

Vitor Amadeu Souza

Introdução ao

React

e

CSS

No ambiente Replit

Parte XXVIII

© 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.

Setembro 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-tec.com.br

Sumário

Capítulo I – Metodologia de desenvolvimento.....	6
1. Introdução.....	6
Capítulo II – Programação em React e CSS.....	7
1. Plotando uma função cúbica com o componente Line.....	7
2. Plotando uma Gaussiana com o componente Line.....	12
3. Plotando uma função sigmoide com o componente Line.....	16

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. Plotando uma função cúbica com o componente Line

Neste exemplo a curva plotada é de terceiro grau, como mostra a próxima tela.



Acompanhe como ficam os arquivos para este exercício.

App.jsx

```
import React from "react";
import Cubic from "./Cubic";

function App() {
  return (
    <div>
      <h1>Gráfico</h1>
      <Cubic a={1} b={0} c={0} d={0} />
    </div>
  );
}

export default App;
```

Cubic.jsx

```
import React from "react";
import { Line } from "react-chartjs-2";
import {
  Chart,
  CategoryScale,
  LinearScale,
  PointElement,
  LineElement,
  Title,
  Tooltip,
  Legend,
} from "chart.js";

// Registrar os componentes do Chart.js
Chart.register(
  CategoryScale,
  LinearScale,
  PointElement,
  LineElement,
  Title,
  Tooltip,
  Legend
);

const Cubic = ({ a, b, c, d }) => {
  const xValues = [];
  for (let i = -50; i <= 50; i++) {
    xValues.push(i);
  }
}
```

```

}
const yValues = xValues.map(
  (x) => a * Math.pow(x, 3) + b * Math.pow(x, 2) + c * x + d
);

// Configuração do gráfico
const data = {
  labels: xValues,
  datasets: [
    {
      label: `y = ${a}x² + ${b}x + ${c}`,
      data: yValues,
      borderColor: "rgba(75, 192, 192, 1)",
      backgroundColor: "rgba(75, 192, 192, 0.2)",
      borderWidth: 2,
      fill: true,
    },
  ],
};

const options = {
  responsive: true,
  plugins: {
    legend: {
      position: "top",
    },
    title: {
      display: true,
      text: "Gráfico de Função Quadrada",
    },
  },
  scales: {
    x: {
      title: {
        display: true,
        text: "x",
      },
    },
    y: {
      title: {
        display: true,
        text: "y",
      },
    },
  },
};

return <Line data={data} options={options} />;
};

export default Cubic;

```

O código apresentado consiste em dois componentes React: App.jsx e Cubic.jsx, que trabalham em conjunto para exibir um gráfico de uma função cúbica utilizando react-chartjs-2 e chart.js. O componente App.jsx é responsável por renderizar a estrutura principal da aplicação, onde o título "Gráfico" é exibido e o componente Cubic é chamado com quatro propriedades (a, b, c, d) definidas como 1, 0, 0 e 0, respectivamente. Esses valores representam os coeficientes da função cúbica no formato $y = ax^3 + bx^2 + cx + d$, que, no exemplo dado, resulta na função $y = x^3$.

Já o componente Cubic.jsx é o responsável pela geração do gráfico. Ele importa e registra os módulos necessários do chart.js, como CategoryScale, LinearScale, PointElement e outros. Em seguida, gera um conjunto de pontos x variando de -50 a 50, armazenando-os em um array chamado xValues. Para cada valor de x, é calculado o valor correspondente de y utilizando a fórmula $y = ax^3 + bx^2 + cx + d$, que resulta em um array yValues.

Com esses pontos x e y, a estrutura de dados do gráfico é definida na constante data, que especifica os rótulos para o eixo x e o conjunto de dados contendo os valores de y, com propriedades de estilo como cor da linha e preenchimento. O rótulo do conjunto de dados foi ajustado para $y = ax^3 + bx^2 + cx + d$, refletindo corretamente