

Angular - Compartilhando dado Pai e Filho

💡 O que é um componente Pai e um componente Filho?

No Angular, **Pai** e **Filho** não são nomes formais.

É uma forma de explicar a **relação entre dois componentes na tela**.

Pensa assim:

- O **Pai** é o componente **que contém** outro dentro dele.
- O **Filho** é o componente **que está dentro** do pai.

É igual quando uma caixa está **dentro** de outra caixa:

a caixa grande é o pai, a caixa dentro dela é o filho.

■ PARTE 1 — Pai → Filho (NOVA SINTAXE usando `input()`)

✓ Como funciona agora?

- Pai envia
- Filho recebe com `minhaProp = input<T>()`

Não existe mais decorador.

◆ Filho (recebe dados)

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

@Component({
  selector: 'app-filho',
  standalone: true,
  template: `
```

```
<p>Mensagem recebida: {{ mensagem() }}</p>  
}  
)  
export class Filho {  
  mensagem = input<string>();  
}
```

📌 Repare:

`mensagem()` → precisa do parênteses porque virou *signal*.

◆ Pai (envia dados)

HTML:

```
<app-filho [mensagem]="'Olá, Ana!'></app-filho>
```

ou com variável:

```
<app-filho [mensagem]="textoDoPai"></app-filho>
```

TS:

```
textoDoPai = "Oi do pai!";
```

✓ Resumo novo Pai → Filho

- Pai usa `[propriedade] = "valor"`
- Filho usa `propriedade = input<T>()`
- No HTML do filho: usar `propriedade()`

■ PARTE 2 — Filho → Pai (NOVA SINTAXE usando `output()`)

Agora outputs também mudaram.

✓ Filho dispara evento

✓ Pai escuta evento

Mas agora usamos `output()`.

◆ Filho (envia para o pai)

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

@Component({
  selector: 'app-filho',
  standalone: true,
  template: `
    <button (click)="enviar()">Enviar para o pai</button>
  `
})

export class Filho {
  avisarPai = output<string>(); // NOVO

  enviar() {
    this.avisarPai.emit("Filho disse olá!");
  }
}
```

◆ Pai (recebe evento)

HTML:

```
<app-filho (avisarPai)="receberMensagem($event)"></app-filho>
```

TS:

```
receberMensagem(msg: string) {
  console.log(msg);
}
```

■ PASSANDO OBJETO (NOVA SINTAXE)

Pai:

TS:

```
usuario = {  
  nome: 'Ana',  
  idade: 20,  
  curso: 'SI'  
};
```

HTML:

```
<app-filho [dadosUsuario]="usuario"></app-filho>
```

Filho:

```
dadosUsuario = input<{ nome: string; idade: number; curso: string }>();
```

HTML:

```
<p>Nome: {{ dadosUsuario().nome }}</p>  
<p>Idade: {{ dadosUsuario().idade }}</p>  
<p>Curso: {{ dadosUsuario().curso }}</p>
```
