

Vitor Amadeu Souza

Introdução ao

React

e

CSS

No ambiente Replit

Parte XXX

© 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-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 racional com o componente Line.....	7
2. Plotando uma função degrau com o componente Line.....	11
3. Plotando uma função logaritmo com o componente Line.....	16
4. Plotando uma função exponencial com o componente Line.....	20

Capítulo I

Metodologia de desenvolvimento

1. Introdução

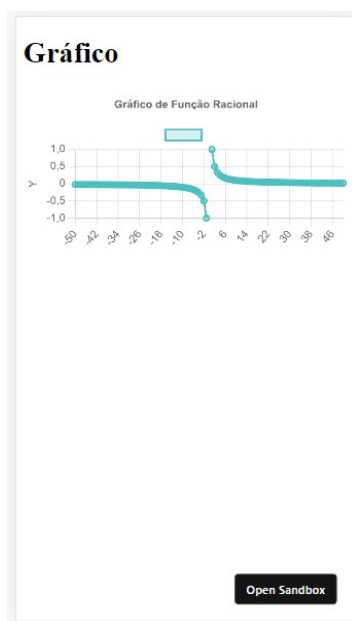
Neste livro, você encontrará uma parte da série sobre o desenvolvimento de projetos utilizando React e CSS. Ao longo desta obra, diversos experimentos serão apresentados segundo o sumário da obra, permitindo o aperfeiçoamento neste framework para a construção de interfaces responsivas juntamente com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

Capítulo II

Programação em React e CSS

1. Plotando uma função racional com o componente Line

Uma função racional é uma função que pode ser expressa como o quociente de dois polinômios, ou seja, uma razão entre duas expressões polinomiais. A forma geral de uma função racional é dada por $f(x)=1/x$. Note que neste caso, para $x=0$ a função é descontínua. Acompanhe o resultado na próxima tela.



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

App.jsx

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

function App() {
  return (
    <div>
      <h1>Gráfico</h1>
      <Rational />
    </div>
  );
}

export default App;
```

Rational.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 Rational = () => {
  const xValues = [];

  // Gerar valores de x de 0 a 50
  for (let i = -50; i <= 50; i++) {
    xValues.push(i);
  }

  yValues = xValues.map((x) => {
    if (x !== 0) {
      return 1 / x;
    }
  });

  // Configuração do gráfico
  const data = {
    labels: xValues,
    datasets: [
      {
        label: '',
        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 Racional",
      },
    },
    scales: {
      x: {
        title: {
          display: false,
          text: "x",
        },
      },
      y: {
        title: {
          display: true,
          text: "y",
        },
      },
    },
  };

```

```

    },
  },
},
};

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

export default Rational;

```

O código utiliza a biblioteca React para exibir um gráfico de uma função racional, composto por dois componentes principais: App e Rational. O componente App é o ponto de entrada do aplicativo e exibe o título "Gráfico" seguido do componente Rational, que renderiza o gráfico. No componente Rational, são usados react-chartjs-2 e chart.js para configurar e desenhar o gráfico. Primeiro, é criado um array xValues que armazena valores de x variando de -50 a 50, e depois são calculados os valores correspondentes de y utilizando a função $y = 1/x$, exceto quando $x = 0$, para evitar a divisão por zero. Os dados do gráfico são definidos em um objeto data, que inclui os valores de x e y, bem como o estilo da linha, como cor e preenchimento. As opções de configuração estão no objeto options, que determina a responsividade, os títulos e a posição da legenda. Para renderizar, é utilizado o componente Line do react-chartjs-2, com as configurações de data e options. No entanto, há um erro no código porque a variável yValues não foi declarada corretamente; para corrigir, é necessário adicionar `let yValues = []` antes de calcular