Vitor Amadeu Souza

Introdução ao

React

e

CSS

No ambiente Replit

Parte XII

© 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. Tela responsiva I.....7

2. Tela responsiva I.....11

3. Tela responsiva III.....18

Capítulo I
Metodologia de desenvolvimento

1. Introdução

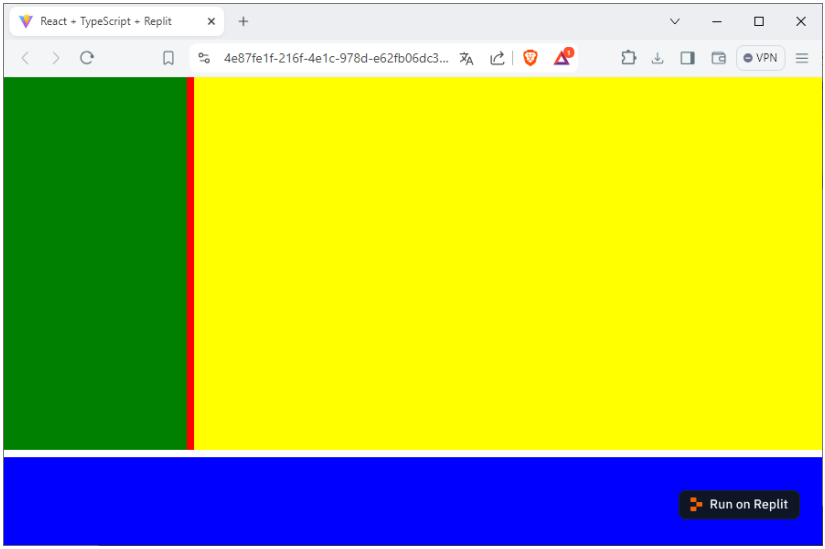
Neste livro, você encontrará uma parte da série sobre o desenvolvimento de projetos utilizando React e CSS. Ao longo desta obra, diversos experimentos serão apresentados segundo o sumário da obra, permitindo o aperfeiçoamento neste framework para a construção de interfaces responsivas juntamente com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

Capítulo II

Programação em React e CSS

1. Tela responsiva I

O objetivo do próximo exemplo é elaborar a página a seguir, que fornecerá suporte para o desenvolvimento do projeto até o término deste capítulo.



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

App.css

```
:root{
  --playlist-gap: 8px;
}

html, body, #root{
  height: 100%;
}

body{
  margin:0;
  padding: var(--playlis-gap);
  box-sizing: border-box;
}

.playlist{
  display: flex;
  flex-direction: column;
  height: 100%;
  gap: var(--playlist-gap);
  box-sizing: border-box;  /*tira o scrool*/
}

.playlist-view{
  background-color: red;
  flex-grow:1;
  display: flex;
  flex-direction: row;
  gap: var(--playlist-gap);
}

.playlist-view-sidebar{
  background-color: green;
```

```
  width: 200px;
  flex-shrink: 0;
}

.playlist-view-main{
  background-color: yellow;
  flex-grow: 1;
  flex-basis: 280px;
  flex-shrink: 0;
}

.playlist-bar{
  background-color: blue;
  height: 96px;
}
```

App.jsx

```
import './App.css'

export default function App() {
  return (
    <div className="playlist">
      <div className ="playlist-view">
        <div className="playlist-view-sidebar"></div>
        <div className="playlist-view-main"></div>
      </div>

      <div className ="playlist-bar"></div>
    </div>
  )
}
```

O código fornecido consiste em dois arquivos, App.css e App.jsx, que trabalham juntos para criar uma interface de usuário usando React e CSS. No arquivo App.css, são definidos os estilos que controlam a aparência e o layout dos elementos na página. O seletor :root define uma variável --playlist-gap com valor de 8 pixels, que será usada para espaçar elementos ao longo da interface. O estilo html, body, #root garante que o HTML, o corpo do documento e o elemento raiz (#root) ocupem 100% da altura da tela, permitindo que a interface preencha toda a janela. O estilo aplicado ao body remove margens, define um padding utilizando a variável --playlist-gap, e usa box-sizing: border-box, o que faz com que padding e bordas sejam incluídos nas dimensões totais dos elementos.

O seletor .playlist aplica esses estilos à div com a classe playlist, definindo-a como um contêiner flexível disposto em coluna, ocupando toda a altura disponível e espaçando seus elementos internos com o valor da variável –playlist-gap. Dentro da classe .playlist-view, que é um contêiner flexível disposto em linha, são definidos dois elementos principais: a barra lateral (.playlist-view-sidebar), que tem largura fixa de 200 pixels e um fundo verde, e o conteúdo principal (.playlist-view-main), que ocupa o espaço restante