

Angular 2

Ángel Villalba Fdez-Paniagua

Table of Contents

1. Angular CLI	1
1.1. Instalación	1
1.2. Creación de contenido	1
1.3. Ejecución	2
1.4. Producción	2
1.5. Generar Elementos de Angular	2
1.6. Testing	3
1.7. Añadir librerías externas	4
1.8. Actualizar proyecto y librerías	4
1.9. Estructura del proyecto	4

1. Angular CLI



Figure 1. Angular cli

Angular CLI es el intérprete de línea de comandos de Angular 2, facilita el inicio de proyectos y la creación del esqueleto, o scaffolding, de la mayoría de los componentes de una aplicación Angular 2.

Angular CLI no es una herramienta de terceros, sino que nos la ofrece el propio equipo de Angular. En resumen, nos facilita mucho el proceso de inicio de cualquier aplicación con Angular, ya que en pocos minutos te ofrece el esqueleto de archivos y carpetas que vas a necesitar, junto con una cantidad de herramientas ya configuradas. Además, durante la etapa de desarrollo nos ofrecerá muchas ayudas, generando el "scaffolding" de muchos de los componentes de una aplicación. Durante la etapa de producción o testing también nos ayudará, permitiendo preparar los archivos que deben ser subidos al servidor, transpilar las fuentes, etc.

1.1. Instalación

Para poder instalarlo necesitamos tener Node y npm instalados. La manera mas sencilla de empezar es descargarnos e instalar la distribución para nuestro sistema operativo de [Node](#), que viene ya con su manejador de paquetes npm.

Para instalar Angular cli, desde la terminal. Una vez instalado, desde la línea de comandos podremos usar el comando `ng`. Para comprobar que todo ha ido bien, ejecutamos `ng -v` (o `ng v` en las últimas versiones).

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

```
$ ng -v
```

```
$ ng v
```

1.2. Creación de contenido

Creamos un directorio para nuestra aplicación y ejecutamos

```
$ ng new hola-angular2
```

Después de unos segundos tendremos nuestro proyecto creado y listo para ejecutarse.

Angular cli nos crea un proyecto bastante mas grande que si lo hacemos nosotros a mano, además añade test, pero en general es el mismo proyecto de Angular 2.

1.3. Ejecución

Para lanzar y probar tu aplicación necesitas otro comando de Angular-CLI. Este comando se ocupa entre otras cosas de todo el proceso necesario para transformar el código TypeScript en JavaScript reconocible por el navegador. También crea un mini servidor estático y además refresca el navegador a cada cambio los fuentes.

```
$ ng serve
```

1.4. Producción

Para la fase de producción necesitaremos los archivos del proyecto (componentes, servicios, directivas...) transpilados a JavaScript, minificados para que el proyecto ocupe el menor espacio posible, se encarga de eliminar el código que es inaccesibles (aquellos que no se llegan a ejecutar nunca en la aplicación).

Para generar estos archivos, y así preparar la aplicación para subir a producción, lanzaremos el siguiente comando:

```
$ ng build --prod
```

1.5. Generar Elementos de Angular

Podemos generar funcionalidades como componentes, servicios, directivas... en nuestro proyecto usando los comandos que nos proporciona Angular-CLI.

Crear un componente:

```
$ ng generate component nombre-componente  
$ ng g c nombre-componente
```

Crear una directiva

```
$ ng generate directive nombre-directiva  
$ ng g d nombre-directiva
```

Crear un pipe

```
$ ng generate pipe nombre-pipe  
$ ng g p nombre-pipe
```

Crear un servicio:

```
$ ng generate service nombre-servicio  
$ ng g s nombre-servicio
```

Crear una clase

```
$ ng generate class nombre-clase  
$ ng g cl nombre-clase
```

Crear un guard

```
$ ng generate guard nombre-guard  
$ ng g g nombre-guard
```

Crear un interface

```
$ ng generate interface nombre-interface  
$ ng g i nombre-interface
```

Crear un enum

```
$ ng generate enum nombre-enum  
$ ng g e nombre-enum
```

Crear un módulo

```
$ ng generate module nombre-module  
$ ng g m nombre-module
```

A esos comandos les podemos añadir las siguientes opciones:

- **-it**: no genera archivo de plantilla.
- **-is**: no genera archivo de estilos.
- **--flat**: genera los archivos en la carpeta actual en lugar de en una nueva carpeta.
- **--spec false**: no genera los archivos de testing (**.spec**) para versiones de AngularCLI < 7.
- **--skipTests**: no genera los archivos de testing (**.spec**) a partir de AngularCLI 7.

1.6. Testing

Angular CLI nos permite ejecutar los archivos de testing a través de comandos.

Si queremos ejecutar los **test unitarios**, es decir, todos aquellos que se crean por defecto con la extensión **.spec.ts**, entonces vamos a lanzar el comando:

```
$ ng test
```

Si queremos ejecutar los **test e2e**, es decir, aquellos que se han definido dentro de la carpeta **e2e** o que tienen la extensión **.e2e-spec.ts**, entonces lanzaremos el comando:

Test End to End

```
$ ng e2e
```

1.7. Añadir librerías externas

Para añadir una librería externa a Angular como puede ser *Bootstrap* o *Angular Material*, había que usar *NPM* o *Yarn* para instalarlas y después añadir los archivos fuente instalados en el archivo **angular.json**.

A partir de Angular 6 se puede usar un nuevo comando de Angular CLI que nos permite **añadir las librerías externas** (aquellas que están soportadas) instalándolas y añadiendo las referencias necesarias al archivo mencionado antes.

```
$ ng add nombre-libreria
```

1.8. Actualizar proyecto y librerías

Además del comando anterior, se ha añadido otro más que se encarga de analizar el ``package.json`` en busca de paquetes que no estén actualizados a la última versión disponible. Una vez lanzado y analizado dicho archivo, nos va a mostrar una lista con aquellos paquetes que deberíamos de actualizar y nos mostrará los comandos para ello.

```
$ ng update
```

A la hora de actualizar de versión las aplicaciones de Angular, nos puede venir bien echar un ojo a la página <https://update.angular.io/> en la que al rellenar la información que se nos pide sobre el proyecto, nos muestra una serie de pasos que debemos de seguir para que la actualización salga bien.

1.9. Estructura del proyecto

Al crear un proyecto de Angular 2, nos crea una carpeta que contiene la siguiente estructura de ficheros:

- **e2e**: aquí se encuentran los archivos de testing End to End, que se ejecutan con la ayuda de Protractor y Jasmine.

- **node_modules**: aquí se encuentran todas las dependencias de nuestro proyecto.
- **src**: es posiblemente la carpeta más importante del proyecto, aquí se encuentra todo el código de la aplicación, y es en la carpeta que pasaremos la mayor parte del tiempo.
 - **app**: en esta carpeta vamos a tener todos los componentes, servicios, plantillas y resto de elementos de la aplicación.
 - **app.module.ts**: este archivo se encarga de entender que componentes y dependencias tenemos en el proyecto.
 - **assets**: aquí guardaremos los assets que no pertenezcan a los componentes de Angular, como imágenes, sonidos...
 - **environments**: en esta carpeta tendremos los archivos encargados de indicar los entornos de trabajo (producción, desarrollo...).
 - **index.html**: este es el archivo donde se inicia la SPA, donde se van a cargar los componentes.
 - **main.ts**: es el primer archivo en ejecutarse, y en el se encuentran los parámetros de configuración de la aplicación y del entorno en el que trabajaremos...
 - **polyfills.ts**: asegura la compatibilidad con los distintos navegadores.
 - **styles.css**: aquí definimos los estilos globales que vamos a darle a la aplicación.
 - **test.ts**: es el punto de entrada a los test unitarios, los haremos con Jasmine.
- **angular-cli.json**: aquí se encuentra la configuración de Angular-CLI.
- **tsconfig.json**: en este archivo se encuentra la configuración de TypeScript, donde se indica como hacer la compilación a JavaScript.
- **karma.conf.js**: aquí se encuentra la configuración de los test unitarios.
- **package.json**: aquí se indican las dependencias del proyecto, las acciones a ejecutar cuando se lanza el proyecto, y alguna información del proyecto.
- **protractor.conf.js**: aquí se encuentra la configuración de los test End to End.

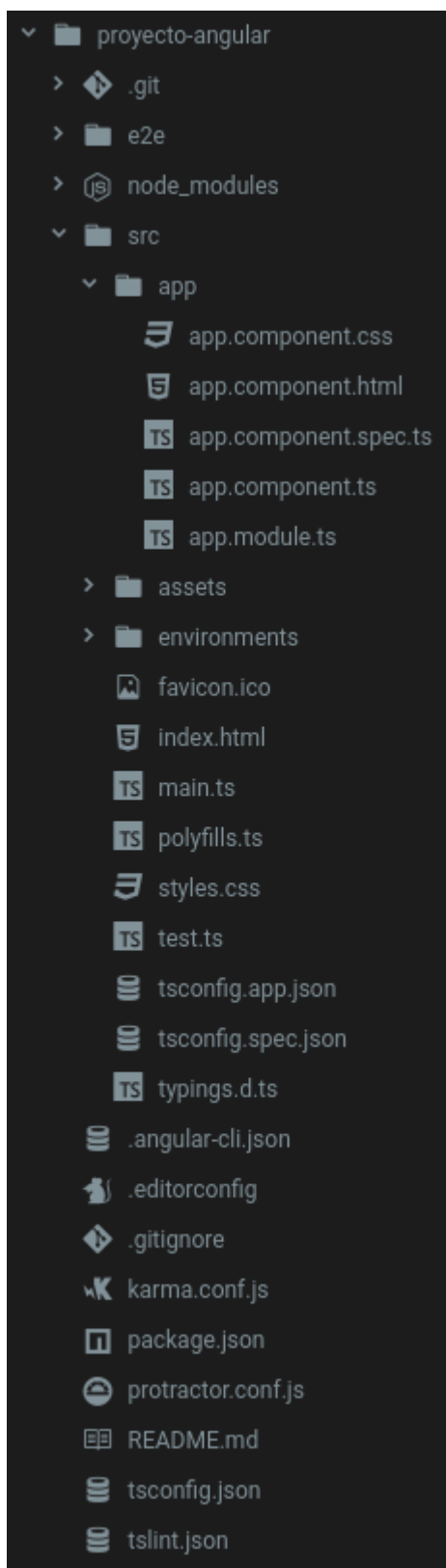


Figure 2. Estructura de proyecto en Angular 2