

Desarrollo de Aplicaciones Web

JavaScript. AngularJS.



AngularJS

AngularJS es una librería escrita en JavaScript, desarrollada en 2012 por Miško Hevery, empeado de Google

Actualmente es “soportada” por Google, como uno de sus proyectos.

La librería se integra con el HTML mediante etiquetas `<script>`:

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.7.5/angular.min.js">
</script>
```

O bajándola del repositorio:

```
<script src="angular.min.js"></script>
```

AngularJS Directivas

AngularJS amplía el uso de etiquetas HTML mediante **directivas y expresiones**.

Directivas AngularJS → Atributos HTML con prefijo ng.

- **ng-app** → Define el ámbito de extensión de una aplicación AngularJS.
- **ng-init** → Inicializa una variable de la aplicación a un valor.
- **ng-model** → Establece un enlace entre una etiqueta de control HTML (formulario.- input, select, textarea) y un dato de la aplicación.
- **ng-bind** → Establece un enlace entre un dato de la aplicación y el innerHTML de una etiqueta HTML de “vista”.

```
<div ng-app ng-init= "name='ETSE'">  
  <p>Introduce un nombre en la caja de texto:</p>  
  <p>Nombre: <input type="text" ng-model= "name"></p>  
  <p ng-bind= "name"></p>  
</div>
```

Introduce un nombre en la caja de texto:

Nombre:

USC

AngularJS Otras directivas

- **ng-repeat** → Clona etiquetas HTML

```
<div ng-app="" ng-init="nombres=['USC','ETSE','ETIS']">
  <ul>
    <li ng-repeat="x in nombres">
      {{ x }}
    </li>
  </ul>
</div>
```

- USC
- ETSE
- ETIS

- **.directive** → Crea una nueva directiva propia. Puede ser invocada a nivel de etiqueta, atributo, clase o comentario, pudiendo ser restringida a uno de esos ámbitos

```
<body ng-app="myApp">
  <n-Directiva></n-Directiva>
  <script>
    var app = angular.module("myApp", []);
    app.directive("nDirectiva", function() {
      return {
        restrict: "E",
        template : "<h1>Nueva Directiva!</h1>"
      };
    });
  </script>
</body>
```

Se escribe en notación *camel* → *nDirectiva*
Se invoca separando por guiones → *n-Directiva*

Nueva Directiva

AngularJS Expresiones

Expresiones AngularJS → Mecanismo de AngularJS para ejecutar acciones y visualizar el resultado de dicha ejecución.

- Las expresiones AngularJS se escriben entre llaves dobles **{{ exp }}**.

```
<div ng-app>
  <p> Mi primera expresión {{ 1 + 1 }} </p>
</div>
```

Mi primera expresión 2

```
<div ng-app="">
  <p>Nombre: <input type="text" ng-model="name"></p>
  <p>{{name}}</p>
  <p ng-bind="name"></p>
</div>
```

Nombre:

Hola DAW

Hola DAW

```
<div ng-app ng-init="miColor='red'">
  <input style="background-color:{{miColor}}" ng-model="mColor">
  <h1 ng-bind="mColor" style="color:{{mColor}}"></h1>
</div>
```

blue

blue

AngularJS Tipos de Datos

Números en AngularJS

```
<div ..... ng-init= "valor1= 10; valor2= 20">  
<p> El valor total es: {{ valor1 + valor2 }} </p>
```

Cadenas en AngularJS

```
<div ..... ng-init= "valor1= 'uno'; valor2= 'dos'">  
<p> El nombre completo es: {{ valor1 + " " + valor2 }} </p>
```

Arrays en AngularJS

```
<div ..... ng-init= "valor= [1, 2, 3, 4]">  
<p> El valor es: {{ valor[1] }} </p>
```

Objetos AngularJS

```
<div ..... ng-init= "obj= {valor1= 'uno', valor2= 'dos'}">  
<p> El dato es: {{ obj.valor1 }} </p>
```

AngularJS Modulos y Controladores

Módulos AngularJS → Constituyen el contenedor para las diferentes partes de una aplicación.

```
<div ng-app="miApl">...</div>
<script>
    var apl = angular.module("miApl", []);
</script>
```

Controladores AngularJS → Objetos JavaScript que sirven para manejar los datos de una aplicación. Los controladores pertenecen siempre a un modulo.

```
<div ng-app="miAp" ng-controller="miCtr">
    {{ nombre + " " + apellido }}
</div>
```

```
<script>
    var app = angular.module("myApp", []);
    app.controller("myCtrl", function($scope) {
        $scope.nombre = "Pablo";
        $scope.apellido = "García";
    });
</script>
```

\$scope → Objeto AngularJS usado con controladores para empaquetar datos y funciones.

```
<script src="miAp.js"></script>
<script src="miCtr.js"></script>
```

```
miAp.js
    var app = angular.module("myApp", []);

miCtr.js
    app.controller("miCtrl", function($scope) {
        $scope.nombre = "Pablo";
        $scope.apellido = "García";
    });
```

AngularJS Filtros

Filtros AngularJS → Mecanismo que permite formatear datos de salida.

- **currency** → Formatea a tipo de moneda
- **date** → Formatea a tipo fecha
- **filter** → Selecciona items de un array
- **json** → Formatea un objeto a una cadena JSON
- **limitTo** → Limita un array/cadena a un número determinado de elementos/caracteres
- **lowercase** → Pasa cadena a minúsculas
- **number** → formatea una cadena como un número
- **orderBy** → Ordena un array, según una expresión
- **uppercase** → Pasa cadena a mayúsculas

```
<div ng-app="miApp" ng-controller="ctr">
  <p>El apellido es: {{ apellido | uppercase }} </p>
</div>

<script>
  angular.module('miApp', []).controller('ctr', function($scope) {
    $scope.nombre = "Pablo",
    $scope.apellido = "García"
  });
</script>
```

El apellido es: GARCÍA

AngularJS vs MVC

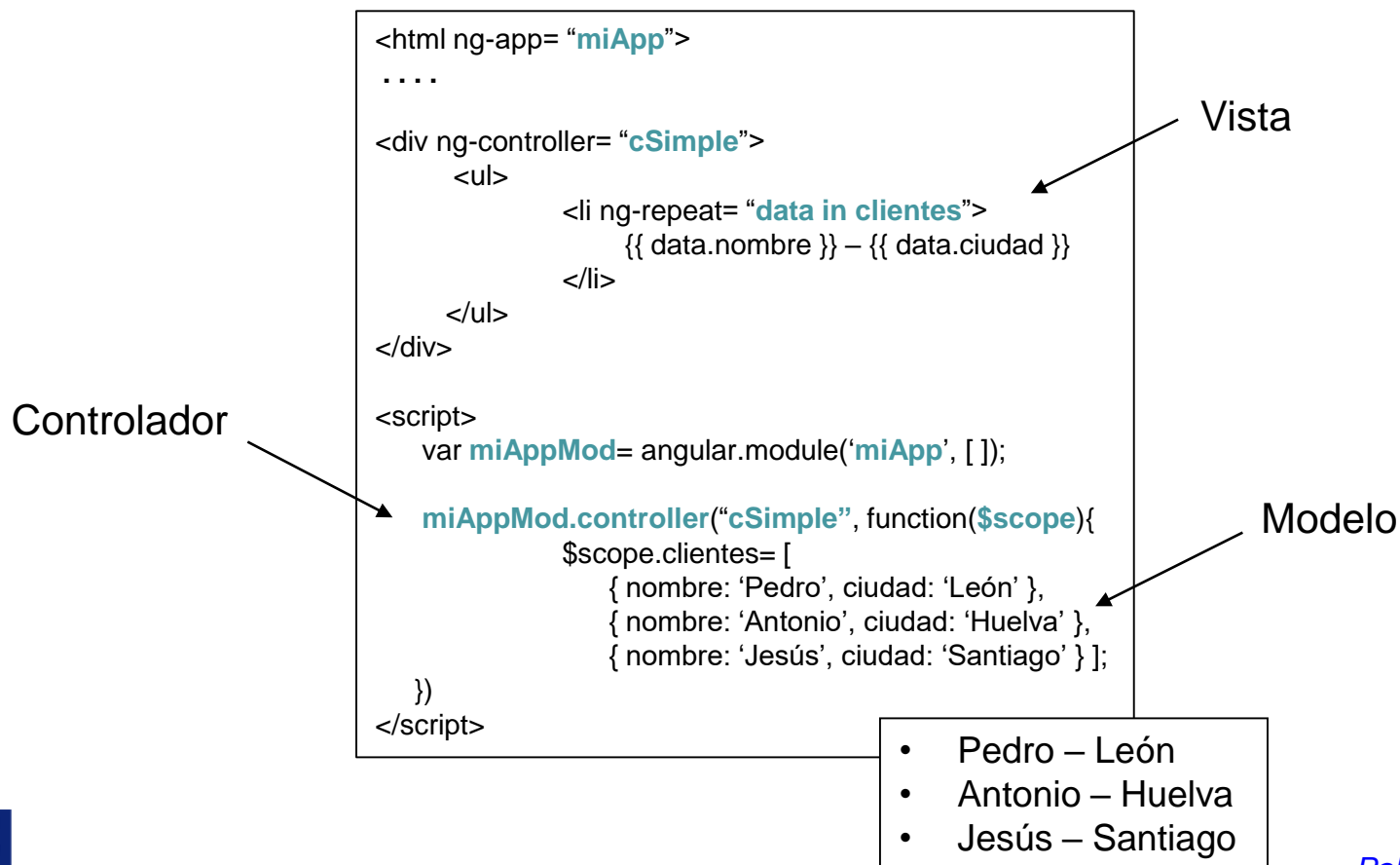


`$scope` → Juega el papel de “modelo” para la “vista” (los datos a visualizar).

El `controlador` sirve de fuente de `datos` para `vista`, sin necesidad de saber nada de cómo esos datos van a ser visualizados.

AngularJS vs MVC

El **controlador** sirve de fuente de **datos** para **vista**, sin necesidad de saber nada de cómo esos datos van a ser visualizados.



AngularJS Routing

Mecanismo que permite la navegación por diferentes páginas, manteniendo una estructura SPA (*Single Page Application*).

Utiliza el módulo *route* de *AngularJs*, que se ha de invocar aparte:

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular-route.js">
</script>
```

Se invoca → `var app = angular.module("miApp", ["ngRoute"]);`

Utiliza el objeto **\$routeProvider** para encapsular las diferentes rutas

Utiliza la directiva **ng-view** para visualizar el contenido

AngularJS Routing

Modulo *ngRoute*

```
<body ng-app="miApp">
  <p><a href="#/">Ciudad</a></p>
  <a href="#usc">Universidad</a>
  <a href="#etse">Centro</a>
  <div ng-view></div>
</body>
```

```
<script>
app.controller("mainCtrl", function($scope) {
  $scope.msg= "Santiago de Compostela es la capital de Galicia";
});
app.controller("uscCtrl", function($scope) {
  $scope.msg= "La USC es la universidad de Santiago";
});
app.controller("etseCtrl", function($scope) {
  $scope.msg= "La ETSE es la escuela de ingenieria de la unive";
});
</script>
```

```
<script>
var app = angular.module("miApp", ["ngRoute"]);
app.config(function($routeProvider) {
  $routeProvider
    .when("/", {
      templateUrl : "main.html",
      controller : "mainCtrl"
    })
    .when("/usc", {
      templateUrl : "usc.html",
      controller : "uscCtrl"
    })
    .when("/etse", {
      templateUrl : "etse.html",
      controller : "etseCtrl"
    });
});
</script>
```

Ciudad

Universidad Centro

Santiago de Compostela

AngularJS Eventos

Es posible añadir a cada etiqueta HTML eventos en forma de directivas *AngularJS*

- *ng-blur*
- *ng-change*
- *ng-click*
- *ng-copy*
- *ng-cut*
- *ng-dblclick*
- *ng-focus*
- *ng-keydown*
- *ng-keypress*
- *ng-keyup*
- *ng-mousedown*
- *ng-mouseenter*
- *ng-mouseleave*
- *ng-mousemove*
- *ng-mouseover*
- *ng-mouseup*
- *ng-paste*

```
<div ng-app="miApp" ng-controller="miCtrl">
  <h1 ng-click="count = count + 1">Pulsar</h1>
  <p>{{ count }}</p>
</div>

<script>
  var app = angular.module('miApp', []);
  app.controller('miCtrl', function($scope) {
    $scope.count = 0;
  });
</script>
```

Pulsar

Número de veces pulsado: 4

AngularJS Servicios

Función *JavaScript* responsable de realizar una tarea concreta dentro de una aplicación *AngularJS*.

AngularJS dispone de 30 [servicios propios](#), aunque el usuario puede crear los suyos a mayores. Todos los servicios propios van precedidos del símbolo \$.

Existen dos mecanismos para crear servicios:

- Método *factory*
- Método *service*

AngularJS Servicios

Método *factory*. - Se crea un objeto dentro de una *factoría* y se devuelve para su uso mediante etiquetas HTML.

```
miApp.factory('serviceName', function(){
    var factoryName={};

    factoryName.metodoName()= function(par_1, par_2, ..., par_n){
        .....
        return result;
    }
    return factoryName;
})
```

AngularJS Servicios

Método *factory*.- Se crea un objeto dentro de una *factoría* y se devuelve para su uso mediante etiquetas HTML.

```
<div ng-app= "miApp" ng-controller= "miCtr">
  <p>Introduce un valor: <input type="number" ng-model="number"/></p>
  <h3 ng-click= "square()">Pulsar</h3>
  <p>Resultado: {{result}}</p>
</div>
```

```
<script>
  var miApp= angular.module("miApp", []);

  miApp.factory('fService', function() {
    var factoryName= {};
    factoryName.cuadrado= function(a){
      return a*a
    }
    return factoryName;
  });

  miApp.controller('miCtr', function($scope, fService){
    $scope.square= function(){
      $scope.result= fService.cuadrado($scope.number);
    }
  });
</script>
```

AngularJS Service con Factory

Introduce un valor:

Pulsar

Resultado: 25

AngularJS Servicios

Método *service*.- Se crea un servicio mediante el método *service* y se le define un método para que realiza algún tipo de funcionalidad.

```
miApp.service('serviceName', function(){  
    this.functionName= function(par_1, par_2, ..., par_n){  
        .....  
        return result;  
    }  
})
```

AngularJS Servicios

Método *service*.- Se crea un servicio mediante el método *service* y se le define un método para que realiza algún tipo de funcionalidad.

```
<div ng-app= "miApp" ng-controller= "miCtr">
  <p>Introduce un valor: <input type="number" ng-model="number"/></p>
  <h3 ng-click= "square()">Pulsar</h3>
  <p>Resultado: {{result}}</p>
</div>
```

```
<script>
  var miApp= angular.module("miApp", []);

  miApp.service('fService', function(){
    this.cuadrado= function(a) {
      return a*a;
    }
  });

  miApp.controller('miCtr', function($scope, fService){
    $scope.square= function(){
      $scope.result= fService.cuadrado($scope.number);
    }
  });
</script>
```

AngularJS Service con Service

Introduce un valor:

Pulsar

Resultado: 16

AngularJS Servicios

Servicios propios → \$http.

\$http.- permite la comunicación con el servidor vía protocolo http para la lectura de datos remotos mediante diferentes métodos asociados:

.delete()
.get()
.head()
.jsonp()
.patch()
.post()
.put()

```
<div ng-app="miApp" ng-controller="miCtr">
  <ul>
    <li ng-repeat="x in miDato">
      {{ x.Nombre + '--> ' + x.Ciudad + ', ' + x.Pais }}
    </li>
  </ul>
</div>
```

```
<script>
  var app = angular.module('miApp', []);
  app.controller('miCtr', function($scope, $http){
    $http.get("prueba.txt").then(function(response){
      $scope.miDato= response.data.registros;
    });
  });
</script>
```

- Pedro Suárez--> Santiago, España
- Ana Gómez--> Londres, UK
- Petter Church--> New York, USA