

# Uso de varios componentes en Angular



## Uso de varios componentes en Angular

Aquí tienes un ejemplo de cómo usar un componente dentro de otro componente en Angular, utilizando `PadreComponent` e `HijoComponent`. En este caso, `PadreComponent` incluirá varias veces el componente hijo para que veamos cómo podemos relacionar componentes.

### Paso 1: Crear el componente `PadreComponent`

Primero, creamos el componente `PadreComponent`, que será un componente standalone y representará al componente padre.

```
ng g c padre --skip-tests
```

Este comando generará un componente llamado `PadreComponent`.

### Paso 2: Crear el componente `HijoComponent`

Ahora, creamos el componente `HijoComponent`, el cual se usará desde la clase `PadreComponent`. Estamos agregando el componente hijo dentro del componente padre, no es necesario, pero es común organizarlos de esta manera.

```
ng g c /padre/hijo --skip-tests
```

### Paso 3: Relacionar los componentes Padre e Hijo

Para usar el `HijoComponent` desde el `PadreComponent`, asegúrate de importarlo en la clase `PadreComponent` en la sección de imports e incluirlo en la plantilla del `PadreComponent`.

#### Código HijoComponent:

Código `hijo.component.html`:

```
<p>Componente Hijo</p>
```

Código `hijo.component.ts`:

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

@Component({
  selector: 'app-hijo',
  standalone: true,
  imports: [],
  templateUrl: './hijo.component.html',
  styleUrls: ['./hijo.component.css']
})
export class HijoComponent {

}
```

#### Código PadreComponent:

Código `padre.component.html`. Observar que incluye ya al `HijoComponent` con el selector `<app-hijo/>`:

```
<p>Componente Padre</p>
<app-hijo/>
<app-hijo/>
<app-hijo/>
```

Código `padre.component.ts`. Observar que importa la clase `HijoComponent` en la sección de imports para poder usarlo sin problemas:

```
import { Component } from '@angular/core';
import { HijoComponent } from "../hijo/hijo.component";

@Component({
  selector: 'app-padre',
  standalone: true,
  imports: [HijoComponent],
```

```

    templateUrl: './padre.component.html',
    styleUrls: ['./padre.component.css']
  })
  export class PadreComponent {

  }

```

## Incluir el PadreComponent en la clase principal AppComponent

Finalmente, podemos utilizar PadreComponent en el punto de entrada principal de la clase AppComponent tanto en la sección de imports de la clase, como en la plantilla.

Código app.component.html. Observar que incluye al PadreComponent con el selector <app-padre/>:

```

<section>
  <div class="container text-center">
    <h1 class="text-center text-warning">{{ titulo }}</h1>
    <app-padre/>
  </div>
</section>

```

Código app.component.ts. Observar que importa la clase PadreComponent en la sección de imports para poder usarlo sin problemas:

```

import { Component } from '@angular/core';
import { RouterOutlet } from '@angular/router';
import { NuevoComponenteComponent } from './nuevo-componente/nuevo-componente.component';
import { ComponenteInlineComponent } from './componente-inline/componente-inline.component';
import { PadreComponent } from './padre/padre.component';

@Component({
  selector: 'app-root',
  standalone: true,
  imports: [
    RouterOutlet,
    NuevoComponenteComponent,
    ComponenteInlineComponent,
    PadreComponent,
  ],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css'],
})
export class AppComponent {
  titulo = 'Universidad Angular';
}

```

```
}

imports: [HijoComponent],
templateUrl: './padre.component.html',
styleUrl: './padre.component.css'
}))
export class PadreComponent {

}
```

## Resultado Esperado

Al ejecutar la aplicación (`ng serve`), verás el resultado de incluir el componente Padre, y a su vez incluyendo el detalle del componente HijoComponent.



Este ejemplo muestra cómo se pueden usar componentes de manera reutilizable dentro de otros componentes en Angular.

Saludos!

Ing. Ubaldo Acosta

Fundador de [GlobalMentoring.com.mx](http://www.globalmentoring.com.mx)