

Vitor Amadeu Souza

Introdução ao

React

e

CSS

No ambiente Replit

Parte VII

© 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.

Agosto 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. Calculadora com React	7
2. Calculadora com React II	13

Capítulo I

Metodologia de desenvolvimento

1. Introdução

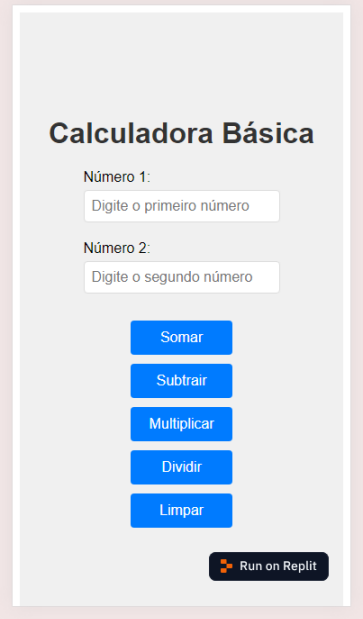
Neste livro, você encontrará uma parte da série sobre o desenvolvimento de projetos utilizando React e CSS. Ao longo desta obra, diversos experimentos serão apresentados segundo o sumário da obra, permitindo o aperfeiçoamento neste framework para a construção de interfaces responsivas juntamente com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

Capítulo II

Programação em React e CSS

1. Calculadora com React

O objetivo do próximo exemplo é elaborar uma calculadora fazendo uso do React, onde para este caso teremos vários `useState()` para verificar o valor informado pelo usuário e realizar uma das 4 operações aritméticas, como apresentado na figura abaixo.



The image shows a web-based calculator interface. At the top, it is titled "Calculadora Básica". Below the title, there are two input fields for numbers. The first is labeled "Número 1:" and contains the placeholder text "Digite o primeiro número". The second is labeled "Número 2:" and contains the placeholder text "Digite o segundo número". Below these inputs are five blue buttons stacked vertically: "Somar", "Subtrair", "Multiplicar", "Dividir", and "Limpar". At the bottom right of the interface, there is a dark button with a Replit logo and the text "Run on Replit".

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

App.css

```
.container {
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  height: 100vh;
  background-color: #f0f0f0;
  font-family: Arial, sans-serif;
}

h1 {
  color: #333;
}

.input-group {
  margin-bottom: 20px;
}

label {
  display: block;
  margin-bottom: 5px;
}

input {
  padding: 8px;
  font-size: 16px;
  border: 1px solid #ddd;
  border-radius: 4px;
}
```

```
    width: 200px;
  }

.button-group {
  display: flex;
  gap: 10px;
  margin-top: 10px;
  flex-direction: column;
}

button {
  padding: 10px 20px;
  font-size: 16px;
  color: white;
  background-color: #007bff;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  transition: background-color 0.3s;
}

button:hover {
  background-color: #0056b3;
}

.result {
  margin-top: 20px;
}
```

App.jsx

```
import React, { useState } from 'react';
import './App.css';
```

```

export default function App() {
  const [num1, setNum1] = useState('');
  const [num2, setNum2] = useState('');
  const [result, setResult] = useState(null);

  const calculate = (operation) => {
    const number1 = parseFloat(num1);
    const number2 = parseFloat(num2);
    let res;

    if(Number.isNaN(number1) || Number.isNaN(number2))
      res = 'Operação inválida';
    else
    {
      switch (operation) {
        case 'sum' :
          res = number1 + number2;
          break;
        case 'subtract':
          res = number1 - number2;
          break;
        case 'multiply':
          res = number1 * number2;
          break;
        case 'divide':
          res=number2!==0?number1/number2:'Erro: Divisão por zero';
      }
    }

    setResult(res);
  };

  const clearFields = () => {
    setNum1('');
    setNum2('');
    setResult(null);
  };

  return (

```