

# Passando valores para um componente



As propriedades adicionadas a um componente são públicas mas é necessário informar ao angular quais dessas propriedades podem receber valores dos seus parents através do property bind. Para isso vamos usar os decorators.

```
import { Component, Input } from '@angular/core'

@Component({
  selector: 'ex-header',
  template: 'ex.html'
})

export class MeuComponente {
  @Input() titulo: string;
}
```

# Diretivas



**Nós já utilizamos diretivas. Na verdade, componentes são diretivas com templates. Portanto, componentes herdam tudo de diretivas.**

# Diretivas



Uma diretiva serve para adicionar comportamento a um elemento mas sem o template de um componente.

**Componentes** → São as mais comuns

**Estruturais** → Mudam o template e a estrutura do DOM

**Atributos** → Associam um atributo a um elemento do DOM e um determinado comportamento é aplicado ao elemento.

# Diretivas estruturais

## ngIf



Permite renderizar um conteúdo caso a expressão associada seja verdadeira.

```
<!-- ngIf -->  
  
<input type="text" [value]="aluno.nome">  
<div *ngIf="aluno.isAprovado">  
  O aluno está aprovado!  
</div>
```



# Diretivas estruturais

## ngFor

Repete o conteúdo de um elemento para cada item de uma coleção de objetos.

```
<!-- ngFor -->  
  
<ul>  
<li *ngFor="let aluno of alunos">{{aluno.nome}}</li>  
</ul>
```

# Diretivas estruturais

## ngSwitch



```
<!-- ngSwitch -->

<div [ngSwitch]="permissao">
  <p *ngSwitchCase="root">Você pode ler e gravar</p>
  <p *ngSwitchCase="user">Você pode ler</p>
  <p *ngSwitchDefault>Você tem permissão!</p>
</div>
```