



PP II - REACT.JS E TAILWIND

Semestre	4º Semestre
Tópicos Abordados	Códigos Introdução
Prioridade de Revisão	Média
Status de Compreensão	Compreendido
Matéria	UCE – Paradigmas de Programação II
Data da Aula	@25 de agosto de 2025
Professor	Daniel
Conteúdo	Conceitos

O que é React.Js

- Biblioteca de Interface de usuário
- Baseada em componentes (pequenas peças de UI)
- Interface declarativa: UI = função do estado

Componentes

Componentes é o nome dado a elementos/funções em JS/TS. Que irá renderizar um HTML e CSS na página WEB

```
function Botao() {
// Declarando Componente
  return <button>Clique aqui</button>;
  // TODO componente deve ter return
}
// Essa função retorna um HTML
```

Estado e Props

- Estado (useState): valores que mudam com interação
- Props/parametro: valores que passamos para os componentes

```
import { useState } from "react";
// Importante UseState para a aplicação

function Contador() {
// Declarando Componente
  const [n, setN] = useState(0);
  // Declarando variavel que fará o estado do componente
  return
  //Retornando...
  <button onClick={() => setN(n+1)}>
    Valor: {n}
  </button>;
  // onClick = {} - Metodo HTML que vigia ação de click, quando chamado, executa algo
  // () => - Função anonima em JS/TS
  // setN - Função declarada que irá definir valor de n
```

```
// n+1 - Somatiza 1 ao n
// RESUMO - Toda vez que onClick é chamado, uma função anônima é executada
// essa função anônima chama outra função, que é responsável por acrescentar um valor a variável
}
```

Vite

Em resumo, o vite é um pré processador para facilitar o uso de bibliotecas JS.

Instalação do Vite

```
npm create vite@latest meu-projeto -- --template react-ts

cd meu-projeto

npm install

npm run dev

# Não executar esse código no terminal!
```

Tailwind CSS

- Framework capaz de criar interface web responsiva
- Responsável apenas pela parte visual da aplicação
- Ele é inserido por meio de classes diretamente no HTML
- Funciona por meio de classes pré definidas!

```
<button
className="bg-blue-600 text-white px-4 py-2 rounded-lg hover:bg-blue-700">
  Clique aqui
</button>
<!--
  <button></button> - TAG HTML capaz de gerar um botão na página
  className - Atributo HTML responsável por armazenar classes
→
```

Classes do Exemplo

- className = "bg-blue-600 text-white px-4 py-2 rounded-lg hover:bg-blue-700"
 - bg-blue-600 → cor de fundo azul no tom 600 (Tailwind tem escala 50–900, quanto maior mais escuro)
 - text-white → cor do texto branca
 - px-4 → padding horizontal = 1rem (16px)
 - py-2 → padding vertical = 0.5rem (8px)
 - rounded-lg → cantos arredondados de tamanho "large"
 - hover:bg-blue-700 → quando o mouse passa em cima, muda o fundo para azul 700 (mais escuro que o 600).

Tailwind CSS por traz dos panos

Com o conceitos de classe CSS, o Tailwind traz classes pré definidas quando instalado.

Classe CSS do Tailwind

```
.blur-sm {
  --tw-blur: blur(var(--blur-sm));
  filter: var(--tw-blur,) var(--tw-brightness,) var(--tw-grayscale,);
```

```
}  
.grayscale {  
  --tw-grayscale: grayscale(100%);  
  filter: var(--tw-blur,) var(--tw-brightness,) var(--tw-grayscale,);  
}
```

/*

Classe é declarada e exportada

Essas classes não vemos no ambiente de desenvolvimento.

Apenas podemos utilizalas

Para poder alterar o conteudo dessa classe, devemos modificar a configuração

Alterar a classe não é o foco do estudo, mas é possível

*/

Chamando no HTML

Após o tailwind ter declarado a classe, ele disponibiliza para o HTML.

```
<div class="blur-sm grayscale">  
</div>
```

<!--

Nem toda classe CSS serve para toda TAG HTML

→

Criar classes

Você pode criar novas classes com sua nomenclatura e tipos, mas para isso é necessário ler a documentação!

Mais Sobre o Tailwind CSS

<https://tailwindcss.com/>