República Bolivariana de Venezuela

Ministerio del poder popular para la educación

Universidad Valle del Momboy – Facultad Estovacuy

San Rafael de Carvajal – Edo. Trujillo

Recursion Company

**AngularJS**

AngularJS (comunalmente llamado "Angular" o "Angular.js"), es un [framework](https://es.wikipedia.org/wiki/Framework) de [JavaScript](https://es.wikipedia.org/wiki/JavaScript) de [código abierto](https://es.wikipedia.org/wiki/C%C3%B3digo_abierto), mantenido por [Google](https://es.wikipedia.org/wiki/Google), que se utiliza para crear y mantener [aplicaciones web](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web) [de una sola página](https://es.wikipedia.org/wiki/Single-page_application). Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de [Modelo Vista Controlador](https://es.wikipedia.org/wiki/Modelo_Vista_Controlador) (MVC), en un esfuerzo para hacer que el desarrollo y las [pruebas](https://es.wikipedia.org/wiki/Pruebas_de_software) sean más fáciles.

La biblioteca lee el [HTML](https://es.wikipedia.org/wiki/HTML) que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Los valores de las variables de JavaScript se pueden configurar manualmente, o recuperados de los recursos [JSON](https://es.wikipedia.org/wiki/JSON) estáticos o dinámicos.

AngularJS se puede combinar con el entorno en tiempo de ejecución [Node.js](https://es.wikipedia.org/wiki/Node.js), el framework para servidor [Express.js](https://es.wikipedia.org/w/index.php?title=Express.js&action=edit&redlink=1) y la base de datos [MongoDB](https://es.wikipedia.org/wiki/MongoDB) para formar el conjunto [MEAN](https://es.wikipedia.org/wiki/MEAN).

## Mejoras del HTML

Este Javascript pretende que los programadores mejoren el HTML que hacen. Que puedan producir un HTML que, de manera declarativa, genere aplicaciones que sean fáciles de entender incluso para alguien que no tiene conocimientos profundos de informática. El objetivo es producir un HTML altamente semántico, es decir, que cuando lo leas entiendas de manera clara qué es lo que hace o para qué sirve cada cosa.

Lógicamente, AngularJS viene cargado con todas las herramientas que los creadores ofrecen para que los desarrolladores sean capaces de crear ese HTML enriquecido. La palabra clave que permite ese HTML declarativo en AngularJS es "directiva", que no es otra cosa que código Javascript que mejora el HTML. Puedes usar el que viene con AngularJS y el que han hecho terceros desarrolladores, puesto que muchas personas están contribuyendo con pequeños proyectos -independientes del propio framework- para enriquecer el panorama de directivas disponibles. Hasta este punto serás un "consumidor de directivas", y finalmente cuando vayas tomando experiencia serás capaz de convertirte en un "productor de directivas", enriqueciendo tú mismo las herramientas para mejorar tu propio HTML.

Los objetivos de diseño:

* [Disociar](https://es.wikipedia.org/wiki/Separaci%C3%B3n_de_conceptos) la manipulación del DOM de la lógica de la aplicación. Esto mejora la capacidad de prueba del código.
* Considerar a las [pruebas](https://es.wikipedia.org/wiki/Pruebas_de_software) de la aplicación como iguales en importancia a la escritura de la aplicación. La dificultad de las pruebas se ve reducida drásticamente por la forma en que el código está estructurado.
* Disociar el lado del cliente de una aplicación del lado del servidor. Esto permite que el trabajo de desarrollo avance en paralelo, y permite la reutilización de ambos lados.
* Guiar a los desarrolladores a través de todo el proceso del desarrollo de una aplicación: desde el diseño de la [interfaz de usuario](https://es.wikipedia.org/wiki/Interfaz_de_usuario), a través de la escritura de la lógica del negocio, hasta las pruebas.

Angular sigue el patrón MVVM (Model View View-Model) de ingeniería de software y alienta la articulación flexible entre la presentación, datos y componentes lógicos. Con el uso de la inyección de dependencias, Angular lleva servicios tradicionales del lado del servidor, tales como controladores dependientes de la vista, a las aplicaciones web del lado del cliente. En consecuencia, gran parte de la carga en el backend se reduce, lo que conlleva a aplicaciones web mucho más ligeras.

# ReactJS

React (también llamada React.js o ReactJS) es una biblioteca Javascript de código abierto para crear interfaces de usuario con el objetivo de animar al desarrollo de [aplicaciones en una sola página](https://es.wikipedia.org/wiki/Single-page_application). Es mantenido por [Facebook](https://es.wikipedia.org/wiki/Facebook), [Instagram](https://es.wikipedia.org/wiki/Instagram) y una comunidad de desarrolladores independientes  y compañías.

React intenta ayudar a los desarrolladores a construir aplicaciones que usan datos que cambian todo el tiempo. Su objetivo es ser sencillo, declarativo y fácil de combinar. React sólo maneja la [interfaz de usuario](https://es.wikipedia.org/wiki/Interfaz_de_usuario) en una aplicación; está construida únicamente para utilizar el patrón de diseño [modelo–vista–controlador](https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador) (MVC), y puede ser utilizada conjuntamente con otras bibliotecas de Javascript o más grandes #MVC como [AngularJS](https://es.wikipedia.org/wiki/AngularJS). También puede ser utilizado con las extensiones de React-based que se encargan de las partes no-UI (no gráficas) de una aplicación web.

Según el servicio de análisis [Javascript](https://es.wikipedia.org/wiki/JavaScript) (en inglés javascript analytics service), Libscore, React actualmente está siendo utilizado en las páginas principales de Imgur, Bleacher Informe, [Feedly](https://es.wikipedia.org/wiki/Feedly), [Airbnb](https://es.wikipedia.org/wiki/Airbnb), SeatGeek, HelloSign, y otras.

**Funcionamiento**

React.js está construido en torno a hacer funciones, que toman las actualizaciones de estado de la página y que se traduzcan en una representación virtual de la página resultante. Siempre que React es informado de un cambio de estado, vuelve a ejecutar esas funciones para determinar una nueva representación virtual de la página, a continuación, se traduce automáticamente ese resultado en los cambios del DOM necesarios para reflejar la nueva presentación de la página.

A primera vista, esto suena como que fuera más lento que el enfoque JavaScript habitual de actualización de cada elemento, según sea necesario. Detrás de escena, sin embargo, React.js hace justamente eso: tiene un algoritmo muy eficiente para determinar las diferencias entre la representación virtual de la página actual y la nueva. A partir de esas diferencias, hace el conjunto mínimo de cambios necesarios en el DOM.

Pues utiliza un concepto llamado el DOM virtual que hace selectivamente sub-árboles de los nodos sobre la base de cambios de estado, desarrollando esto, con la menor cantidad de manipulación DOM posible, con el fin de mantener los componentes actualizados, estructurando sus datos.

**Ionic Framework**

Ionic es un SDK de código abierto para el desarrollo de aplicaciones móviles hibridas. Esta contruida por encima de AngularSJ y Apache Cordova, Ionic provee herramientas y servicios para el desarrollo de aplicacions móviles hibridas utilizando tecnologías Web como CCS, HTML5 y Sass. Las aplicaciones pueden ser construidas con esas tecnologías Web y pueden ser destribuidas en las tiendas de aplicaciones nativas instaladas en los dispositivos aprovechando Cordova.

El Ionic Creator es un programa de interfaz drag-and-drop (Agarrar y soltar). La App Builder para Ionic es también soportado por Appery.io.

**El MVC**

El MVC (Model-View-Controller o Modelo-Vista-Controlador), es un patrón de diseño que separa los datos, la lógica y las interfaces de usuario. Como su nombre indica, está separado en tres componentes: Modelo, Vista y Controlador. Está basado en la ideología de separación de conceptos y cumple perfectamente con los objetivos de los patrones de diseño.

1.- Modelo

Es la capa encargada de los datos, es decir, la que se encarga de hacer peticiones a las bases de datos para enviar o recibir información. Estas bases de datos pueden estar alojadas de forma local en nuestra app o de forma remota en un servidor externo.

2.- Vista

Se trata del código que nos permitirá presentar los datos que el modelo nos proporciona, como ejemplo podríamos decir que en una aplicación es el código HTML que nos permite mostrar la salida de los datos procesados.

3.- Controlador

Es la capa que sirve de enlace entre la vista y el modelo. Envía comandos al modelo para actualizar su estado, y a la vista correspondiente para cambiar su presentación.

En el caso MVVM (Modelo Vista VistaModelo) la interacción entre la vista y el controlador será en los dos sentidos, el controlador muestra los datos en la vista y si en la vista hay un cambio de datos, se actualiza el modelo automáticamente.

**Principales características**

1.- Alto rendimiento

La velocidad es importante. Tan importante que sólo se nota cuando no está en tu app. Ionic está construido para ser rápido gracias a la mínima manipulación del DOM, con cero jQuery y con aceleraciones de transiciones por hardware.

**AngularJS & Ionic**

Ionic utiliza AngularJS con el fin de crear un marco más adecuado para desarrollar aplicaciones ricas y robustas. Ionic no sólo se ve bien, sino que su arquitectura central es robusta y seria para el desarrollo de aplicaciones. Trabaja perfectamente con AngularJS.

3.- Centro nativo

Ionic se inspira en las SDK de desarrollo móviles nativos más populares, por lo que es fácil de entender para cualquier persona que ha construido una aplicación nativa para iOS o Android. Lo interesante, como sabéis, es que desarrollas una vez, y compilas para varios.

4.- Bonito diseño

Limpio, sencillo y funcional. Ionic ha sido diseñado para poder trabajar con todos los dispositivos móviles actuales. Con muchos componentes usados en móviles, tipografía, elementos interactivos, etc.

5.- Un potente CLI

Con un sólo comando podrás crear, construir, probar y compilar tus aplicaciones en cualquier plataforma.

**Cloud Computing**

Cloud Computing se ha convertido en un concepto comodín. Se puede usar en múltiples contextos para referirse a cosas de lo más dispares. Pero parece que hay consenso respecto a las tres clases fundamentales del Cloud Computing. A saber:

* **Software as a Service (SaaS)**: En español Software como Servicio. Modelo de distribución de software donde una empresa sirve el mantenimiento, soporte y operación que usará el cliente durante el tiempo que haya contratado el servicio. El cliente usará el sistema alojado por esa empresa, la cual mantendrá la información del cliente en sus sistemas y proveerá los recursos necesarios para explotar esa información. Ejemplos: [Salesforce](http://www.salesforce.com), [Basecamp](http://www.basecamphq.com).
* **Infrastructure as a Service (Iaas)**: En español Infraestructura como Servicio. Modelo de distribución de infraestructura de computación como un servicio, normalmente mediante una plataforma de virtualización. En vez de adquirir servidores, espacio en un centro de datos o equipamiento de redes, los clientes compran todos estos recursos a un proveedor de servicios externo. Una diferencia fundamental con el hosting virtual es que el provisionamiento de estos servicios se hacen de manera integral a través de la web. Ejemplos: [Amazon Web Services EC2](https://aws.amazon.com/ec2/) y [GoGrid](http://www.gogrid.com/).
* **Platform as a Service (PaaS)**: En español Plataforma como Servicio. Aunque suele identificarse como una evolución de SaaS, es más bien un modelo en el que se ofrece todo lo necesario para soportar el ciclo de vida completo de construcción y puesta en marcha de aplicaciones y servicios web completamente disponibles en la Internet. Otra característica importante es que no hay descarga de software que instalar en los equipos de los desarrolladores. PasS ofrece mútliples servicios, pero todos provisionados como una solución integral en la web. Aunque algunos servicios de Amazon Web Services como [SimpleDB](https://aws.amazon.com/simpledb/) y [SQS](https://aws.amazon.com/sqs/) yo los considero PaaS, esta afirmación puede ser discutida. Otro ejemplo es [Google App Engine](https://code.google.com/appengine/).
* **Backend as a service** (BaaS), también conocido como “mobile backend as a service”, es un modelo para proporcionar a los desarrolladores web y de [aplicaciones móviles](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_m%C3%B3vil) una forma de vincular estas aplicaciones al [almacenamiento en nube](https://es.wikipedia.org/wiki/Almacenamiento_en_nube) (cloud storage), servicios analíticos y/o otras características tales como la gestión de usuarios, la posibilidad de enviar [notificaciones push](https://es.wikipedia.org/wiki/Tecnolog%C3%ADa_Push) y la integración con [servicios de redes sociales](https://es.wikipedia.org/wiki/Servicio_de_red_social). Estos servicios se prestan a través de la utilización de [kits personalizados de desarrollo de software](https://es.wikipedia.org/wiki/Kit_de_desarrollo_de_software) (SDK) y las [interfaces de programación de aplicaciones](https://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones) (API). BaaS es un modelo relativamente reciente en la computación en la nube, donde la mayoría de empresas datan del 2011 o posterior.