Exemplo de Projeto React com TypeScript

Passo 1: Configuração do Projeto

1. Crie um novo projeto React com TypeScript usando Vite:

```
npm create vite@latest color-box -- --template react-ts
cd color-box
npm install
npm run dev
```

2. Estrutura do Projeto:

Certifique-se de que a estrutura básica do projeto é semelhante a esta:

```
color-box/
??? node_modules/
??? public/
??? src/
?   ??? App.tsx
?   ??? index.css
?   ??? main.tsx
?   ??? components/
?       ??? ColorBox.tsx
?       ??? ColorInput.tsx
??? .gitignore
??? index.html
??? package.json
```

??? tsconfig.json

??? vite.config.ts

Passo 2: Implementação dos Componentes

1. Componente ColorBox (ColorBox.tsx):

Este componente exibe uma caixa cuja cor de fundo é alterada com base no valor fornecido pelo usuário. O useEffect é utilizado para aplicar a cor de fundo sempre que a cor selecionada mudar.

Código:

```
import { useEffect, useState } from 'react';

function ColorBox({ color }: { color: string }) {
    const [bgColor, setBgColor] = useState<string>('white');

    useEffect(() => {
        setBgColor(color);
    }, [color]);

    return (
        <div style={{ backgroundColor: bgColor, width: '200px', height: '200px', border: '1px solid black' }}>
            <p style={{ textAlign: 'center', lineHeight: '200px', color: 'black' }}>Cor Atual: {bgColor}</p>
        </div>
```

```
    );
  }

  export default ColorBox;
```

2. Componente ColorInput (ColorInput.tsx):

Este componente permite ao usuário digitar o nome de uma cor ou um código hexadecimal de cor. O valor é passado para o ColorBox para alterar a cor de fundo.

Código:

```
  function ColorInput({ color, setColor }: { color: string, setColor: (value: string) => void }) {
    return (
      <div>
        <label htmlFor='colorInput'>Digite uma cor ou código hexadecimal: </label>
        <input
          type='text'
          id='colorInput'
          value={color}
          onChange={(e) => setColor(e.target.value)}
        />
      </div>
    );
```

```
}

export default ColorInput;
```

3. Componente Principal (App.tsx):

O componente principal App gerencia o estado da cor e o passa para os componentes ColorBox e ColorInput. O useEffect em ColorBox será chamado sempre que o valor da cor for alterado.

Código:

```
import './App.css';
import { useState } from 'react';
import ColorBox from './components/ColorBox';
import ColorInput from './components/ColorInput';

function App() {
  const [color, setColor] = useState<string>('white');

  return (
    <div className='App'>
      <h1>Mudar Cor de Fundo com useEffect</h1>
      <ColorInput color={color} setColor={setColor} />
      <ColorBox color={color} />
```

```
        </div>
      );
  }

  export default App;
```

Passo 3: Executar o Projeto

1. Execute o servidor de desenvolvimento:

```
npm run dev
```

2. Verifique a aplicação no navegador:

Acesse o endereço fornecido pelo Vite (geralmente http://localhost:3000) para ver os componentes em ação.

Explicação do Código

- ColorBox Component:

Usa useState para manter o estado da cor de fundo (bgColor).

Usa useEffect para atualizar a cor de fundo da caixa sempre que o valor da cor fornecida pelo usuário (prop color) mudar.

- ColorInput Component:

Permite ao usuário digitar o nome de uma cor ou um código hexadecimal. Esse valor é passado como prop para o

ColorBox, que então atualiza a cor de fundo.

- App Component:

Gerencia o estado da cor (color) e o distribui entre ColorBox e ColorInput. A mudança no valor da cor desencadeia o useEffect no ColorBox, alterando dinamicamente o estilo de exibição.