



Angular - Componentes

O QUE É UM COMPONENTE NO ANGULAR?

Um **componente** é a **unidade básica de construção** de uma aplicação Angular.

👉 Pensa assim:

Um site em Angular é montado como LEGO.

Cada pecinha é um componente.

Exemplos de componentes comuns:

- cabeçalho (header)
- menu lateral
- rodapé
- card de produto
- tela de login
- formulário
- botão customizado

Cada **parte visível** da sua aplicação é um componente.

COMO UM COMPONENTE É FORMADO?

Um componente Angular SEMPRE tem 4 partes:

- 1 **Classe TypeScript** (`.ts`) → lógica
- 2 **HTML** (`.html`) → estrutura visual
- 3 **CSS/SCSS** (`.css`) → estilo

4 **Decorator @Component** → diz ao Angular que isso é um componente

Quando você cria um componente, o Angular gera arquivos assim:

```
meu-componente/
  meu-componente.ts    → lógica
  meu-componente.html  → visual
  meu-componente.css   → estilos
  meu-componente.spec.ts → testes (pode ignorar)
```

COMO FUNCIONA UM COMPONENTE NA PRÁTICA?

Vamos ver o “formato” de um componente:

```
import { Component } from '@angular/core';

@Component({
  selector: 'meu-componente',
  standalone: true,
  templateUrl: './meu-componente.html',
  styleUrls: ['./meu-componente.css']
})
export class MeuComponente {
  titulo = 'Olá!';
}
```

Explicando:

- `@Component({...})` → diz ao Angular: “Isso é um componente!”
- `selector` → nome da tag que você vai usar no HTML
 - exemplo: `<meu-componente></meu-componente>`
- `templateUrl` → arquivo HTML do componente
- `styleUrl` → arquivo CSS
- `standalone: true` → componentes modernos sem módulo

- `export class MeuComponente` → a lógica do componente

```
<header-component></header-component>
<menu-component></menu-component>
<conteudo-principal></conteudo-principal>
<footer-component></footer-component>
```

Isso monta a página inteira.

COMO COLOCAR UM COMPONENTE NA PÁGINA?

Se o componente tem este selector:

```
selector: 'card-produto',
```

Então você usa assim no HTML:

```
<card-produto></card-produto>
```

Simples assim.

COMO IMPORTAR COMPONENTES (ANGULAR MODERNO)

Quando você quer usar um componente dentro de outro, você **importa ele** dentro do decorador.

Exemplo:

```
@Component({
  selector: 'pagina-inicial',
  standalone: true,
  imports: [CardProdutoComponent], // IMPORTANDO
  templateUrl: './pagina.html',
```

```
})  
export class Paginalnicial {}
```

Isso permite você colocar no HTML:

```
<card-produto></card-produto>
```

✓ Criar um componente:

```
ng g c components/nome --standalone
```

✓ Usar o componente:

1. IMPORTAR no componente pai:

```
imports: [CardProdutoComponent]
```

1. Usar a tag:

```
<card-produto></card-produto>
```

3 Importar no componente pai

No componente onde você quer usar o novo componente, você faz:

Passo A — importar no topo

```
import { NomeDoComponente } from './caminho/nome-do-componente';
```

Passo B — adicionar na lista **imports**

```
@Component({  
  ...
```

```
    imports: [NomeDoComponente],  
})
```

Exemplo completo:

```
import { Component } from '@angular/core';  
import { NomeDoComponente } from './components/nome-do-component  
e/nome-do-componente';  
  
@Component({  
  selector: 'app-root',  
  standalone: true,  
  imports: [NomeDoComponente],  
  templateUrl: './app.html',  
})  
export class App {}
```

4 Usar no HTML

Agora você usa o selector normalmente:

```
<app-nome-do-componente></app-nome-do-componente>
```