



ANGULAR

DESARROLLO WEB ENTORNO CLIENTE



CONTENIDO

- 1.- ¿Qué es Angular?
- 2.- Arquitectura
- 3.- Estructura
- 4.- Objetivos Conseguidos

¿QUÉ ES ANGULAR?



Es un **framework** de código abierto desarrollado por Google para crear aplicaciones web de una sola página.

CARACTERÍSTICAS

- Es modular y escalable, lo que permite que se adapte a cualquier proyecto.
- El lenguaje de programación que se utiliza es **TypeScript** y está orientado al front-end.
- Separa en plantillas la parte del front-end de la parte del back-end permitiendo usar otras herramientas más optimizadas para sacar mayor rendimiento a la aplicación.
- Integración con herramientas de testing y con Ionic, lo que facilita la creación de web responsive.

ARQUITECTURA



La arquitectura de una aplicación Angular se basa en módulos:

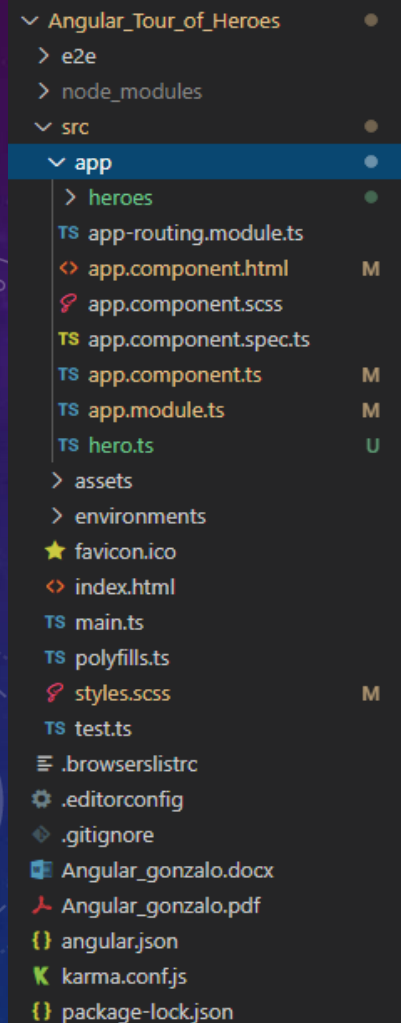
- El conjunto de módulos es **NgModules**.
- Se necesita al menos un componente raíz que permita el arranque de nuestra aplicación (**rootModulo**) que envuelve todo lo demás.

Dentro de los módulos nos encontramos:

- Componentes que se componen de -> **Metadatos** y **Plantillas** (html y css).
- Las **directivas** con sus metadatos para poder interactuar y se relacionen las plantillas.
- Los **servicios** que se inyectan a los componentes a medida que se vaya necesitando en el proyecto.

ARQUITECTURA

Nuestra aplicación

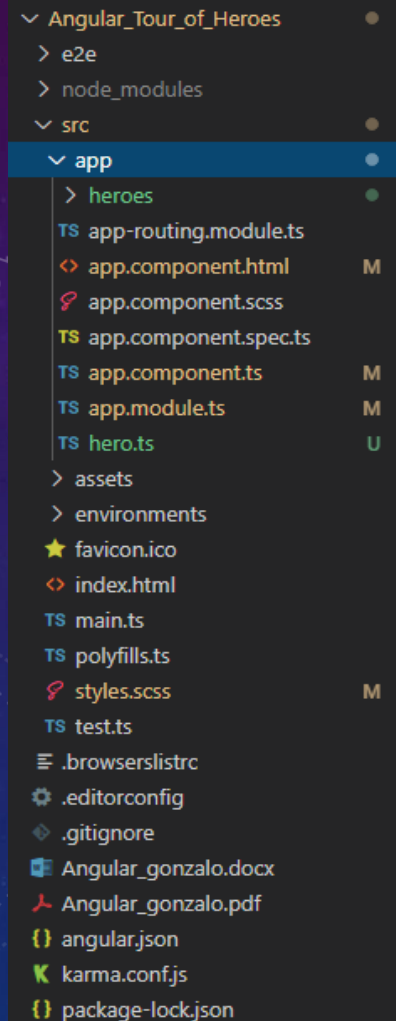


Aquí encontramos el módulo raíz: **app.module.ts**

```
TS app.module.ts X
Angular_Tour_of_Heroes > src > app > TS app.module.ts > AppModule
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3 import { FormsModule } from '@angular/forms'; //<-- NgModel lives here
4
5 import { AppRoutingModule } from './app-routing.module';
6 import { AppComponent } from './app.component';
7 import { HeroesComponent } from './heroes/heroes.component';
8
9
10
11
12 @NgModule({
13   declarations: [
14     AppComponent,
15     HeroesComponent
16   ],
17   imports: [
18     BrowserModule,
19     FormsModule,
20     AppRoutingModule
21   ],
22   providers: [],
23   bootstrap: [AppComponent]
24 })
25 export class AppModule { }
26
```

ARQUITECTURA

Nuestra aplicación



Aquí encontramos el componente raíz: **app.component.ts**

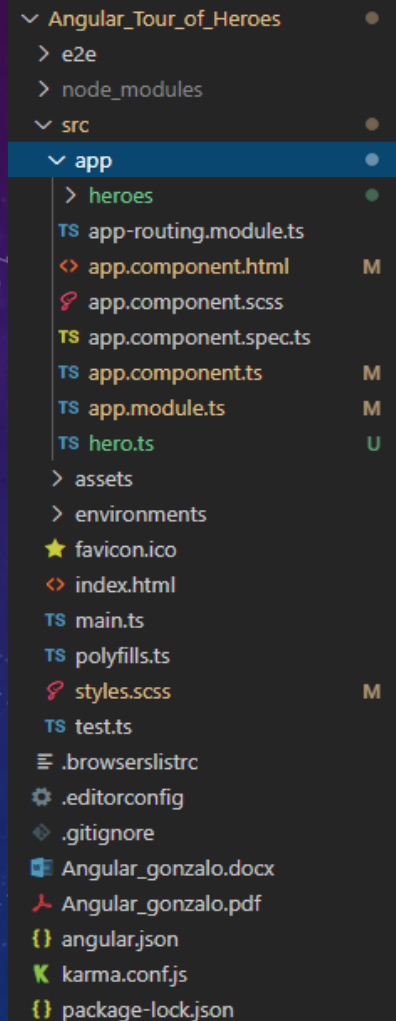
TS *app.component.ts* X

Angular_Tour_of_Heroes > src > app > TS app.component.ts > AppComponent > title

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.scss']
7 })
8 export class AppComponent {
9   title = 'Tour of heroes';
10 }
11
```

ARQUITECTURA: VISTA

Nuestra aplicación



Aquí encontramos la plantilla html: [app.component.html](#)

app.component.html X

Angular_Tour_of_Heroes > src > app > app.component.html > app-heroes

```
1 <h1>{{title}}</h1>
2 <app-heroes></app-heroes>
```


ESTRUCTURA: ARCHIVOS



Archivos de configuración para el espacio de trabajo: Todos los proyectos Comparten un conjunto de ficheros de configuración.

`editorconfig` → Configuración para editores de código.

`angular.json` → Los valores predeterminados de configuración de **CLI** para todos los proyectos en el espacio de trabajo.

`package.json` → Configura las dependencias del paquete **npm** que están disponibles.

`package-lock.json` → Proporciona información sobre la versión de todos los paquetes instalados.

`src/` → Archivos de origen para el proyecto de aplicación de nivel raíz.

`node_modules/` → Proporciona paquetes **npm** a todo el espacio de trabajo.

`tsconfig.json` → La configuración básica de **TypeScript** para proyectos en el espacio de trabajo. Todos los demás archivos de configuración heredan de este archivo base.

`tslint.json` → Configuración de **TSLint** predeterminada para proyectos en el espacio de trabajo.

ESTRUCTURA: ARCHIVOS



Las subcarpetas contienen el origen de la aplicación y la configuración específica de la Aplicación.

- app/** → Contiene los archivos de componentes en los que se definen la lógica y los datos de su aplicación.
- assets/** → Contiene imágenes y otros archivos de activos que se copiarán tal cual cuando compile su aplicación.
- environments/** → Contiene opciones de configuración de compilación para entornos de destino particulares.
- favicon.ico** → Un icono que se utilizará para esta aplicación en la barra de marcadores.
- index.html** → La página HTML principal que se muestra cuando alguien visita su sitio.
- main.ts** → El principal punto de entrada de la aplicación.
- polyfills.ts** → Proporciona scripts de **polyfill** para compatibilidad con el navegador.
- styles.sass** → Muestra una lista de archivos CSS que proporcionan estilos para un proyecto.
- test.ts** → El principal punto de entrada para sus pruebas unitarias.

ESTRUCTURA: ARCHIVOS



La app/ contiene la lógica y los datos del proyecto. Los componentes, plantillas y estilos.

`app/app.component.ts` → Define la lógica del componente raíz de la aplicación, denominado **AppComponent**.

`app/app.component.html` → Define la plantilla HTML asociada a la raíz **AppComponent**.

`app/app.component.css` → Define la hoja de estilo CSS base para la raíz **AppComponent**.

`app/app.component.spec.ts` → Define una prueba unitaria para la raíz **AppComponent**.

`app/app.module.ts` → Define el módulo raíz, llamado **AppModule**, que le dice a Angular cómo ensamblar la aplicación.

¿QUÉ OBJETIVOS HAS CONSEGUIDO?



- Saber que es Angular.
- Como está organizado su arquitectura.
- Cuál es su configuración.
- Crear un proyecto desde cero.
- Iniciar el ejemplo Tour de Heroes.
- Crear un componente.
- Crear una interfaz.



¡¡ GRACIAS !!

Gonzalo García Mateos

- Bibliografía: <https://angular.io/start>
- Github: https://github.com/Kurtgon/Angular_Tour_of_Heroes