

Ejercicios del tema 2

Tenemos que realizar una serie de comprobaciones previas antes de comenzar con nuestros proyectos en Angular. El primer paso es instalar NodeJS dentro de nuestra máquina. Podemos hacerlo a través de la propia web de Node.

<https://nodejs.org/es/>

Node.js® es un entorno de ejecución para JavaScript construido con el **motor de JavaScript V8 de Chrome**. Node.js usa un modelo de operaciones E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente. El ecosistema de paquetes de Node.js, **npm**, es el ecosistema mas grande de librerías de código abierto en el mundo.

Important security releases, please update now!

Descargar para macOS (x64)

8.9.4 LTS

Recomendado para la mayoría

9.3.0 Actual

Últimas características

[Otras Descargas](#) | [Cambios](#) | [Documentación del API](#)

[Otras Descargas](#) | [Cambios](#) | [Documentación del API](#)

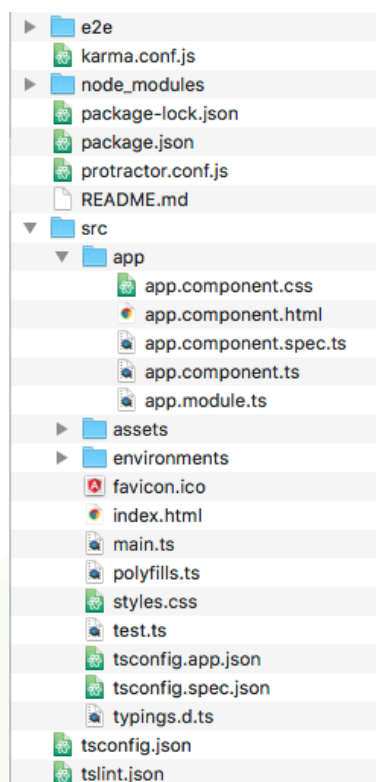
Para comprobar si hemos instalado correctamente Node, podemos ejecutar el comando **node -v** dentro de nuestro terminal.

Para poder compilar el código TypeScript y convertirlo en código Javascript legible por la mayoría de navegadores, debemos instalar la herramienta **tsc**. En próximos temas, veremos más en profundidad el uso de esta herramienta.

npm install -g typescript

Dentro de los recursos del tema se puede encontrar el fichero llamado **Recurso tema 2.zip**, el cual incluye la estructura para la creación de una aplicación sencilla con Angular. Se puede revisar

la estructura de directorios de la aplicación, para comprobar que contiene todos los ficheros que se han analizado en el tema.



Se trata de una aplicación formada por un único componente (app.component), dentro del cual, definimos el bloque de código que vamos a mostrar. La creación de este componente, la encontramos dentro del fichero raíz (index.html).

Se especifica un fichero concreto **main.ts** para arrancar nuestra aplicación, ya que podría ser que dentro de nuestro mismo proyecto se quisiera trabajar con aplicaciones destinadas a ejecutarse en dispositivos diferentes. En nuestro caso, vamos a trabajar con el navegador.

El resto de ficheros, definen las configuraciones que vamos a aplicar a los diferentes elementos de nuestro proyecto.

El primer paso que deberíamos dar para poder arrancar la aplicación sería el de obtener las dependencias definidas dentro de **package.json**. Las dependencias se instalan automáticamente creando el proyecto con angular-cli.

No obstante si tuviéramos el proyecto base podríamos añadir las dependencias situándonos en el directorio de nuestra aplicación, ejecutando el comando:

npm install

El resultado de esta ejecución es la generación del directorio **node_modules** con todas las librerías necesarias para nuestro proyecto. Una vez tengamos todos los requisitos y dependencias instaladas, podemos ejecutar nuestra aplicación a través del comando

npm start

Este comando se encarga de compilar todos aquellos ficheros de tipo TypeScript que contenga nuestro proyecto. Además arranca una instancia de nuestra aplicación para que podamos acceder a ella a través del navegador.

Si todo ha ido correctamente, se debe abrir una ventana del navegador con el resultado de nuestra aplicación:

Prueba a realizar cambios en el **template** de la clase AppComponent (app.component.ts) para ver como, en tiempo real, sin detener la aplicación, se visualizan los cambios sobre tu navegador.

En los próximos temas, iremos profundizando en los diferentes aspectos que conforman nuestras aplicaciones Angular.

```
Mac-mini-de-raul:prueba raul$ npm start

> prueba@0.0.0 start /Users/raul/Documents/prueba
> ng serve

** NG Live Development Server is listening on localhost:4200, open your browser
on http://localhost:4200 **
Hash: 47f5d7ecc18467efc669
Time: 13618ms
chunk {0} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 160 kB {4}
[initial] [rendered]
chunk {1} main.bundle.js, main.bundle.js.map (main) 5.27 kB {3} [initial] [re
ndered]
chunk {2} styles.bundle.js, styles.bundle.js.map (styles) 10.5 kB {4} [initia
l] [rendered]
chunk {3} vendor.bundle.js, vendor.bundle.js.map (vendor) 2.18 MB [initial] [
rendered]
chunk {4} inline.bundle.js, inline.bundle.js.map (inline) 0 bytes [entry] [re
ndered]
webpack: Compiled successfully.
```

Para estos ejercicios y los siguientes temas se recomienda subir el proyecto comprimido sin la carpeta node_modules.

