

ESTILO GLOBAL

Ao falar sobre "estilo global" em um projeto React, geralmente se refere ao uso de estilos que afetam globalmente toda a aplicação, em oposição a estilos modulares ou locais que se aplicam apenas a componentes específicos. Existem algumas vantagens em usar estilos globais em um projeto React:

Consistência Visual: o uso de estilos globais pode garantir uma consistência visual em toda a aplicação. Se você padronizar a aparência de certos elementos (como fontes, cores, margens) globalmente, isso ajuda a manter uma identidade visual coesa em toda a aplicação.

Facilidade de Manutenção: ao centralizar estilos globais, torna-se mais fácil fazer alterações e manutenções. Se houver a necessidade de ajustar um estilo específico, você pode fazer isso em um único local em vez de ter que atualizar estilos em vários componentes.

Rápida Implementação de Alterações de Design: mudanças de design que se aplicam a toda a aplicação podem ser implementadas de maneira mais eficiente com estilos globais. Isso é particularmente útil para ajustes de última hora ou para alterações que precisam ser implementadas rapidamente.

Melhor Performance em Algumas Situações: em alguns casos, a aplicação de estilos globais pode resultar em um desempenho mais eficiente, especialmente se você estiver lidando com um grande número de estilos ou componentes. Isso ocorre porque os estilos globais podem ser processados de maneira mais eficiente pelos motores de renderização.

No entanto, é importante equilibrar o uso de estilos globais com a modularidade. O excesso de estilos globais pode levar a problemas de colisão de estilos e dificultar a compreensão do código. Em alguns casos, uma abordagem híbrida, combinando estilos globais com estilos locais para componentes específicos, pode ser a solução mais adequada. Para resolver este problema, iremos utilizar CSS-in-JS!!!

TEMA DA APLICAÇÃO

Dentro da pasta **src**, crie uma pasta chamada **styles**.

Endereço: src/styles

A pasta 'styles' será responsável por armazenar todo o estilo global do projeto. Na pasta styles, crie um arquivo chamado **theme.js**.

Endereço: src/styles/theme.js

```
export default {  
  COLORS: {  
    BACKGROUND_900: "#232129",  
    BACKGROUND_800: "#312E38",  
    BACKGROUND_700: "#3E3B47",  
    WHITE: "#F4EDE8",  
    ORANGE: "#FF9000",  
    GRAY_100: "#999591",  
    GRAY_300: "#666360",  
    RED: "#FF002E"  
  }  
}
```

Esse objeto é exportado como padrão (usando export default), o que significa que, ao importar esse módulo em outro arquivo, você pode usar qualquer nome para se referir a esse objeto.

COLORS: É uma propriedade do objeto que contém várias constantes de cores organizadas de forma lógica. Cada constante é associada a um nome significativo para representar a cor que ela armazena.

Cores:

- BACKGROUND_900, BACKGROUND_800, BACKGROUND_700: Essas constantes representam diferentes tons de cor de fundo, provavelmente para serem usadas como cores de plano de fundo em elementos da interface.
- WHITE: Representa a cor branca.
- ORANGE: Representa uma cor alaranjada.
- GRAY_100, GRAY_300: Representam diferentes tons de cinza.
- RED: Representa uma cor vermelha.

Cada valor de cor é uma string no formato hexadecimal, que é uma representação comum de cores na web.

Essas constantes podem ser importadas em outros arquivos para garantir consistência nas cores usadas em diferentes partes do projeto. Por exemplo, se você estiver utilizando componentes em vários arquivos, poderá importar essas constantes para garantir que as cores utilizadas sejam consistentes em toda a aplicação. Isso facilita a manutenção e a atualização das cores, pois você só precisa fazer alterações em um local central.

No arquivo `main.js`, faça a importação do `ThemeProvider` da biblioteca `styled-components`.

```
import { ThemeProvider } from 'styled-components'
```

O `ThemeProvider` é um componente fornecido pela biblioteca `styled-components` que permite a criação de temas para estilização em uma aplicação React. Ele é usado para fornecer um tema (um conjunto de estilos) a todos os componentes estilizados dentro de uma determinada parte da árvore de componentes.

A ideia principal por trás do `ThemeProvider` é permitir a fácil troca de temas em toda a aplicação, proporcionando uma maneira consistente de gerenciar os estilos com base em um tema específico. Isso é especialmente útil em aplicações maiores, onde a consistência visual e a capacidade de alternar entre temas (como temas claros e escuros) são importantes.

Faça a importação do `theme`:

```
import theme from './styles/theme'
```

Por fim, faça com que o `ThemeProvider` seja o elemento Pai das páginas que desejam aplicar o tema desejado. Todos os componentes estilizados dentro do `<ThemeProvider>` terão acesso ao tema definido. Os componentes estilizados podem acessar o tema usando a propriedade `theme` passada automaticamente.

```
import React from 'react'
import ReactDOM from 'react-dom/client'

import { ThemeProvider } from 'styled-components'
```

```
import theme from '../styles/theme'

import { Home } from '../pages/Home/index'

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <ThemeProvider theme={theme}>
      <Home />
    </ThemeProvider>
  </React.StrictMode>,
)
```

Nos arquivos de estilos, para aplicar as variáveis definidas no tema global, utiliza a seguinte sintaxe.

`{{ theme }} => theme.COLORS.NOME_DA_VARIÁVEL;`

Exemplo aplicando a cor de fundo:

`background-color: {{ theme }} => theme.COLORS.BACKGROUND_800;`

CREATE GLOBAL STYLE

O `createGlobalStyle` é uma função fornecida pela biblioteca `styled-components` que permite a criação de estilos globais em uma aplicação React. Estilos globais são estilos que se aplicam a toda a aplicação, independentemente de onde os componentes estilizados estejam renderizados na árvore de componentes.

Ao contrário dos estilos locais, que são encapsulados em componentes específicos, os estilos globais afetam todos os elementos na aplicação, proporcionando uma maneira conveniente de definir estilos que não estão vinculados a um componente específico.

Dentro da pasta `styles`, crie um arquivo chamado **global.js**.

Endereço: `src/styles/global.js`

```
import { createGlobalStyle } from 'styled-components'

export default createGlobalStyle`
  * {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
  }

  body {
    background-color: ${({ theme }) => theme.COLORS.BACKGROUND_800};
    color: ${({ theme }) => theme.COLORS.WHITE};
  }

  a {
    // Remover o underline dos link
    text-decoration: none;
  }

  button, a {
    cursor: pointer;
    transition: filter 0.2s; //criar transições suaves e animações
  }
`
```

```
button:hover, a:hover {  
  filter: brightness(0.9); // leve escurecida  
}
```

No arquivo main.js, faça a importação do GlobalStyles na pasta styles e adicione o componente dentro do ThemeProvider, para ter acesso aos estilos.

```
import React from 'react'  
import ReactDOM from 'react-dom/client'  
  
import GlobalStyles from './styles/global'  
import { ThemeProvider } from 'styled-components'  
  
import theme from './styles/theme'  
  
import { Home } from './pages/Home/index'  
  
ReactDOM.createRoot(document.getElementById('root')).render(  
  <React.StrictMode>  
    <ThemeProvider theme={theme}>  
      <GlobalStyles/>  
      <Home />  
    </ThemeProvider>  
  </React.StrictMode>,  
)
```