

# TRABAJO GRUPAL

## 1. Tutorial ngular

Aplicaciones Distribuídas en Internet

Grado en Ingeniería Informática

Antero Guarinos Caballero  
Gema Lozano Jiménez  
Álvaro Prieto Martín-Buitrago

# ÍNDICE

1- Introducción	2
2- Angular vs React vs Vue	3
3- Componentes	4
4- Servicios	4
5- Modelos	4
6- Enrutamiento	5
7- Seguridad	5
8- Bibliografía	6

# 1- Introducción

**Angular** es un *framework opensource* desarrollado por Google para facilitar la creación y programación de aplicaciones web de una sola página, las webs SPA (*Single Page Application*).

Angular separa completamente el frontend y el backend en la aplicación, evita escribir código repetitivo y mantiene todo más ordenado gracias a su patrón MVC (Modelo-Vista-Controlador) asegurando los desarrollos con rapidez, a la vez que posibilita modificaciones y actualizaciones.

En una web SPA aunque la velocidad de carga puede resultar un poco lenta la primera vez que se abre, navegar después es totalmente instantáneo, ya que se ha cargado toda la página de golpe.

Solamente es una ruta la que se tiene que enviar al servidor, y Angular lo que hace ‘por debajo’ es cambiar la vista al navegar para que dé la apariencia de una web normal, pero de forma más dinámica.

Entre otras ventajas, este *framework* es modular y escalable adaptándose a nuestras necesidades y al estar basado en el estándar de componentes web, y con un conjunto de interfaz de programación de aplicaciones (API) permite crear nuevas etiquetas HTML personalizadas que pueden reutilizarse.

El lenguaje principal de programación de Angular es *Typescript*, y así toda la sintaxis y el modo de hacer las cosas en el código es el mismo, lo que añade coherencia y consistencia a la información, permitiendo por ejemplo, la incorporación de nuevos programadores, en caso de ser necesarios, ya que pueden continuar su trabajo sin excesiva dificultad.

Como ya se ha indicado, las plantillas de Angular almacenan por separado el código de la interfaz del usuario (front-end) y el de la lógica de negocio (back-end), que entre otros beneficios permite utilizar mejor otras herramientas anteriormente existentes.

Y por si fuera poco, los principales editores y entornos de desarrollo integrado (IDEs) ofrecen ya extensiones para poder trabajar con este framework con mayor comodidad.

Por su programación reactiva, la vista se actualiza automáticamente tras realizar los cambios. Además Angular dispone de asistente por línea de comandos para poder crear proyectos base y también se integra bien con herramientas de *testing* y con Ionic, lo que facilita la creación de web-responsive, es decir, adaptadas a móviles.

Este aspecto cada día adquiere mayor importancia tanto por el creciente uso de estos dispositivos para acceder a internet como por la penalización que Google realiza de aquellas páginas que no facilitan su visita en cualquier dispositivo.

## 2- Angular vs React vs Vue

### Curva de aprendizaje

Cuando hablamos de la curva de aprendizaje nos referimos a una gráfica que describa qué tan complicado es aprender algo y qué tan pronto puedo obtener beneficios y resultados del esfuerzo que estoy dedicando a aprender algo.

En lo que respecta a estos 3 frameworks, Angular es el que se separa de los otros 3 por ser una tecnología que toma más tiempo dominar y en la que quizás sentirte cómodo pueda tomar más tiempo.

React y Vue en este apartado son muy similares, con una pequeña ventaja para Vue porque es un ecosistema mucho más uniforme. Si bien existen muchísimas librerías muy buenas para React, Vue tiene la ventaja de que el mismo equipo que desarrolla el framework trabaja en las librerías más populares y útiles del ecosistema de Vue, en términos de aprendizaje esto se traduce en que Vue y sus librerías son muy similares por lo que sentirse productivo en esta tecnología es mucho más fácil.

### Aplicaciones de gran escala

Aquí el primero en caer será Vue, porque aunque internamente se usa en Netflix e incluso Facebook, no tiene los grandes nombres que vienen con Angular y React.

En lo que respecta a Angular y React, la competencia es mucho más cerrada, comenzando por el fuerte impulso que reciben de Google y Facebook respectivamente. Ambas librerías son usadas tanto en Google como en Facebook mismo, además de otras empresas enormes como Uber, PayPal, The New York Times, entre otros más.

### Experiencia de desarrollo

Las 3 tienen muy buenas herramientas para el desarrollo, buena integración con los editores de texto más populares y por supuesto herramientas diseñadas para mejorar tu flujo de trabajo.

En Angular todos los frameworks se crean usando Typescript, un lenguaje que mejora considerablemente la experiencia de desarrollo de aplicaciones JavaScript como las de Angular, sin embargo, el desarrollo usando Typescript también es muy popular en React y Vue, aunque no es una norma como en Angular.

React se lleva la batalla por el desarrollo de herramientas como las React Developer Tools, una extensión para el navegador que, precisamente, extiende el soporte para depurar aplicaciones creadas con React.

### Configuración del entorno de trabajo

En este punto destacan Vue y React, porque no necesitas nada más que importar algunas librerías y listo, puedes usarlas sin problemas, de hecho, Vue destaca mucho más porque a diferencia de React, está diseñado para funcionar en cualquier entorno. Y aunque React también puede hacerlo, tienes que traer al menos algunas herramientas como JSX, porque sin ellas sería muy raro escribir código de React.

Angular es un framework mucho más robusto con el que te tienes que comprometer mucho más para poder aprovecharlo.

### 3- Componentes

El comando que usaremos para crear un nuevo componente para nuestra app será:

**ng generate component *name***

o en su defecto la abreviación:

**ng g c *name***

Los componentes son la parte principal de una aplicación en angular. Estos componentes se componen, valga la redundancia:

- De un fichero HTML
- Un fichero CSS
- Una clase TypeScript que define el comportamiento.
- Puede contener ficheros CSS adicionales.

### 4- Servicios

Los componentes no deben buscar o guardar datos directamente y ciertamente no deben presentar datos falsos a sabiendas. Deben centrarse en presentar datos y delegar el acceso a los datos a un servicio.

El comando que usaremos para crear un nuevo servicio para nuestra app será:

**ng generate service *name***

o en su defecto la abreviación:

**ng g s *name***

Los servicios se componen de 2 archivos:

- archivo.service.spec.ts
- archivo.service.ts

### 5- Modelos

Básicamente es una clase que se exporta que contiene todos los campos de una entidad de la base de datos. Los modelos se suelen crear en una carpeta aparte al igual que los servicios, componentes, etc y en la misma altura que estos. No existe en Angular un generador de este tipo de modelos.

Estos modelos para ser utilizados se importan en los typescript de donde quieras utilizarlo y, al asignar a una variable el tipo correspondiente al modelo de la bd, le aparecerán recomendados todos los campos de esta, facilitando así todo.

## 6- Enrutamiento

El enrutamiento o rutas en Angular es la manera en la que navegamos entre las vistas de nuestra aplicación, en una web normal nosotros navegamos entre páginas HTML, pero en Angular navegamos entre vistas que hemos generado a base de módulos y componentes.

Hay varias maneras de crear las rutas, ya sea con Angular CLI o manualmente:

**ng g module app-routing**

## 7- Seguridad

Cuando queremos que determinadas áreas de nuestra aplicación web estén protegidas y solo puedan ser accedidas en determinadas situaciones. Para controlar el acceso de los usuarios a las diferentes vistas de una aplicación se utilizan los guards, pudiendo ser configurados como nosotros queramos para controlar estos accesos.

Dentro de los guards hay 4 tipos principales:

- **CanActivate:** Mira si el usuario puede acceder a una página determinada.
- **CanActivateChild:** Mira si el usuario puede acceder a las páginas hijas de una determinada ruta.
- **CanDeactivate:** Mira si el usuario puede salir de una página, es decir, podemos hacer que aparezca un mensaje, por ejemplo, de confirmación, si el usuario tiene cambios sin guardar.
- **CanLoad:** Sirve para evitar que la aplicación cargue los módulos perezosamente si el usuario no está autorizado a hacerlo.

El más usado de todos y el que explicaremos aquí es el CanActive; que siempre devuelve true o false, dependiendo de si el usuario puede acceder o no. Y en esta función puedes controlarlo de la manera que tu quieras.

Una vez se crean, para ser utilizados, se añaden a la ruta deseada que hemos creado en el app.routing.module.

## 8- Bibliografía

Introducción:

<https://www.qualitydevs.com/2019/09/16/que-es-angular-y-para-que-sirve/>

Comparativa:

<https://codigofacilito.com/articulos/angular-react-vue>

Componentes:

<https://angular.io/guide/component-overview#:~:text=Components%20are%20the%20main%20building.is%20used%20in%20a%20template>

Servicios:

<https://angular.io/guide/architecture-services>

Modelos:

<https://desarrolloweb.com/articulos/practica-angular-modulos-componentes-servicios.html>

Enrutamiento:

<https://aristotekean.medium.com/enrutamiento-en-angular-996f14e89293#:~:text=El%20Enrutamiento%20o%20rutas%20en%20Angular%20es%20la%20manera%20en.base%20de%20m%C3%B3dulos%20y%20componentes>

Seguridad:

<https://codingpotions.com/angular-seguridad>