# Estructura de un Proyecto Angular

Cuando se crea un nuevo proyecto Angular con Angular CLI, se genera una estructura de archivos y carpetas organizada para facilitar el desarrollo y la escalabilidad de la aplicación. En este documento, exploraremos en detalle la estructura de un proyecto Angular y la función de cada archivo y carpeta.

---

## 1. Estructura General

Al crear un proyecto con `ng new mi-proyecto-angular`, la estructura generada se verá similar a la siguiente:

```

mi-proyecto-angular/

├── node\_modules/ # Dependencias instaladas con npm

├── src/ # Código fuente del proyecto

│ ├── app/ # Componentes, módulos y servicios

│ ├── assets/ # Archivos estáticos como imágenes, fuentes y CSS global

│ ├── environments/ # Configuración para diferentes entornos (desarrollo, producción)

│ ├── index.html # Archivo HTML principal

│ ├── main.ts # Punto de entrada de la aplicación

│ ├── polyfills.ts # Scripts para compatibilidad con navegadores antiguos

│ ├── styles.scss # Archivo de estilos global de la aplicación

│ ├── test.ts # Configuración para pruebas unitarias

└── ├── favicon.ico # Ícono de la aplicación

├── angular.json # Configuración general del proyecto Angular

├── package.json # Listado de dependencias y scripts npm

├── tsconfig.json # Configuración de TypeScript

├── tsconfig.app.json # Configuración de TypeScript específica para la aplicación

├── tsconfig.spec.json # Configuración de TypeScript para pruebas unitarias

└── README.md # Documentación inicial del proyecto

```

---

## 2. Explicación de Archivos y Carpetas

### 📁 `node\_modules/`

Contiene todas las dependencias del proyecto instaladas a través de npm. No se debe modificar manualmente, ya que npm se encarga de gestionar estos archivos.

### 📁 `src/`

El código fuente de la aplicación se encuentra en esta carpeta. Contiene subcarpetas clave:

#### 📁 `app/`

* - Contiene los \*\*componentes, módulos, servicios\*\* y otros elementos centrales de la aplicación.
* - Dentro de `app/`, Angular genera por defecto un módulo raíz (`app.module.ts`) y un componente raíz (`app.component.ts`).
* - Ejemplo de estructura dentro de `app/`:

```

app/

├── app.component.html # Plantilla del componente principal

├── app.component.ts # Lógica del componente principal

├── app.component.scss # Estilos del componente principal

├── app.module.ts # Módulo raíz de la aplicación

└── app-routing.module.ts # Configuración de rutas (si se habilitó el enrutamiento)

```

#### 📁 `assets/`

Carpeta para almacenar recursos estáticos como imágenes, fuentes y archivos JSON. Todo lo que se coloque aquí estará accesible en `http://localhost:4200/assets/`.

#### 📁 `environments/`

Contiene archivos para definir variables específicas de entorno, por ejemplo:

* - `environment.ts` (modo desarrollo)
* - `environment.prod.ts` (modo producción)

Esto permite cambiar configuraciones según el entorno de ejecución.

#### 📄 `index.html`

Es el archivo HTML principal de la aplicación. Angular inyecta aquí el código de la aplicación generada.

#### 📄 `main.ts`

Es el punto de entrada de la aplicación. Se encarga de \*\*bootstrapear\*\* (inicializar) el módulo principal `AppModule`.

```ts

import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';

platformBrowserDynamic().bootstrapModule(AppModule);

```

#### 📄 `polyfills.ts`

Contiene configuraciones para soportar navegadores más antiguos.

#### 📄 `styles.scss`

Archivo global de estilos de la aplicación. Aquí se pueden importar fuentes, variables SCSS o estilos globales.

#### 📄 `test.ts`

Archivo de configuración para pruebas unitarias con Karma y Jasmine.

---

## 3. Archivos de Configuración del Proyecto

### 📄 `angular.json`

Archivo de configuración principal del proyecto. Define opciones como:

* - \*\*Configuración de build y compilación.\*\*
* - \*\*Configuración de servidores y proxies.\*\*
* - \*\*Configuración de estilos globales.\*\*

### 📄 `package.json`

Contiene la lista de dependencias y scripts npm. Ejemplo:

```json

"dependencies": {

"@angular/core": "^16.0.0",

"rxjs": "^7.0.0"

},

"scripts": {

"start": "ng serve",

"build": "ng build",

"test": "ng test"

}

```

### 📄 `tsconfig.json`

Archivo de configuración de TypeScript. Define reglas como:

* - Versión de ECMAScript (`target: ES6`)
* - Configuración de módulos (`module: CommonJS`)
* - Rutas de salida (`outDir: ./dist`)

### 📄 `README.md`

Archivo de documentación inicial del proyecto, útil para describir la aplicación y cómo ejecutarla.

---

## Conclusión

Comprender la estructura de un proyecto Angular facilita el desarrollo, mantenimiento y escalabilidad. Cada carpeta y archivo tiene una función específica que permite organizar el código de manera eficiente y estructurada.