



Angular

**19 de febrero 2019.**

<https://es.slideshare.net/secret/4TT04IERdzkPJT>

L.I. Hugo G Cuéllar Martínez.  
hugo.cuellar@comunidad.unam.mx



**DGTIC**



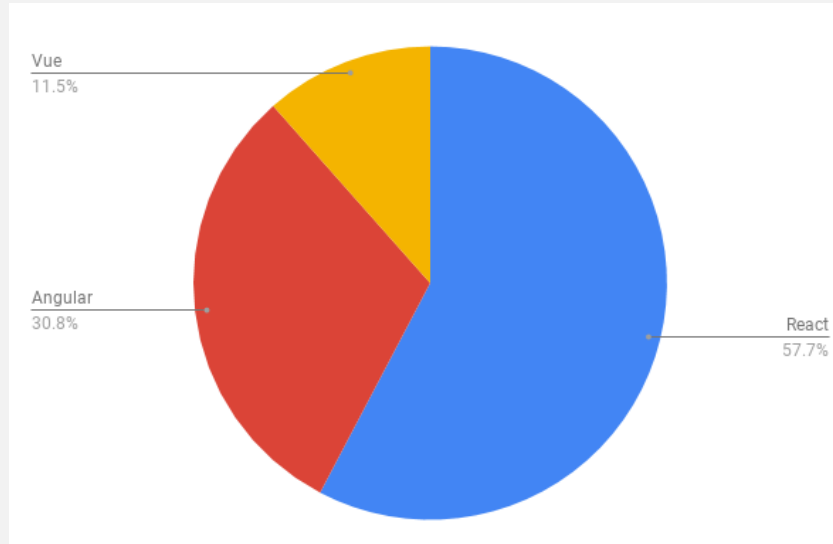
## Framework de Javascript Desarrollado por

La finalidad de Angular es facilitarnos el desarrollo de aplicaciones web SPA y además darnos herramientas para trabajar con los elementos de una web de una manera más sencilla y optima.

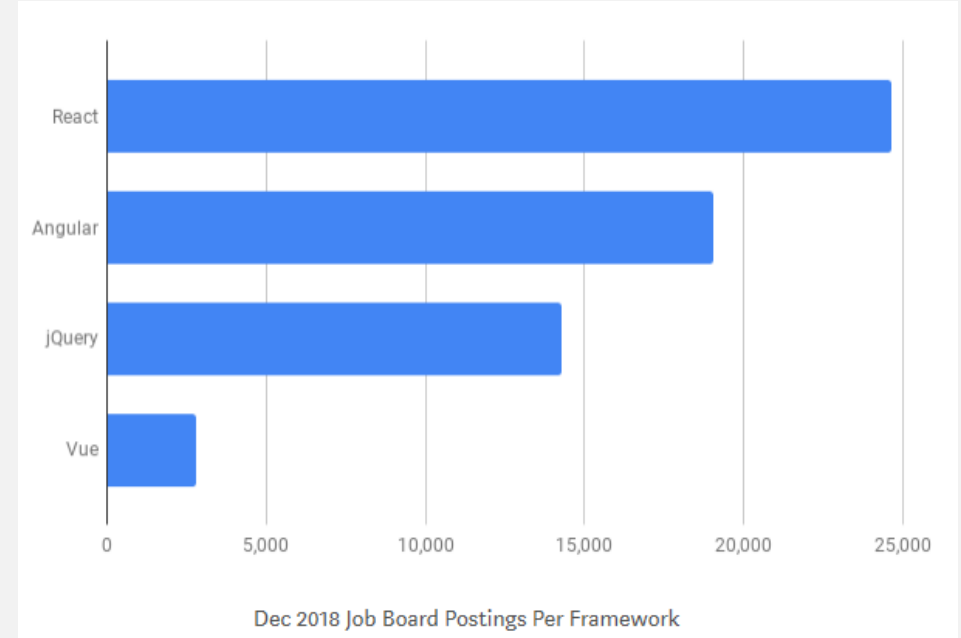
<https://angular.io>



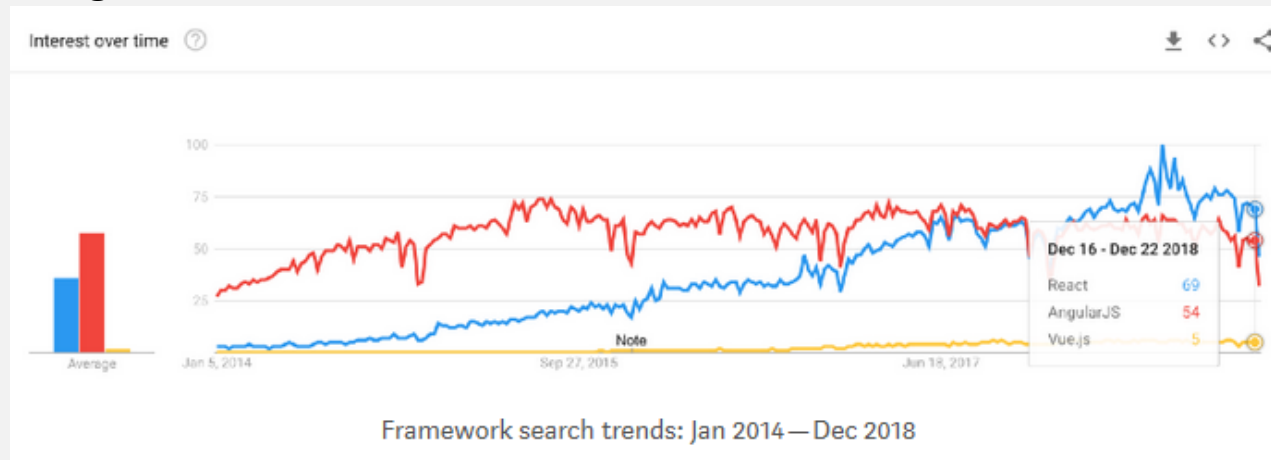
## Descargas 2018



## Indeed.com Job Postings



## Google Search Trends



<https://medium.com/javascript-scene/top-javascript-frameworks-and-topics-to-learn-in-2019-b4142f38df20>



# Requisitos

## NodeJS

Concebido como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node está diseñado para construir aplicaciones en red escalables.

Construido con el motor de JavaScript V8 de Chrome.

(<https://v8.dev/> )

Ventajas:

- Rápido.
- Construido para funcionar de manera asíncrona.
- Comunicación activa.
- Funciona muy bien con base de datos no relacionales.

<https://nodejs.org>



# Requisitos

## TypeScript

- Lenguaje de programación desarrollado por Microsoft
- El código es predecible y fácil de depurar
- Hace fácil la organización del código en proyectos o apps grandes ya que es un lenguaje OO
- Transpila a JS atrapando todo tipo de errores antes de pasar a ejecución.

<https://www.typescriptlang.org>



# Requisitos

## **ECMAScript**

- Especificación de lenguaje de programación publicada por ECMA International.
- Actualmente está aceptado como el estándar ISO 16262.

<http://www.ecma-international.org/publications/standards/Ecma-262.htm>



# Características

## **SPA (Single Page Application)**

- Web de una sola página
- Navegación entre secciones y páginas, la carga de datos, se realiza de manera dinámica, asíncronamente haciendo llamadas al servidor (backend) sin actualizar la página en ningún momento.
- Lograr la mayor fluidez posible en UX
- La comunicación entre cliente y servidor se realiza de forma transparente al usuario
- Da una sensación de no abandonar la página
- Busca asimilar a las app de escritorio



# Características

## Cross Platform

- Progressive Web Apps.
- Native.
- Desktop.

## Speed and Performance

- Code Generation.
- Universal.
- Code Splitting.

## Productivity

- Templates.
- Angular CLI.
- IDEs.

## Full Development Story

- Testing.
- Animation.
- Accessibility.





# Características

## Data-Binding

- Vista y modelo en relación constante.
- Todo cambio visual se actualiza en tiempo real en el modelo y viceversa.
- Evita que el desarrollador esté encargado de la sincronía entre el modelo y la vista.



# Características

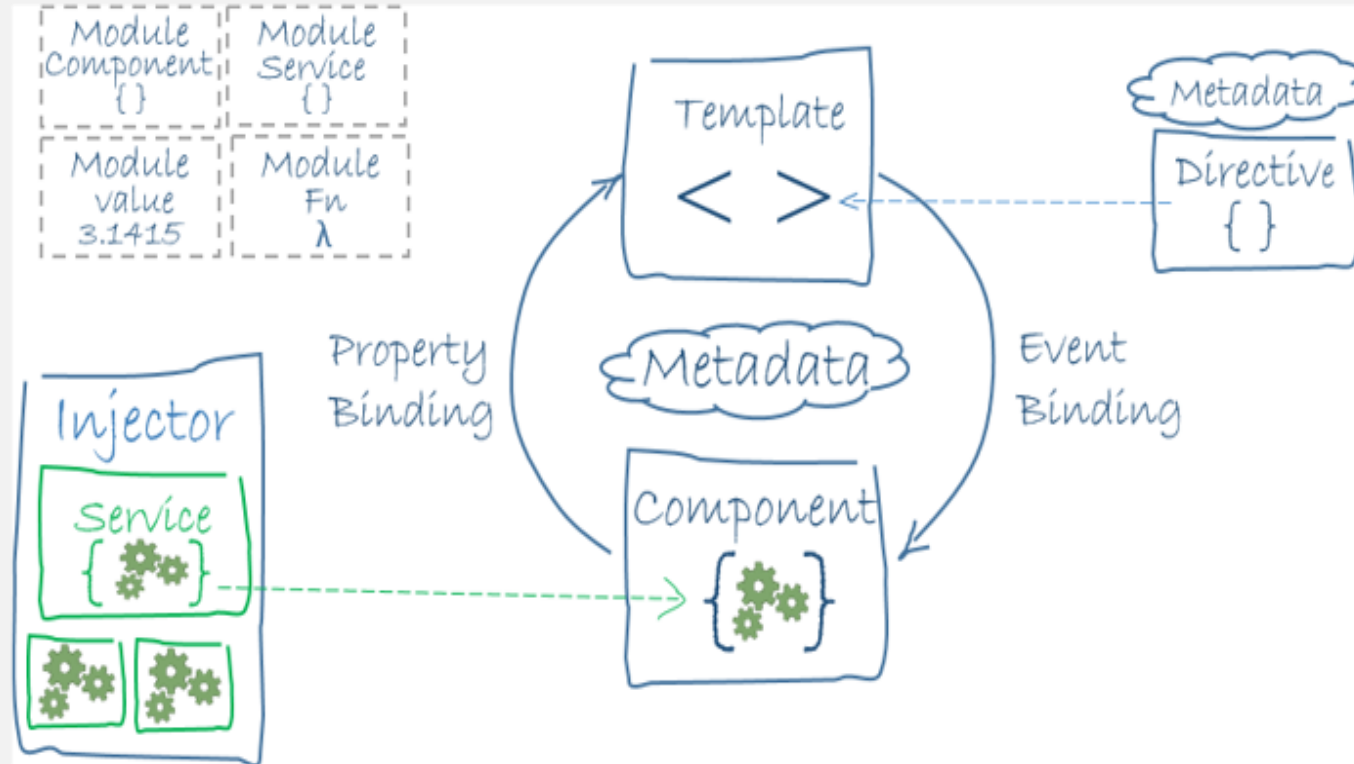
## Dependency Injection

- Es un patrón de diseño OO
- Nos dice que los objetos necesarios en una clase serán suministrados y que por lo tanto no necesitamos que la propia clase cree estos objetos.
- Ejemplo

Si necesitamos utilizar un servicio en un controlador, al crearlo debemos especificar nuestra dependencia al servicio y no crear un objeto del servicio.



# Arquitectura





# Módulos

- Facilitan la inyección de dependencias
- Módulos de ES: exportar e importar

```
//app/app.component.js
export class AppComponent {
  //aquí va la definición del componente
}
```

```
//app/main.js
import { AppComponent } from './app.component';
```

- ES6 <http://ccoentraets.github.io/es6-tutorial/>



# Decoradores

- Son funciones que modifican las clase de JS
- Añaden metadatos
- Le dicen a las clases como deben de trabajar.
- Ejemplo -> @Component



# Metadatos

- Le dice a Angular como procesar una clase
- Ejemplo:

```
@Component ({  
  selector: 'etiqueta para instanciar el componente',  
  templateUrl: 'ruta de la vista de este componente'  
})
```



# Librerías

- Colección de librerías JS
- Sólo importas lo que necesitas
- Ejemplo

@angular/core

Para importar Component

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



# Directivas

- Marcadores en un DOM que indican si un elemento tiene un comportamiento específico
- Manipular atributos
- Pequeña funcionalidad en forma de atributo HTML
- Ejemplo

`*ngIf=""`

`(click)=""`





# Plantillas

- También llamados “templates”
- Define la vista de un componente
- HTML (con algunas diferencias)



# Instalación

## Prerequisitos

NodeJS

<https://nodejs.org/es/download/>

```
npm -v  
node -v
```

Instalar globalmente el Angular CLI con el manejador de paquetes de node.

```
npm install -g @angular/cli@latest  
ng -version
```



# Creación de un proyecto

Crear un proyecto

```
ng new my-app
```

Iniciar la app

```
cd my-app
```

```
ng serve
```

```
npm start
```

¡Listo! Ahora abre tu navegador en:

<http://localhost:4200>

<http://127.0.0.1:4200>



# Instalar un proyecto

Para instalar un proyecto que bajamos de cualquier sistema de control de versiones o de la web y que no contenga la carpeta `node_modules` ejecutamos el siguiente comando.

```
npm install
```

Con este comando descargara las dependencias que están descritas en el archivo `package.json`



# Comandos adicionales del CLI



DGTIC

<https://cli.angular.io>

**ng new** genera un nuevo proyecto

<https://github.com/angular/angular-cli/wiki/new>

**ng serve** inicia el servidor

<https://github.com/angular/angular-cli/wiki/serve>

**ng generate** genera componentes, rutas, servicios pipes.

<https://github.com/angular/angular-cli/wiki/generate>

ng generate component

***--inline-style (-s)*** no genera un archivo css aparte.

***--inline-template(-t)*** no genera un archivo html aparte.

***--skipTests=true/false*** genera o no genera un archivo de prueba unitaria karma. Por default es true.



# Comandos adicionales del CLI



DGTIC

**ng build** compila la aplicación en un directorio.

<https://github.com/angular/angular-cli/wiki/build>

**--base-href** url base para la aplicación.

**--prod** configuración para producción. (Automáticamente optimiza para aot)

**--optimization** habilita optimizaciones.

**--build-optimizer** –habilita @angular-devkit/build-optimizer cuando se usa la opción aot

**--aot** compila usando Ahead of Time

Con AOT, el navegador descarga una versión precompilada de la aplicación. El navegador carga el código ejecutable para que pueda procesar la aplicación de inmediato, sin esperar a compilar la aplicación primero.

<https://angular.io/guide/aot-compiler>



Gracias

L.I. Hugo G Cuéllar Martínez.

[hugo.cuellar@comunidad.unam.mx](mailto:hugo.cuellar@comunidad.unam.mx)

Curso de NodeJS: <https://codigofacilito.com/cursos/nodejs>

URL Taller Angular código.

[https://github.com/comando132/node\\_redisybd](https://github.com/comando132/node_redisybd)

[https://github.com/comando132/angular\\_redisybd](https://github.com/comando132/angular_redisybd)