0 - HELLO

> Hola Angular CLI

Angular en su versión 7 es la plataforma perfecta para el desarrollo profesional de aplicaciones modernas. **El Angular CLI es la herramienta** adecuada para generar aplicaciones Angular. Juntos son imbatibles en cuanto a velocidad en desarrollo y a potencia en ejecución.

El comúnmente conocido como **AngularCLI** o *angular cli* o *angular-cli* o _ CLI a secas_ es la herramienta de línea de comandos estándar para **crear, depurar y publicar aplicaciones Angular**. En su actual **versión 7** es más potente y versátil que nunca y es muy sencillo dominar los aspectos básicos.

> Código asociado a este artículo en *GitHub*: AcademiaBinaria/angular-board/

1. Instalación de Angular CLI 7

Angular es una plataforma de desarrollo dogmática y llave en mano. Para empezar, como en casi cualquier desarrollo **necesitarás NodeJS y su manejador de de paquetes *npm***. Tenerlos actualizados es un mandamiento básico para un desarrollador web. Comprueba tu versión con el siguiente comando:

```
node -v
```

Empieza con una **instalación global** que te permita usar la herramienta desde cualquier directorio. Comprueba la versión instalada y accede a la ayuda en línea. La ayuda está disponible tanto de modo general como para cada comando que vayas a usar.

Instrucciones para instalar Angular CLI.

```
$ npm i -g @angular/cli@latest
$ ng version
$ ng help
$ ng new --help
```

2. Crear y ejecutar una aplicación Angular 7

Una vez que hayas instalado el CLI de manera global ya puedes empezar a usarlo en tu directorio de trabajo. El primer comando será `ng new` que te va a **generar toda una aplicación funcional** y las configuraciones necesarias para su depuración, pruebas y ejecución.

Como novedad, en las últimas versiones, el CLI te preguntará por algunas opciones para crear tu aplicación. Eso es porque no todos los desarrollos son iguales. Se puede configurar el tipo, el estilo y muchas más cosas, tanto de forma interactiva como mediante opciones en línea de comandos.

Lo más habitual es usar la configuración que viene por defecto, pero también se pueden crear soluciones a medida. Te muestro unos ejemplos para que pruebes y te familiarices con la herramienta. Para más información mira la documentación del comando ng new.

2.1 Normal

```
ng new normal
cd normal
npm start
```

2.2 Minimalista

```
ng new minimalista -s -S -t
cd minimalista
npm start
```

2.3 Profesional

```
ng new profesional -p acme --routing true
cd profesional
npm start
```

2.4 Empresarial

```
ng new empresarial --create-application false
cd empresarial
ng generate application compras -p acme --routing true
npm start
```

Una vez finalizada la instalación de todas las librerías necesarias puedes bajar a la carpeta recién creada y ejecutar el comando standard de *npm* para el arranque de cualquier aplicación: `npm start`. Si todo va bien, en unos segundo podrás visitar http://localhost:4000 para ver en marcha la aplicación. Lo veremos con más detalle en el apartado de configuración.

> ejemplo: La aplicación que sirve de ejemplo a este tutorial fue creada con este comando:

```
ng new angular-board --routing true -s -S
```

3. Estructura de una aplicación Angular

Una vez generada y comprobada la ejecución, toca estudiar cómo es la estructura de la aplicación. Para ello revisa carpeta a carpeta. Las malas noticias son que hay **una enorme cantidad de ficheros y carpetas**, las buenas son que como verás, casi todo es **configuración e infraestructura**.

3.1 Visual Studio Code

Para ver y editar los ficheros te vale cualquier editor de código, pero yo uso y te recomiendo VSCode. Es un **gran editor, gratuito y multiplataforma**. Viene con un terminal integrado y puedes mejorarlo instalando extensiones desde una galería del propio editor.

Te recomiendo instalar un paquete de extensiones ya configurado y preparado para el desarrollo de aplicaciones con *Angular*, se llama Angular Essentials. Con eso y el Material Icon Theme verás *Angular en colores*.

3.2 Carpetas y Ficheros principales

Volviendo a la **estructura de ficheros y carpetas** te encontrarás con muchos archivos de distintos tipos. Si eres completamente nuevo en Angular, te llamará la atención las extensiones `.ts`. Son para ficheros *TypeScript*, una evolución del *JavaScript* con facilidades para el programador. Por ahora sólo tienes que familiarizarte con estos:

- **angular.json** *: configuración del propio CLI. La madre de todos los configuradores*
- **package.json** *: dependencias de librerías y scripts*
- **src/** *: la carpeta donde están los archivos fuentes*
    - **index.html** *: un fichero HTML índice estándar*
    - **main.ts** *: fichero TypeScript de arranque de la aplicación*
    - **app/** *: la carpeta con el código específico de tu aplicación*
        - **app.module.ts** *: la aplicación es un árbol de módulos, y este es su raíz*
        - **app.component.ts** *: la página es un árbol de componentes, y este es su raíz*
        - **app.component.html** *: el componente tiene una parte visual, esta es su vista*

Echa un vistazo a estos ficheros, pronto los modificaremos para sentirnos programadores.

4. Edición

Angular CLI instala y configura un conjunto de herramientas que te harán la vida más fácil. Entre otras, destaca la capacidad de **recargar la aplicación en caliente** en cuanto guardas tu trabajo como programador. En esta última versión, la 7, se ha mejorado el proceso y es realmente rápido.

Para probarlo sólo tienes que dejar arrancada la aplicación con el comando `npm start`; **cambiar un fichero de código y comprobar el resultado** en el navegador. Te propongo empezar como en cualquier otro lenguaje; por el famoso *hola mundo*.

4.1 Hola Mundo

Abre el fichero `app.component.ts` y busca dentro de él una clase llamada `AppComponent`. Encontrarás que tiene una propiedad `title`. Asígnale el saludo de rigor: `title = 'actibot: hello world ;-)';`. Guarda y comprueba cómo tu navegador **se habrá actualizado automáticamente**.

Toda **esta magia depende de una cadena de comandos** que lanzan distintas herramientas previamente instaladas y configuradas por el CLI. Entre ellas cabe mencionar a *WebPack*, un coloso que afortunadamente viene instalado y preparado para funcionar de manera transparente.

4.2 npm start

Esta es una lista no exhaustiva de lo que sucede.

1. npm start
2. ng serve
3. webpack server en http://localhost:4270
    1. vigilancia de cambios sobre la carpeta src/
    2. live reload
    3. compilado de la aplicación
    4. recarga del navegador

Cambia a tu antojo el `app.component.ts` o su hermano `app.component.html` y comprueba los cambios de manera inmediata en tu navegador.

5. Configuración

El CLI 7 viene con pilas incluidas, se puede usar desde el primer momento. Sólo quedan pequeñas mejoras que hacer. Por ejemplo ajustar el `package.json` y agregar librerías de terceros.

5.1 Package.json

El `package.json` es el fichero estándar de *npm* donde se almacenan las **dependencias de terceros**. Contiene las librerías que necesita la aplicación para ejecutarse, por ejemplo todas las de *Angular 7*. Y también las herramientas que necesita el programador, por ejemplo el propio *AngularCLI*;

```
{
  "dependencies": {
    "@angular/core": "^7.2.0"
  },
  "devDependencies": {
    "@angular/cli": "7.2.0"
  }
}
```

Otro uso del `package.json` es servir de **contenedor de scripts** para automatizar tareas de operaciones rutinarias. Por ejemplo, el comando estándar `npm start` ejecutará el contenido asignado en el fichero *json*, originalmente `ng serve`. Esto lanza el servidor de pruebas con sus opciones por defecto.

Pero el **comando ng serve** admite muchas configuraciones. Te propongo que uses esta para activar un modo de compilación más rápido y seguro, y para que se abra el navegador de forma automática en cuanto lances el servidor. Y además te recomiendo que uses un puerto específico para cada aplicación.

Otras configuraciones en producción

```
{
  "start": "ng serve --aot -o --port 4270"
}
```

5.2 Estilos y librerías de terceros

Las librerías que vienen de fábrica tienen todo lo necesario para crear aplicaciones. Pero raro es el caso en que no necesitemos **algún que otro producto de terceros**. Ya sean utilidades como *date-fns*, librerías gráficas como *chart.js* o la aplicación de estilos y componentes visuales de *frameworks como Bootstrap o MaterialDesign*. Pero todos se instalan de igual forma. Descargándolos con *npm* y adjuntándolos en el `angular.json`.

> En este tutorial te propongo usar una hoja de estilos muy simple que mejora la apariencia de cualquier aplicación sin necesidad de usar clases propias. Se llama *MiniCSS* y es apropiada para prototipos, pruebas o pequeños proyectos.

Se descargan e instalan de manera estándar.

```
npm install mini.css --save
```

Para que se incluyan en la distribución hay que ir a la configuración del *CLI*. Entonces se agrega dentro del fichero `.angular.json` a la colección de *styles* o de *scripts* que corresponda.

```
{
  "styles": ["src/styles.css", "./node_modules/mini.css/dist/mini-
default.min.css"]
}
```

Estas colecciones de archivos los usa el *cli* a través de *webpack* para incluirlos **minificados y concatenados en un fichero *bundle* sustituyendo a las clásicas etiquetas html**.Todo, el html y sus estilos, se construirá en el cliente a partir de instrucciones JavaScript. De esta forma el fichero `index.html` apenas tendrás que tocarlo, salvo para algunas etiquetas de meta información.

```
<meta name="description" content="A sample project for learning Angular ;-)" />
<meta name="keywords" content="Angular Sample Tutorial Ejemplo" />
<meta name="author" content="Alberto Basalo" />
```

Una cosa más, los cambios en los ficheros de configuración no se auto recargan. Tienes que parar la servidor y volver a lanzarlo para apreciar el estilo *MiniCSS*.

5.3 Environments

La carpeta `environments/` contiene dos ficheros, y puede contener más, para cada entorno de distribución necesario. En código siempre importaremos el fichero base, pero durante la compilación el CLI lo sustituirá por el adecuado.

```
title = environment.appName + 'hello world ;-)';
```

5.4 Assets

Los ficheros de la carpeta `assets/` se copian tal cual al despliegue. Es un buen lugar para logos, imágenes y ficheros de datos estáticos.

Por ejemplo, en el `app.component.html` he sustituido la imagen incrustada del Angular con un logo propio.

```
<img width="100" src="./assets/logo.png" />
```

6. Angular 7, el CLI 7 y su ecosistema

Algunos consejos y herramientas útiles que rodean al mundo Angular.

Pero si la línea de comandos te suena muy antigua y lo tuyo son las interfaces gráficas, estás de suerte. El reciente proyecto Angular Console te permite generar y ejecutar comandos desde una cómoda interfaz gráfica.

Una mejora digna de mención es la capacidad de presupuestar tamaños de los ficheros generados. De esa forma podemos controlar el peso y tiempo de descarga de la aplicación.

Para garantizar la limpieza del código conviene usar herramientas como Prettier y configurarlas para su compatibilidad con Angular.

Los últimos toques antes de publicar pueden incluir el *script de analytics* en el `index.html` y ajustes de retro-compatibilidad en el fichero `polyfills.ts`. Luego un comando y listo para publicar en *github pages*.

> Te recomiendo que te familiarices y uses mucho estos *scripts* en el `package.json` para poder lanzarlos más tarde.

```
{
  "scripts": {
    "build:prod": "ng build --prod",
    "build:pub": "ng build --prod --output-path docs --base-href
https://academiabinaria.github.io/angular-board/",
    "e2e": "ng e2e",
    "http-server": "http-server ./dist/angular-board/ -c-1 -p4271 -a localhost -
o",
    "lint": "ng lint",
    "ng": "ng",
    "pub": "npm run build:pub && npm run push",
```

```
    "push": "git add * && git commit -m 'pub' && git push",
    "start:prod": "npm run build:prod && npm run http-server",
    "start": "ng serve --aot -o --port 4270",
    "test": "ng test"
  }
}
```

Comprueba las ejecuciones de los distintos *scripts*. Con `npm start` no se generan ficheros físicos. Todos es en memoria para mayor velocidad de re-compilación mientras desarrollas. En cambio `npm run build:prod` creará una carpeta `./dist/angular-board` en la que dejará los archivos necesarios para ejecución. Por último `npm run pub` los prepara para enviar compilados a la carpeta estándar `./docs` listos para publicarse en las *github pages*.

Otros enlaces de interés sobre el ecosistema Angular.

- Extensiones Esenciales
- Prettier
- Angular Console
- Angular Material
- Bootstrap
- Augury
- Apollo GraphQL
- Ionic
- Angular Console

Para complementar tu conocimiento te recomiendo la documentación de Angular/CLI y este artículo avanzado que trata la configuración del CLI en profundidad Angular CLI under the hood