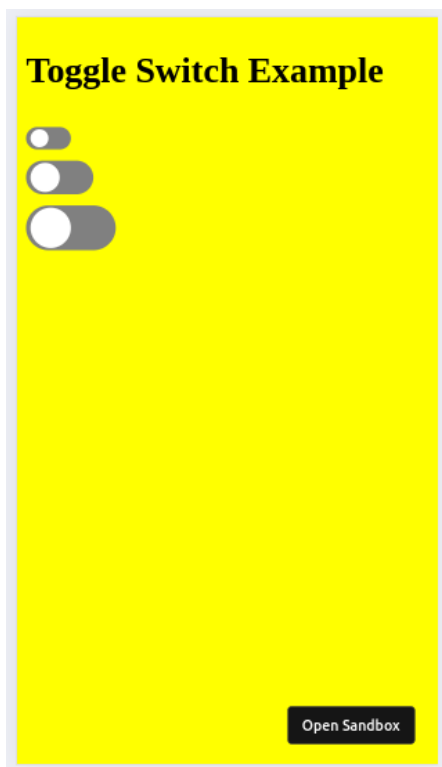
Neste livro, você encontrará uma parte da série sobre o desenvolvimento de projetos utilizando React e CSS. Ao longo desta obra, diversos experimentos serão apresentados segundo o sumário da obra, permitindo o aperfeiçoamento neste framework para a construção de interfaces responsivas juntamente com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

Capítulo II

Programação em React e CSS

1. Componente ToggleSwitch

A proposta deste exemplo é construir um componente ToggleSwitch, que permita alternar entre os estados on-off.



Acompanhe como ficam os arquivos para este exercício.

App.jsx

```
import React, { useState } from "react";
import ToggleSwitch from "./ToggleSwitch";
import "./styles.css";

function App() {
  const [isOn, setIsOn] = useState(false);

  const handleToggle = () =>
  {
    setIsOn(!isOn);
  };

  return (
    <div className="container">
      <h1>Toggle Switch Example</h1>
      <ToggleSwitch isOn={isOn} onToggle={handleToggle} size="small" />
      <ToggleSwitch isOn={isOn} onToggle={handleToggle} size="medium" />
      <ToggleSwitch isOn={isOn} onToggle={handleToggle} size="large" />
    </div>
  );
}

export default App;
```

styles.css

```
body {  
  background-color: yellow;  
}
```

```
.container  
{  
  display: flex;  
  flex-direction: column;  
  gap: 10px;  
}
```

ToggleSwitch.jsx

```
import React from "react";  
import styled from "styled-components";  
  
// Define os estilos baseados em props  
const SwitchWrapper = styled.label`  
  display: inline-flex;  
  align-items: center;  
  cursor: pointer;  
  user-select: none;  
  position: relative;  
  width: ${({ size }) =>  
    size === "small" ? "40px" : size === "medium" ? "60px" : "80px"};  
  height: ${({ size }) =>  
    size === "small" ? "20px" : size === "medium" ? "30px" : "40px"};  
`;  
;
```

```
const HiddenCheckbox = styled.input.attrs({ type: "checkbox" })`
  position: absolute;
  opacity: 0;
  cursor: pointer;
  width: 0;
  height: 0;
`;
```

```
const Slider = styled.span`
  position: absolute;
  cursor: pointer;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  background-color: ${({ isOn }) => (isOn ? "green" : "gray")};
  border-radius: ${({ size }) =>
    size === "small" ? "10px" : size === "medium" ? "15px" : "20px"};
  transition: background-color 0.2s;
  transition: transform 0.2s;
```

```
&:before {
  content: "";
  position: absolute;
  height: ${({ size }) =>
    size === "small" ? "16px" : size === "medium" ? "26px" : "36px"};
  width: ${({ size }) =>
    size === "small" ? "16px" : size === "medium" ? "26px" : "36px"};
  border-radius: 50%;

  left: ${({ isOn }) => (isOn ? "calc(100% - 20px)" : "4px")};
```