

# Estilização do Conteúdo Projetado no Angular

Quando usamos `ng-content` para projetar conteúdo dentro de um componente, podemos aplicar **estilos diferentes** para diferentes partes do conteúdo projetado.

No entanto, é importante entender como o **Encapsulamento de Estilos** no Angular funciona e como podemos estilizar os elementos corretamente.

---

**OBS: Faça a estilização dos conteúdos no próprio arquivo scss do conteúdo.**

## Como Funciona o Encapsulamento de Estilos no Angular?

Por padrão, o Angular encapsula os estilos dos componentes para evitar que vazem para outros componentes. Isso significa que os estilos definidos no CSS de um componente **não afetam diretamente o conteúdo projetado via `ng-content`**.

### Problema: Estilização Não Aplicada

Se tentarmos estilizar o conteúdo projetado **dentro do CSS do componente pai**, os estilos **não funcionarão**:

```
/* No arquivo card.component.scss */
p {
  color: red; /* ❌ Isso NÃO funcionará nos parágrafos projetados */
}
```

Para corrigir isso, temos algumas abordagens:

---

### ◆ 1. Usando `::ng-deep` para Forçar a Estilização

O `::ng-deep` permite que os estilos do componente **afetem o conteúdo projetado**.

### Arquivo: **card.component.scss**

```
::ng-deep p {  
  color: red;  
  font-weight: bold;  
}
```

### Arquivo: **card.component.html**

```
<div class="card">  
  <ng-content></ng-content> <!-- Qualquer conteúdo projetado será estilizado -->  
</div>
```

### Arquivo: **app.component.html**

```
<app-card>  
  <p>Sou um conteúdo projetado!</p>  
</app-card>
```

✅ O parágrafo dentro de `<app-card>` **ficará vermelho e negrito**.

📌 **Nota:** `::ng-deep` está **deprecated** e pode ser removido em versões futuras do Angular. Evite seu uso sempre que possível!

---

## ◆ 2. Estilizando Diretamente no Componente Pai

Uma abordagem melhor é **forçar a estilização diretamente no CSS global**.

### Arquivo: **styles.scss (Global)**

```
app-card p {  
  color: red;  
  font-weight: bold;  
}
```

✅ Isso **garante** que os parágrafos dentro de `<app-card>` sejam estilizados corretamente.

---

### ◆ 3. Usando **select** no **ng-content** para Diferentes Estilos

Podemos projetar diferentes conteúdos e aplicar **estilos específicos** para cada um deles.

#### Arquivo: **card.component.html**

```
<div class="card">
  <div class="header">
    <ng-content select=".titulo"></ng-content>
  </div>

  <div class="content">
    <ng-content></ng-content>
  </div>

  <div class="footer">
    <ng-content select=".rodape"></ng-content>
  </div>
</div>
```

#### Arquivo: **card.component.scss**

```
.header {
  background-color: lightblue;
  padding: 10px;
  text-align: center;
}

.content {
  padding: 15px;
}

.footer {
  background-color: lightgray;
  padding: 10px;
  text-align: right;
}
```

#### Arquivo: **app.component.html**

```
<app-card>
  <h2 class="titulo">Título do Card</h2>
  <p>Este é o conteúdo principal.</p>
  <p class="rodape">Rodapé do Card.</p>
</app-card>
```

✅ O título será estilizado com fundo azul, o conteúdo normal e o rodapé com fundo cinza.

---

## ◆ 4. Usando **ViewEncapsulation.None** para Desativar o Encapsulamento

Se quiser aplicar estilos livremente sem encapsulamento, podemos usar:

📁 Arquivo: **card.component.ts**

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

```
@Component({
  selector: 'app-card',
  templateUrl: './card.component.html',
  styleUrls: ['./card.component.scss'],
  encapsulation: ViewEncapsulation.None // ❌ Remove encapsulamento de estilos
})
export class CardComponent {}
```

✅ Agora, qualquer estilo definido no **card.component.scss** será aplicado a todos os elementos projetados.

---

## 🎯 Conclusão

🔥 Com essas técnicas, conseguimos **controlar a estilização do conteúdo projetado** de maneira eficiente:

- ✅ **::ng-deep** (evite usar, pois está obsoleto).
- ✅ Estilizar no CSS global (**styles.scss**).
- ✅ Usar **select** no **ng-content** para organizar os estilos.

✅ Desativar o encapsulamento com `ViewEncapsulation.None` (se necessário).

💡 Agora você pode criar componentes **reutilizáveis, estilizados e organizados!**

