TEMA 1

Daniel Estévez  @  & 

 & 

Plantillas (template)

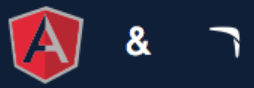
- En Angular, un fichero html con markup añadido se le llama “template”

```
<body ng-app="gnaApi">
    <content></content>
    <foot></foot>
    <script src="scripts/vendor/angular.js"></script>

    <!-- API MODULE→
    <script src="scripts/gnaApi.js"></script>
    <script src="scripts/directives.js"></script>
    <script src="scripts/services.js"></script>
    <!-- -->
</body>
```

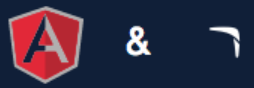
Vistas (view)

- Cuando Angular carga la aplicación, parsea este nuevo markup de los templates usando el “compilador” ($compiler).
- El cargado, transformado y renderizado DOM es a lo que llamamos vista



Directivas (directives)

- El primer tipo de markup son las directivas.
- Añaden un comportamiento a elementos o atributos en el HTML



Directivas (directives)

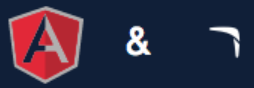
- El atributo ng-app esta linkado a una directiva que directamente inicializa la aplicación Angular
- La directiva ng-model guarda/actualiza el valor del campo de un input en una variable

```
<body ng-app>
     <h3> Primera App Angular: Sumador </h3>
     Numero A.<input type="number" autocomplete="off" ng-model="numberA"/>
     Numero B.<input type="number" autocomplete="off" ng-model="numberB"/>
     <div id="display">
          <div id="suma"><h3> A + B: {{numberA + numberB}} </h3></div>
     </div>
</body>
```

Directivas (directives)

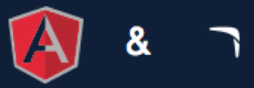
- Angular ofrece un montón de directivas
- ngRepeat instancia un template por elemento en una colección
- Cada instancia del template tiene su propio ámbito de aplicación

```
<ul class="nav navbar-nav nav-cover">
    <li class='{{ section.animation }}' ng-repeat="section in sections">
        <a href="{{section.hash}}">{{ section.text }}</a>
    </li>
</ul>
```



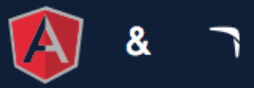
Expresiones ({{}})

- El segundo tipo de nuevo markup es la doble llave
- Cuando el compilador encuentra este markup, evalúa la expresión de dentro y lo reemplaza por esta.
- Una expresión en un template en un fragmento de código javascript que permite leer y escribir en variables
- Dichas variables no son globales, sino que pertenecen a un $scope determinado.



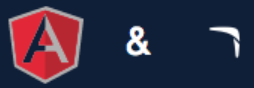
Scope ($scope)

- Igual que las variables dentro de una función javascript viven en un scope, Angular provee un scope para las variables, accesible en las expresiones.



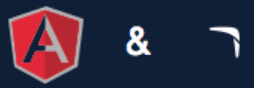
Modelo (model)

- Y nos referimos a los valores almacenados en las variables del scope como el modelo.



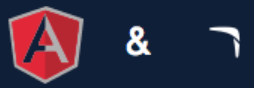
Data Binding

- El data-binding en Angular es la sincronización automática de los datos entre los componentes del modelo y la vista
- La forma en que implementa angular de enlace de datos le permite tratar el modelo como la única fuente-de-verdad
- La vista es una proyección del modelo en todo momento Cuando el modelo cambia, la vista refleja el cambio, y viceversa.



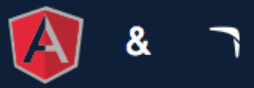
Data Binding

- Clásico
  - La mayoría de los sistemas de plantillas de datos se unen en una sola dirección
  - Se funde el template y el modelo en la vista
  - Después, los cambios en el modelo o secciones relacionadas de la vista no se reflejan automáticamente en la vista



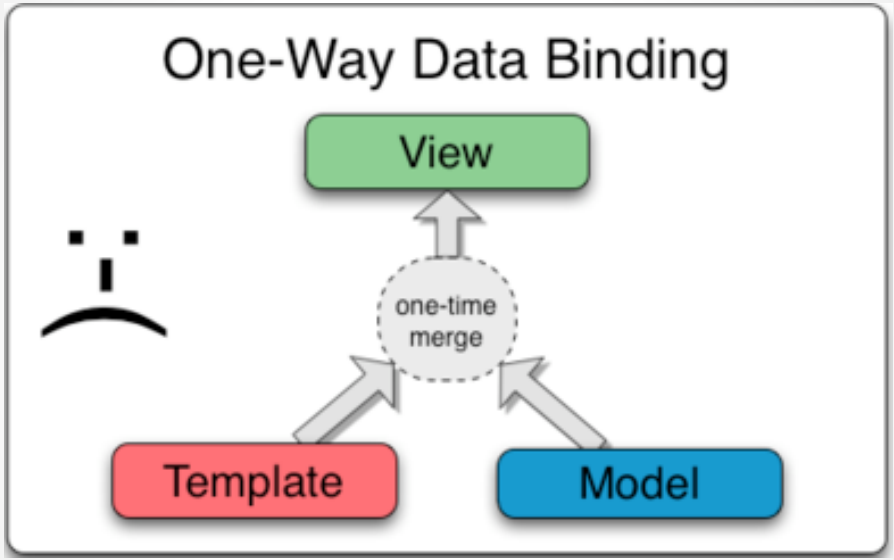
Data Binding

- Clásico
  - Peor aún, los cambios que realice el usuario a la vista no se reflejan en el modelo.
  - El desarrollador tiene que escribir el código que se sincroniza constantemente la vista con el modelo y el modelo con la vista.



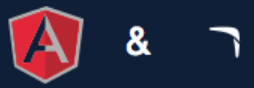
Data Binding

- Clásico



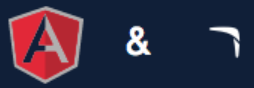
Data Binding

- Angular
  - En Angular es diferente
  - El template (HTML sin compilar) se compila en el navegador y se produce una vista en vivo.
  - Cualquier cambio a la vista se reflejan inmediatamente en el modelo, y cualquier cambio en el modelo se propagan a la vista.



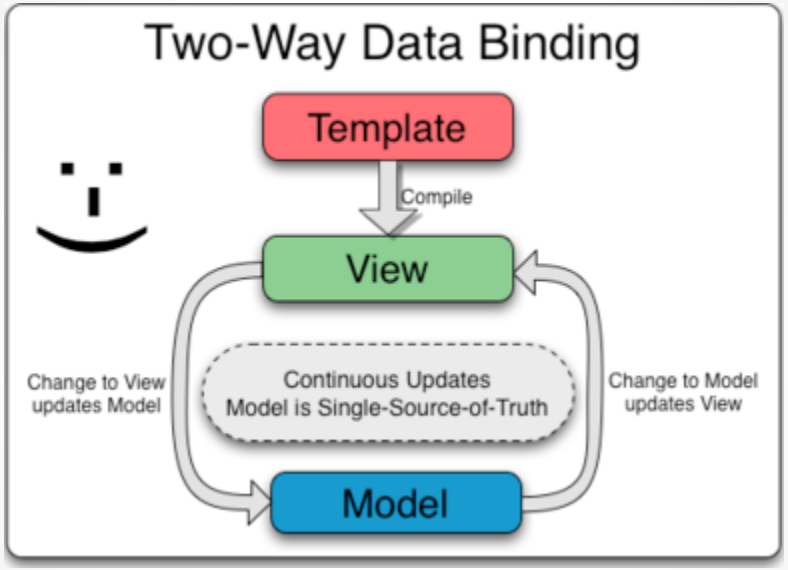
Data Binding

- Angular
  - La vista es simplemente una proyección del modelo, el controlador está completamente separada de la vista y la desconoce.
  - Esto hace más fácil de testear el controlador de manera aislada sin la vista y la dependencia DOM/navegador relacionado.



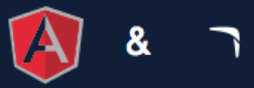
Data Binding

- Angular



Creando una app Angular

- Para crear una aplicación Angular necesitamos 2 cosas:
  - Tener Angular cargado en nuestro index
  - Añadir la directiva ngApp a nuestro markup



Ejercicios

1. Utilizando "primera-app-mio.html", crea una app Angular
2. Utiliza ngModel para comprobar cómo funciona el maravilloso doble data binding de angular
3. Añade alguna expresión que evalúe esos valores que hemos recogido gracias a ngModel