



Curso  
Desarrollo de Software

# FRONT END

NIVEL AVANZADO



## Conceptos básicos de Angular

### Estructura

Cuando empecemos un proyecto en [Angular](#), es muy importante estructurar las carpetas para que sean escalables y evitar desorden en el desarrollo. Además, al crear nuestro proyecto, Angular crea muchas carpetas y archivos que son imprescindibles saber para qué existen, ya que pueden surgir errores sobre ellos.

- **e2e:** esta carpeta, denominada “end to end”, engloba unas series de ficheros cuya función es la realización de test automáticos, como si un usuario interactuase con la app. Se ejecuta con el comando `ng e2e`.
- **node\_modules:** es la carpeta que contiene todas las dependencias de nuestro proyecto.
- **Src:** es el directorio donde trabajaremos nuestros módulos. Además, es el más importante, ya que contiene todo el código.
  - Carpeta `src/app`  
Aquí se encuentra la carpeta `app` donde se ubica toda la implementación de los componentes principales, junto a su template html y archivos de estilos css.
  - Carpeta `src/assets`  
Contendrá todos los asset y archivos adicionales para hacer que el proyecto funcione.
  - Carpeta `src/environments`  
Donde se encuentran las configuraciones y variables de entorno para poner el proyecto tanto en desarrollo como en producción.
  - Archivo `src/favicon.ico`  
Es el archivo del icono del proyecto.
  - Archivo `src/index.html`  
Es el archivo de la página principal del proyecto.
  - Archivo `src/main.ts`  
Es el archivo Type Script inicial del proyecto donde podrás configurar todas las configuraciones globales del proyecto.
- **editorconfig:** es la configuración de nuestro editor de código.
- **.gitignore:** son las carpetas o archivos que debe ignorar el git cuando lo añadamos al repositorio.
- **angular.json:** contiene la configuración de Angular. Además, incluye rutas, versiones, etc.

- package.json: es la configuración de nuestra aplicación. Contiene el nombre de la app, las dependencias necesarias para su correcta ejecución y muchas otras cosas.
- REARME.md: aquí podemos añadir información sobre la aplicación. Este archivo es leído por GIT y los muestra en el repositorio.
- tsconfig.json: contiene la configuración TypeScript.
- tslint.json: se utiliza para que el código sea sostenible y se mantenga.

**Actividad:** Explora las carpetas que tiene Angular.

## Hoja de estilos

Puedes aplicar todo lo que sabes sobre hojas de estilo CSS, selectores, reglas y *media queries* directamente a [Angular](#).

Para cada componente Angular que escriba, puede definir no solo una plantilla HTML sino también los estilos CSS que acompañan a esa plantilla, especificando los selectores, reglas y consultas de medios que necesite.

Una forma de hacer esto es estableciendo la propiedad de estilos en los metadatos del componente. La propiedad de estilos toma una matriz de cadenas que contienen código CSS.

Ejemplo:

```
@Component({
  selector: 'app-root',
  template: `
    <h1>Tour of Heroes</h1>

    <app-hero-main [hero]="hero"></app-hero-main>
  `,
  styles: ['h1 { font-weight: normal; }']
})
export class HeroAppComponent {
  /* . . . */
}
```

Los estilos especificados en los metadatos de `@Component` se aplican sólo dentro de la plantilla de ese componente.

## Ventajas de Angular

[AngularJS](#) extiende HTML añadiendo funcionalidad y permite crear plantillas dinámicas poderosas. Implementa binding de datos de dos vías, conectando tu HTML (vistas) a los objetos de JavaScript (modelos) sin problemas. También permite generar componentes reusables abstrayendo toda la lógica de manipulación del DOM de modo que cualquier actualización en su modelo se refleja inmediatamente en la vista, sin necesidad de ningún tipo de manipulación DOM o el control de eventos (por ejemplo, con jQuery).

Como ves AngularJS ofrece una gran cantidad de ventajas, las principales son:

- Puedes usar TypeScript pero no es obligatorio: Tanto si eres un desarrollador old school como si prefieres trabajar con un estilo propio, Angular no te obliga a escribir tu código usando TypeScript pero es una práctica que recomendamos. ¿Por qué complicarse? ¿Si algo funciona no es mejor no tocarlo? Puede ser. Si vienes del mundo de Java quizá te guste más trabajar en ECMAScript pero debo recordar que existen diferentes versiones y cada versión ofrece varias formas diferentes de declarar un objeto, lo cual puede confundirnos. Esto genera un serio problema de inconsistencia. Al usar TypeScript no tenemos este problema. TypeScript evita la confusión y ofrece un mantenimiento más fácil de las aplicaciones.
- No hace complicado aquello que debe ser fácil: AngularJS es un framework y no una biblioteca por lo que incluye muchas más opciones y funcionalidades de serie. Esto supone un importante ahorro de recursos y permite a empresas reducir sus tiempos de producción.
- Componentes web. Reutilizar nuestro código supone una enorme ventaja. Angular adopta el estándar de los componentes web. Un conjunto de APIs que permiten crear nuevas etiquetas HTML personalizables y reutilizables que podemos utilizar en otros proyectos con independencia del navegador que utilicemos. ¿Y qué beneficio nos ofrece esto? ¡Más tiempo libre!
- Es open source. No vamos a profundizar mucho sobre sus bondades, pero en la actualidad el código abierto es clave en el desarrollo de la nueva economía colaborativa. Gracias al software de fuentes abiertas es posible que otras personas aprendan a partir de nuestro código, aporten soluciones o desarrollen las suyas a partir de las aportaciones de otros.
- Buen soporte y una enorme cantidad de recursos disponibles. A diferencia de otros lenguajes podemos usar desde el bloc de notas a los principales editores e IDEs para trabajar con este framework. Podemos desarrollar nuestra aplicación desde cero o a partir de una plantilla. Al almacenar por separado el código de la capa lógica y de presentación podemos sacar partido y editar este código con el IDE que mejor nos convenga. Pero lo mejor es su comunidad de usuarios y desarrolladores. Siempre volcados en ayudar, resulta muy fácil

encontrar respuestas a nuestras consultas. Y si todavía nos quedan dudas podemos consultar la excelente documentación disponible.