

O que é um componente no Angular?

Imagine que você está construindo uma casa com blocos. Cada bloco é uma **peça pequena e independente**, mas quando você junta todos, eles formam uma casa completa.

No Angular, **componentes** são como esses blocos:

- **Cada componente é uma parte independente de uma aplicação.**
- Ele é responsável por uma **pequena área da tela**, como um botão, menu, ou até uma página inteira.
- Componentes ajudam a organizar o código, tornando mais fácil criar, manter e entender aplicações maiores.

Por exemplo:

- Um componente de "menu" cuida de mostrar o menu da aplicação.
- Um componente de "rodapé" exibe informações no final da página.

Estrutura de um componente

Um componente Angular é composto de 3 partes principais:

1. **HTML**: Define o que será exibido na tela (estrutura).
2. **CSS/SCSS**: Define como o conteúdo será estilizado (cores, fontes, espaçamento).
3. **TypeScript (TS)**: Define o comportamento do componente (o que ele faz e como funciona).

No Angular 19, a forma de declarar componentes em módulos mudou um pouco em relação a versões anteriores. Embora a abordagem do **@NgModule** tenha sido simplificada, a necessidade de agrupar componentes e outros elementos dentro de módulos ainda é fundamental para uma aplicação bem estruturada.

Aqui está como declarar componentes em módulos no Angular 19:

Passo 1: Criação de um Módulo

Em versões anteriores, você criaria um arquivo de módulo onde declararia todos os componentes que o módulo utiliza. A estrutura é similar à antiga, mas agora você pode usar a configuração diretamente no componente, e não apenas no módulo. Vamos ver um exemplo para entender como você faria isso.

Exemplo:

Imagine que você tem um **componente principal (AppComponent)** e quer declarar outro componente dentro de um módulo específico, como um **componente secundário (Comp2Component)**.

1. Arquivo do Componente Secundário (**comp2.component.ts**):

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

@Component({
  selector: 'app-comp2',
  template: `<p>Este é o Componente 2!</p>`,
})
export class Comp2Component {}
```

2. Arquivo do Componente Principal (**app.component.ts**):

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

import { Comp2Component } from './comp2/comp2.component'; // Importando
o componente secundário

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  imports: [Comp2Component], // Declarando o componente
  secundário no módulo
})
export class AppComponent {}
```

```
        title = 'projeto-inicial';  
    }  
}
```

Passo 2: Agrupando Componentes em Módulos

Embora no exemplo acima você possa diretamente importar e usar o **Comp2Component** no **AppComponent**, ainda existe a opção de criar um **módulo** dedicado para gerenciar esses componentes.

3. Arquivo do Módulo (**app.module.ts**):

```
import { NgModule } from '@angular/core';  
  
import { BrowserModule } from '@angular/platform-browser';  
  
import { AppComponent } from './app.component';  
  
import { Comp2Component } from './comp2/comp2.component';  
  
@NgModule({  
  declarations: [  
    AppComponent, // Componente principal  
    Comp2Component // Componente secundário  
  ],  
  imports: [  
    BrowserModule, // Módulo necessário para inicializar a aplicação  
  ],  
  providers: [],  
  bootstrap: [AppComponent], // Define o componente principal a ser  
    inicializado  
})  
  
export class AppModule {}
```

Passo 3: Usando Componentes em Módulos Externos

Se você estiver criando um componente que deseja **usar em outro módulo**, basta exportá-lo do módulo onde ele foi declarado e importá-lo no módulo onde você deseja usá-lo.

4. Arquivo de Módulo Externo (**comp2.module.ts**):

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

import { CommonModule } from '@angular/common';

import { Comp2Component } from './comp2.component';

@NgModule({

  declarations: [Comp2Component],

  imports: [CommonModule], // Dependências externas como CommonModule

  exports: [Comp2Component], // Exporta o componente para ser utilizado em
  outros módulos

})

export class Comp2Module {}
```

5. Importando no Módulo Principal:

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

import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';

import { Comp2Module } from './comp2/comp2.module'; // Importando o
módulo externo

@NgModule({

  declarations: [AppComponent],
```

```
imports: [BrowserModule, Comp2Module], // Agora podemos usar o
'Comp2Component' aqui

providers: [],

bootstrap: [AppComponent],

}))

export class AppModule {}
```

Resumo do Processo:

- **Declarar Componentes:** Para declarar componentes dentro de um módulo, você os coloca na lista **declarations** do módulo, como mostrado no **AppModule**.
- **Importar Componentes:** Quando você deseja usar um componente em outro módulo, você precisa importá-lo. Se o componente estiver em um módulo externo, exporte-o no módulo de origem e importe-o no módulo de destino.
- **Usar Componentes:** Dentro do arquivo de template HTML do componente, você pode usar a tag **<app-comp2></app-comp2>** para referenciar qualquer componente importado.