

ANGULAR

Es un framework para JavaScript que nos ayuda a desarrollar aplicaciones web SPA, es decir, una aplicación que no se necesita recargar la página en ningún momento y que está completamente separada del **backend**. La comunicación con el *Backend* es mediante peticiones AJAX y mediante comunicación asincrónica.

En caso de tenerlo instalado debemos hacer lo siguiente:

```
Desinstalar angular/cli:  
- npm uninstall -g @angular/cli
```

```
Borrar cache: - npm cache verify - npm cache clear --force
```

INSTALACIÓN DE ANGULAR

```
npm install -g @angular/cli
```

Generar un nuevo proyecto angular

```
ng new "nombre de la carpeta"
```

```
Arrancar el proyecto: ng serve --open (se debe estar dentro de la carpeta creada)
```

CONCEPTOS TEÓRICOS DE ANGULAR

1- Componente

Un componente al final va a controlar un trozo de pantalla o de la vista. Todo lo que se puede ver en pantalla es controlado y gestionado por este tipo de elementos. La lógica de un componente dentro de una clase en Angular es que da soporte a una vista interactuando con ella a través de una API con propiedades y métodos.

El componente hace de mediador entre la vista a través de la plantilla y la lógica de la app donde se incluirá el modelo de datos, es decir una especie de controlador.

2- Plantillas

Las plantillas van a definir la vista de los componentes.

Son HTMLs y tienen sintaxis especial de Angular. Trabajando con el databinding y las directivas.

3- Decoradores y metadatos

Con los decoradores (patrón de diseño) vamos a configurar dinámicamente atributos/metadatos de las clases y componentes.

Los metadatos van a describir a las clases pero también describen relaciones, por ejemplo si tenemos un componente y una plantilla el metadato sirve para encargarse de decirle a Angular que ese componente y esa plantilla van juntos, entre otras muchas cosas.

4- Servicios

Son clases con un objetivo claro, facilita la reutilización, son un tipo de elemento dentro de la arquitectura de Angular y mediante la inyección de dependencias los podemos usar en otros componentes principales.

5- Providers

Son servicios que nos proveen de datos o funcionalidades mediante sus métodos. Existen providers/servicios propios de Angular o creados por nosotros mismos.

6- Directivas

Son funcionalidades aplicables al DOM y a los elementos HTML en las plantillas de un componente. Por ejemplo una directiva puede servir para controlar que un div se muestre o no, recorrer un array en la vista (directivas estructurales, estructuras condicionales y de control) o incluso también para interactuar con el modelo de datos del componente.

Basicamente son nuevos atributos para aplicarle a cualquier cosa en nuestra plantilla/vista.

COMPONENTE

Es una parte de nuestra aplicación. Puedo tener un componente para el header, otro componente para un sidebar, otro para un calendario. Todo en Angular está formado por componentes.

Los componentes se crearán en la carpeta "app", en el archivo app.component.ts

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

@Component({ //Decorador
  selector: 'app-root',
  templateUrl: ['.app/component.html'] //Vista del componente
  styleUrls: ['.app/component.css']
})
export class AppComponent {
  title = 'app';
}
```