Angular

Referencias:

* <https://angular.io/cli>

**Propiedades de las aplicaciones desarrolladas en Angular:**

* **Single Page Application (SPA)**

Angular permite generar aplicaciones **SPA**, es decir, la página se carga una sola vez y lo que se van actualizando son las vistas contenidas en la página mediante el manejo del routing.

* Las aplicaciones desarrolladas en angular se generan a partir de **COMPONENTES**
* Los COMPONENTES están formados de Plantillas, Clases, Metadatos.
* Los **Módulos** están formados de componentes y se utilizan para agrupar funcionalidades.
* Flujo de datos **UNIDIRECCIONAL** (Del componente padre al componente hijo)
* Flujo de datos **BIDIRECCIONALIDAD** (Los datos fluyen en ambos sentidos padre-hijo, hijo-padre)
* **Inyección de Dependencias**

Lo cual, genera código desacoplado.

Creación de objetos mediante constructor/setter

* **Directivas**

Las directivas extienden la funcionalidad del HTML usando una nueva sintaxis.

Etiquetas html con atributos con los que se les agrega funcionalidad.

Existen dos tipos de directivas:

* + Atributos
  + Estructurales
* Angular trabaja con TypeScript

Code.ts 🡪Transpila Code.ts 🡪 Code.js

* CLI

Mediante la herramienta de CLI se generan los componentes de una aplicación desarrollada en Angular.

**TypeScript:**

* Es un lenguaje tipado, es decir, todos los objetos tienen un tipo de dato
* Maneja el concepto de Orientada a Objetos.
* Herencia
* Declaración de variables:
  + let 🡪 ámbito de bloque – locales – Dentro del bloque donde fueron declaradas.
  + var 🡪 ámbito global
  + const 🡪 ámbito de bloque – locales – Deben ser inicializadas.
* Maneja el concepto de **hoisting**

No importa donde se declaren las variables en TypeScript, se hace un recorrido previo a su utilizarlas para identificar su declaración. Influencia el ámbito de las variables.

Ejemplo:

console.log(x);

var x = 3;

En consola se imprime: undefined 🡪 Tiene que ver el ámbito global de var x=3;

console.log(z);

let z = 5;

En consola se imprime: Se genera un error 🡪 Tiene que ver el ámbito de bloque.

console.log(y);

const y = 10;

En consola se imprime: Se genera un error. 🡪 Tiene que ver al ámbito de bloque.

* **Constantes**

Las constantes no se pude cambiar su valor.

Ejemplo:

const x = 10;

x = 11; 🡪 Esto genera un error.

const figura = {nombre: “Cuadrado”, area: 10}

figura = {nombre: “Circulo”, area: 15} 🡪 Genera un error

figura.nombre = “Circulo” 🡪 OK

No se puede asignar un nuevo valor a figura, pero si se pueden modificar sus atributos (estado interno).

**Elementos importantes de un componente:**

* app.component.css 🡪 Se definen los estilos
* app.component.html 🡪 Se define el template
* app.component.ts 🡪 Se definen los elementos del componente.
* app.module.ts 🡪 Se dan de alta los componentes del modul.

**NOTAS:**

A partir de Angular 17 ya no se genera por default el **app.module.ts** en los proyectos, para generar el **app.module.ts** se debe crear el proyecto con la instrucción:

ng new <appName> --standalone=false

**Comandos de CLI**

* ng new <appName> 🡪 Crear una nueva aplicación
* ng g component <componentName> 🡪 Crea un nuevo componente

**Comandos para levantar aplicaciones Angular:**

* **ng serve**