

# ANGULAR: PRINCIPAIS DIRETIVAS DESCOMPLICADAS



# INTRODUÇÃO

As diretivas são uma parte fundamental do Angular, permitindo que você manipule o DOM de maneira eficiente e declarativa. Existem dois tipos principais de diretivas no Angular: **Diretivas de Atributo** e **Diretivas Estruturais**. Neste Ebook vamos explorar cada uma delas com exemplos práticos.



# Diretivas de Atributo

As diretivas de atributo são usadas para alterar a aparência ou o comportamento de um elemento, componente ou outra diretiva. Elas são usadas para manipular as propriedades dos elementos. A seguir veremos algumas das diretivas de atributo mais comuns.



# ngClass

A diretiva ngClass permite adicionar ou remover classes CSS dinamicamente.

Exemplo de uso:

```
app.component.ts  
  
<button  
  [ngClass]="{'active': isActive, 'disabled': isDisabled}">  
  Clique Aqui  
</button>
```

Neste exemplo isActive e isDisabled são variáveis no componente. Se isActive for true, a classe active será aplicada. Se isDisabled for true, a classe disabled será aplicada.

# ngStyle

A diretiva ngStyle permite aplicar estilos CSS dinamicamente no elemento ao qual ela é aplicada.  
Exemplo de uso:

app.component.html

```
<button (click)="toggleActive()">Toggle Active</button>
<div [ngStyle]="{'color': isActive ? 'green' : 'red'}">
  Este texto muda de cor dinamicamente
</div>
```

app.component.ts

```
export class MeuComponente {
  isActive: boolean = false;

  toggleActive() {
    this.isActive = !this.isActive;
  }
}
```



# ngStyle

Neste exemplo, cada vez que o botão é clicado, a função `toggleActive` é chamada, alternando o valor de `isActive` entre `true` e `false`, e consequentemente, a cor do texto dentro do `<div>` muda entre verde e vermelho com base no valor da variável `isActive`.

Se a variável `isActive` for `true`, a cor do texto será verde (`green`). Caso contrário, se `isActive` for `false`, a cor do texto será vermelha (`red`).

# ngModel

A diretiva ngModel é usada para criar uma ligação bidirecional entre os dados do componente e o elemento do formulário no template. Isso significa que quando o usuário altera o valor do formulário, o valor da variável no componente também é atualizado, e vice-versa.

Exemplo de uso:

```
app.component.html
<input [(ngModel)]="username" placeholder="Digite seu nome">
<p>Olá, {{username}}! </p>
```



# ngModel

Neste exemplo username é uma variável no componente. O [(ngModel)] vincula o valor do input à variável username. Quando o usuário digita algo no campo de texto, a variável username é atualizada automaticamente.



# Diretivas Estruturais

As diretivas estruturais alteram a estrutura do DOM, adicionando ou removendo elementos com base em condições. A seguir veremos algumas das diretivas estruturais.



# ngIf

A diretiva ngIf adiciona ou remove um elemento baseado em uma expressão booleana.

Exemplo de uso:

```
app.component.html
<div *ngIf="isVisible">
  Este div só aparece se isVisible for true
</div>
```

Neste exemplo, isVisible é uma variável no componente. Se isVisible for true, o elemento será renderizado.

# ngFor

A diretiva ngFor cria uma instância de um template para cada item em uma lista.

Exemplo de uso:

```
app.component.html
<ul>
  <li *ngFor="let item of items">
    {{ item }}
  </li>
</ul>
```

Neste exemplo, items é um array no componente. Para cada item em items, um <li> será renderizado com o valor do item.

# ngSwitch

A diretiva ngSwitch exibe um dos vários elementos possíveis, com base em uma expressão.

Exemplo de uso:

```
app.component.html  
  
<div [ngSwitch]="status">  
  <div *ngSwitchCase="'active'">Status: Ativo</div>  
  <div *ngSwitchCase="'inactive'">Status: Inativo</div>  
  <div *ngSwitchDefault>Sem status</div>  
</div>
```

Neste exemplo, status é uma variável no componente. Dependendo do valor de status, um dos templates será renderizado.

# CONCLUSÃO

Diretivas são uma parte essencial do Angular, permitindo que você manipule o DOM de maneira eficaz. Com as diretivas de atributos, você pode alterar estilos e classes dinamicamente, enquanto as diretivas estruturais permitem criar ou remover elementos do DOM com base em condições e listas.

Compreender as diferenças e os usos de diretivas de atributo e diretivas estruturais é crucial para criar aplicativos Angular robustos e eficientes.

