Vitor Amadeu Souza

Introdução ao

React

e

CSS

No ambiente Replit

Parte III

© 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.

Cerne
Conhecimento para o Desenvolvimento

cerne-tec.com.br

Sumário

Capítulo I – Metodologia de desenvolvimento.....6

1. Introdução.....6

Capítulo II – Programação em React e CSS.....7

1. Adicionando um botão.....7

2. Botão com link.....11

3. flex-direction.....15

4. Disposição de elementos.....16

5. flex-grow.....20

Capítulo I
Metodologia de desenvolvimento

1. Introdução

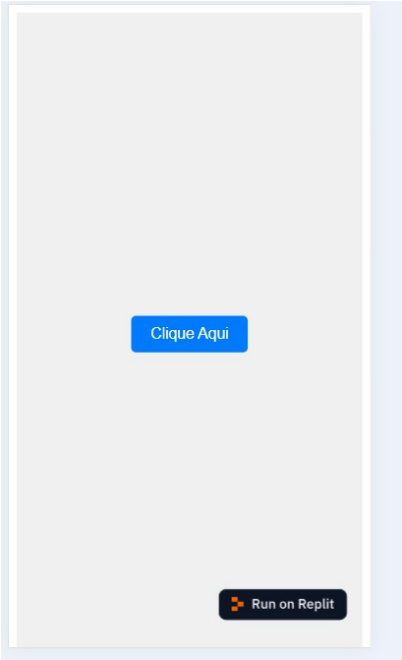
Neste livro, você encontrará uma parte da série sobre o desenvolvimento de projetos utilizando React e CSS. Ao longo desta obra, diversos experimentos serão apresentados segundo o sumário da obra, permitindo o aperfeiçoamento neste framework para a construção de interfaces responsivas juntamente com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

Capítulo II

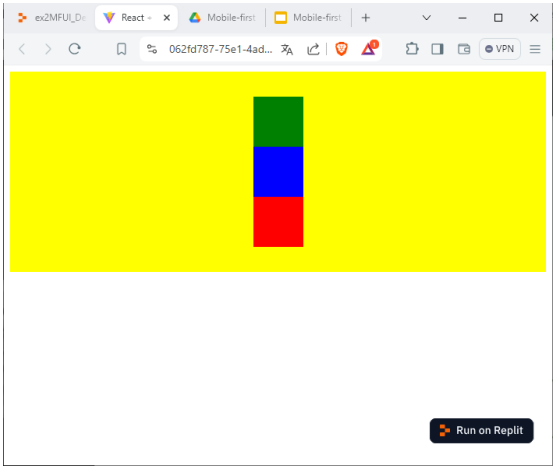
Programação em React e CSS

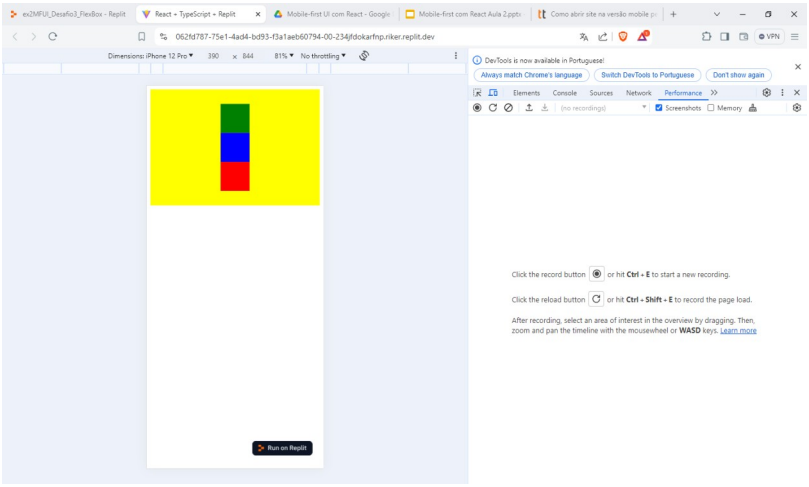
1. Adicionando um botão

O objetivo do próximo exemplo é definir um botão centralizado, como o apresentado na próxima figura.



Para observar a tela em diversos formatos de aparelhos, pressione F12 na tela de apresentação do browser. Veja que ao fazer isso, temos disponível diversas opções de dispositivos no qual pode ser verificado se o design de fato está responsivo.





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

App.css

```
.flex.flex-container {
  display: flex;
  justify-content: center; /* Centraliza horizontalmente */
  align-items: center;    /* Centraliza verticalmente */
  height: 100vh;          /* Altura da viewport */
  background-color: #f0f0f0;
}

.flex-button {
  padding: 10px 20px;
  font-size: 16px;
  background-color: #007bff;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  transition: background-color 0.3s;
}

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

App.jsx

```
import './App.css'

export default function App() {
```

```
  return (
    <div className="flex-container">
      <button className="flex-button">Clique Aqui</button>
    </div>
  )
}
```

Os códigos apresentados consistem em dois arquivos: App.jsx, que é um componente React, e App.css, que define os estilos CSS para os elementos da aplicação. No arquivo App.jsx, o componente App retorna uma estrutura HTML, composta por um com a classe flex-container. Dentro deste , há um botão (<button>) com a classe flex-button e o texto "Clique Aqui".

O arquivo App.css define os estilos para esses elementos. A classe flex-container transforma o em um contêiner flexível (display: flex;), centralizando seu conteúdo tanto horizontal quanto verticalmente, utilizando as propriedades justify-content: center; e align-items: center;. A altura do contêiner é definida como 100% da altura da viewport (`height: 100vh;`), e o fundo do contêiner é um cinza claro (background-color: #f0f0f0;).

A classe flex-button estiliza o botão dentro do contêiner. O botão recebe um preenchimento interno (padding: 10px 20px;), um tamanho de fonte de 16 pixels, um fundo azul (background-color: