

NGULARJS  
by Google

# Bienvenidos !!!

## *Programación con Angular JS*

Instructor: Borja Cabeza

Herramientas: Adobe Brackets 

Donwload desde <http://brackets.io>

Duración: 24 Horas

# 1. Introducción a Angular JS

¿ Qué es Angular JS ?

¿ Qué es la arquitectura MVC ?

Angular JS vs jQuery

Angular JS vs bootstrap

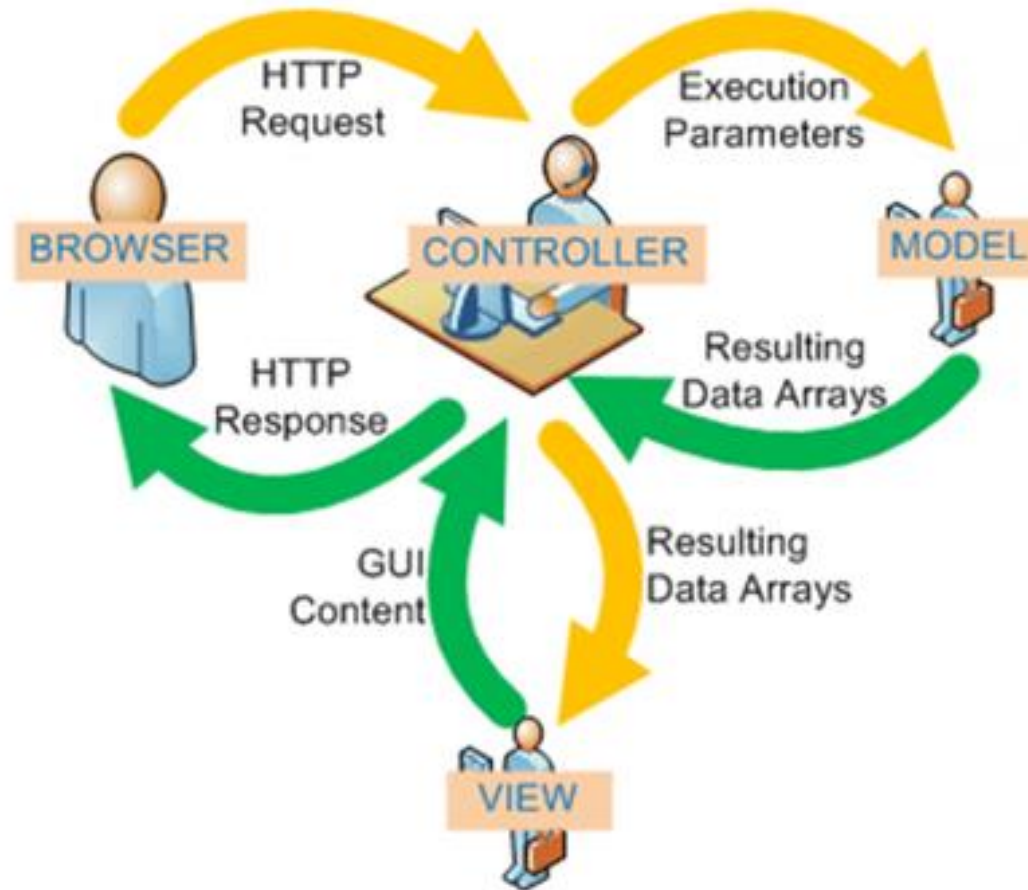
Angular JS vs IONIC

Angular JS vs ngCordova

Plantillas

# ¿ Qué es la arquitectura MVC ?

## Model-View-Controller



# Angular JS vs ...

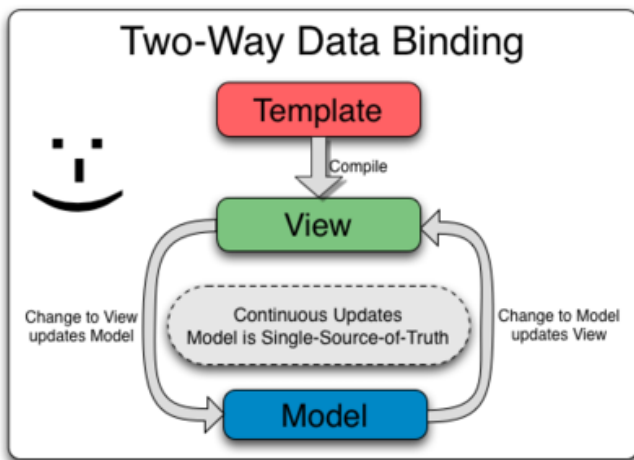


## 2. Características Angular JS

- Framework popular mantenido por Google
- Sin componentes gráficos
- Ligero y eficiente (155kb)
- Sintaxis sencilla
- Coexistencia con otros frameworks sin problemas de incompatibilidad
- Data Binding
- Inyección de dependencias
- Directivas

# Data Binding

Mediante el Data Binding Angular JS es capaz de transformar una plantilla estática en dinámica.



## Data Binding

- Une dos elementos en tiempo real
- El cambio en los datos se reflejan en tiempo real en la vista
- No requiere eventos o funciones adicionales para su funcionamiento

# Inyección de dependencias

Angular JS es modular. Se compone de una librería básica y otras extras especializadas.

## Index of /1.5.3/

---

<a href="#">../</a>	25-Mar-2016 20:35	-
<a href="#">docs/</a>	25-Mar-2016 20:35	-
<a href="#">i18n/</a>	25-Mar-2016 20:35	-
<a href="#">angular-1.5.3.zip</a>	25-Mar-2016 20:35	10250035
<a href="#">angular-animate.js</a>	25-Mar-2016 20:35	149421
<a href="#">angular-animate.min.js</a>	25-Mar-2016 20:35	25367
<a href="#">angular-animate.min.js.map</a>	25-Mar-2016 20:35	69998
<a href="#">angular-aria.js</a>	25-Mar-2016 20:35	14958
<a href="#">angular-aria.min.js</a>	25-Mar-2016 20:35	3807
<a href="#">angular-aria.min.js.map</a>	25-Mar-2016 20:35	8409
<a href="#">angular-cookies.js</a>	25-Mar-2016 20:35	9750
<a href="#">angular-cookies.min.js</a>	25-Mar-2016 20:35	1444
<a href="#">angular-cookies.min.js.map</a>	25-Mar-2016 20:35	3396
<a href="#">angular-csp.css</a>	25-Mar-2016 20:35	343
<a href="#">angular-loader.js</a>	25-Mar-2016 20:35	17202
<a href="#">angular-loader.min.js</a>	25-Mar-2016 20:35	1729
<a href="#">angular-loader.min.js.map</a>	25-Mar-2016 20:35	3838
<a href="#">angular-message-format.js</a>	25-Mar-2016 20:35	37844
<a href="#">angular-message-format.min.js</a>	25-Mar-2016 20:35	10306
<a href="#">angular-message-format.min.js.map</a>	25-Mar-2016 20:35	28130
<a href="#">angular-messages.js</a>	25-Mar-2016 20:35	27840
<a href="#">angular-messages.min.js</a>	25-Mar-2016 20:35	2911
<a href="#">angular-messages.min.js.map</a>	25-Mar-2016 20:35	7736
<a href="#">angular-mocks.js</a>	25-Mar-2016 20:35	103355
<a href="#">angular-resource.js</a>	25-Mar-2016 20:35	31396
<a href="#">angular-resource.min.js</a>	25-Mar-2016 20:35	4486
<a href="#">angular-resource.min.js.map</a>	25-Mar-2016 20:35	10975
<a href="#">angular-route.js</a>	25-Mar-2016 20:35	37889
<a href="#">angular-route.min.js</a>	25-Mar-2016 20:35	4582
<a href="#">angular-route.min.js.map</a>	25-Mar-2016 20:35	11513
<a href="#">angular-sanitize.js</a>	25-Mar-2016 20:35	25655
<a href="#">angular-sanitize.min.js</a>	25-Mar-2016 20:35	5828
<a href="#">angular-sanitize.min.js.map</a>	25-Mar-2016 20:35	9886
<a href="#">angular-scenario.js</a>	25-Mar-2016 20:35	1458347
<a href="#">angular-touch.js</a>	25-Mar-2016 20:35	26523
<a href="#">angular-touch.min.js</a>	25-Mar-2016 20:35	3942
<a href="#">angular-touch.min.js.map</a>	25-Mar-2016 20:35	10877
<a href="#">angular.js</a>	25-Mar-2016 20:35	1141759
<a href="#">angular.min.js</a>	25-Mar-2016 20:35	155877
<a href="#">angular.min.js.map</a>	25-Mar-2016 20:35	421029
<a href="#">errors.json</a>	25-Mar-2016 20:35	9444
<a href="#">version.json</a>	25-Mar-2016 20:35	307
<a href="#">version.txt</a>	25-Mar-2016 20:35	5

---

## Módulos

- Código fácil de mantener
- Aplicativos más ligeros y con mayor rendimiento
- Módulos a medida



### 3. Creando un Proyecto con Angular JS

- Instalación y descarga de Angular JS
- Configurar mi Aplicación Web utilizando **ng-app**
- Inicializar la Aplicación Web mediante la sentencia javascript ***angular.module***
- Crear las primeras expresiones **{{ }}**
- Crear la primera plantilla con expresiones dinámicas y configurar un controlador **ng-controller**
- Utilización del objeto **Scope**
- Utilización de la directiva **ng-model** y **ng-click**



# Laboratorio 1

Vamos a practicar como crear una aplicación y un controlador de Angular JS y el uso de expresiones. Utiliza las directivas *ng-app*, *ng-controller* y *ng-model*.



## Instrucciones para realizar el Laboratorio

- Completa el código Javascript
- Completa el código HTML



## Ficheros para realizar el Laboratorio

- Descarga de la URL:

<https://github.com/borjacabeza/Curso-Angular-JS/blob/master/Laboratorios/Lab1/index.html>

Tiempo Estimado: 20 minutos

## 4. Filtros en expresiones

- Sirven para dar formato a textos y número
- Sirven para buscar entre los datos manipulados
- Sirven para ordenar los datos manipulados
- Se pueden combinar hasta obtener el resultado deseado
- Se pueden crear filtros personalizados

Formato HTML: *{{ expresión | filtro | filtro }}*

Formato Javascript: *\$filter('filtro')()*



# Filtros para formatear contenido

- Usamos **lowercase** y **uppercase** para transformar alfanuméricos:

```
<p>{{ t1 | uppercase }}</p>
```

```
$scope.t1 = "Titulo de la Página";  
$scope.t2 = $filter('uppercase')($scope.t1);
```

- Usamos **currency** para la representación de divisas.
- Usamos **number** para la representación de números formateando su parte decimal.

# Filtros para formatear fechas y objetos

- Usamos **date** para formatear la representación de fechas:

```
<p>{{ fecha | date : format : timezone }}</p>
```

```
$scope.f1 = new Date();  
$scope.f2 = $filter('date')($scope.f1, 'dd/MM/yyyy');  
$scope.f3 = $filter('date')(new Date(), 'hh:mm:ss');  
$scope.f4 = $filter('date')(new Date(), 'medium');
```

- Usamos **json** para serializar un objeto javascript a JSON

# Filtros para representación de elementos

- Usamos **limitTo** para determinar el número de elementos que se representan:

```
<p>{{ expresión | limitTo : ítems : comienzo }}</p>
```

```
$scope.r1 = $filter('limitTo')('abcdef', 4);  
$scope.r2 = $filter('limitTo')([8, 9, 1, 0], 3, 1);
```

# Filtros personalizados

- Creamos un nuevo filtro personalizado utilizando el método *filter*.

```
var miApp = angular.module('filtros', []);

miApp.filter('capitalize' function () {
    var temp = function(datos, arg) {

        //Escribe el código de tu filtro

    };

    return temp;
});
```

# Laboratorio 2

Vamos a practicar como crear filtro personalizado en Angular JS. Utiliza el metodo *filter*.



## Instrucciones para realizar el Laboratorio

- Completa el código Javascript
- Utiliza tres parámetros. Uno para datos y dos para argumentos
- Controla el valor *undefined* de los parámetros



## Ficheros para realizar el Laboratorio

- Descarga de la URL:

<https://github.com/borjacabeza/Curso-Angular-JS/blob/master/Laboratorios/Lab2/index.html>

Tiempo Estimado: 30 minutos



# Filtros para búsqueda y filtrado de datos

- Usamos **filter** para la búsqueda o filtrado de los datos.
- Mediante data binding hacemos posible que sea en tiempo real.

```
{{ expresión | filter : expresión }}
```

- Utilizaremos la directiva *ng-repeat* para mostrar colecciones que posteriormente filtraremos



# Filtros para ordenar de datos

- Usamos **orderBy** para ordenar los datos.
- Mediante data binding hacemos posible que sea en tiempo real.

```
{{ expresión | orderBy : expresión : reverse }}
```

- Utilizaremos la directiva *ng-repeat* para mostrar colecciones que posteriormente filtraremos



## 5. Controladores de Angular JS

- Un aplicación puede tener uno o varios controladores
- Los controladores mantienen el principio de encapsulamiento frente a otros controladores de la aplicación
- Creamos un controlador con el método *controller*
- Activamos un controlador con la directiva *ng-controller*
- El objeto *\$scope* hace de enlace entre los controladores y las vistas
- Todos los controladores comparten el objeto *\$rootScope*



# Laboratorio 3

Vamos a practicar una aplicación con dos controladores y los objetos *\$scope* y *\$rootScope*.



## Instrucciones para realizar el Laboratorio

- Completa el código Javascript
- Añade las operaciones al array controlado por el *\$rootScope*
- Muestra los datos del array en el segundo controlado usando *ng-repeat*

## Ficheros para realizar el Laboratorio

- Descarga de la URL:

<https://github.com/borjacabeza/Curso-Angular-JS/blob/master/Laboratorios/Lab3/index.html>

Tiempo Estimado: 30 minutos

## 6. Gestión de eventos Angular JS

- Los eventos en Angular JS son controlados mediante directivas.
- Los eventos en Angular JS se enlazan con funciones definidas en el objeto *\$scope* del controlador

### Directivas para gestión eventos

*ng-click*: Cuando se produce el evento click.

*ng-dblclick*: Cuando se produce el evento doble click.

*ng-blur*: Cuando se produce el evento blur.

*ng-change*: Cuando se cambia el contenido de un <INPUT> o <SELECT> pero no cuando se cambia a consecuencias del propio modelo.

*ng-cut*: Cuando se produce el evento cut.

*ng-keydown*: Cuando se produce el evento keydown.

*ng-keyup*: Cuando se produce el evento keyup.

*ng-keypress*: Cuando se produce el evento keypress.

*ng-mousedown*: Cuando se produce el evento mousedown.

*ng-mouseenter*: Cuando se produce el evento mouseenter.

*ng-mouseleave*: Cuando se produce el evento mouseleave.

*ng-mousemove*: Cuando se produce el evento mousemove.

*ng-mouseover*: Cuando se produce el evento mouseover.

*ng-mouseup*: Cuando se produce el evento mouseup.



# Laboratorio 4

Vamos a practicar una aplicación la gestión de diferentes eventos.



## Instrucciones para realizar el Laboratorio

- Completa el código Javascript
- Añade las directivas necesarias para controlar los eventos



## Ficheros para realizar el Laboratorio

- Descarga de la URL:

<https://github.com/borjacabeza/Curso-Angular-JS/blob/master/Laboratorios/Lab4/index.html>

Tiempo Estimado: 30 minutos

## 7. Directivas de Angular JS

- Son atributos insertados en las etiquetas HTML que habilitan conductas complejas para la etiqueta y su contenido
- Para pasar las validación de W3C añadimos el prefijo *data-* al nombre de las directivas ya que realmente se trata de atributos personalizado de HTML

[http://validator.w3.org/#validate by input](http://validator.w3.org/#validate_by_input)

- La directiva *ng-bind* cumple la misma función que `{{ }}`
- Podemos crear directivas personalizadas y definir como se deben utiliza, *A*tributo, *E*tiqueta, *C*lase o *M*Comentario.



# Directivas más comunes en Angular JS

- Usamos *ng-app* para declarar una aplicación.
- Usamos *ng-controller* para definir el trozo de HTML afectado o contralo por un controlador
- Usamos *ng-model* para enlazar elementos del modelo con <INPUT> o elementos de formulario
- Usamos *ng-bind* para indicar contenidos de las etiquetas HTML, igual que *{{ }}*
- Usamos *ng-repeat* para repetir código HTML en función del contenido de un array de elementos
- Usamos *ng-show* para mostrar o ocultar elementos





# Directivas más comunes en Angular JS II

- Usamos *ng-disable* y *ng-readonly* para habilitar o deshabilitar elementos HTML
- Usamos *ng-maxlength* y *ng-minlength* para el número de caracteres de un `<INPUT>`
- Usamos *ng-pattern* para indicar expresiones regulares de un `<INPUT>`
- Usamos *ng-true-value* y *ng-false-value* para especificar valores en los `<INPUT>` de tipo *checked*
- Usamos *ng-options* para rellenar un `<SELECT>`



# Directivas más comunes en Angular JS III

- Usamos *ng-if* muestra un elemento si es *true* o no lo muestra si el *false*
- Usamos *ng-switch* determinar que conjunto de código HTML se muestra en función de un valor

