Vitor Amadeu Souza

Introdução ao

React

e

CSS

No ambiente Replit

Parte XV

© 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. Exemplo de imagem e texto responsivo 7
2. Layout responsivo com flexbox 10
3. Tipografia fluida 13
4. Imagens responsivas 16
5. Tela responsiva 18

Capítulo I
Metodologia de desenvolvimento

1. Introdução

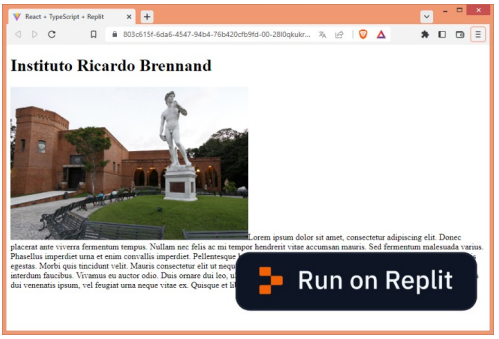
Neste livro, você encontrará uma parte da série sobre o desenvolvimento de projetos utilizando React e CSS. Ao longo desta obra, diversos experimentos serão apresentados segundo o sumário da obra, permitindo o aperfeiçoamento neste framework para a construção de interfaces responsivas juntamente com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

Capítulo II

Programação em React e CSS

1. Exemplo de imagem e texto responsivo

O objetivo deste exemplo é apresentar uma imagem e texto em diversos formatos de tela, como mostra as figuras abaixo.



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

App.css

```
h1 {
  font-size: 2rem;
}

@media screen and (max-width: 785px) {
  h1 {
    font-size: 1rem;
  }
}

/* Estilos padrão para imagens */
img {
  width: 100%; /* Imagem ocupa 100% da largura do contêiner */
  height: auto; /* Mantém a proporção da imagem */
}

/* Para telas maiores que 768px */
@media (min-width: 768px) {
```

```
  img {
    width: 50%; /* Imagem ocupa 50% da largura do contêiner */
  }
}

/* Para telas maiores que 1200px */
@media (min-width: 1200px) {
  img {
    width: 100%; /* Imagem ocupa 100% da largura do contêiner */
  }
}
```

App.jsx

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

export default function App() {

  return (

    <div>
      <h1>Instituto Ricardo Brennand</h1>
      <img  src="/irb.jpg" />
      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec
placerat ante viverra fermentum tempus. Nullam nec felis ac mi tempor
hendrerit vitae accumsan mauris. Sed fermentum malesuada varius. Phasellus
imperdiet urna et enim convallis imperdiet. Pellentesque habitant morbi
tristique senectus et netus et malesuada fames ac turpis egestas. Morbi
quis tincidunt velit. Mauris consectetur elit ut neque dictum, at
tincidunt velit rutrum. Nullam at neque sit amet justo interdum faucibus.
Vivamus eu auctor odio. Duis ornare dui leo, ultricies dictum enim
tincidunt ut. In sodales, ex et efficitur efficitur, eros dui venenatis
ipsum, vel feugiat urna neque vitae ex. Quisque et libero quam.
    </div>

  );
}
```

O código apresentado define um componente React chamado App, que utiliza um arquivo CSS para estilizar o conteúdo de acordo com o tamanho da tela. No CSS, o título (h1) é estilizado para ter um tamanho de fonte de 2rem em telas maiores. Para telas com largura máxima de

785px, o tamanho da fonte é reduzido para 1rem, adaptando-se melhor a dispositivos menores. Quanto às imagens, inicialmente elas ocupam 100% da largura do contêiner, mantendo a proporção original. No entanto, em telas com largura mínima de 768px, as imagens passam a ocupar 50% da largura do contêiner. Para telas maiores que 1200px, a largura das imagens volta a ser 100%. O arquivo App.jsx utiliza o CSS para renderizar uma página com um título "Instituto Ricardo Brennand" e uma imagem (/irb.jpg), acompanhados de um texto exemplo. O uso das media queries no CSS garante que tanto o título quanto a imagem sejam responsivos, ajustando-se adequadamente a diferentes tamanhos de tela.

2. Layout responsivo com flexbox

Alguns layouts responsivos podem ser alcançado sem usar consultas de mídia (media query). Por exemplo, podemos usar a propriedade flex-wrap em um contêiner flexível para envolver automaticamente os elementos na próxima linha se a janela de visualização for muito estreita, a fim de evitar a necessidade de rolagem horizontal, como sugere as figuras abaixo.