

Angular.pdf



quiico_



Desarrollo de interfaces



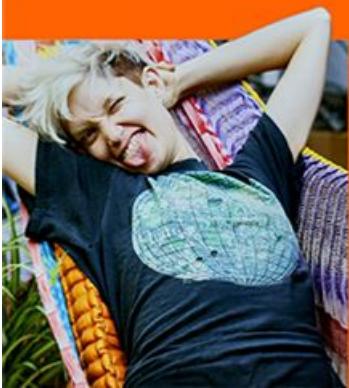
1º Desarrollo de Aplicaciones Multiplataforma



Estudios España



[Accede al documento original](#)



Una cuenta que no te pide **nada.**

Ni siquiera que apruebes. De momento.

(Estudia y no nos des ideas)

Cuenta NoCuenta

[Saber más](#)

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANCA NV se encuentra adherida al Sistema de Garantía de Depósitos Holandés con una garantía de hasta 100.000 euros por depositante. Consulta más información en [ing.es](#)

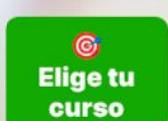
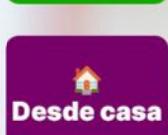




CARPE DIEM

Organiza tu futuro: estudia hoy para destacar mañana.

En Carpe Diem te esperan cursos adaptados a ti.
Una forma fácil y real de avanzar profesionalmente.



Angular

2º DAM Curso 2024/2025

Asignatura: Desarrollo de Interfaces

- Profesor: Manuel Mauri
- Alumno: Francisco José Alonso de Caso Ortiz Nº2



WUOLAH

Índice

Índice.....	1
¿Qué es Angular?.....	2
Características principales.....	2
Ventajas de Angular.....	2
Arquitectura de Angular.....	2
Componentes.....	3
Características clave de un componente:.....	3
Uso en la plantilla principal:.....	3
Ciclo de vida de un componente.....	3
AppComponent.....	4
Características del AppComponent:.....	4
¿Qué hace el AppComponent?.....	4
Uso de HTML y CSS en Angular.....	5
1. HTML en Angular:.....	5
2. CSS en Angular:.....	5
Creación de un Nuevo Proyecto Expo.....	6
Instalar Angular en un proyecto ya creado.....	6
Comandos básicos de Angular CLI.....	7
Directivas comunes.....	7
Data Binding en Angular.....	8

¿Qué es Angular?

Angular es un framework de desarrollo web de código abierto creado por Google. Está diseñado para desarrollar aplicaciones web modernas, especialmente aplicaciones de una sola página (SPA).

Características principales

1. **TypeScript:** Angular está escrito en TypeScript, un superconjunto de JavaScript que mejora la productividad y reduce errores.
2. **Arquitectura basada en componentes:** La aplicación se divide en piezas reutilizables llamadas componentes.
3. **Data Binding:** Proporciona una conexión entre el modelo de datos y la vista:
 - **One-Way Binding:** Los datos fluyen en una sola dirección (modelo a vista).
 - **Two-Way Binding:** Los datos se sincronizan entre modelo y vista.
4. **Directivas:** Permiten manipular el DOM de manera declarativa.
5. **Inyección de dependencias:** Facilita la gestión de servicios compartidos en toda la aplicación.
6. **Ruteo:** Incluye un sistema de navegación para crear SPA con varias vistas.
7. **RxJS:** Manejo de programación reactiva y observables.

Ventajas de Angular

- Desarrollo estructurado y mantenible.
- Gran ecosistema de herramientas y librerías.
- Soporte para aplicaciones multiplataforma.
- Alta productividad gracias a TypeScript y CLI.
- Comunidad activa y soporte a largo plazo por parte de Google.

Arquitectura de Angular

1. **Módulos:** Agrupan componentes, directivas, pipes y servicios.
 - Módulo principal: `AppModule`.
2. **Componentes:** Controlan una parte de la interfaz de usuario (vista).
 - Tienen un archivo HTML, CSS y TypeScript.
3. **Plantillas:** Define la estructura de la interfaz (HTML).
4. **Directivas:** Instrucciones especiales en las plantillas:
 - **Estructurales:** Alteran el DOM (por ejemplo, `*ngIf`, `*ngFor`).
 - **Atributos:** Cambian el comportamiento de un elemento (por ejemplo, `[disabled]`, `ngClass`).
5. **Servicios:** Contienen lógica compartida y manejo de datos.
6. **Inyección de dependencias:** Los servicios se inyectan donde son necesarios.
7. **Ruteo:** Define la navegación entre vistas mediante el módulo `RouterModule`



Organiza tu futuro: estudia hoy para destacar mañana.

En Carpe Diem te esperan cursos adaptados a ti.
Una forma fácil y real de avanzar profesionalmente.



Ciclo: Desarrollo de Aplicaciones Multiplataforma

Módulo Profesional: Desarrollo de Interfaces

Componentes

Un **componente en Angular** es una pieza fundamental que controla una parte específica de la interfaz de usuario. Cada componente se compone de tres elementos principales:

1. **HTML**: Define la estructura o plantilla de la interfaz del componente.
2. **CSS/SCSS**: Aplica estilos específicos al componente.
3. **TypeScript**: Contiene la lógica y los datos que alimentan la plantilla.

Características clave de un componente:

- **Reutilizables**: Pueden utilizarse en varias partes de la aplicación.
- **Aislados**: Cada componente puede tener su propio diseño y lógica, sin interferir con otros.
- **Decorador @Component**: Declara la clase como un componente y define la configuración, como la plantilla y los estilos.

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

@Component({
  selector: 'app-ejemplo', // Nombre del componente (para usarlo en el HTML)
  templateUrl: './ejemplo.component.html', // Archivo HTML asociado
  styleUrls: ['./ejemplo.component.css'] // Archivo CSS asociado
})
export class EjemploComponent {
  titulo = 'Hola Mundo'; // Variable de la lógica
  saludar() {
    console.log('¡Hola desde Angular!');
  }
}
```

Uso en la plantilla principal:

Para usar el componente, se inserta su **selector** en el archivo HTML de otro componente:

```
<app-ejemplo></app-ejemplo>
console.log('¡Hola desde Angular!');
```

Ciclo de vida de un componente

1. **ngOnInit**: Inicializa el componente después de crear la instancia.
2. **ngOnChanges**: Responde a cambios en las propiedades de entrada.

3. **ngOnDestroy**: Limpia recursos antes de destruir el componente.

AppComponent

El **AppComponent** es el **componente raíz** que se crea automáticamente al iniciar un proyecto Angular con el comando ng new. Es el punto de entrada principal de la aplicación y el contenedor donde se renderizan todos los demás componentes.

Características del AppComponent:

1. **Es el componente inicial:**
 - Es definido en el módulo principal (**AppModule**) como el componente principal de la aplicación.
 - Su selector (**<app-root>**) se encuentra en el archivo **index.html** para iniciar la aplicación.
2. **Estructura básica:** Al crear un nuevo proyecto Angular, el **AppComponent** contiene:
 - Un decorador **@Component** que define el selector, la plantilla HTML y el archivo de estilos.
 - Un archivo de clase TypeScript para gestionar la lógica.
3. **Archivos asociados:**
 - **app.component.ts**: Contiene la clase TypeScript con la lógica.
 - **app.component.html**: Plantilla que define la interfaz de usuario del componente.
 - **app.component.css**: Archivo CSS que define los estilos del componente.
 - **app.component.spec.ts**: Archivo para pruebas unitarias del componente.

```

● ● ●

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

@Component({
  selector: 'app-root', // Selector del componente
  templateUrl: './app.component.html', // Archivo HTML asociado
  styleUrls: ['./app.component.css'] // Archivo CSS asociado
})
export class AppComponent {
  title = 'Mi Aplicación Angular'; // Propiedad de ejemplo
}

```

¿Qué hace el AppComponent?

- Sirve como base para construir el resto de la aplicación.
- Desde este componente se cargan otros componentes mediante su selector (por ejemplo, **<app-header>** o **<app-footer>**).
- Actúa como el "cerebro inicial" de la aplicación Angular y define la estructura base donde se inyectan otros componentes.

**OFERTAS
BLACK FRIDAY**

msi®

Esta oferta

**es como una doble victoria
tuya: disfrútala ahora porque
no pasa dos veces.**

Ver ofertas



Tu viejo portátil ya dio lo que tenía que dar. Pásate a MSI: rápido, potente y sin dramas. Lo enciendes y estás listo para todo. Aprovecha las ofertas y despídete del modo “se cuelga cada dos por tres”.

Uso de HTML y CSS en Angular

Angular utiliza HTML y CSS para definir y estilizar la interfaz de usuario de la aplicación:

1. HTML en Angular:



```
<div>
  <h1>{{ titulo }}</h1>
  <button (click)="saludar()">Saludar</button>
</div>
```

Los componentes contienen plantillas HTML que definen la estructura de la interfaz.

Se pueden utilizar directivas de Angular dentro del HTML para crear comportamientos dinámicos.

2. CSS en Angular:

Cada componente puede tener su propio archivo CSS asociado para estilos específicos.

Los estilos pueden ser locales al componente o globales a la aplicación.

Angular admite preprocesadores como SCSS o LESS para estilos avanzados.

```
h1 {
  color: blue;
}

button {
  background-color: green;
  color: white;
}
```

Para aplicar clases condicionalmente, se utiliza `[ngClass]`:



```
<div [ngClass]="{ 'activo': esActivo }">Contenido</div>
```

Portátiles desde
549€



msi
BLACK FRIDAY



Ciclo: Desarrollo de Aplicaciones Multiplataforma
Módulo Profesional: Desarrollo de Interfaces

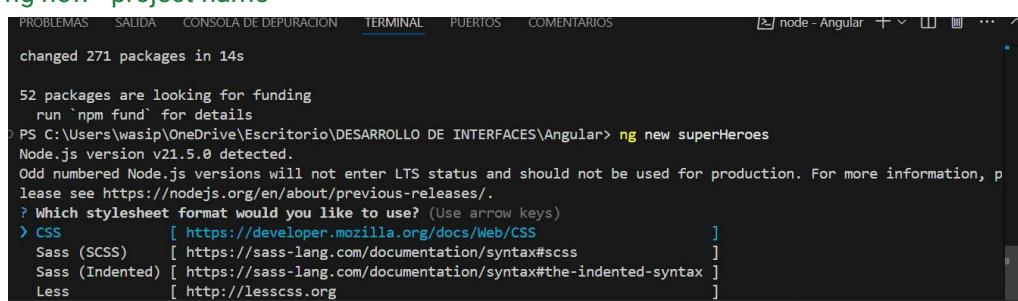
Creación de un Nuevo Proyecto Expo

1. Instala angular:

```
npm install -g @angular/cli
```

2. Inicializa el proyecto:

```
ng new <project-name>
```



```
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS COMENTARIOS node - Angular ... changed 271 packages in 14s 52 packages are looking for funding run `npm fund` for details PS C:\Users\wasip\OneDrive\Escritorio\DESARROLLO DE INTERFACES\Angular> ng new superHeroes Node.js version v21.5.0 detected. Odd numbered Node.js versions will not enter LTS status and should not be used for production. For more information, please see https://nodejs.org/en/about/previous-releases/. ? Which stylesheet format would you like to use? (Use arrow keys) > CSS [ https://developer.mozilla.org/docs/Web/CSS ] Sass (SCSS) [ https://sass-lang.com/documentation/syntax#scss ] Sass (Indented) [ https://sass-lang.com/documentation/syntax#the-indented-syntax ] Less [ http://lesscss.org ]
```

Debes elegir CSS

3. Muévete a la carpeta del proyecto recién creado:

```
cd my-first-angular-app
```

4. Inicia el proyecto en el entorno de desarrollo:

```
npm start
```

Instalar Angular en un proyecto ya creado

1. Inicializa el proyecto si no lo está (si no tienes package.json):

```
npm init -y
```

2. Instala Angular CLI (si no está instalado globalmente):

```
npm install -g @angular/cli
```

3. Añade Angular al proyecto:

```
ng new . --skip-install
```

- Esto creará los archivos básicos de Angular en el directorio actual sin sobrescribir nada.

4. Instala las dependencias:

```
npm install
```

5. Inicia el servidor de desarrollo:

```
ng serve
```

Con esto, Angular estará configurado dentro del proyecto existente.

ENCUENTRA EL TUYO
#01



VER OFERTAS

Francisco José Alonso de Caso Ortiz 2ºDAM Nº2 Curso 2024/2025

WUGLAH

6

Comandos básicos de Angular CLI

- **Crear un componente:**

```
ng generate component nombre-componente
```

Cuando creas un componente has de indicarle su:

- selector → identificador único que permite renderizar el componente como una etiqueta HTML personalizada.
- imports
- templateUrl → a qué html pertenece
- styleUrls → a qué style/css pertenece

- **Crear un servicio:**

```
ng generate service nombre-servicio
```

- **Crear un módulo:**

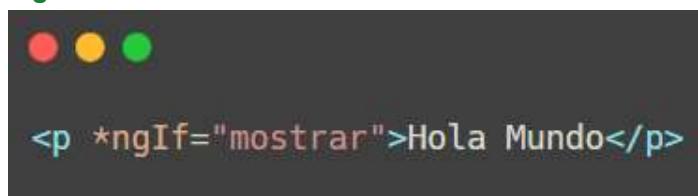
```
ng generate module nombre-modulo
```

- **Construir la aplicación para producción:**

```
ng build --prod
```

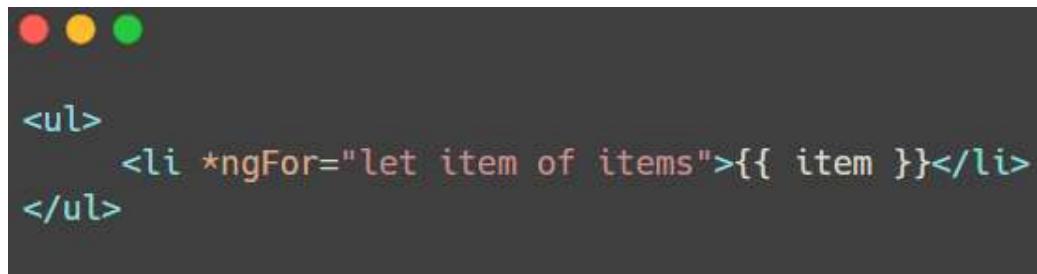
Directivas comunes

1. ***ngIf:** Renderiza elementos condicionalmente.



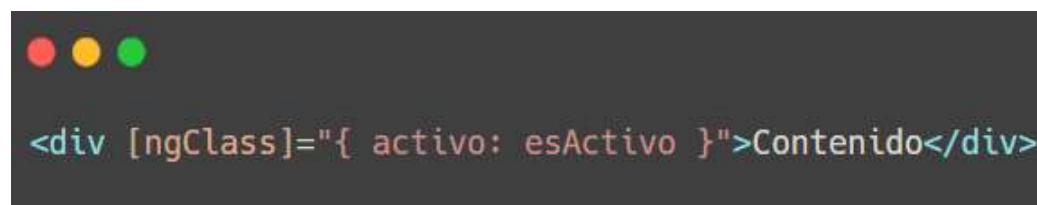
```
<p *ngIf="mostrar">Hola Mundo</p>
```

2. ***ngFor:** Itera sobre listas.



```
<ul>
  <li *ngFor="let item of items">{{ item }}</li>
</ul>
```

3. **[ngClass]:** Aplicar clases CSS condicionales.



```
<div [ngClass]="{ activo: esActivo }">Contenido</div>
```

- Es decir, si se cumple mi condición muestro las cosas de una manera, y sino, de otra.
- Doble comilla para los atributos y comillas simples para las clases.

Data Binding en Angular

1. Interpolación:

Muestra datos del componente en la plantilla:



```
<h1>{{ titulo }}</h1>
```

2. Property Binding:

Enlaza propiedades de elementos HTML:



```
<input [value]="nombre">
```

3. Event Binding:

Escucha eventos de usuario:



```
<button (click)="saludar()">Click</button>
```

4. Two-Way Binding:

Sincroniza datos entre la vista y el modelo (requiere `FormsModule`):



```
<input [(ngModel)]="nombre">
```