



Angular Básico

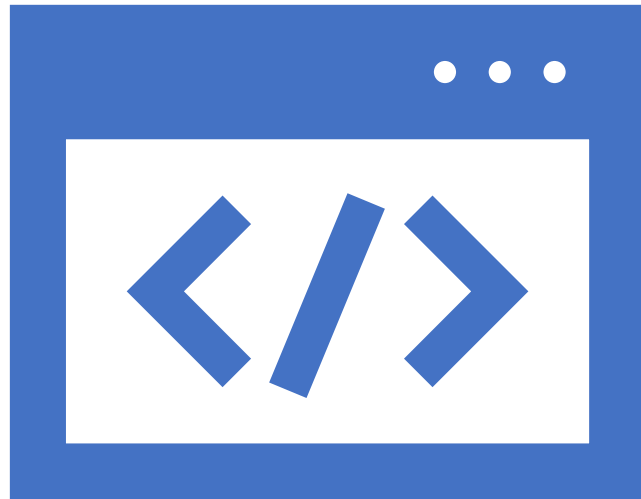
Web Application Development





Introduccion

- ¿Qué es Angular?
- ¿Por qué aprender Angular?
- Evolución de Angular



¿Qué es Angular?

- Es un Framework Frontend de JavaScript, Ayuda a construir interfaces web interactivas y modernas.
- Es una colección de herramientas y funciones que rodean al Framework, tal como una CLI, gestor de proyectos, Herramientas para Debugging, Plugins para IDE, entre otras herramientas.

A decorative graphic on the left side of the slide. It features a large, stylized blue checkmark centered within a white circle. This circle is surrounded by several concentric, semi-transparent rings in shades of light blue and green, creating a layered, circular effect.

¿Por qué aprender Angular?

- Simplifican el proceso de creación de interfaces conforme un proyecto vaya creciendo.
- Escribes Código Declarativo
- Separación de conceptos vía Componentes
- principios y conceptos OOP
- Usas TypeScript de forma nativa

Evolución de Angular

Es un framework que sigue evolucionando e innovando, aunque lo hace de una manera muy estable y compatible con versiones anteriores.

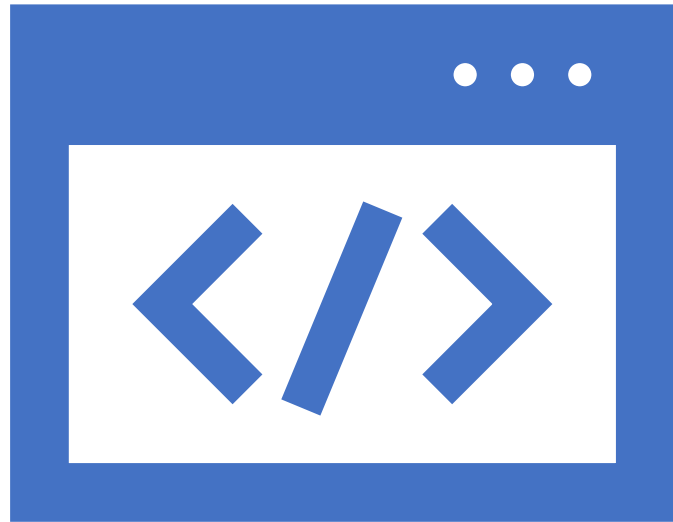
El equipo de Angular tiene una política de actualización de versiones cada 6 meses.

Las actualizaciones son incrementales, esto significa que Angular 2 es la base “Vanilla” y sus posteriores versiones hacen mejoras “opcionales” a la versión base.



Instalación de Angular

- Requisitos previos: Node.js y npm.
- Instalación de Angular CLI.
- Creación de un nuevo proyecto.
- Estructura del proyecto Angular




Requisitos previos:
Node.js y npm.

- Node.js es un entorno de ejecución para JavaScript.
- Permite ejecutar código JavaScript en el servidor.
- En esta presentación, aprenderemos a instalar Node.js en Windows.

Run JavaScript Everywhere

Node.js® is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts.

Download Node.js (LTS) 

Create an HTTP Server Write Tests Read and Hash a File Streams

```
1 // server.mjs
2 import { createServer } from 'node:http';
3
4 const server = createServer((req, res) => {
5   res.writeHead(200, { 'Content-Type': 'text/plain' });
6   res.end('Hello World!\n');
7 });
8
9 // starts a simple http server locally on port 3000
10 server.listen(3000, '127.0.0.1', () => {
11   console.log('Listening on 127.0.0.1:3000');
12 });
13
14 // run with `node server.mjs`
```

Paso 1:
Descargar
Node.js

- Visita el sitio web oficial de Node.js: <https://nodejs.org/>
- Haz clic en el botón de descarga para Windows.



Paso 2: Ejecutar el Instalador

- Una vez descargado el archivo, haz doble clic para ejecutar el instalador.
- Se abrirá el asistente de instalación de Node.js.
- Lee el acuerdo de licencia.
- Haz clic en "Next".
- Elige la carpeta donde se instalará Node.js.
- Haz clic en "Next".
- Elige los componentes a instalar.
- Deja las opciones predeterminadas y haz clic en "Next".
- Revisa la configuración y haz clic en "Install".
- Una vez finalizada la instalación, haz clic en "Finish".
- Node.js ya está instalado en tu sistema.

```
PS C:\Users\[redacted]> node -v
v20.15.0
PS C:\Users\[redacted]> npm -v
10.7.0
```

Verificar Instalación

- Abre una terminal (CMD o PowerShell).
- Escribe **node -v** y **npm -v** para verificar la instalación.
- Se debe ver las versiones de Node.js y npm instaladas.

Paso 3: Instalar Angular CLI

```
PS C:\Users\[redacted]> npm install -g @angular/cli
changed 224 packages in 9s
41 packages are looking for funding
  run `npm fund` for details
```

- Angular CLI es una herramienta de línea de comandos para inicializar, desarrollar, y mantener aplicaciones Angular.
- En la terminal, ejecuta el siguiente comando:

Comando: **npm install-g @angular/cli**

La bandera "-g" indica que se está instalando este paquete a nivel Global.

Paso 4: Crear un Nuevo Proyecto Angular

- Navega a la carpeta donde deseas crear el proyecto.
- Ejecuta el comando: Comando: **ng new my-app**
- La convención indica que se escriba el nombre del proyecto en minúsculas y separado con guiones (Kebab case)
- Sigue las instrucciones en pantalla para configurar el nuevo proyecto

```
PS C:\Users\[redacted]\Escritorio> ng new my-app

Would you like to share pseudonymous usage data about this project with the Angular Team
at Google under Google's Privacy Policy at https://policies.google.com/privacy. For more
details and how to change this setting, see https://angular.dev/cli/analytics.

Yes

Thank you for sharing pseudonymous usage data. Should you change your mind, the following
command will disable this feature entirely:

  ng analytics disable --global

Global setting: enabled
Local setting: No local workspace configuration file.
Effective status: enabled
? Which stylesheet format would you like to use? CSS [ https://developer.mozilla.org/docs/Web/CSS ]
? Do you want to enable Server-Side Rendering (SSR) and Static Site Generation (SSG/Prerendering)? Yes
```

Paso 4: Crear un Nuevo Proyecto Angular

- En caso de que tenga un error por falta de permisos, ejecuta el comando:

Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy Unrestricted

```
PS C:\Users\██████████\Escritorio> ng new my-app
ng : File C:\Users\██████████\AppData\Roaming\npm\ng.ps1 cannot be
loaded because running scripts is disabled on this system. For
more information, see about_Execution_Policies at
https://go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:1
+ ng new my-app
+ ~~~
    + CategoryInfo          : SecurityError: (:) [], PSSecurityException
    + FullyQualifiedErrorId : UnauthorizedAccess

PS C:\Users\██████████\Escritorio> Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy Unrestricted
```

Paso 5: Ejecutar la Aplicación Angular

- Navega a la carpeta del proyecto:
- Ejecuta el siguiente comando para iniciar el servidor de desarrollo:
 - **ng serve**
- Abre tu navegador y ve a **http://localhost:4200/** para ver tu aplicación



Hello, my-app

Congratulations! Your app is running. 🎉

[Explore the Docs](#)

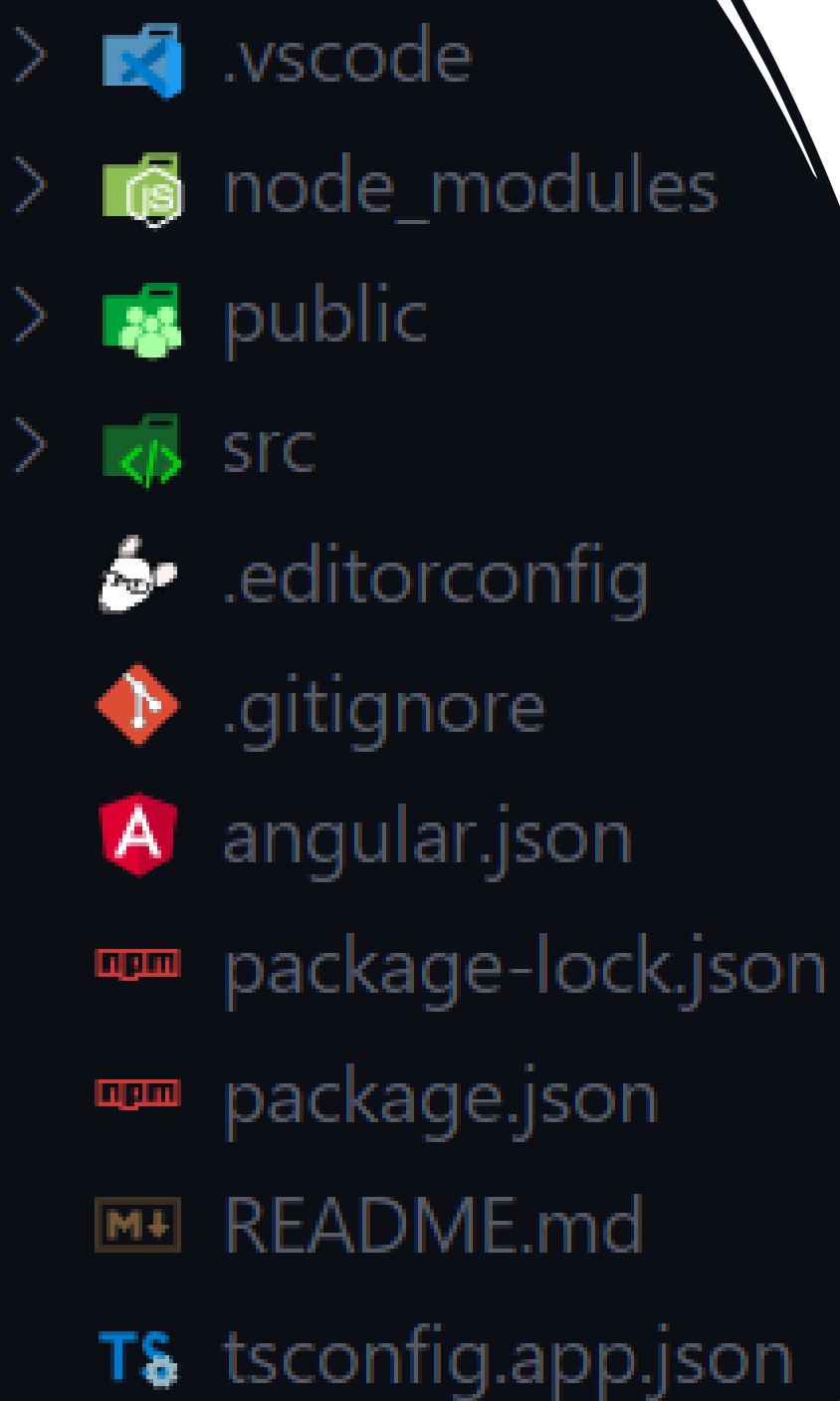
[Learn with Tutorials](#)

[CLI Docs](#)

[Angular Language Service](#)

[Angular DevTools](#)





Estructura de un Proyecto en Angular

Estructura de un Proyecto en Angular

README.md - Descripción del proyecto y primera vista dentro de un repositorio, además de detalles necesarios para levantar los servicios.

tsconfig.* - toda la configuración de TypeScript se hace mediante estos ficheros.

package.json - indica todas las dependencias de Angular que se tienen en este proyecto.

package-lock.json - fichero que se actualiza automáticamente con las versiones utilizadas en el proyecto.

angular.json - describe toda la arquitectura del Proyecto, así como la configuración de este mismo.

.gitignore - indica directorios y ficheros a ignorar en caso de ser guardado en un gestor de versiones como git.

Estructura de los directorios de la aplicación

- **/src** - Carpeta Principal donde se encuentran los archivos de código del proyecto.
- **/src/app** - Carpeta donde se encuentran los componentes Principales de la aplicación.
- **/public** - Todo fichero que se encuentre en esta carpeta será de fácil acceso mediante una URL relativa.
- **/node modules** - Todas las dependencias de Angular se guardan en este fichero.
- **/.vscode** - configuración del editor de texto a usar (Visual Studio Code en este ejemplo). Esta es la única carpeta ajena a Angular.

Estructura de la carpeta **/src**

- **/src/style.css** - Estilos globales de la aplicación.
- **/src/main.ts** - Es el punto de entrada de la aplicación.
- **/src/index.html** - Es el fichero que inicializa la aplicación.
- **/src/app** - Carpeta donde se encuentran los componentes Principales de la aplicación.

```
src/  
  app/  
    core/  
    shared/  
    components/  
    services/  
    models/  
    app.module.ts  
    app.component.ts  
    app.component.html  
    app.component.css  
    app.component.spec.ts
```

Estructura de la
carpeta /src/app

Estructura de los archivos en /src/app

app.module.ts - Este archivo es el módulo principal de la aplicación. Define los componentes, directivas, pipes y servicios que pertenecen a este módulo y los que deben ser importados para que la aplicación funcione.

app.component.ts - Define el componente principal de la aplicación. Incluye la lógica de negocio y las propiedades necesarias para la interfaz de usuario.

app.component.html - La plantilla HTML del componente principal. Define la estructura visual de la interfaz de usuario.

app.component.css - Archivo de estilos CSS del componente principal. Define la apariencia visual del componente.

app.component.spec.ts - Archivo de pruebas unitarias para el componente principal.

Estructura de la carpeta /src/app

- **/core** - Contiene servicios y otras piezas reutilizables que solo se deben cargar una vez en toda la aplicación, como guardias, interceptores y servicios globales.
- **/shared** - Contiene componentes, directivas y pipes reutilizables en toda la aplicación. También puede incluir módulos compartidos.
- **/components** - Agrupa los componentes específicos de la aplicación. Cada componente suele tener su propia carpeta con sus archivos ".ts", ".html", ".css" y ".spec.ts".
- **/services** - Contiene los servicios que manejan la lógica de negocio y la comunicación con APIs externas.
- **/models** - Define las interfaces y clases que representan datos utilizados en la aplicación.