# Configuración del Entorno para Desarrollar con Angular

Para desarrollar aplicaciones con Angular, es fundamental configurar correctamente el entorno de desarrollo. En este documento, abordaremos la instalación y configuración de \*\*Node.js\*\*, \*\*Angular CLI\*\* y \*\*TypeScript\*\*, asegurando que los desarrolladores tengan todo lo necesario para comenzar con Angular sin problemas.

---

## 1. Instalación de Node.js

### ¿Por qué es necesario Node.js?

Node.js es un entorno de ejecución para JavaScript del lado del servidor. Angular utiliza Node.js para manejar su sistema de paquetes (npm) y ejecutar herramientas como Angular CLI.

### Descargar e instalar Node.js

1. Ir a la página oficial de [Node.js](https://nodejs.org/).

2. Descargar la última versión LTS (Long-Term Support) recomendada para estabilidad.

3. Instalar el paquete siguiendo las instrucciones del instalador.

4. Verificar la instalación ejecutando en la terminal:

```sh

node -v

```

```sh

npm -v

```

Estas líneas mostrarán las versiones instaladas de Node.js y npm respectivamente.

### Configuración de npm

Para evitar problemas de permisos con npm, se recomienda actualizar el gestor de paquetes y configurar el uso de paquetes globales:

```sh

npm install -g npm@latest

```

Opcionalmente, configurar una carpeta global para npm:

```sh

mkdir ~/.npm-global

npm config set prefix '~/.npm-global'

export PATH=~/.npm-global/bin:$PATH

```

---

## 2. Instalación de Angular CLI

### ¿Qué es Angular CLI?

Angular CLI (Command Line Interface) es la herramienta oficial para la gestión y automatización de proyectos Angular. Facilita la creación de componentes, servicios, módulos, la compilación y el despliegue de aplicaciones.

### Instalación de Angular CLI

Ejecutar el siguiente comando en la terminal:

```sh

npm install -g @angular/cli

```

Para verificar la instalación:

```sh

ng version

```

Esto mostrará la versión de Angular CLI instalada junto con la versión de Angular y otros paquetes relacionados.

### Actualización de Angular CLI

Para mantener Angular CLI actualizado:

```sh

npm update -g @angular/cli

```

---

## 3. Instalación y Configuración de TypeScript

### ¿Por qué TypeScript?

Angular utiliza \*\*TypeScript\*\*, un superconjunto de JavaScript que añade tipado estático y otras características avanzadas, mejorando la mantenibilidad del código.

### Instalación de TypeScript

Ejecutar:

```sh

npm install -g typescript

```

Para verificar la instalación:

```sh

tsc -v

```

Esto mostrará la versión instalada de TypeScript.

### Configuración de TypeScript

Es recomendable crear un archivo `tsconfig.json` en el proyecto para definir configuraciones clave. Un ejemplo básico de configuración:

```json

{

"compilerOptions": {

"target": "ES6",

"module": "CommonJS",

"strict": true,

"outDir": "./dist",

"rootDir": "./src"

}

}

```

#### Explicación de cada propiedad:

* - \*\*`target`: "ES6"\*\* → Define la versión de ECMAScript a la que se compilará el código TypeScript. `ES6` permite el uso de características modernas de JavaScript, como `let`, `const`, y funciones flecha.
* - \*\*`module`: "CommonJS"\*\* → Especifica el sistema de módulos a utilizar. `CommonJS` es común en entornos Node.js y define cómo se importan y exportan módulos en la aplicación.
* - \*\*`strict`: true\*\* → Activa el modo estricto en TypeScript, lo que obliga a escribir un código más seguro, con validaciones más estrictas sobre el tipado y uso de variables.
* - \*\*`outDir`: "./dist"\*\* → Especifica la carpeta donde se generarán los archivos JavaScript compilados a partir del código TypeScript.
* - \*\*`rootDir`: "./src"\*\* → Define la carpeta raíz donde se encuentran los archivos TypeScript del proyecto. Esto ayuda a organizar el código fuente.

Estas configuraciones permiten que el código sea más eficiente, seguro y compatible con las mejores prácticas de desarrollo en Angular.

---

## 4. Creación de un Proyecto Angular

Una vez configurado el entorno, podemos crear un proyecto Angular con:

```sh

ng new mi-proyecto-angular

```

Este comando:

* - Crea la estructura del proyecto.
* - Instala las dependencias necesarias.
* - Configura TypeScript, Webpack y otros archivos importantes.

Para ejecutar la aplicación:

```sh

cd mi-proyecto-angular

ng serve

```

Esto iniciará el servidor de desarrollo en `http://localhost:4200/`.

---

## Conclusión

Siguiendo estos pasos, los desarrolladores tendrán un entorno completamente funcional para trabajar con Angular. Con Node.js, npm, Angular CLI y TypeScript correctamente instalados y configurados, estarán listos para empezar a construir aplicaciones escalables y modernas con Angular.