

Desarrollo de aplicaciones web en el cliente con Angular



Autores

Miguel Ángel García Picó
Rafael Serrano Aracil
Jose Francisco Vera Girona

Curso 2020-21

Universidad de Alicante
Grado en Ingeniería Informática
Aplicaciones Distribuidas en Internet

Índice

Introducción	2
¿Qué es Angular?	2
Conceptos de Angular	3
Módulos	3
Componentes	3
Servicios e inyección de dependencias	3
Inicio rápido	4
Preparación del entorno	4
Creación de un proyecto	4
Desarrollando una aplicación con Angular	6
Introducción	6
Un vistazo a la estructura del proyecto	6
Creando los componentes	7
Navegación de la aplicación	7
Definición de servicios	7
Seguir aprendiendo Angular	8
Bibliografía	9

Introducción

Para este trabajo en grupo, hemos decidido realizar un pequeño tutorial sobre el framework para desarrollo de aplicaciones web Angular. Este documento presenta algunos conceptos básicos del mismo y se complementa con un vídeo en el que desarrollamos paso a paso una aplicación y el código de dicha aplicación.

Los enlaces para el vídeo y el código son, respectivamente:

<https://youtu.be/yYcF6Sfelxk>

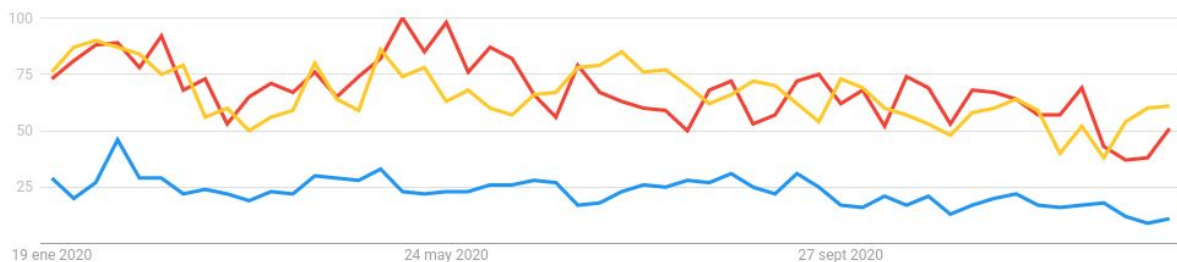
<https://github.com/serranorafa/adi-angular-app>

¿Qué es Angular?

Angular es un framework de diseño de aplicaciones y una plataforma de desarrollo para crear aplicaciones de página única (SPA) eficientes y complejas desarrollado en TypeScript y mantenido por Google con la idea de facilitar el desarrollo de aplicaciones y sus pruebas. Con una única forma de construir aplicaciones permite su despliegue para cualquier plataforma objetivo entre PC y dispositivos móviles, tanto como aplicaciones web como nativas.



Ahora mismo Angular está a la vanguardia junto a React en frameworks del mismo estilo, según nos indica la gráfica de tendencias de Google de los últimos 12 meses siendo azul Vue, rojo Angular y amarillo React:



Para empezar a trabajar con Angular y comprender sus virtudes será preferible tener conocimientos previos de CSS, HTML y TypeScript.

Conceptos de Angular

Antes de empezar, tenemos que comprender una serie de conceptos en los que se apoya Angular para la organización de su código y su funcionamiento:

Módulos

Los `NgModule` son la versión de Angular de los módulos de JavaScript. Permiten establecer un conjunto que una los componentes que estén relacionados entre sí.

Estos módulos pueden llamar a otros módulos para importar funcionalidades, permitiendo la organización y la gestión de aplicaciones complejas y diseñando en torno a la reusabilidad. Esto permite utilizar `lazy-load` para cargar módulos bajo demanda aligerando así la velocidad de arranque de la aplicación.

Cada aplicación tiene un módulo raíz llamado `AppModule`.

Componentes

Los componentes de la aplicación definen una clase con datos de aplicación y lógica, además de una plantilla HTML que tiene etiquetas personalizadas que permiten preparar la página antes de servirla accediendo a los datos del programa (`property binding`). También permite la actualización de datos del programa mediante decisiones del usuario (`event binding`), tratándose así de un enlazado de datos en doble sentido.

Así, cada componente está definido por tres archivos:

- `nombre.component.ts`, que define el nombre a través del que otros componentes podrán insertarlo en sus plantillas, la plantilla asociada al componente y su hoja de estilos, si la tuviera.
- `nombre.component.css`, que es la hoja de estilos del componente.
- `nombre.component.html`, que es la plantilla del componente.

Servicios e inyección de dependencias

Aún si los componentes ofrecen la posibilidad de definir la lógica de la aplicación en ellos, lo ideal es delegar toda funcionalidad no relacionada con la vista en una clase servicio.

Una clase servicio es una clase que tiene un propósito claro y es compartida por los componentes de la aplicación. Se puede mejorar aún más la reusabilidad de estas clases permitiendo la inyección de los proveedores correspondientes para que funcione bien bajo diferentes circunstancias.

Inicio rápido

Preparación del entorno

En primer lugar, necesitamos disponer del entorno de desarrollo de Javascript Node.js. Podemos comprobar si está instalado y su versión con la orden:

```
$ node -v
```

En caso de no tenerlo instalado, lo podemos instalar (en Ubuntu) con:

```
$ sudo apt install nodejs
```

Haremos lo propio con npm, el gestor de paquetes de Node que usaremos para gestionar las dependencias necesarias para nuestros proyectos:

```
$ npm -v  
$ sudo apt install npm
```

Por último, necesitaremos la interfaz de línea de comandos Angular CLI, la cual utilizaremos para inicializar, desarrollar y mantener aplicaciones de Angular desde la terminal:

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

Ahora podremos invocar esta herramienta con el ejecutable ng:

```
$ ng help
```

Creación de un proyecto

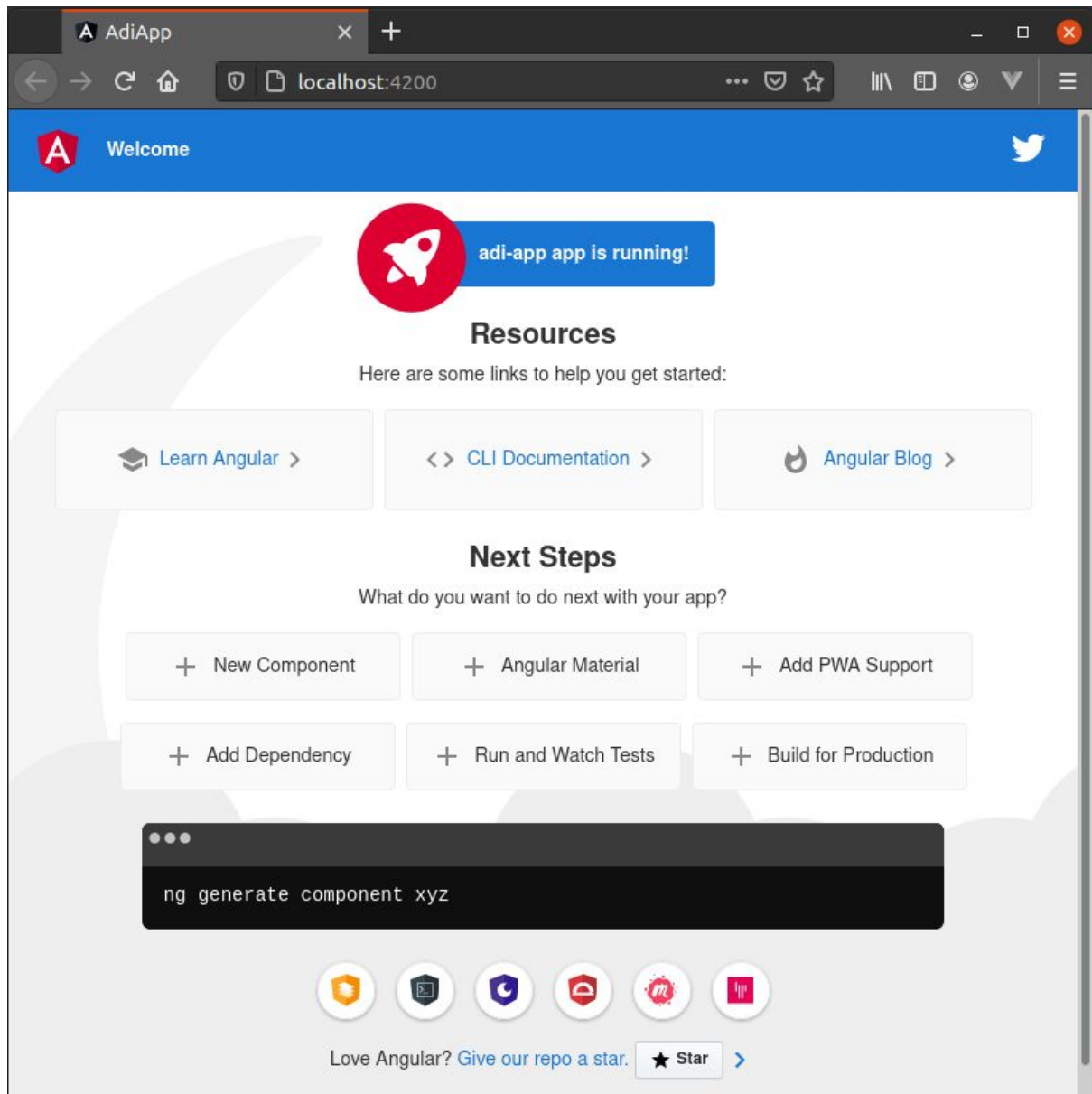
Para crear un nuevo espacio de trabajo y una aplicación que nos servirá de plantilla utilizamos Angular CLI:

```
$ ng new adi-app
```

Cuando nos pregunte si queremos utilizar “Angular routing” le diremos que sí, el resto de opciones las dejaremos por defecto. Angular CLI se encarga de instalar los paquetes npm y las dependencias necesarias. Una vez hecho esto, podemos ejecutar esta aplicación inicial desde el directorio del proyecto:

```
$ cd adi-app  
$ ng serve -o
```

La opción -o abre directamente el navegador en la dirección `http://localhost:4200/`, donde deberíamos ver lo siguiente:



Cabe destacar que no será necesario reiniciar el servidor cada vez que hagamos cambios en los ficheros del proyecto, pues ya se encarga de hacerlo la herramienta automáticamente.

Con esto ya tenemos lo que será el esqueleto de nuestra aplicación, su estructura de directorios y los ficheros mínimos necesarios para su funcionamiento.

Desarrollando una aplicación con Angular

Para aprender sobre el funcionamiento de Angular (y, seguramente, de cualquier herramienta) la mejor forma es ponerse manos a la obra. Así que vamos a construir la típica lista de la compra, una aplicación sencilla para poder centrarnos más en las funcionalidades de Angular que en la complejidad del código. Luego, para ampliar un poco, veremos cómo consumir una API desde Angular.

En [este enlace](#) tenemos un vídeo explicativo del desarrollo de la aplicación.

El código completo se puede consultar en GitHub en [este enlace](#).

En los siguientes apartados explicamos, a modo de guía, el proceso de desarrollo de una aplicación web con Angular.

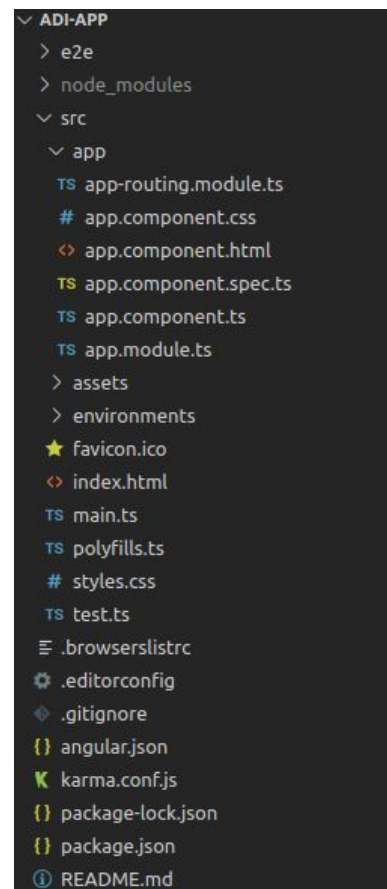
Introducción

Una vez tenemos claros los conceptos descritos anteriormente, partiremos de la aplicación básica generada con la orden `ng new`. En primer lugar, tendremos que familiarizarnos con los ficheros y directorios que se han generado.

Un vistazo a la estructura del proyecto

Los ficheros y directorios más importantes y que debemos conocer por ahora son:

- El fichero “package.json” contiene metadatos e información de las dependencias del proyecto, le da a npm la información necesaria para identificar el proyecto y sus dependencias.
- El fichero “angular.json” contiene la configuración del espacio de trabajo.
- El directorio “node_modules” contiene las dependencias de nuestra aplicación.
- El directorio app contiene el módulo raíz. Dentro de este podemos encontrar lo siguiente:
 - `app.component.ts`: el fichero del componente app.
 - `app.component.html`: la plantilla del componente app.
 - `app.component.css`: la hoja de estilo del componente.
 - `app-routing-module.ts`: módulo separado donde definiremos las rutas de nuestra aplicación.



Creando los componentes

Una aplicación en Angular se divide en componentes, que el desarrollador crea para realizar las distintas funcionalidades del proyecto. Para crear un nuevo componente se hará con la siguiente orden:

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

Una vez se ejecuta este comando, se crea una nueva carpeta dentro del directorio app con el nombre del nuevo componente. Dicho componente se dividirá en los 4 archivos con extensiones distintas: el .css contendrá el estilo del componente, el .html contendrá la plantilla, el .specs.ts tendrá dentro los tests que el desarrollador quiera probar del componente y el .ts define el apartado del script.

Navegación de la aplicación

La navegación de la aplicación se definirá, como ya se ha comentado con anterioridad, en el fichero app-routing-module.ts del directorio app. En este fichero, se define una constante llamada Routes donde a cada ruta que se defina se le asignará un componente de los existentes.

Definición de servicios

En el caso de querer que nuestra aplicación consulte datos de un API remoto, lo más indicado es crear un servicio para ello. Un servicio se creará con el siguiente comando:

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

Al ejecutar la orden se crearán dos archivos dentro del directorio app, uno con el nuevo servicio y otro para definir los tests. Una vez hecho esto, ya podremos crear métodos dentro del servicio que accedan al API remoto mediante alguna herramienta como HttpClient, propia de Angular. Finalmente, para acceder al servicio desde cualquiera de los componentes, habrá que importar el servicio, crear una instancia del mismo en el constructor del componente, y ya podremos llamar a sus métodos desde esta instancia.

Seguir aprendiendo Angular

Puesto que este trabajo es una introducción a Angular, hay muchos aspectos del framework que se han quedado en el tintero. A partir de aquí, proponemos los siguientes pasos y conceptos para seguir profundizando y poder convertirnos en expertos desarrolladores de Angular:

- Ampliar conocimiento en el consumo de APIs y la realización de peticiones HTTP desde Angular.
- Profundizar en la sintaxis de plantillas de Angular.
- Mejoras en la interfaz de usuario con alguna librería de componentes gráficos como por ejemplo “material UI”.
- Uso de directivas.
- Profundizar en el routing y navegación.
- Seguridad de aplicaciones.
- Debugging.

Bibliografía

Documentación de Angular. (2021). <https://angular.io/docs>

Angular (framework). Recuperado el 13 de enero de 2021. En *Wikipedia*.
[https://es.wikipedia.org/wiki/Angular_\(framework\)](https://es.wikipedia.org/wiki/Angular_(framework))

Comparativa frameworks Google Trends. 11 de enero de 2021.
<https://trends.google.es/trends/explore?geo=ES&q=Vue,Angular,React>

Free Code Camp. The Best Angular Examples. 29 de noviembre de 2019.
<https://www.freecodecamp.org/news/the-best-angular-examples/>

Dayana Jabif. Angular Tutorial: Learn Angular from scratch step by step. 16 de agosto de 2020. <https://angular-templates.io/tutorials/about/learn-angular-from-scratch-step-by-step>
<https://www.djamware.com/post/5b87894280aca74669894414/angular-httpclient-678910-consume-rest-api-example>