

Introducción a AngularJS



¿Qué es AngularJS?

Es un proyecto de **código abierto**, realizado en **JavaScript**, (desarrollado, mantenido e impulsado por la gente de Google) que contiene un conjunto de **librerías** útiles para el desarrollo de **aplicaciones web** y propone una serie de **patrones de diseño** para llevarlas a cabo.

En pocas palabras, es lo que se conoce como un **framework** para el desarrollo, en este caso sobre el lenguaje **JavaScript** con programación del lado del cliente.

¿Qué es un framework?

La palabra inglesa "framework" (marco de trabajo) define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

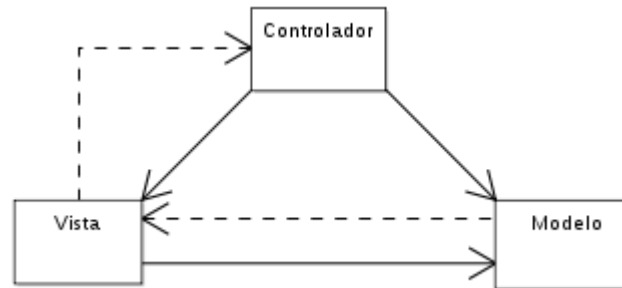
En el desarrollo de software, un framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

Características de un framework

- Inversión de Control
- Comportamiento por defecto
- Extensibilidad
- Son incompletos, no usables al 100%, deben ser completados
- Dirigen la solución o proyecto

Patrón de diseño MVC



El **modelo–vista–controlador (MVC)** es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el **modelo**, la **vista** y el **controlador**, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

Patrón de diseño MVC en AngularJS

Vistas: Será el HTML y todo lo que represente datos o información.

Controladores: Se encargarán de la lógica de la aplicación y sobre todo de las llamadas "Factorías" y "Servicios" para mover datos contra servidores o memoria local en HTML5.

Modelo de la vista: En Angular el "Modelo" es algo más de aquello que se entiende habitualmente cuando te hablan del MVC tradicional, o sea, las vistas son algo más que el modelo de datos. Toda esa información, que es útil para el programador pero que no forma parte del modelo del negocio, es a lo que llamamos el "scope" que es el modelo en Angular.

¿Qué ofrece AngularJS?

Client-side template

El sistema de plantillas en AngularJS es diferente del utilizado en otros frameworks. Por lo general es el servidor el encargado de mezclar la plantilla con los datos y devolver el resultado al navegador. En AngularJS el servidor proporciona los contenidos estáticos (plantillas) y la información que se va a representar (modelo) y es el cliente el encargado de mezclar la información del modelo con la plantilla para generar la vista.

Data binding

Con AngularJS podemos sincronizar el modelo y la vista automáticamente utilizando ciertas **directives** del framework. Esta sincronización es bidireccional, es decir, la información se sincroniza tanto si cambia el valor en la vista como si lo hace el valor en el modelo.

¿Qué ofrece AngularJS? (Cont.)

Directives

Las directives son el plato fuerte de AngularJS. Mediante el uso de las mismas podemos extender la sintaxis de HTML y darle el comportamiento que deseemos. Podemos crear **directives** a nivel de elemento, de atributo, de clase y de comentario.

Filters

Los **filters** nos permiten modificar el modo en el que se va a presentar la información al usuario.

La utilización de los mismos es similar a los Pipeline de Unix:

```
{{ expresion | filtro }}
```

Donde *expresion* puede ser cualquier tipo de expresión de AngularJS, como una variable del *\$scope*, y *filtro* el nombre del filtro que le queremos aplicar a la expresión.

¿Qué ofrece AngularJS? (Cont.)

Services

Los **services** son los encargados de comunicarse con el servidor para enviar y obtener información que después será tratada por los **controllers** para mostrarla en las vistas.

Esta parte es más compleja de explicar con un ejemplo, por el momento nos basta con saber que los **services** se pueden dividir en tres categorías: **services**, **factories** y **providers**.

Uno de los **services** incluidos en el framework es [\\$resource](#), el cual nos permite encapsular la interacción con servicios RESTful sin tener que tratar directamente con las llamadas http.

Dos mundos en AngularJS



Ahora tenemos que examinar AngularJS bajo otra perspectiva, que nos facilite entender algunos conceptos y prácticas habituales en el desarrollo. Para ello dividimos el panorama del framework en dos áreas.

Parte del HTML: Es la parte declarativa, con las vistas, así como las directivas y filtros que nos provee AngularJS, así como los que hagamos nosotros mismos o terceros desarrolladores.

Parte Javascript puro: Que serán los controladores, factorías y servicios.

AngularJS - Recursos

Sitio Oficial: <https://angularjs.org/>

Libros:

[AngularJS](#) by Brad Green and Shyam Seshadri

[Mastering Web App Development](#) by Pawel Kozlowski and Pete Bacon Darwin

[AngularJS Directives](#) by Alex Vanston

[Recipes With AngularJS](#) by Frederik Dietz

[Developing an AngularJS Edge](#) by Christopher Hiller

[ng-book: The Complete Book on AngularJS](#) by Ari Lerner

[AngularJS : Novice to Ninja](#) by Sandeep Panda