

# INTRODUCCIÓN



```
@Component ng serve --Open  
ng generate component [component-name]  
npm install -g @Angular/cli  
ng generate class [class-name]  
ng generate environments[options]  
ng generate interface[name] [type][options]
```

---

# ¿Qué es Angular?

*Angular es una plataforma de desarrollo construida sobre TypeScript.*



---

## Como plataforma incluye

- *Un marco basado en componentes para crear aplicaciones web escalables*
  - *Una colección de bibliotecas bien integradas que cubren una amplia variedad de funciones como:*
    - *El enrutamiento*
    - *La administración de formularios*
    - *La comunicación cliente-servidor*
  - *Un conjunto de herramientas para desarrolladores que le ayudarán a desarrollar, crear, probar y actualizar su código.*
-

---

# ¿Para qué sirve **Angular**? 🤔

---

- Desarrollar aplicaciones web interactivas y escalables
  - Crear interfaces de usuario reactivas
  - Organizar el código por componentes reutilizables
  - Gestionar rutas y navegación entre vistas sin recargar la página
  - Conectar con APIs RESTful fácilmente
-

---

# Historia de Angular

---

- **2009 – AngularJS:**

- Creado por Google. Framework basado en JavaScript con arquitectura MVC y binding bidireccional.
- 🖱️ Bueno para apps pequeñas, pero difícil de escalar.

- **2016 – Angular 2+:**

- Reescritura total usando TypeScript y arquitectura por componentes.
- Mayor rendimiento, modularidad y soporte móvil.

- **Angular moderno (v4 en adelante):**

- Actualizaciones semestrales, mejoras constantes (CLI, Ivy, Standalone Components).

- **Actualidad:**

- Framework robusto y escalable, ampliamente usado en empresas y aplicaciones web complejas.
-

Diferencias

AngularJS vs Angular

Característica	AngularJS	Angular
Arquitectura	MVC + \$scope	Componentes y módulos
Lenguaje	JavaScript	TypeScript
Data binding (Vinculación de datos)	De dos vías	De una sola vía (por defecto)
Rendimiento	Lento en aplicaciones de gran escala	Más rápido gracias a técnicas de compilación y su motor Ivy
Escalabilidad	Limitada	Alta
Herramientas modernas	Limitadas	CLI, Testing, RxJS
Compatibilidad	Obsoleto	Activo y mantenido

Comparación

Angular vs su competencia

Característica	Angular	Vue.js	React
Tipo	Framework completo	Framework progresivo	Librería UI
Lenguaje principal	TypeScript	JavaScript + Templates	JavaScript + JSX (Opcional TS)
Curva de aprendizaje	Alta	Baja	Media
Flexibilidad	Estructura rígida	Equilibrada	Alta flexibilidad
Escalabilidad	Muy Alta	Alta	Alta
Comunidad	Grande	Grande y creciente	Enorme
Ideal para...	Proyectos grandes, corporativos	Apps rápidas de desarrollar	Interfaces interactivas y dinámicas

# Estructura de trabajo en **Angular**

La estructura básica al generar un proyecto de Angular es la siguiente:

```
my-app/
├── src/
│   ├── app/
│   │   ├── components/ → Componentes reutilizables
│   │   ├── pages/      → Componentes principales (pantallas o vistas)
│   │   ├── services/   → Servicios para lógica de negocio y consumo de APIs
│   │   ├── models/     → Interfaces y clases TypeScript (tipos de datos)
│   │   ├── app.module.ts → Módulo principal del proyecto
│   │   └── app.component.ts / .html / .css → Componente raíz
│   ├── assets/         → Imágenes, estilos globales, fuentes, etc.
│   ├── environments/   → Configuración para desarrollo y producción
│   └── index.html       → Archivo HTML principal
├── angular.json         → Configuración del proyecto
├── package.json         → Dependencias del proyecto
└── tsconfig.json        → Configuración de TypeScript
```

**SPA** (Single Page Application), no recarga toda la página cada vez que el usuario navega.

Angular **carga una única vez el archivo `index.html`** y luego maneja toda la navegación de manera dinámica desde el navegador mediante **su sistema de rutas (Router)**, que **permite mostrar diferentes componentes dentro de una misma página**, actualizando solo el contenido necesario **a través de una etiqueta especial llamada `<router-outlet>`**.

# Angular

## Ventajas

---

Ofrece **todas las herramientas necesarias para el frontend**, sin depender de muchas librerías externas.

**Testabilidad:** Facilita la identificación y resolución rápida de problemas.

**Arquitectura basada en componentes:** facilita el desarrollo modular, reutilizable y mantenible del código.

Integración con **TypeScript:**

- Tipado estático: detecta errores en tiempo de desarrollo.

- Autocompletado inteligente: mejora la productividad.

- Depuración más sencilla: facilita identificar fallos.

## Desventajas

---

**Framework complejo**

Requiere **mucho código y configuración** en comparación con otros frameworks

**Complejidad para SEO**

**Rendimiento inicial más lento:** el tamaño del bundle y la carga inicial de una app Angular puede ser mayor

Demasiada estructura impuesta: puede sentirse **rígido para proyectos simples**, donde otras soluciones más flexibles serían más apropiadas.



---

# ¿Qué se necesita para correr un proyecto en Angular?

---

## 1.- Instalar Node.js y npm

esto se hace desde [nodejs.org](https://nodejs.org)

## 2.- Instalar Angular CLI

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

## 3.- crear el proyecto ya sea clonandolo de un repositorio o directamente.

```
ng new my-proyecto-angular
```

## 4.- Instalar Dependencias

```
npm install
```

## 5.- Iniciar el servidor de desarrollo

```
ng serve
```

## 6.- Abrir en el navegador

👉 <http://localhost:4200>

---

---

*Gracias por su Atención*

---