



COMPONENTES DINÁMICOS

Angular



ANGULAR

COMPONENTES DINÁMICOS



ANGULAR

- Componentes dinámicos → son componentes que se crean, cargan o insertan en la vista durante la ejecución, en lugar de declararlos directamente en el HTML mediante etiquetas como <app-mi-componente>
- Alternativas:
 - Mediante ngComponentOutlet
 - Mediante **createComponent**
- Permiten el envío de datos desde hacia y desde el componente dinámico.



ANGULAR

- Componente contenedor (html):
 - Se declara un contenedor lógico con **ng-container**:

```
<p>Componente 1 {{ contador }}</p>

<button (click)="insertar()">Cargar</button>
<ng-container #contenedor></ng-container>
```



ANGULAR

■ Componente contenedor (ts):

- Se obtiene la referencia al contenedor lógico:

```
export class Componente1 {
    container = viewChild.required("contenedor", { read: ViewContainerRef });
    contador = 0;
    public insertar() {
        this.container().createComponent(
            Componente2,
            {
                bindings: [
                    inputBinding("abierto", () => true),
                    inputBinding("nombre", () => "Hello"),
                    outputBinding("cierre", (valor: number) => {
                        this.contador += valor;
                        this.container().clear();
                    })
                ]
            }
        );
    }
}
```



ANGULAR

■ Componente contenedor (ts):

- Se crea el componente sobre el contenedor lógico:

```
export class Componente1 {
    container = viewChild.required("contenedor", { read: ViewContainerRef });
    contador = 0;
    public insertar() {
        this.container().createComponent(
            Componente2,
            {
                bindings: [
                    inputBinding("abierto", () => true),
                    inputBinding("nombre", () => "Hello"),
                    outputBinding("cierre", (valor: number) => {
                        this.contador += valor;
                        this.container().clear();
                    })
                ]
            }
        );
    }
}
```



ANGULAR

■ Componente contenedor (ts):

- Se declaran los ***binding*** que enlazan el componente dinámico con el contendor:

```
export class Componente1 {
    container = viewChild.required("contenedor", { read: ViewContainerRef });
    contador = 0;
    public insertar() {
        this.container().createComponent(
            Componente2,
        {
            bindings: [
                inputBinding("abierto", () => true),
                inputBinding("nombre", () => "Angular"),
                outputBinding("cierre", (valor: number) => {
                    this.contador += valor;
                    this.container().clear();
                })
            ]
        }
    );
}
```

Tipos de binding:
Input
Output
TwoWay



ANGULAR

- Componente dinámico (html):
 - Puede emitir eventos que llegan al contendor:

```
<h1>Componente 2</h1>
<button (click)="cierre.emit(2)">Cerrar</button>
```



ANGULAR

■ Componente dinámico (ts):

- Declara las entradas y salidas del componente (conectores con el contenedor):

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

@Component({
  selector: 'app-componente-2',
  imports: [],
  templateUrl: './componente-2.html',
  styleUrls: ['./componente-2.css'],
})
export class Componente2 {
  abierto = input<boolean>(); //Entrada reactiva
  nombre = input<string>(); //Entrada reactiva
  cierre = output<number>(); //Salida reactiva
}
```