

Projeto: Calculadora de Gorjeta

Este guia apresenta o passo a passo para criar um projeto de **Calculadora de Gorjeta** utilizando React. O projeto é organizado em componentes reutilizáveis, CSS Modules e uma página principal para gerenciar o layout.

Estrutura do Projeto

Organize os arquivos do projeto da seguinte forma:

```
src/  
├── components/  
│   ├── InputField/  
│   │   ├── InputField.js  
│   │   └── InputField.module.css  
│   ├── ResultDisplay/  
│   │   ├── ResultDisplay.js  
│   │   └── ResultDisplay.module.css  
│   └── Button/  
│       ├── Button.js  
│       └── Button.module.css  
├── pages/  
│   └── TipCalculatorPage.js  
├── App.js  
├── App.module.css  
└── index.js
```

Passo 1: Criar o arquivo **index.js**

O arquivo **index.js** é o ponto de entrada da aplicação.

```
import React from "react";  
import ReactDOM from "react-dom";  
import App from "./App";  
import "./index.css";  
  
ReactDOM.render(  
  <React.StrictMode>  
    <App />  
  </React.StrictMode>,  
  document.getElementById("root")  
)
```

Passo 2: Criar o arquivo principal **App.js**

O arquivo **App.js** será responsável por renderizar a página principal.

```
import React from "react";
import TipCalculatorPage from "../pages/TipCalculatorPage";

function App() {
  return <TipCalculatorPage />;
}

export default App;
```

Passo 3: Criar a página principal **TipCalculatorPage.js**

Esta página organiza os componentes e gerencia os estados.

```

import React, { useState } from "react";
import InputField from "../components/InputField/InputField";
import ResultDisplay from "../components/ResultDisplay/ResultDisplay";
import Button from "../components/Button/Button";
import styles from "../TipCalculatorPage.module.css";

function TipCalculatorPage() {
  const [valorConta, setValorConta] = useState("");
  const [percentualGorjeta, setPercentualGorjeta] = useState("");
  const [resultado, setResultado] = useState(null);

  const calcularGorjeta = () => {
    const valor = parseFloat(valorConta);
    const valorGorjeta = (valor * parseFloat(percentualGorjeta)) / 100;
    const total = valor + valorGorjeta;
    setResultado({ gorjeta: valorGorjeta.toFixed(2), total:
total.toFixed(2) });
  };

  return (
    <div className={styles.pageContainer}>
      <h1>Calculadora de Gorjeta</h1>
      <InputField
        label="Valor da Conta:"
        value={valorConta}
        onChange={(e) => setValorConta(e.target.value)}
      />
      <InputField
        label="Porcentagem da Gorjeta (%):"
        value={percentualGorjeta}
        onChange={(e) => setPercentualGorjeta(e.target.value)}
      />
      <Button text="Calcular" onClick={calcularGorjeta} />
      {resultado && (
        <ResultDisplay gorjeta={resultado.gorjeta} total={resultado.total}
      />
      )}
    </div>
  );
}

export default TipCalculatorPage;

```

CSS para a página principal (**TipCalculatorPage.module.css**)

```
.pageContainer {  
  text-align: center;  
  margin: 50px auto;  
}  
  
h1 {  
  font-size: 24px;  
  margin-bottom: 20px;  
}
```

Passo 4: Criar o componente **InputField**

Componente para entradas de valores.

Código **InputField.js**

```
import React from "react";  
import styles from "../InputField.module.css";  
  
function InputField({ label, value, onChange }) {  
  return (  
    <div className={styles.inputContainer}>  
      <label>{label}</label>  
      <input type="number" value={value} onChange={onChange} />  
    </div>  
  );  
}  
  
export default InputField;
```

CSS para **InputField.module.css**

```
.inputContainer {  
  margin: 10px 0;  
}  
  
label {  
  display: block;  
  margin-bottom: 5px;  
}  
  
input {  
  padding: 5px;  
  font-size: 14px;  
  width: 200px;  
}
```

Passo 5: Criar o componente **ResultDisplay**

Componente para exibir os resultados (gorjeta e total).

Código **ResultDisplay.js**

```
import React from "react";
import styles from "./ResultDisplay.module.css";

function ResultDisplay({ gorjeta, total }) {
  return (
    <div className={styles.resultContainer}>
      <p>Gorjeta: R${gorjeta}</p>
      <p>Total: R${total}</p>
    </div>
  );
}

export default ResultDisplay;
```

CSS para **ResultDisplay.module.css**

```
.resultContainer {
  margin-top: 20px;
}

p {
  font-size: 18px;
}
```

Passo 6: Criar o componente **Button**

Botão reutilizável e estilizado.

Código **Button.js**

```
import React from "react";
import styles from "./Button.module.css";

function Button({ text, onClick }) {
  return (
    <button onClick={onClick} className={styles.button}>
      {text}
    </button>
  );
}

export default Button;
```

CSS para **Button.module.css**

```
.button {
  padding: 10px 20px;
  font-size: 16px;
  background-color: #4caf50;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

.button:hover {
  background-color: #45a049;
}
```

Passo 7: Configuração final

Após montar os arquivos, inicie o projeto com:

```
npm run dev
```
