

# Desarrollo y programación de Aplicaciones Avanzadas Angular

## 8 Comunicación entre componentes

**CORE**  
*networks*



# Componentes in-line template and styles

Para componentes muy sencillos se puede prescindir de la plantilla y/o el archivo CSS, incorporando el código a los metadatos:

```
@Component({  
  selector: 'app-nombre',  
  template: `  
    <!-- código HTML -->  
  `,  
  styles: [  
    ' /* reglas CSS */ '  
  ]  
})
```

# Componentes in-line template and styles

Se puede generar con Angular CLI:

```
ng g c componenteEnLinea -s -t
```

# Anidación de componentes

Los componentes se pueden anidar con la jerarquía que se necesiten a través de las etiquetas con la identificación del selector.

```
<app-padre>  
  <app-hijo></app-hijo>  
</app-padre>
```

# ng-content

Este elemento es utilizado para introducir datos de padres a hijos de manera simple a través de las plantillas HTML

```
<ng-content>
...
<elemento-hijo></elemento-hijo>
...
</ng-content select="query">
```

# Encapsulamiento

Es posible configurar el nivel de encapsulamiento del componente, por ejemplo para usar Shadow DOM.

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

@Component({
  ...
  encapsulation: ViewEncapsulation.Emulated, (default)
                                     // .ShadowDom
                                     // .None
  ...
})
```

# @Input: de componentes padres a hijos

A través del decorador @Input se pueden pasar propiedades de padres a hijos:

padre.component.html

```
<hijo [propiedadHijo]="propiedadPadre"></hijo>
```

hijo.component.ts

```
import { ..., Input, ... } from '@angular/core';  
  
@Input() propiedadHijo;
```

# @Output: de componentes hijos a padres

A través del decorador @Input se pueden pasar propiedades de padres a hijos:

hijo.component.ts

```
import { ..., Output, EventEmitter ... } from '@angular/core';

@Output() propiedadHijo: EventEmitter<any> new EventEmitter();

metodoHijo(parametros) {
    this.propiedadHijo.emit(parametros);
}
```



# @Output: de componentes hijos a padres

padre.component.html

```
<hijo (propiedadHijo)="metodoPadre($event)"></hijo>
```

padre.component.ts

```
metodoPadre(event) {  
    // codigo  
}
```