



UNIVERSIDAD TÉCNICA DE BABAHOYO

FACULTAD DE ADMINISTRACIÓN, FINANZAS INFORMÁTICA

F.A.F.I.

***“EXAMEN COMPLEXIVO DE GRADO O DE FIN DE CARRERA
PRUEBA PRÁCTICA INGENIERÍA EN SISTEMAS”***

Tema:

**ANÁLISIS COMPARATIVO DE TECNOLOGÍAS FRONT END
ANGULAR JS VS REACT JS, EN EL MODELO DE PROCESOS PARA
EL DESARROLLO DE APLICACIONES WEB**

Autor:

LUIS ALI ANCHUNDIA MEDRANO

Tutor:

ING. Iván Rubén Ruiz Parrales

LOS RÍOS - BABAHOYO - ECUADOR

RESUMEN

El objetivo de este proyecto es la realización de un análisis comparativo entre estos frameworks: AngularJS vs. ReactJ, con esta comparación permite dar a conocer al lector las características que proporciona AngularJS y que lo ha llevado a ser uno de los líderes del mercado. Es un estudio que, además, incluye el desarrollo de un caso práctico para mostrar el funcionamiento de esta herramienta.

Cabe mencionar que existen herramientas que están orientadas para desarrollar proyectos pequeños y otras que son más escalables, óptimos y tienen un enfoque para aplicaciones de índole empresarial. Realizar proyectos con ayuda de un Framework puede deberse a varios motivos como puede ser que el Framework se adapta a una arquitectura establecida, estas herramientas constituyen algunos Frameworks y/o librerías basados en componentes web como lo son Angular JS y ReactJS. Algunos en la actualidad están basados en JavaScript se usan en el desarrollo de la mayoría de las aplicaciones web y siempre están apareciendo herramientas novedosas que buscan liderar el mercado. Además, con la constante evolución de las tecnologías web, se pueden materializar ideas que años atrás sería impensable realizar, debido a que no existía los recursos para construirlas.

Para culminar este documento acerca del desarrollo web, más específico en los frameworks cliente MVC basados en JavaScript puedo decir que el crecimiento que han tenido estas tecnologías en la última década ha sido muy grande, esto debido a las grandes herramientas que le entregan al desarrollador y por ende la gran comunidad que se genera entorno a ellas.

PALABRAS CLAVE: Frameworks, AngularJS, ReactJS, Librerías, flexibilidad, compilador, contenedores, arquitectura, prototipo.

ABSTRACT

The objective of this project is to carry out a comparative analysis between these frameworks: AngularJS vs. ReactJ, with this comparison, allows the reader to know the features that AngularJS provides and that has led it to be one of the market leaders. It is a study that also includes the development of a practical case to show how this tool works.

It is worth mentioning that there are tools that are oriented to develop small projects and others that are more scalable, optimal and have a focus on business applications. Carrying out projects with the help of a Framework can be due to several reasons, such as the fact that the Framework adapts to an established architecture, these tools constitute some Frameworks and/or libraries based on web components such as Angular JS and ReactJS. Some are currently based on JavaScript, they are used in the development of most web applications and new tools are always appearing that seek to lead the market. In addition, with the constant evolution of web technologies, ideas can be materialized that years ago would be unthinkable, because the resources to build them did not exist.

To conclude this document about web development, more specifically in MVC client frameworks based on JavaScript, I can say that the growth that these technologies have had in the last decade has been very great, this is due to the great tools that they provide to the developer and therefore the great community that is generated around them.

KEYWORDS: Frameworks, AngularJS, ReactJS, Libraries, flexibility, compiler, containers, architecture, prototype.

INTRODUCCIÓN

Actualmente existen muchas herramientas de desarrollo que permiten disminuir tiempos de trabajo de un proyecto, brindan facilidades de implementación o facilitan incluso la programación por mantener una estructura óptima. Estos Frameworks y librerías se encuentran desarrollados en diferentes lenguajes de programación como PHP, Java, JavaScript, Python, entre otros. Sin embargo, muchos de ellos tienen una estructura preestablecida y también están enfocados para un solo ambiente y al querer realizar una migración de un ambiente a otro se vuelve complejo. Cabe mencionar que existen herramientas que están orientadas para desarrollar proyectos pequeños y otras que son más escalables, óptimos y tienen un enfoque para aplicaciones de índole empresarial. Realizar proyectos con ayuda de un Framework puede deberse a varios motivos como puede ser que el Framework se adapta a una arquitectura establecida, estas herramientas constituyen algunos Frameworks y/o librerías. En el mundo de desarrollo Web existe un debate de qué herramientas representan mayor aporte para un proyecto es por esta razón que se ve necesario realizar un estudio comparativo entre los principales Frameworks y librerías como AngularJS y ReactJS, con el fin de brindar un mejor conocimiento de acorde con sus características y ventajas que cada uno de estos puede ofrecer. Es de gran importancia determinar todos los aspectos más sobresalientes de cada una de las herramientas en cuestión ya que para un futuro se puede sustentar de manera adecuada la elección de un Framework u otra herramienta para desarrollar una aplicación web, móvil o incluso una aplicación híbrida.

DESARROLLO

A continuación, se presentarán las tecnologías predominantes a la hora de desarrollar una aplicación web, más en específico las que hace los frameworks AngularJS y ReactJS, para tener una idea global de la estructura y metodologías que esta ocupa. JavaScript es el lenguaje de programación interpretado más usado actualmente en el desarrollo de aplicaciones Web, contiene una sintaxis muy similar a Java y a C, pero no guarda relación directa con ninguno de estos lenguajes de programación, puesto que no es un lenguaje orientado a objetos, ya que está basado en prototipos, ósea las clases nuevas se generan clonando las clases base y de esta forma se extiende su funcionalidad. (Flanagan, 2021)

Al ser un lenguaje de programación interpretado no necesita que los programas sean compilados para que se ejecuten ya que el lenguaje funciona del lado del cliente, cualquier código escrito en JavaScript es directamente transparente y visible por cualquier navegador web, puesto que él es el encargado de interpretar dicho código sin la necesidad de procesos intermedios. (Pérez, 2021)

El código JavaScript se puede encontrar dentro de las etiquetas en las páginas web. Por lo general se insertan entre: `<script></script>`. Entre algunos de los servicios que se encuentran realizados con JavaScript en Internet se encuentran:






-  Correo
-  Chat
-  Buscadores de Información
-  Sitos de consulta
-  Páginas de comunicación



Figura. Logo JavaScript.

Fuente: <https://adictoaltrabajo.com>

Ventajas

Entre las ventajas más destacables podemos mencionar las siguientes:

- ✚ Útil para el desarrollo de páginas web dinámicas.
- ✚ Excelente intermediario para la validación de datos de un formulario en el lado del cliente.
- ✚ Es un lenguaje sencillo y liviano.
- ✚ Consume poca memoria.
- ✚ Tiene gran cantidad de efectos visuales.
- ✚ Compatible con la gran mayoría de navegadores modernos y fácil manejo de datos.
- ✚ Se recomienda para la creación de aplicaciones web.
- ✚ El código se ejecuta en el lado del cliente, por lo que el servidor no es solicitado más de lo debido.
- ✚ No necesita de un compilador para ejecutarse, sino es el navegador el que interpreta el código.

Desventajas

Una de las desventajas más notable de JavaScript es la seguridad, el código es visible y puede ser leído por cualquiera. Es verdad que hoy en día existen varios estándares de seguridad que restringen la ejecución de código por parte de los navegadores, pero aun así se puede ejecutar código malicioso que dañe, robe o destruya información del lado del cliente. Los <script> tienen capacidades limitadas, por razones de seguridad, por lo cual no es posible hacer todo con JavaScript, sino que es necesario usarlo conjuntamente con otros lenguajes evolucionados, posiblemente más seguros, como Java. Existe un determinado número de etiquetas, por lo que no se pueden crear más.

Arquitectura MVC

Es un patrón de arquitectura de las aplicaciones software, separa la lógica de negocio de la interfaz de usuario:

- Facilita la evolución por separado de ambos aspectos
- Incrementa reutilización y flexibilidad Modelo-Vista-Controlador
- Un modelo
- Varias vistas
- Varios controladores

Las vistas y los controladores suelen estar muy relacionados, los controladores tratan los eventos que se producen en la interfaz gráfica (vista). Esta separación de aspectos de una aplicación da mucha flexibilidad al desarrollador. El patrón de arquitectura "modelo vista controlador", es una filosofía de diseño de aplicaciones compuesta por tres capas:

Capa Modelo

Esta capa del modelo define la lógica de negocio (la base de datos pertenece a esta capa). En esta sección estará la estructura de datos, junto con las clases relacionadas a la base de datos y los métodos respectivos.

Capa Vista

La vista es lo que se muestra al usuario para que pueda interactuar con aplicación, en esta capa nos encontramos con los temas o plantillas. Proporcionará al cliente múltiples páginas web dinámicas, visualizándose para él como simples páginas HTML. Esto permite que personas con conocimientos de HTML puedan desarrollar sus aplicaciones de forma sencilla.

Capa Controlador

El controlador es un bloque de código que realiza llamadas al modelo para obtener los datos y se los pasa a la vista para que los muestre al usuario. (Pressman, 2022)

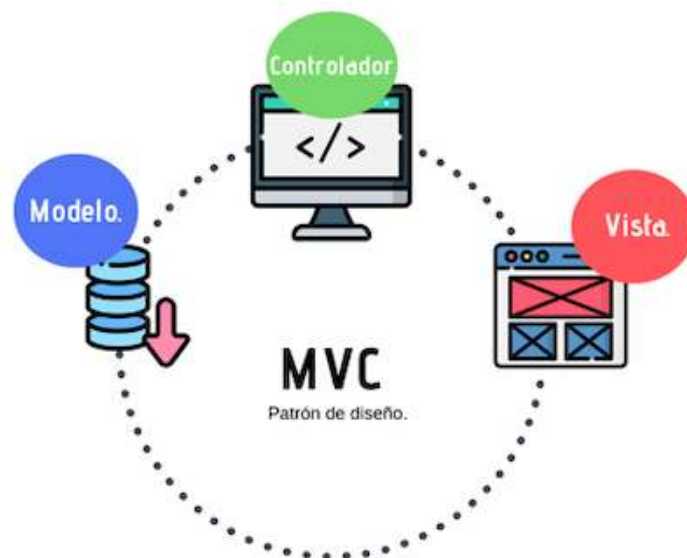


Figura. Patrón de Diseño MVC.

Fuente: <https://blog.nearsoftjobs.com>

Framework AngularJS

Diseñado para aplicaciones web dinámicas, mantenido por Google, utiliza HTML como lenguaje de plantillas el que extiende para expresar los componentes de su aplicación, luego AngularJS enseña al navegador la nueva sintaxis a lo que llamamos directivas. Los conceptos de enlace de datos e inyección de dependencias permiten reducir líneas de código y su ejecución es realizada dentro del navegador. Angular es 100% JavaScript del lado del cliente, no es un simple sistema de plantillas, la razón se debe a “bidirectional data binding”, enlace de datos bidireccional, que posee y se ejecuta de forma automática. La plantilla se compila en el navegador y el paso de compilación da como resultado una vista en vivo. Ya no se necesita que los desarrolladores deban sincronizar constantemente la vista con el modelo o viceversa. Además, tampoco es necesario dejar de lado jQueryUI o Bootstrap. Angular trabaja muy bien con librerías de componentes de terceros. Ayuda a la gestión de lo que se conoce como aplicaciones de una sola página (SPA). (AngularJS, 2019)

Angular permite el uso de cualquier tipo de backend, desde Java, Python, Ruby, C# entre otros. Al trabajar con Angular es necesario utilizar una forma de comunicación de ida y vuelta con el servidor, peticiones HTTP XML o JSON.



Figura. Logo AngularJS.

Fuente: <https://docs.angularjs.org/>, 2016.

Cronología de AngularJS

AngularJS nació en 2009 como parte de un producto comercial más grande, llamado GetAngular. Poco después, Misko Hevery, uno de los ingenieros que fundaron GetAngular, en 3 semanas logro regresar una aplicación web que constaba de 17mil líneas de código utilizando GetAngular, algo que normalmente se tarda 6 meses, este producto logro reducir el tamaño de la aplicación a unas 1000 líneas de código lo que logro convencer a Google para que patrocine este proyecto. (AngularZone, 2019)

La biblioteca lee el 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. Esta basado en clases tipo "Componentes", cuyas propiedades son las usadas para hacer el binding de los datos. En dichas clases tenemos propiedades (variables) y métodos (funciones a llamar).

- ✚ **Generación de código:** Angular convierte tus plantillas en código altamente optimizado para las máquinas virtuales de JavaScript de hoy en día, ofreciéndote todas las ventajas del código escrito a mano con la productividad de un framework.
- ✚ **Universal:** Ejecuta la primera vista de tu aplicación en node.js, .NET, PHP, y otros servidores para renderizado de forma casi instantánea obteniendo solo HTML y CSS. También abre posibilidades para la optimización del SEO del sitio, incluyendo configuración.
- ✚ **División del código:** Las aplicaciones de Angular se cargan rápidamente gracias al nuevo enrutador de componentes. Este ofrece una división automática de códigos para que los usuarios solo carguen el código necesario para procesar la vista que solicitan.

A continuación, presentaremos ciertos conceptos usados en AngularJS con sus respectivas descripciones:

Concepto	Descripción
Template	HTML con marcado adicional
Directivas	HTML extendido con atributos y elementos personalizados
Model	Datos que se muestran al usuario en la vista
Scope	En este se almacena el modelo para que los controladores, directivas y expresiones puedan acceder a él
Expressions	Variables de acceso y funciones del ámbito
Compilador	Analiza la plantilla y crea una instancia, directivas y expresiones
Filtro	Formatea el valor de una expresión para la visualización al usuario
View	Lo que ve el usuario (DOM)
Data Binding	Datos de sincronización entre el modelo y la vista
Controller	Lógica de negocio detrás de las vistas
Dependency Injection	Patrón de diseño de como los componentes se apoderan de sus dependencias
Injector	Contenedor de inyección de dependencias
Module	Contenedor para las diferentes partes de una aplicación, controladores, servicios, filtros, directivas que configura el inyector
Service	Lógica de negocio reutilizable independiente de vistas

Tabla. Conceptos más usados en AngularJS.

Fuente: Luis Anchundia

Características más destacadas de AngularJS

Scope: es uno de los puntos fundamentales de AngularJS, dentro de este se tiene acceso a las variables y los métodos que enlazan al controlador con la vista. Es decir, todo controlador debe poseer un `$Scope` que le permita interactuar con la vista, en la estructura Model-View-Controller, este objeto se convierte en el modelo, en la figura 5 se ilustra la interacción entre sus componentes. (Ahmed, 2019)

Data Binding: También conocido como Enlace de Datos”, Sin duda uno de los puntos fuertes de AngularJS. El binding es enlazar los datos del `$Scope` con lo que mostramos en el HTML de forma automática, producida en dos sentidos.

- ✚ One-way binding. - Los datos van únicamente desde el `$Scope` hacia la parte visual, (modelo hacia la vista). Se consigue mediante la sintaxis de llaves dobles.

{{Datos}}

Si se modifica el dato del modelo (`$Scope`), se actualiza automáticamente la vista.

- ✚ Two-way binding. - Los datos fluyen no solo desde el `$Scope` hacia la vista, sino también desde la vista hacia el `$Scope`, se implementa mediante la directiva `ngModel`. Cuando el modelo cambia, el dato que está dentro de la caja de texto cambia y cuando el usuario cambie el valor de la caja de texto el `$Scope` se actualiza automáticamente.

Directivas: Son marcadores en un elemento DOM (atributo, comentario o clase CSS) estos dan aviso al compilador HTML de angular (`$compile`), para que sea adjuntado el comportamiento especificado para ese elemento DOM o hasta se podría transformar el elemento DOM y sus hijos. Es la manera en la que se puede extender el HTML, permitiéndole agregar código; pueden ser basadas en nombres de elementos, atributos, nombres de clase, así

como los comentarios, se pueden dar en el nombre de atributo, nombre de la etiqueta, comentarios, o el nombre de la clase de atributos.

Expresiones: Se utilizan para representar datos dentro de cualquier parte de un documento HTML, estos pueden ser cadenas de texto, números, operaciones matemáticas, variables o funciones, su sintaxis es:

{{Dato}}

Módulos: Contenedores de distintas partes de la aplicación. Se puede crear la cantidad de módulos que se vaya a utilizar, la idea es desarticular el código y tratar de agruparlo ya sea por características, por funcionalidades, por componentes reusables. El desacoplar el código facilita mantenerlo y escalarlo.

Servicios: Son los objetos que permite obtener información, un servicio no interactúa con la propia página, sino con otros servicios, con un servidor que pueda estar en otro host. Se caracteriza principalmente porque solo hay una única instancia, pero puede ser utilizado más de una vez.

Routing: En la inicialización de la aplicación definimos las rutas con los pares Controlador-Vista. De esta manera podremos definir la ruta que muestra la aplicación mediante `#{'Ruta'}`. En la figura 26 se ve el archivo `app.js` que muestra el uso de Routing en AngularJS. Mediante la llamada `$routes.when` definimos la URL y el par controlador-vista a ejecutar. En este caso al controlador encargado de mostrar los detalles de una oportunidad le pasamos como parámetro el identificador de la oportunidad: `opId`. Con `otherwise` podemos definir una ruta por defecto haciendo `redirect` en cualquier ruta que no corresponda a las ya definidas. En el ejemplo si la ruta no la encuentra irá al home que es lo mismo que ir a `#/index`.

COMPARATIVA

JavaScript uno de los lenguajes más utilizados en la actualidad, debido a que es un lenguaje interpretado, esto hace que el código no necesite ser compilado, sino simplemente puede ser ejecutado en un navegador sin la necesidad de procesos intermedios. Su uso, potencia las páginas haciéndolas dinámicas, en otras palabras, extiende HTML para realizar funciones que no se podían realizar comúnmente, es por ello que se han creado varios Frameworks JavaScript para la ayuda del desarrollo de aplicaciones web y de esta manera facilitar su utilización.

Se realizó el análisis del rendimiento de AngularJS, para luego hacer una comparación con algunos de los frameworks basados en JavaScript más usados del momento, más específicamente a los que se ajustan al modelo Cliente MVC. En esta sección se muestra una breve descripción de cada framework y las principales ventajas y desventajas de cada uno. AngularJS es un framework de JavaScript de código abierto, mantenido por Google, que ayuda con la gestión de lo que se conoce como aplicaciones de una sola página (SPA), que extiende el tradicional HTML con etiquetas propias.



Figura. Logo AngularJS.

Fuente: <https://docs.angularjs.org/>, 2016.

Ventajas

- ✚ Ligero y con buena gestión de dependencias.
- ✚ Potente sistema de plantillas, extendiendo vocabulario de HTML básico.
- ✚ El concepto de directivas, que permite crear nuevos tags customizados que incorporan tanto funcionalidad como capa visual.
- ✚ Posee un potente enlace de interfaces (UI-Binding).
- ✚ Posee buenas herramientas para hacer debug.
- ✚ Inyección automática de dependencias.
- ✚ Desacoplamiento del DOM de JavaScript
- ✚ Internacionalización i18n y l10n.

Desventajas

- ✚ Curva de aprendizaje muy elevada.
- ✚ La escritura de directivas es compleja, es la parte más difícil de escribir código en Angular.

Se muestra una foto de las áreas en las que destaca frente a las que es más débil. Hay que tener en cuenta que no todas las áreas tienen la misma importancia, como se puede ver en los pesos asignados a los criterios de la comparación.

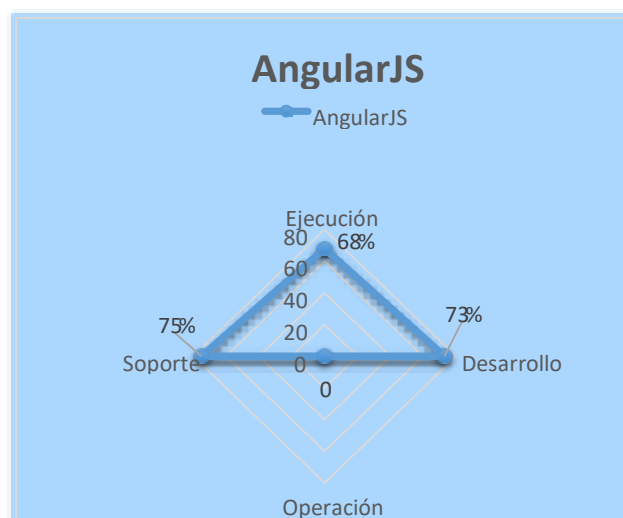


Figura. Gráfico resumen de áreas en evaluación para AngularJS.

Fuente: Stuardo, 2016.

React, también conocida como React.js o ReactJS es una biblioteca JavaScript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página. Es mantenido por Facebook y la comunidad de software libre. En el proyecto hay más de mil desarrolladores libres.

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 en una aplicación; React es la Vista en un contexto en el que se use el patrón MVC (Modelo-Vista-Controlador) o MVVM (Modelo-vista-modelo de vista). También puede ser utilizado con las extensiones de React-based que se encargan de las partes no-UI (que no forman parte de la interfaz de usuario) de una aplicación web.

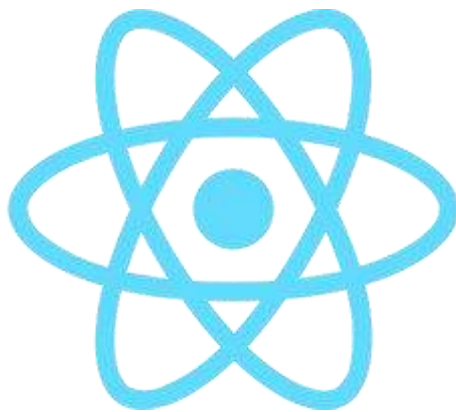


Figura. Logo ReactJS.

Fuente: <https://docs.reactjs.org>.

VENTAJAS

- ✚ Fácil de aprender. React es mucho más sencillo de aprender por su simplicidad en términos de sintaxis.
- ✚ Los ingenieros tan sólo necesitan echar mano de su habilidad para escribir HTML. No hay necesidad de aprender TypeScript con en el caso de Angular.
- ✚ Gran nivel de flexibilidad y máxima ‘responsividad’.
- ✚ DOM Virtual (Modelo de Objetos del Documento) que permite convertir documentos en formato HTML, XHTML o XML en un árbol más manejable para los navegadores mientras parsean diferentes elementos de la aplicación.
- ✚ Combinado con ES6/7, ReactJS puede trabajar con alta carga fácilmente.
- ✚ Vínculo de datos ‘hacia abajo’ lo que significa que con este tipo de flujo de datos los elementos ‘hijo’ no puede afectar a los datos ‘padre’.
- ✚ Librería JavaScript 100% código abierto con frecuentes actualizaciones y mejoras de desarrolladores de todo el mundo.
- ✚ Framework muy ligero ya que los datos del lado del usuario pueden representarse en el servidor simultáneamente.
- ✚ La migración entre versiones es en general muy sencilla, con Facebook proporcionado codemods (modificaciones de código) para automatizar gran parte del proceso.

DESVENTAJAS

- ✚ Ausencia de documentación oficial. La alta velocidad de desarrollo de ReactJS apenas deja lugar a una documentación apropiada, la cual es algo caótica ya que diferentes desarrolladores contribuyen sin un enfoque común.
- ✚ No existe un estándar de desarrollo, de modo que tenemos demasiadas elecciones a tomar.
- ✚ Requiere mucho tiempo para dominarlo, así como un profundo conocimiento de cómo integrar interfaces de usuario en frameworks MVC.

TABLA COMPARATIVA DE CARACTERÍSTICAS DE FRAMEWORKS

Parámetros	AngularJS	ReactJS
Tipo	Marco de trabajo completo	Biblioteca JavaScript
Tipo de DOM	DOM real	DOM virtual
Vinculación de datos	Vinculación de datos bidireccional	Vinculación de datos unidireccional
Escrito en	TypeScript	JavaScript
Plantillas	JSX + J% (ES5/ES6)	HTML + TypeScript
Abstracción	Medio	Fuerte
Inclusión de la biblioteca JavaScript en el código fuente	No es posible	Posible
Modelo	Modelo MVC	DOM virtual
Prueba y depuración	Solución completa en una sola herramienta	Necesita un conjunto adicional de herramientas
Libertad	Limitada	Permite elegir las bibliotecas, la arquitectura y las herramientas

Tabla. Comparativa de características de AngularJS vs. ReactJS.

Fuente: Luis Anchundia

ANGULAR VS REACT: UN ANALISIS DE COMPARACIÓN EN PROFUNDIDAD

Del análisis comparativo de tecnologías front end angular js y react js realizado se detalla las siguientes características.

Las similitudes entre Angular y React incluyen:

Arquitectura: Tanto Angular como React tienen una arquitectura basada en componentes. Estos componentes son reutilizables dentro de otros componentes, lo que los hace infinitamente reciclables. En particular, un componente es una parte de una interfaz de usuario. Por ejemplo, los componentes podrían ser un cuadro de diálogo de inicio de sesión con texto, un campo de contraseña o un botón de inicio de sesión.

Fuente abierta: Ambos son de código abierto. Como resultado, React y Angular tienen grandes comunidades de desarrolladores que enriquecen regularmente los recursos.

Popular: Los desarrolladores utilizan principalmente estas dos tecnologías para crear aplicaciones de página única. Eso afirma que puede crear aplicaciones de una sola página para obtener soluciones digitales mejores y más rápidas.

Desarrollo ambiente: Angular y React se utilizan para desarrollar la interfaz de un aplicación móvil o web.

Representación: Angular y React proporcionan una representación eficiente del lado del cliente y del lado del servidor.

Rendimiento: Angular y React ofrecen un rendimiento similar. La diferencia depende principalmente de la perspectiva del usuario.

Fácil de actualizar: Tanto Angular como React ofrecen una fácil actualización. Mientras que Angular usa CLI, React depende de bibliotecas externas.

CONCLUSIONES

Se aplicó el modelo a las herramientas en estudio entre Angular JS y React JS, sin embargo, la primera consiguió superar en la capacidad de aprendizaje y en la facilidad de operación que este ofrece. Además, supera en otras características como el enlace de datos, la coexistencia con otros lenguajes, lo cual acumula ventajas a su favor.

Para culminar este documento acerca del desarrollo web, más específico en los frameworks cliente MVC basados en JavaScript puedo decir que el crecimiento que han tenido estas tecnologías en la última década ha sido muy grande, esto debido a las grandes herramientas que le entregan al desarrollador y por ende la gran comunidad que se genera entorno a ellas.

Gran parte de desarrolladores están aprovechando las ventajas de estos frameworks, en especial AngularJS, como se pudo observar, en el estudio realizado recopilando numerables experiencias de desarrolladores con distintos niveles de experiencia, este fue el mejor evaluado entre los tres frameworks estudiados y con muchas razones, su gran escalabilidad y modularidad ayudan al desarrollador a generar aplicación rápidamente y con la menor cantidad de código.

Es cierto que la curva de aprendizaje es pronunciada, no queda más que decir que por algo AngularJS fue patrocinado por el gigante Google y lo usa en varias de sus aplicaciones. Angular inicio el desarrollo web moderno y lo que hoy conocemos como frontend, angular es una de las tecnologías más demandada para el desarrollo web y móvil en el mundo.

BIBLIOGRAFÍA

Ahmed, Z. (2019). *Learning AngularJS Part 4: Modular Design of AngularJS Application*.

Obtenido de <http://conceptf1.blogspot.cl/2014/05/learning-angularjspart5>

AngularJS. (2019). *AngularJS by Google*. Obtenido de <https://angularjs.org/>

AngularZone. (2019). *AngularJs History*. *AngularZone*. Obtenido de <https://angularzone.wordpress.com/angularjs-history/>

Ember. (2019). *A framework for creating ambitious web applications*. Obtenido de <http://emberjs.com>

Express. (2019). *Infraestructura web rápida, minimalista y flexible para Node.js*. Obtenido de <http://expressjs.com/es/>

Flanagan, D. (2021). *JavaScript la Guía Definitiva*. Obtenido de https://books.google.com.ec/books/about/JavaScript.html?id=mriFGgAACAAJ&redir_esc=y

Pérez, D. (2021). *Qué es JavaScript*. Obtenido de <http://www.maestrosdelweb.com/que-es-javascript/>

Pressman, R. S. (2022). *Software engineering: a practitioner's approach*. Palgrave.