



Florida
Università

Introducción a Angular

Curso 2025/26

Paco Segura

¿Qué es Angular?

- Framework de desarrollo en entorno cliente creado y mantenido por Google.
 - <http://angular.io>
 - “Angular is a platform and framework for building single-page client applications using HTML and TypeScript”
 - Framework -> “marco de trabajo” para desarrollar aplicaciones.
 - Frameworks de JS -> Angular, React, Vue...
-

¿Qué es Angular?

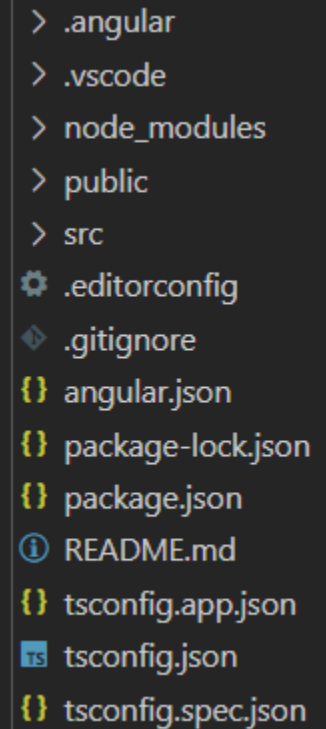
- Basado en desarrollo de componentes web.
 - ¿Por qué surge Angular?
 - Para homogeneizar los procedimientos de desarrollo de aplicaciones en cliente.
 - Con DOM puro, cada desarrollador podía aportar una solución diferente.
-

Angular CLI:

- CLI: Command Line Interface
 - Utilidad de consola para automatizar tareas repetitivas
 - Angular CLI
 - <https://angular.dev/tools/cli>
 - Comando: ng
 - ng help
 - ng new
 - ng generate
 - ng serve
 - ng build
-

Angular:

- **Prerrequisitos:** última versión de [Node.js LTS](#)
- **Instalación:**
 - `npm install -g @angular/cli`
 - `ng new my-app`
 - `ng serve` -> lanza la aplicación.

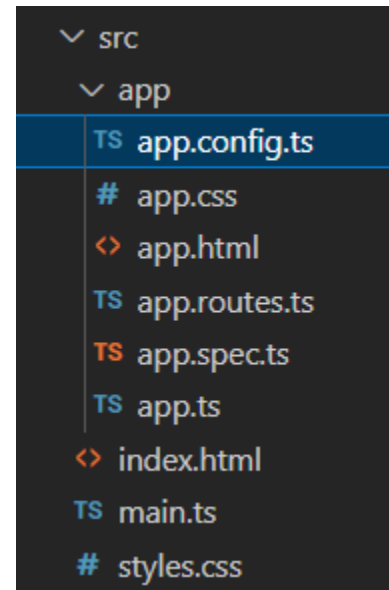


A screenshot of a file explorer showing the project structure of an Angular application. The files and folders are listed with their respective icons: a folder icon for directories and a file icon for files. The files are: .angular, .vscode, node_modules, public, src, .editorconfig, .gitignore, angular.json, package-lock.json, package.json, README.md, tsconfig.app.json, tsconfig.json, and tsconfig.spec.json.

- > .angular
- > .vscode
- > node_modules
- > public
- > src
- ⚙ .editorconfig
- 📄 .gitignore
- { } angular.json
- { } package-lock.json
- { } package.json
- 📖 README.md
- { } tsconfig.app.json
- ts tsconfig.json
- { } tsconfig.spec.json

Angular:

- Archivos *.ts -> archivos TypeScript
- Carpeta src



Angular:

- **main.ts** -> punto de entrada a Angular.
 - **index.html** -> punto de entrada a la aplicación (donde se renderiza todo el código que creamos).
 - **app.html** -> Donde trabajaremos.
 - **app.ts** -> Donde crearemos los componentes.
 - **styles.css** -> estilos globales de la aplicación.
-

Componentes Web:

- ¿Qué son Componentes Web?
 - Desarrollo Software basado en componentes: enfoque basado en la reutilización de entidades llamadas “componentes de software”
 - Componente: contenido autónomo que tiene un patrón común que se repite. Puede tener subcomponentes.
-

Componentes Web:

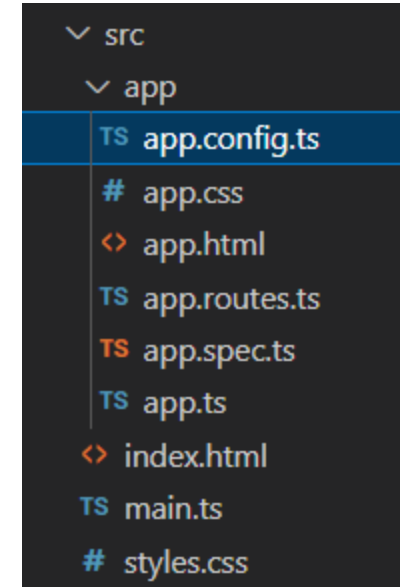
- Componentes: evitan tener que copiar, pegar y personalizar cada vez el contenido.
 - Como trabajar con diferentes piezas que reutilizamos cada vez para construir la aplicación.
-

Componentes Web:



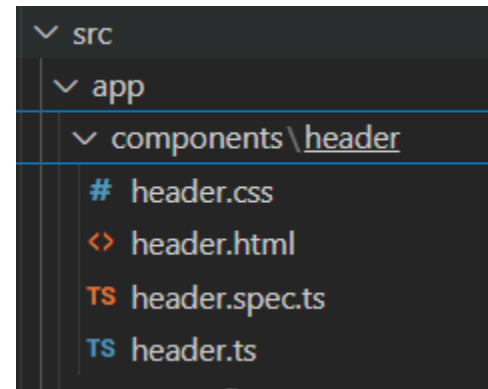
Componentes en Angular:

- Carpeta src/app
- **app.html** -> donde instanciamos los componentes.



Componentes en Angular:

- Creación de componentes: los componentes los ubicaremos en una carpeta llamada componentes. Utilizamos el siguiente comando:
 - `ng generate component components/nombre_componente`
- Crea una carpeta '**components**' y otra carpeta con el nombre del componente y 4 archivos.



Componentes en Angular:

- En este caso hemos creado un componente Header. Se nos ha creado automáticamente una carpeta 'header' dentro de la carpeta 'components'. En ella tenemos cuatro archivos. Trabajaremos con tres:
 - **header.css:** donde ubicaremos los estilos css de ese componente.
 - **header.html:** donde ubicaremos el código html del componente.
 - **header.ts:** donde ubicaremos las funcionalidades, en una clase TypeScript.
-

Componentes en Angular:

- Veamos una aplicación práctica para el componente header.

- **header.css:**

```
header{  
  width: 1000px;  
  font-size: 35px;  
  margin: auto;  
  background-color: grey;  
  padding: 20px;  
  display: flex;  
  flex-direction: row-reverse;  
}
```

Componentes en Angular:

- header.html:

```
<header>Hello World!</header>
```

Componentes en Angular:

- **header.ts:** utilizamos el selector 'app-header' para poder instanciarlo en el archivo **app.component.html**.

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-header',
  imports: [],
  templateUrl: './header.html',
  styleUrls: ['./header.css'],
})
export class Header {

}
```


Componentes en Angular:

- **app.ts**: importamos el componente Header.

```
import { Component, signal } from '@angular/core';
import { RouterOutlet } from '@angular/router';
import { Header } from '../components/header/header';

@Component({
  selector: 'app-root',
  imports: [RouterOutlet, Header],
  templateUrl: './app.html',
  styleUrls: ['./app.css']
})
export class App {
  protected readonly title = signal('project');
}
```

Componentes en Angular:

- **app.html.** Instanciamos el componente <app-header>

```
<app-header></app-header>
```
