

# Guía Básica De AngularJS

De vez en cuando se anuncia una nueva herramienta y así  como «nace» de repente, se olvida. El lanzamiento de **AngularJs**, el cual ha estado presente desde el año 2009 desde su creación por **Misko Hevery**, ha estado ganando bastante atención por parte de los programadores y desarrolladores en los últimos meses.

La gente está hablando bastante de este lenguaje de programación, los [desarrolladores](#) lo han ido integrando en sus desarrollos, y diversos autores han escrito libros sobre el tema. Entonces, ¿Qué es AngularJS y por qué deberías de usarlo en todo desarrollo web? ¿Es un cambio que vale la pena hacer? Claro que lo es! Deja que te guíemos en el por que debes hacerlo ;)

Antes de continuar, es recomendable que te sientas cómodo con JavaScript antes de comenzar de lleno en AngularJS. Si no estás familiarizado con MVC y [DOM](#), te sugerimos leer más sobre estos paradigmas antes de seguir adelante, de lo contrario es posible que te confundas con la mayor parte de la terminología utilizada en este artículo.

## ¿Qué es AngularJS?

AngularJS no es más que otro **Framework de JavaScript**. Claro, también puedes usar Backbone, Ember, y el más usado [jQuery](#), pero **AngularJS** es diferente en muchos sentidos.

# Unión De Datos Y Desarrollo Para Aplicaciones De Una Sola Página (SPA)

En primer lugar, **AngularJS es un Framework** que enlaza datos, que se construye específicamente para las [SPAs](#). Esto quiere decir que puedes desarrollar fácilmente una aplicación sin necesidad de usar cualquier otra biblioteca, puesto que ya tiene todo lo que necesitas. También mantiene la sincronización para el modelo y la vista.

La belleza del **desarrollo de un SPA** es que imita una experiencia de escritorio en el que la página sigue siendo el mismo en un todo, con sólo las vistas que están cambiando junto con la URL. Es como si acabaras de abrir una aplicación de escritorio y tener todo lo que necesitas.

Otra cosa es que a diferencia de otros SPA, el historial de navegación se mantiene. Por ejemplo, si deseas hacer clic en el botón Atrás para volver a la vista anterior, AngularJS en realidad le llevará de nuevo a la vista anterior. La mayoría de las SPA no funcionan de esta manera.

## Implementación Del Modelo-Vista-

# Controlador

AngularJS implementa el paradigma de MVC de una manera hermosa. La mayoría de los Framework que usan MVC requieren separar la aplicación en módulos y luego escribir el código que los unirá. Mientras que el razonamiento detrás de esto es hacer que el código funcione más flexible y sea reutilizable, esto lleva a muchos horrores de codificación, especialmente para los desarrolladores perezosos.

# Animación

Por supuesto, una aplicación de una sola página no puede verse bien sin las apropiadas animaciones. Como se mencionó anteriormente, AngularJS es un [Framework con múltiples funciones](#) que tiene todas las funciones que necesitas en la construcción de aplicaciones. Como tal, proporciona una manera fácil de introducir animación en todas las vistas de la misma manera como lo hace jQuery. He [aquí](#) un buen ejemplo de cómo AngularJS maneja animaciones.

Pero eso es sólo la superficie de AngularJS. Aquí hay más de lo que puede hacer:

- Validación de datos
- Inyección de librerías
- Manejar lógica personalizada
- Directivas de múltiples elementos
- Compartir datos entre los controladores
- Mejorar el HTML
- Manipulación DOM con la ayuda de jqLite
- AJAX
- Pruebas
- y muchos más...

# Una Comparación

Ahora, te invitamos a dar un vistazo de **como funciona AngularJS** comparándolo con JavaScript regular y jQuery.

## JavaScript

Sin el uso de cualquier biblioteca JavaScript, así es como se ve cuando muestra los datos de entrada en tiempo real.

```
[html]

<html>
<head>
<title>JavaScript</title>
</head>

<body>
Name: <input id="textInput" type="text" />
<br/>
Your name is <span id="name"></span>

<script>
var textInputElement = document.getElementById('textInput'),
nameElement = document.getElementById('name');
textInputElement.addEventListener('keyup',function(){
var text = textInputElement.value;
nameElement.innerHTML = text;
});
</script>
```

```
</body>
```

```
</html>
```

```
[/html]
```

# jQuery

Con jQuery, la visualización de los datos de entrada se hace mucho más simple ya que la mayoría de la comunicación es [manejada por jQuery](#). Por lo tanto, permite escribir menos código.

```
[html]
```

```
<html>
```

```
<head>
```

```
<title>jQuery</title>
```

```
<script
```

```
src="http://code.jquery.com/jquery-2.1.1.min.js"></script>
```

```
</head>
```

```
<body>
```

```
Name: <input id="textInput" type="text"/>
```

```
<br/>
```

```
Your name is <span id="name"></span>
```

```
<script>
```

```
$('#textInput').on('keyup',function(){
```

```
$('#name').html($('#textInput').val());
```

```
});
```

```
</script>
```

```
</body>
```

```
</html>
```

```
</ html>
```

```
[/html]
```

# AngularJS

AngularJS lleva todo a un nivel superior. No sólo es un Framework ligero, la forma de escribir el código HTML se hace también más fácil.

```
[html]
```

```
<!DOCTYPE html>
```

```
<html ng-app>
```

```
<head>
```

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

```
<title>AngularJS</title>
```

```
</head>
```

```
<body>
```

```
Name: <input type="text" ng-model="name" />
```

```
<br />
```

```
Your name is {{ name }}
```

```
</body>
```

```
</html>
```

```
[/html]
```

Como se ha demostrado anteriormente, la belleza de AngularJS radica en escribir menos código, **manteniendo la integridad de la aplicación**. Otra cosa notable es que no tienes que manipular los controladores con el fin de realizar cambios en la vista.

# Limitaciones de AngularJS

AngularJS no es todo un camino de arco iris. Si estás apuntando para crear una aplicación que hace cálculos simples, como una calculadora, un juego de rompecabezas, animaciones, formas dinámicas y similares, entonces **AngularJS es el Framework** que buscas.

Sin embargo, si estás construyendo una aplicación grande como una herramienta de gestión, es posible que desees omitir el **uso de AngularJS** ya que no está desarrollado para ser usado en conjunto con otros Framework. AngularJS se desarrolló para la creación rápida de prototipos, específicamente para aplicaciones genéricas, pero hay casos en los que puedes utilizarlo para **construir aplicaciones de mayor escala**.

# Donde Aprender AngularJS

Para aprender los **fundamentos de AngularJS** puedes tomar un curso gratuito en CodeSchool, el cual está [patrocinado por Google](#). Allí te enseñan a **cómo programar en AngularJS**. No olvides visitar el [canal de YouTube de AngularJS](#) donde los propios desarrolladores publican tutoriales y noticias, la información esta toda en inglés ;)

Pero si eres más del tipo de documentación para desarrolladores, es posible que desees revisar la [documentación de la API de AngularJS](#). Para las personas que son desarrolladores en JavaScript, esta documentación debe ser bastante fácil para poder aprender este lenguaje de programación. No necesitas reinventar la rueda, ya que hay una gran cantidad de módulos que se pueden usar y mejorar en el [repositorio de ngmodules.org](#).

# AngularJS En Acción

## NgSweetAlert

Es un [reemplazo bastante elegante](#) para el tan conocido «Alert» de JavaScript.



## Gráficos Con AngularJs

Como se mencionó anteriormente, se puede utilizar para AngularJS para cálculos simples. Usando [Angular-nvD3](#), puedes personalizar los gráficos de acuerdo a sus necesidades.



# AngularJS Para Rompecabezas De Deslizado Y Buscador De Palabras

Este sencillo [rompecabezas](#) muestra lo flexible y simple que puede llegar a ser la **programación con AngularJS**. Este ejemplo también lo puedes encontrar en [Github](#).



## Finalmente

Sin duda **AngularJS** es un **Framework con un gran alcance** que puede ayudar a los desarrolladores acelerar el proceso de desarrollo de aplicaciones web. El **uso de AngularJS** se está volviendo más y más popular al pasar los días, y es recomendable que todo desarrollador y diseñador web se una a esta sentencia.