

Vitor Amadeu Souza

Introdução ao

React

e

CSS

No ambiente Replit

Parte VI

© 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. fr	7
2. Grid layout	7
3. Trabalhando com useState	11
4. Saudação personalizada	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

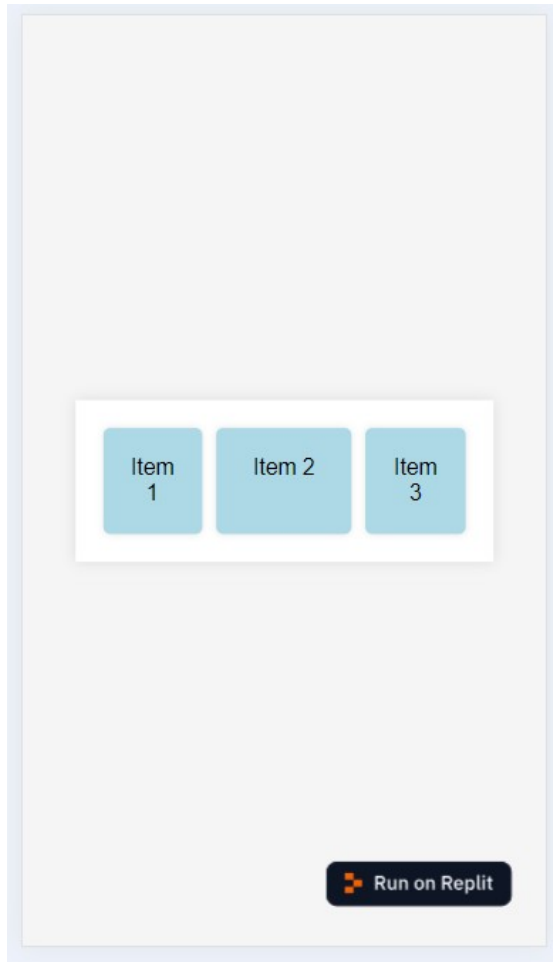
Programação em React e CSS

1. fr

A unidade fr (abreviação de "fraction") no CSS Grid Layout representa uma fração do espaço disponível no contêiner do grid. Em outras palavras, fr é uma unidade de medida flexível que permite dividir o espaço disponível entre os itens do grid de forma proporcional. Quando você usa a unidade fr, está basicamente dizendo ao navegador para dividir o espaço disponível entre os itens do grid com base no número de frações que você especificou. Por exemplo, se você tiver três colunas definidas como 1fr 1fr 1fr, o espaço disponível será dividido igualmente em três partes.

2. Grid layout

O objetivo do próximo exemplo é elaborar uma tela responsiva que atenda a demanda do exemplo, como apresentado na figura abaixo.



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

App.css

```
body, html, #root {
  height: 100%;
  margin: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  background-color: #f5f5f5;
  font-family: Arial, sans-serif;
}

.grid-container {
  display: grid;
  grid-template-columns: 1fr 2fr 1fr;
  gap: 10px;
  width: 80%;
  background-color: white;
  padding: 20px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  box-sizing: border-box;
}

.item {
  background-color: lightblue;
  padding: 20px;
  text-align: center;
  border-radius: 5px;
  box-shadow: 0 0 5px rgba(0, 0, 0, 0.1);
}
```

```
}
```

App.jsx

```
import './App.css'  
  
export default function App() {  
  return (  
    <div className="grid-container">  
      <div className="item">Item 1</div>  
      <div className="item">Item 2</div>  
      <div className="item">Item 3</div>  
    </div>  
  );  
}
```

O código cria um layout centralizado usando CSS Grid para organizar três itens em colunas de tamanhos diferentes. O CSS define que o corpo da página (body), o HTML e o elemento com id #root ocupem toda a altura da tela, removendo margens padrão. O conteúdo é centralizado tanto vertical quanto horizontalmente utilizando Flexbox, com um fundo cinza claro e a fonte Arial aplicada ao texto.

A classe .grid-container é configurada como um contêiner de grade (grid) com três colunas, onde a primeira e a terceira colunas ocupam 1 fração do espaço disponível, enquanto a coluna do meio