UNIVERSIDAD PRIVADA DEL NORTE

FACULTAD DE INGENIERÍA

Carrera de Ingeniería de Sistemas Computacionales

“ANALISIS DE FRAMEWORK JAVASCRIPT Y MVC CON SERIALIZADOR DE DATOS Y SU INCIDENCIA EN RENDIMIENTO EN LOS ÚLTIMOS 4 AÑOS”

Trabajo de investigación para optar al grado de:

Bachiller en Ingeniería de Sistemas Computacionales

Autor:

Juan Carlos Espinoza Pereda

Asesor:

Ing. Mg. Neicer Campos Vasquez

Lima - Perú

2018

DEDICATORIA

El presente trabajo de investigación está dedicado a Dios,

Ya que es el ser que me ilumina día a día para lograr y

Alcanzar mis metas.

A mi querida madre Santos Josefina Pereda Alvarado,

Que con su apoyo y cariño desinteresado hizo que yo

Cumpla cada uno de mis objetivos.

A mí querido padre Eulogio Espinoza Menor,

Que desde el cielo es mi guía y por los sabios concejos,

Nobleza y perseverancia que me brindó para conseguir

Lo que uno se propone y anhela.

A mis familiares y amigos que de una u otra manera contribuyeron

Con un granito de arena y compartiendo experiencias

Para lograr mis metas.

AGRADECIMIENTO

Agradezco a la Universidad Privada del Norte

Que mediante su plana docente me ayudaron

A cumplir una de mis metas que es terminar

Con éxito la carrera de

Ingeniería de Sistemas Computacionales.

A los docentes del curso de proyecto de

Tesis y tesis respectivamente por sus asesorías

Y conocimientos brindados para poder

Cumplir de manera satisfactoria la carrera.

Tabla de contenido

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

Se realizó un análisis de los Framework JavaScript y MVC con Serializador de datos y su incidencia en rendimiento en los últimos 04 años. En esta revisión sistemática se tomó como fuente 188 documentos encontrados en diferentes base de datos y repositorios tales como: Redalyc, DOAJ, Directory of Open Access Journals, Dialnet, etc. Del total de documentos se aplicaron filtros y se trabajó con 15 documentos finales. Se incluyó artículos con idioma español, inglés y portugués. Los países de donde se obtuvo los documentos finales fueron Ecuador, USA, Chile, Croacia, Suecia, Rumania, Nepal, México, Costa Rica, Brasil, Perú. Los años fueron en el periodo 2015 al 2018. Los resultados muestran los beneficios de la utilización de un Framework en base a comparaciones y pruebas, este Framework es AngularJS con un segundo que le sigue en beneficios que es ReactJS, también se menciona que JSON ofrece mayor ventaja en serialización/deserialización en la transferencia y tratamiento de datos. Se concluye con esta revisión sistémica que el uso de un Framework JavaScript y MVC con serializador de datos incide positivamente en el rendimiento tales como mejora el tiempo de respuesta, mayor usabilidad, mejor mantenibilidad y mayor portabilidad. Se hace una recomendación para investigaciones futuras que puedan realizar pruebas en otros navegadores diferentes a Firefox y Google Chrome.

PALABRAS CLAVES: JavaScript, MVC, Serialización/Deserialización, JSON, AngularJS.

CAPÍTULO I. INTRODUCCIÓN

Anteriormente la demanda en el acceso a las web eran considerablemente bajas y la lentitud en la transferencia de data y el tiempo de respuesta no era tan notorio debido a la poca saturación de las redes de internet. Actualmente las aplicaciones web han contribuido al crecimiento y desarrollo tecnológico en diferentes campos de nuestro país, así mismo las arquitecturas de las aplicaciones web en las últimas décadas han evolucionado notablemente y la demanda de acceso de usuarios ha incrementado considerablemente, esto trae de forma paralela acciones que tienen que realizarse para satisfacer dicha demanda de los usuarios. Cuando ya se repotenciaron los servidores por el lado del hardware a un nivel máximo, se mejoraron las infraestructuras, soporte de redes y telecomunicaciones; ya se tiene que pensar en otras alternativas las cuales implica optimización y mejoramiento del código para mejorar el rendimiento, agilizar la transferencia de datos, tiempo de respuesta y tener acceso a las web de forma ligera. Son muchas las herramientas, técnicas y Frameworks que los programadores utilizan para desarrollar aplicaciones web de manera óptima.

"Es muy importante que en la actualidad los desarrolladores de aplicaciones web incluyan técnicas y algoritmos óptimos de serialización/deserialización y transmisión de datos que permitan la manipulación de diferentes tipos de objetos para poder manejarlos como texto y así facilitar tus transferencia, para luego volver a convertirlo en el objeto original, este proceso ofrece una gran facilidad y ventaja en la comunicación y transporte de data en las aplicaciones web" (Mora, 2016). "*Una de las mayores ventajas de la*

serialización es que proporciona almacenamiento e intercambio de datos, manteniendo además información importante entre aplicaciones" (Ochoa, 2016).

Existen Frameworks del lado del servidor como del lado del cliente. Para el presente trabajo de investigación nos avocaremos a la programación del lado del cliente; muchas veces no es recomendable sobrecargar al servidor debido a la gran cantidad de concurrencias de usuarios que acceden a un sitio web en forma paralela. Esto obliga a los desarrolladores a trabajar del lado del cliente con la inserción de código para manipular la carga de datos a través del lenguaje JavaScript.

"Las interfaces web son muy importantes en cuanto a rendimiento y consumo de recursos, en ocasiones se requiere que respondan a gran velocidad, en otras que no consuman tantos recursos de hardware del servidor y en otras que al momento de hablar de rendimiento sean muy prácticos" (Peña & Cambisaca, 2015). Son muchas ya las empresas que están implementando aplicaciones Web con arquitectura MVC a la vez que existen diferentes Framework JavaScript para agilizar los procesos de carga, transferencia de datos y mejorar el rendimiento. Estudios demuestran que tener aplicaciones web con demasiados tiempos de respuestas y lentitud repercute de forma directa en nuestras empresas.

"La gran mayoría de aplicaciones web usa un Framework JavaScript para construir un sitio web interactivo, accesible, con elementos dinámicos y cambiantes en segundos, estos Frameworks manipulan el objeto de modelo de documento del archivo HTML (DOM) del lado del cliente" (Svensson, 2015).

(Echeverria, 2016) "*Afirma que cuando la web toma demasiado tiempo en responder, éste puede ser un factor determinante para que los usuarios tomen la decisión de no volver a entrar a dicha página. Menciona algunos casos como: de todos los usuarios que miran un video con problemas de lentitud el 80% no lo ve, el tiempo promedio de carga de una página web es 4,9 segundos, La publicidad de Insideline aumentó en 3% y la visita de sus páginas incremento en 17% al lograr mejorar el tiempo de respuesta de 9 a 1,4 segundos, Shospzilla también logro mejorar su tiempo de respuesta de 7 a 2 segundos y aumentó las visitas a su web en 25% y la publicidad en 112%, Amazon investigó que se pierde el 1% de ingresos por 0,1 segundos de retraso en la web, de similar manera Facebook menciona que puede haber saturación de tráfico del 3% si hubiera un incremento en demora de 0,5 segundos en sus páginas, Yahoo menciona que se presentan caídas en el tráfico de la red de entre 5% y 9% por la demora de 0,4 segundos y Google Maps aumentó las transacciones de solicitudes en 30% reduciendo el tamaño de sus archivos en 30%".*

(Zurita & Zavala, 2016) "*Menciona que el sitio web del Diario El Comercio colapso en las elecciones PERU - 2011 por la elevada tasa de acceso de los usuarios que querían informarse de los resultados electorales".* (Santos, 2018) "*Evidencia que desarrollar en MVC aumenta la productividad de los programadores y requiere menos tiempo de codificación que realizar la misma tarea en WebForms".*

"Las aplicaciones web presentan mejoras de usabilidad y tiempo de respuesta al usar un JavaScript Framework" (Márquez, 2016). "*A medida que los navegadores son más poderosos, JavaScript es más popular con un 45% de crecimiento desplegado; así mismo*

MVC ofrece ventajas arquitectónicas con JavaScript por estar mejor organizado y fácil de mantener" (Enache, 2015). En una investigación de maestría (Correia, 2016) "*Menciona que para mejorar los accesos a la web se debe reducirse lo que se envía al explorador ya que cuando se accede a una web, se cargan imágenes, archivos JavaScript, CSS, etc. Esto obliga a: minificar recursos eliminando de espacios, líneas en blanco y comentarios que no son necesarios y utilizar la caché del browser ya que la primera vez que se accede a una web se cargan los recursos localmente y si se accede por segunda vez la carga es rápida".* "*La minificación es una técnica referida a reducir el tamaño de los recursos de una página web (HTML, CSS Y JavaScript) eliminando bytes innecesarios, espacios adicionales y en blanco, saltos de línea, sangrías, comentarios no necesarios para lograr acelerar la descarga, tiempo de ejecución y hacer más ligera una web"* (Bendezú & Figueroa, 2017).

(Manish, 2016) Describe que "*MVC (modelo-vista-controlador) es un patrón de software arquitectónico que desacopla esencialmente varios componentes de una aplicación web en modelo, vista y controlado, incluye servicios necesarios para construir aplicaciones web con un mínimo de codificación".*

Cada Framework JavaScript crece con una gran comunidad que continúa mejorando las fuentes del código y genera una base de conocimiento que ayuda a los nuevos desarrolladores a usar y utilizar el Framework en el desarrollo de aplicaciones web. "*En el diseño de cada Framework JavaScript se enfatiza un conjunto distinto de características y priorizado las principales que están relacionadas con la facilidad de codificación, tarea, automatización, uso de recursos y rendimiento. El usuario final*

evaluará una aplicación que se carga rápidamente y no presente problema alguno " (Hans, 2015). En tal sentido ésta investigación pretende estudiar y analizar si existe una mejora considerable en el aplicativo web según lo expuesto anteriormente.

CAPÍTULO II. METODOLOGÍA

Las Revisiones Sistemáticas clasifican las publicaciones de forma sistemática, aplican estadísticas para resolver controversias de otras investigaciones que son limitadas: sujetos de estudio, sesgos de investigador (Garcia, y otros, 2014).

El presente trabajo de investigación esta sesgado a la siguiente pregunta *¿Cuáles son los estudios de Frameworks JavaScript y MVC con Serializador de datos y su incidencia en el rendimiento en los últimos 4 años?*

Para nuestro trabajo de investigación de la revisión de la literatura emplearemos la metodología PRISMA (Elementos de informes preferidos para revisiones sistemáticas y meta-análisis) que es la evolución del QUOROM (calidad de los informes del meta-análisis) (Liberati, y otros, 2009).

El proceso de selección se realizó en diferentes motores de búsqueda y repositorios loas cuales fueron: Redalyc, DOAJ Directory of Open Access Journals, Dialnet, ECORFAN, UNIANDES, Peumo, Repositorio Institucional de la Escuela Superior Politécnica de Chimborazo y Google Académico; se establecieron protocolos de búsqueda y algunos operadores booleanos: [("*ASP.NET webforms MVC*")], [("*transferencia de datos web mvc*") AND ("*2018*") AND ("*ciencias de la información*") AND ("*español*") OR ("*inglés*")], [("*javascript mvc*") OR ("*tiempo respuestas usuario*") OR ("*time answers user*")], [("*serialización/deserialización*") OR ("*serialization/deserialization*")], [("*ejecución eficiente web*") OR ("*efficient web execution*")], [(("*análisis comparativo angular ember*") AND ("*javascript angularjs ember*")) OR (("*angular comparative*

analysis ember") AND ("*javascript angularjs ember*"))], [("*Análisis comparativo Web*") AND ("*2017-2018*") AND [("*JavaScript*")]], [("*comparison framework applications development*") AND ("*2018*")], [("*serialize MVC*")], [("*Performance JavaScript MVC framework*")], [("*Análisis Framework lado cliente*") AND ("*2016 TO 2018*") AND ("*tiempo respuesta*")], [("*mejora tecnológica javascript framework*") AND ("*framework*")], [("*tiempo respuestas usuario*")], [("*researchjournals /sistemas_computacionales_y_tics*")], [("*análisis de framework ASP.NET MVC 5*") AND ("*Gómez Cadena, Jimmy Fabricio*")], [("*(tacapps) javascript framework investigation*")], [("*Performance javascript frameworks SPA*") AND ("*Findel Dávila, Hans. 2015*") AND ("*2015 TO 2018*")], [("comparing performance javascript frameworks") AND ("Ladan, Zlatko")].

Se detalla a continuación los criterios de búsquedas:

Grupo Tiradentes Portal de Periódicos

[("ASP.NET webforms MVC")]

Redalyc

[("transferencia de datos web MVC") AND ("2018") AND ("ciencias de la información") AND ("español") OR ("inglés")]].

DOAJ Directory of Open Access Journals

[("javascript mvc")].

[("tiempo respuestas usuario") OR ("time answers user")]

Dialnet

[(“serialización/deserialización”) OR (“serialization/deserialization”)].

[(“ejecución eficiente web”) OR (“efficient web execution”)].

Repositorio Institucional de la Escuela Superior Politécnica de Chimborazo

[(“análisis comparativo angular ember”) AND (“javascript angularjs ember”)].

[(“angular comparative analysis ember”) AND (“javascript angularjs ember”)].

Repositorio Institucional de la Universidad de las Fuerzas Armadas ESPE

[(“Análisis comparativo Web”) AND (“2017-2018”) AND (“JavaScript”)].

Portal Of Scientific Journals of Croatia

[("comparison framework applications development") AND ("2018")]

The Repository ST. Cloud State University

[(“serialize MVC”)]

Digitala Vetenskapliga Arkivet

[(“Performance JavaScript MVC framework”)]

[(“Comparing performance JavaScript frameworks”) AND (“Ladan, Zlatko”)]

Repositorio Institucional de la Universidad Señor de Sipán

[("Análisis Framework lado cliente") AND ("2016 TO 2018") AND ("tiempo respuesta")]

Peumo

[("mejora tecnológica javascript framework") AND ("framework")].

Además de busco un artículo en **Google Académico** lo cual nos llevó al siguiente repositorio:

ECORFAN y se hizo el siguiente filtro

[("researchjournals/sistemas_computacionales_y_tics")]

Repositorio Digital Universidad Técnica del Norte

[("Análisis de Framework ASP.NET MVC 5") AND ("Gómez Cadena, Jimmy Fabricio")]

Defense Technical Information Center

[("(TACAPPS) JAVASCRIPT FRAMEWORK INVESTIGATION")]

Repositorio de la Pontificia Universidad Católica de Chile

[("Performance javascript frameworks SPA") AND ("Findel Dávila, Hans. 2015") AND ("2015 TO 2018")]

Redalyc (18 documentos encontrados), Grupo Tiradentes Portal de Periódicos (01 documento encontrado), DOAJ (65 documentos encontrados), Dialnet (59 documentos encontrados), Repositorio Institucional de la Escuela Superior Politécnica de Chimborazo (18 documentos encontrados), Repositorio Institucional de la Universidad de las Fuerzas

Armadas ESPE (05 documentos encontrados**)**, Portal de Revistas Científicas de Croacia (01 documento encontrado), The Repository ST. Cloud State University (01 documento encontrado), Digitala Vetenskapliga Arkivet (02 documentos encontrados), Repositorio Institucional de la Universidad Señor de Sipan (10 documentos encontrados), Peumo (03 documentos encontrados), ECORFAN (01 documento encontrado), Repositorio Digital Universidad Técnica del Norte (01 documento encontrado), Defense Technical Information Center (02 documentos encontrados), Repositorio de la Pontificia Universidad Católica de Chile (01 documento encontrado).

En total se encontraron 188 documentos; se buscó información desde al año 2015 hasta el 2018; los idiomas solo fueron español, inglés y con alguna excepción portugués. Cabe resaltar que se realizaron búsquedas diversas ya que la información referente a este tema de investigación es muy sesgada encontrarla en una sola base de datos y/o gestor de búsqueda; Además se encontraron artículos con gran similitud referente al tema de investigación.

Para el proceso de inclusión se tomó en cuenta la lectura completa de los resúmenes y una revisión de la descripción de título, el año de publicación, aplicación de los criterios básicos de temas que guarden relación con el tema de investigación. Del total de 188 documentos encontrados, se lograron descartar 128 ya que no pasaron el primer filtro: 93 no tenían tema apropiado para el trabajo, 21 no tenían un orden consistente en el resultado del estudio y 14 eran idiomas diferentes a español, inglés y portugués. Se filtraron 60 documentos listos para la siguiente etapa de exclusión.

Se procedió a dar una lectura más completa de los 60 documentos preseleccionados para filtrar solo aquellos que tengan relación e información relevante con el tema de investigación del presente trabajo. Se descartaron 45 documentos: 25 no tenían contenido relevante para el estudio, 12 no tenían orden especifico y de poca validez y 08 de descarto por múltiples publicaciones. Se incluyeron 15 documentos finales para el trabajo de investigación.

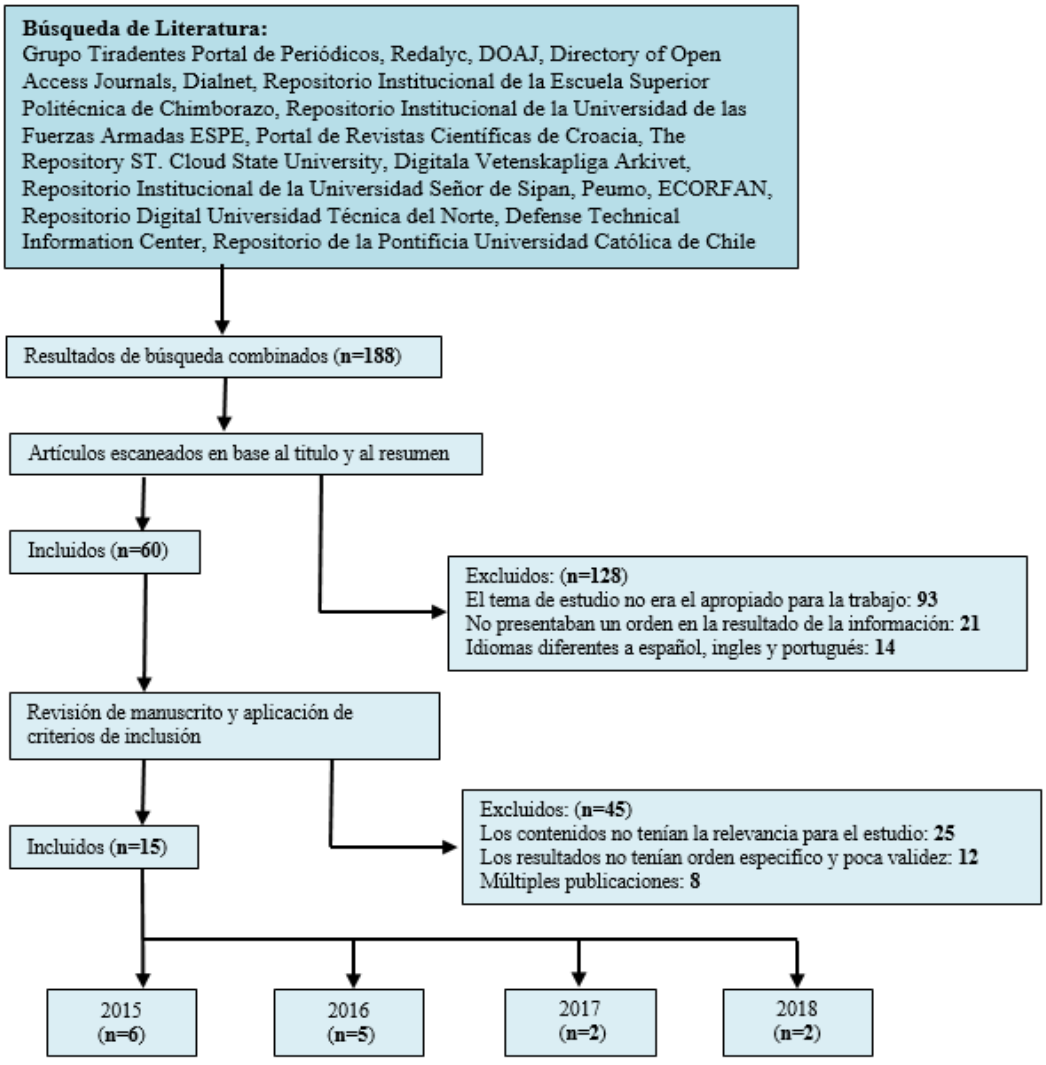

Figura 1. Diagrama de Flujo PRISMA. Fuente. Elaboración Propia

CAPÍTULO III. RESULTADOS

El total de documentos encontrados fueron 188, de estos se excluyeron 128 en la primera etapa, posteriormente se descartaron 45 documentos; quedando posteriormente con 15 documentos para la revisión de la literatura. La Tabla 1 muestra los documentos seleccionados para el presente trabajo de investigación.

Tabla 1
Documentos incluidos para el trabajo de investigación

Fuente	Diseño Metodológico	País	Universidad y/o Institución	Año	Resumen
(Santos, 2018)	Revisión	Brasil	Interfaces Científicas - Exatas e Tecnológicas	2018	En el mundo corporativo actual, la plataforma ASP.NET es ampliamente utilizada para el desarrollo de aplicaciones Web orientadas a objetos, se trata de una tecnología que acopla varios modelos en un único ambiente.
(Kaluža, Troskot, & Vukelić, 2018)	Revisión	Croacia	Portal Of Scientific Journals of Croatia	2018	Las aplicaciones web modernas, debido a las funcionalidades que proporcionan en sus interfaces de usuario, tienen una compleja estructura
(Moreno, 2017)	Tesis	Ecuador	Universidad de las Fuerzas Armadas	2017	JavaScript toma fuerza en desarrollo web. Análisis comparativo Angular 2 y React JS considerado normas de calidad ISO/IEC 9126
(Craig, Tiffany, Norbert, Jaiden, Liqiao, & Ross, 2017)	Informe	USA	Army Armament Research, Development And Engineering Center	2017	El futuro de las demandas de computación táctica es una nueva e innovadora solución para abordar una variedad de necesidades operativas desafiantes
(Mora, 2016)	Revisión	Costa Rica	Universidad Nacional	2016	Métodos de serialización/deserialización de objetos que usa la notación JSON (siglas en inglés de JavaScript Object Notation, o Notación de Objetos de JavaScript)
(Zurita &	Tesis	Perú	Universidad	2016	Análisis y Evaluación de

Zavala, 2016)			Señor de Sipan		frameworks de lado del cliente y del servidor, midiendo tiempos de respuesta y carga del servidor.
(Márquez, 2016)	Tesis	Chile	Universidad Técnica Federico Santa María	2016	Aplicación prototipo funcional sobre la interfaz frontend de ERPS, se realizaron mejoras funcionales utilizando arquitectura REST y un Javascript Framework.
(Ochoa, 2016)	Revista	México	Revista de Sistemas Computacionales y TIC's	2016	Ventajas de la serialización es que proporciona almacenamiento e intercambio de datos, manteniendo información importante entre aplicaciones. Si la información se transmite en binario o XML se facilita la comunicación entre datos
(Manish, 2016)	Revisión	Nepali	ST. Cloud State University	2016	Desarrollo de página estática hasta la dinámica como aplicaciones de negocios electrónicas y redes sociales. Mostrar cómo MVC con ExtJS ha cambiado los patrones de desarrollo web.
(Gómez, 2015)	Tesis	Ecuador	Universidad Técnica del Norte	2015	Las aplicaciones web han abierto un extenso campo de investigación y estudio de nuevas herramientas, metodologías y arquitecturas.
(Hans, 2015)	Tesis	Chile	Pontificia Universidad Católica De Chile	2015	La arquitectura web ha evolucionado en los últimos años. La necesidad de proveer una mejor experiencia de usuario ha forzado a los desarrolladores a agregar más código en el lado del cliente (JavaScript)
(Peña & Cambisaca, 2015)	Tesis	Ecuador	Escuela Superior Politécnica de Chimborazo	2015	Se analizó frameworks JavaScript MVC (Modelo, Vista, Controlador) AngularJs y EmberJs. Analizar e interpretar resultados para análisis comparativo de frameworks.
(Enache, 2015)	Revisión	Rumania	University of Galati	2015	Building full-blown web applications in JavaScript is not only feasible, but increasingly popular. Based on trends HTTP Archive, deployed JavaScript code size has grown 45% over the course of the year.
(Svensson, 2015)	Revista	Usa	School of Computing	2015	Sitios web de hoy son interactivos y elementos cambiantes por segundo. A menudo, un framework

					JavaScript se utiliza para construir un sitio web
(Ladan, 2015)	Revisión	Suecia	Linnaeus University	2015	JavaScript se usa en la web junto con HTML y CSS, en muchos casos usando marcos para JavaScript tales como jQuery y Backbone.js

Nota. Fuente: Elaboración propia.

Introducción a los conceptos

MVC

Es un modelo de desarrollo (patrón de diseño) que propuso descomposición y demarcación de una aplicación en tres componentes interdependientes: el modelo, la vista y el controlador. Los conceptos de MVC son abstractos, para entender mejor estos conceptos hacemos una relación con lo real. (Enache, 2015).

```
<div class="row form-group">
    <label class="col-md-4 col-sm-3 control-label no-padding-right">
        Tipo Documento:
    </label>
    <div class="col-md-8 col-sm-9 col-xs-12">
        <div class="clearfix">
            <select id="xTipoDocumento" class="form-control col-md-12" onchange="valida_longitud();" >
            </select>
        </div>
    </div>
</div>
```

Figura 2. Model = HTML. Fuente. Elaboración Propia.

```
html {
  position: absolute;
  width: 1008;
}
body {
  background-color: #E9E9E9;
}

.sidebar-fixed:before {
  left: 0;
}
```

Figura 3. View = CSS. Fuente. Elaboración Propia.

Figura 4. Controller = Browser. Fuente. Elaboración Propia

Modelo

Contiene la lógica de negocio de la aplicación la misma que puede ser prevista en cualquier almacén de datos, encapsula el estado de la aplicación, es independiente del controlador y la vista.

Vista

Contiene la presentación de la capa de modelado, encargada de la visualización de la interfaz de usuario de la aplicación, puede acceder al modelo pero nunca cambiar su estado, puede ser notificado cuando hay un cambio de estado en el modelo.

Controlador

"Es la capa intermedia que interactúa entre la capa de modelo y vista". (Gómez, 2015, págs. 8-9).

Vistas Razor

"Motor de vistas, fácil de usar, contienen bloque de código para realizar peticiones al servidor y generar y mostrar el HTML". (Gómez, 2015)

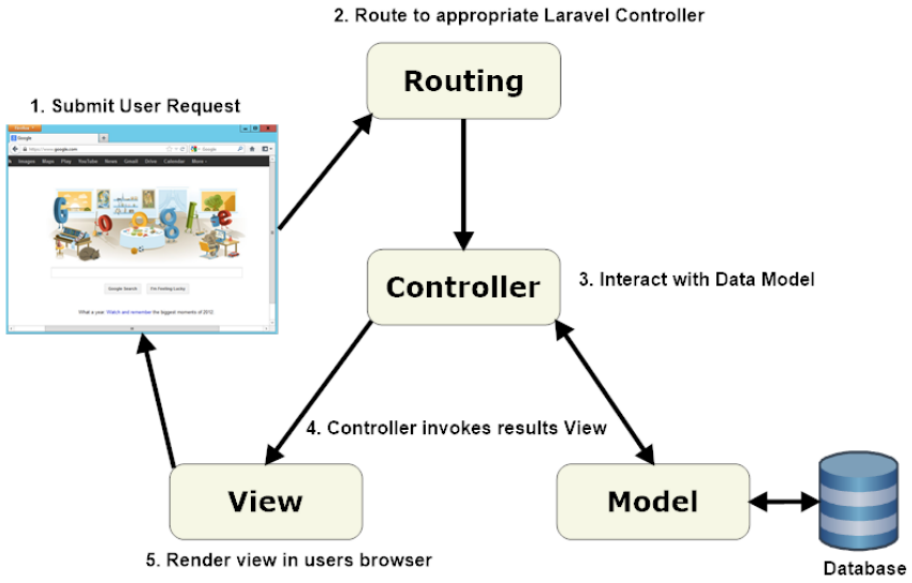

Figura 5. Esquema Model Vista Controlador. Fuente (Enache, 2015)

HTML5

HyperText Mark-up Language 5 es la versión más reciente del lenguaje y empezó a ser desarrollado por el Web Hypertext Application TechnologyWorking Group (WHATWG). "*En el año 2006 el W3C toma interés en HTML5 y en el año 2007 se forma un grupo de trabajo con el WHATWG para desarrollar las especificaciones de éste. La W3C quiere crear un estándar sólido y duradero con pocas modificaciones entre las versiones. El 28 de octubre del año 2014 el W3C publicó la recomendación de HTML5 y el 8 de octubre del 2015 se publica el borrador en progreso de HTML5.1."* (Márquez, 2016)

JavaScript

"Lenguaje de programación interpretado más usado en desarrollo de aplicaciones Web, JavaScript no es un lenguaje orientado a objetos. Se basa en prototipos, las clases nuevas se generan clonando las clases base. No necesita compiladores, se ejecuta del lado del cliente, es trasparente y visible por cualquier navegador web". (Peña & Cambisaca, 2015)

jQuery

Es una biblioteca pequeña y rápida con muchas funciones, es simple de usar, funciona en múltiples navegadores web. Microsoft y Nokia lo utilizan en sus plataformas. Maneja Ajax (JavaScript y XML asíncronos) del lado del cliente para aplicaciones web asíncronas. (Ladan, 2015)

Framework AngularJS

Creado para aplicaciones web dinámicas, Utiliza HTML para representar la información. AngularJS trabaja con directivas, permite reducir líneas de código mediante enlace de datos, inyecciones de dependencia; es 100% JavaScript del lado del cliente. Es compatible con Java, Phyton, Ruby, C# o cualquier otro lenguaje. La sincronización vista - modelo es automática y en tiempo real, usa peticiones HTTP XML o JSON. El "bidireccional data binding", ofrece ventajas considerable: **One-way binding:** desde el Scope hacia la parte visual (modelo - vista). **Two-way binding**: desde Scope hacia la vista y viceversa. (Peña & Cambisaca, 2015)

```
<html ng-app>
<body>
<h1> Hola {{5 + 2}} </h1>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.15/angular.min.js"></script>
</body>
</html>
```

Figura 6. Ejemplo AngularJS Utilizando CDN Google. Fuente (Peña & Cambisaca, 2015)

Framework EmberJS

Es de código abierto, basado en MVC y aplicaciones web de lado del cliente, permite a los programadores crear aplicaciones single-page (una sola página). Implementa clases y procedimientos que ayudan a controlar data, permite definir plantillas; automatiza los cambios, si el objeto JavaScript cambia el DOM cambia y viceversa. Trabaja con 3 dependencias. "jquery.js", "handlebars.js" y "ember.js" (Peña & Cambisaca, 2015)

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title> Primera App con Ember.js </title>
<script src="jquery.min.js"></script>
<script src="handlebars.js"></script>
<script src="ember.js"></script>
<script>
window.App = Ember.Application.create();
</script>
</head>
<body>
</body>
</html>
```

Figura 7. Ejemplo EmberJS en una página índex. Fuente (Peña & Cambisaca, 2015)

Framework Aurelia

Aurelia es uno de frameworks JavaScript más nuevos y tiene algunas similitudes con Angular 2.0. Tiene una arquitectura MVVM y está destinado a ser altamente modular donde solo incluyes lo que necesitas. Utiliza Polyfills que permite a navegadores web antiguos utilizar módulos más nuevos para trabajar. También utiliza enlaces bidireccionales como lo hace Angular, lo que hace que trabajar con los datos sea fácil y flexible (Svensson, 2015).

Framework BackboneJS

Es uno de los primeros frameworks de JavaScript que se hizo muy popular, pero actualmente su popularidad ha ido disminuyendo, usa el mismo enfoque que Ember para manejo de objetos e intercambios de datos, proporciona una estructura a las aplicaciones web al proporcionando modelos con combinación de valores clave y eventos personalizados, colecciones con abundante de API de enumerables funciones, vistas con

manejo de eventos y se puede conectar con API existente como una interfaz RESTful JSON (Svensson, 2015).

Framework Knockout

Knockout es uno de los primeros marcos de JavaScript que se hizo popular en el comienzo de 2011 y sigue siendo bastante popular según las tendencias de Google. Está categorizado como Marco MVVM (Model View ViewModel) donde ViewModel observa los cambios de datos en la Vista o el Modelo. Ayuda a crear contenido rico y receptivo, mostrar y editar las interfaces de usuario con un modelo de datos subyacente limpio (Svensson, 2015).

Framework Mithril

No es el marco más popular en este momento, todavía es bastante nuevo en comparación con algunos otros marcos famosos como Knockout, Backbone y Ember. Mithril es un pequeño trabajo de marco MVC que debe ser lo más simple posible y ligero. Mithril aprovecha la experiencia del desarrollador con los frameworks MVC del lado del servidor, es muy similar en alcance a AngularJS. La principal diferencia es como manejan sus plantillas (Svensson, 2015, pág. 16).

Framework VueJS

Uno de los frameworks de JavaScript más nuevos y está creciendo rápidamente en popularidad. VueJS tiene mucho en común con Angular 2.0, que es una de las razones de su popularidad. Tiene una clara separación entre directivas y componentes que han sido un

tema confuso para muchos usuarios de Angular, es fácil y muy similar al estándar JavaScript (Svensson, 2015, pág. 17).

Framework ExtJS

Es un framework JavaScript de código fuente altamente robusto, escalable y de código abierto. Hay diferentes utilidades que hacen que el Modelo de Objetos de Documento (DOM) manipulación y DOM transversal sea muy estable y fácil. Permite diseñar mejor y más rápido para luego validarlo con varios usuarios; no presenta problemas de compatibilidad con los navegadores y está orientada a objetos; también se le conoce como Sencha (Manish, 2016).

The Document Object Model (DOM)

El **modelo de objeto de documento** (DOM) es una interfaz de programación para HTML, XML y SVG (archivos vectoriales). Los elementos se convierten en un objeto que tiene métodos, propiedades y valores. Crea un árbol de nodos de interfaz estructurados. El navegador recibe el texto del documento (HTML, XML o SVG). Se puede acceder a los nodos de DOM con un lenguaje de programación como JavaScript, pero éste no es el único sino el más popular (Svensson, 2015).

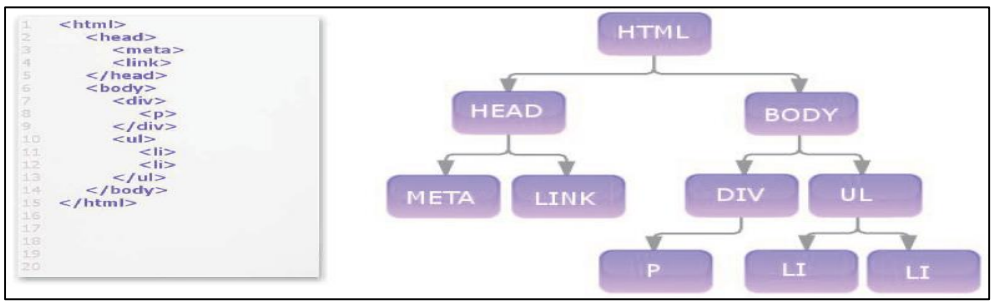

Figura 8. HTML convertido a un árbol de modelo de objetos de documento (DOM). Fuente (Svensson, 2015)

Serialización/deserialización

Se define como el proceso de convertir un objeto en un formato leíble, que se pueda transportar entre aplicaciones. La deserialización convierte el formato leíble a su forma original. Consiste en transformar una variable como una tabla en una cadena de caracteres, convierte una secuencia de bytes y lo trasmite a archivo, base de datos o memoria; el objeto serializado almacena datos como el tipo de objeto y estado. (Ochoa, 2016).

JSON (JavaScript Object Notation)

Es un formato de codificación de datos eficaz que permite intercambios rápidos de cantidades pequeñas de datos entre los exploradores de cliente como Internet Explorer, Google Chrome y servicios web, por lo cual se le considera una gran librería de desarrollo web para la serialización/deserialización y la transmisión de datos (Mora, 2016, pág. 121). Formato ligero de intercambio de datos, puede ser leído por cualquier lenguaje de programación, permitiendo intercambiar información entre diferentes tecnologías. (Peña & Cambisaca, 2015)

```
Objetos   {"estudiante":"José Zoto" , "edad":22 }
Array     {
            "estudiantes": [      { "estudiante":"Martín Rojas" , "edad":21 },
                                  { "estudiante":"Rosa Juarez" , "edad":19 },
                                  { "estudiante":"Maria Bertiz" , "edad":24 }
                            ]
          }
```

SOAP (Simple Object Access Protocol)

Es un protocolo basado en XML, sirve para transportar llamadas a procedimientos. Si hay una propiedad que contiene una referencia a otra clase, serializará también esta última. (Ochoa, 2016).

Clasificación de los Estudios

Se clasificó y se categorizo a los estudios por país, tipo de investigación y año de publicación. Las Tablas 2, 3 y 4 muestran las clasificaciones respectivas.

Tabla 2
Estudios clasificados por País

País	Porcentaje %
Ecuador	20.00
USA	13.33
Chile	13.33
Croacia	06.67
Suecia	06.67
Rumania	06.67
Nepal	06.67
México	06.67
Costa Rica	06.67
Brasil	06.67
Perú	06.67

Nota. Fuente (Elaboración propia)

Tabla 3
Estudios clasificados por Tipo de Investigación

Tipo de Investigación	Porcentaje %
Cuantitativo cuasi experimental	53.33
Cualitativo descriptivo	26.67
Cuantitativo experimental	13.33
Cuantitativo y Cualitativo cuasi experimental	06.67

Nota. Fuente (Elaboración propia)

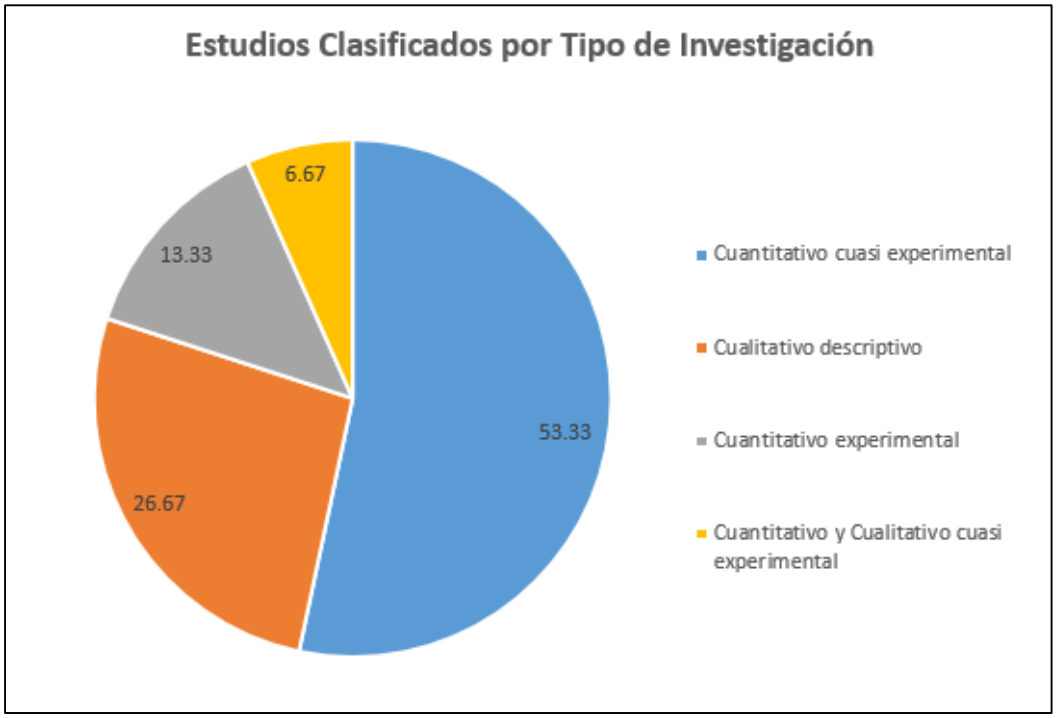

Figura 9. Porcentaje de Estudios por Tipo de Investigación. Fuente. elaboración propia

Tabla 4
Estudios clasificados por Año

Año de Publicación	Porcentaje %
2015	40.00
2016	33.33
2017	13.33
2018	13.33

Nota. Fuente (Elaboración propia)

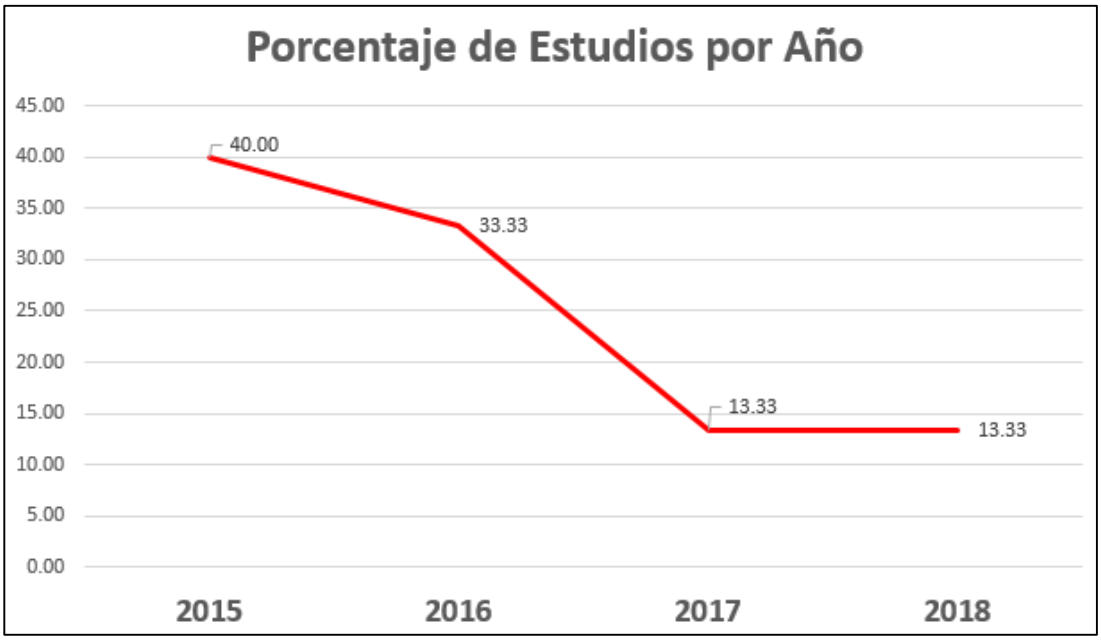

Figura 10. *Porcentaje de estudios por Año*

Resultados de Estudios

Año 2015

(Gómez, 2015) En su estudio realizado demuestra que ASP.NET MVC presenta mejor puntuación que WebForm tradicional. Los criterios empleados fueron curva de aprendizaje, escalabilidad, compatibilidad, rendimiento, tendencia de uso, soporte técnico, fácil desarrollo, pruebas unitarias, control de html y arquitectura. Las métricas de valoración fueron de 00 a 04 donde 04 es la valoración más alta. ASP.NET MVC obtuvo mayor valoración. La Tabla 5 muestra el cuadro comparativo de dicha investigación.

Tabla 5

Comparativa entre ASP.NET MVC vs Webforms

MÉTRICA	ASP.NET MVC	ASP.NET WEB FORMS
Curva de aprendizaje	01	04
Escalabilidad	04	02
Compatibilidad	04	03
Rendimiento	04	02
Tendencia de uso	03	03
Soporte técnico	04	04
Fácil desarrollo	01	04
Pruebas unitarias	03	01
Control de HTML	04	03
Arquitectura	04	02
Total	**32**	**28**

Nota. Fuente: (Gómez, 2015)

(Enache, 2015) Menciona en su estudio que cuando el problema es grandes volúmenes de datos y la información a procesar es compleja es óptimo utilizar arquitecturas MVC; algunos de estos Framework incluyen conceptos como autenticación, autorización, caché, bases de datos, encriptación, ubicación, API, etc. MVC permite suministran sistemas que incluyen un diseño de patrón óptimo de uso, conocido, apropiado, redundancia mínima, escalabilidad, fácil de mantenimiento, reducir costos y previsibilidad del desarrollo.

En una tesis realizada por (Hans, 2015) se hizo los tests de algunos Frameworks JavaScript (Angular, Backbone, Ember, Marionette, React) en dos navegadores Google Chrome y Mozilla Firefox. Para realizar las pruebas se agregaron en simultaneo 1000 tareas al navegador web para ver la capacidad de estrés del framework, posteriormente se eliminaron uno por uno las tareas para ir analizando el performance; finalmente se incrementó nuevamente en 5000 tareas para realizar las pruebas. También se realizó pruebas de carga de datos con una muestra de 1000 registros. Los resultados arrojaron un valor óptimo para ReactJS. La Figura 9 muestra el resultado de esta prueba.

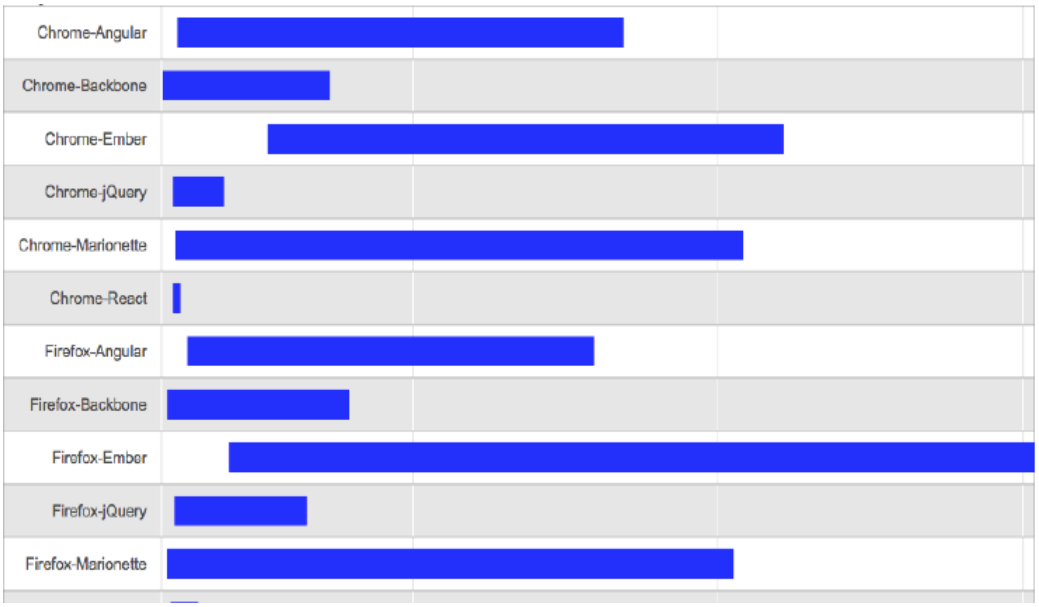

Figura 11 Tiempo de procesamiento y carga de datos con 1000 registros. Fuente (Hans, 2015)

(Ladan, 2015) Realizó un estudio de JavaScript, jQuery y Backbone.js. Referente al tiempo: tiempo de ejecución con tiempo medio al agregar una tarea, tiempo de ejecución con tiempo medio al eliminar una tarea y tiempo de ejecución con tiempo medio al editar una tarea. Referente a tamaño en Kilobytes: asignación de kilobytes con valor medio al cargar la página de tareas, asignación de kilobytes con valor medio al agregar una tarea, asignación de kilobytes con valor medio al eliminar una tarea y asignación de kilobytes con valor medio al editar una tarea. Los resultados fueron favorables para JavaScript. Las Tablas 6 y 7 muestran los resultados.

Tabla 6
Valores medios para todos los tiempos de CPU en milisegundos

Test Subject	Loading	Adding	Removing	Editing
JavaScript	006.1	07.0	06.9	10.3
jQuery	347.2	08.5	09.4	49.8
Backbone.js	330.6	33.3	16.2	37.4

Nota. Fuente (Ladan, 2015)

Tabla 7
Valores medios para asignaciones de pila en kilobytes

Test Subject	Loading	Adding	Removing	Editing
JavaScript	272.0	17.1	01.7	05.9
jQuery	526.2	68.8	24.3	30.6
Backbone.js	632.0	56.9	25.2	38.1

Nota. Fuente (Ladan, 2015)

En un estudio realizado entre AngularJS y EmberJS se analizó los resultados del porcentaje de consumo o utilización de los recursos del CPU: procesador CPU, memoria Ram, disco duro, tarjeta de red y tarjeta gráfica. Se emplearon métodos GET Y POST para las diferentes pruebas. Se asignó una ponderación del 20% a todos los indicadores. A menor consumo o utilización de recursos, se obtiene un mayor rendimiento y viceversa;

por lo que se infirió que AngularJS presenta mejor rendimiento. La Tabla 8 muestra los resultados del estudio (Peña & Cambisaca, 2015).

Tabla 8
Porcentajes de Consumo de Recursos, por ponderación de indicador

Indicadores	**Ponderación**	**AngularJS**		**EmberJS**	
		%	**%Consumo**	**%**	**%Consumo**
CPU	20%	40,22	08,044	59,78	11,956
Memoria RAM	20%	67,66	13,532	86,83	17,366
Disco	20%	55,23	11,046	54,63	10,926
Red	20%	19,21	03,842	55,30	11,060
Tarjeta Gráfica	20%	73,21	14,642	89,29	17,858

Nota. Fuente (Peña & Cambisaca, 2015)

(Svensson, 2015) Realizó un estudio para determinar cuál es el mejor framework JavaScript, los framework seleccionados fueron (Angular 1.5, Angular 2.0, Aurelia, Backbone.js, Ember, Knockout, Mithril y Vue). Las pruebas fueron: crear 1000 filas, eliminar todas las 1000 filas, actualizar todas las 1000 filas, actualizar 1/3 de las 1000 filas y eliminar 1/3 de las 1000 filas. Se realizó en Google Chrome versión 50.0.2661.94, Windows 8 64bit, Intel Core i5-42000 CPU 1.60GHz 2.30Ghz 12 GB RAM. Los resultados fueron favorables para el Framework JavaScript AngularJS 2.0. Los resultados se muestran en la Tabla 9.

Tabla 9
Tiempo promedio en milisegundos de las diferentes pruebas

	Knockout	Ember	Backbone	Angular 1.5	Mithril	Vue	Aurelia	Angular 2.0
Crear 1000 filas	792	1228	1386	733	665	631	570	475
Actualizar 1000 filas	576	1133	1208	151	467	207	201	149
Eliminar todas las 1000 filas	71	99	63	113	38	62	91	78
Actualizar 1/3 de 1000 filas	309	516	513	133	360	169	169	132
Eliminar 1/3 de 1000 filas	207	333	143	358	484	360	369	145

Nota. Fuente (Svensson, 2015)

Año 2016

En su trabajo de tesis (Zurita & Zavala, 2016) realizo un análisis comparativo de Frameworks tanto del lado del cliente y como del lado del servidor. Se emplearon criterios como funcionalidad, confiabilidad, eficiencia, mantenimiento y portabilidad; también se contó con opiniones de expertos en el tema para tener un nivel mejor de certeza. Los Frameworks fueron clasificados por el lado del cliente: AngularJS y BackboneJS y por el lado del servidor: Lavarel y Symfony. Para nuestro caso de estudio solo haremos referencias a los Frameworks del lado del cliente. AngularJS presentó mayor puntaje promedio de características de calidad. La Tabla 10 muestra los promedios de medición de características de la calidad.

Tabla 10
Promedios de Medición de características de la calidad

Característica	AngularJS	BackboneJS
Funcionalidad	09.5	09.5
Confiabilidad	09.5	09.1
Eficiencia	09.7	09.5
Mantenimiento	10.0	08.8
Portabilidad	10.0	09.8

Nota. Fuente (Zurita & Zavala, 2016)

(Manish, 2016) Afirma que MVC tiene beneficios porque desacopla una aplicación en diferentes elementos y permite a un desarrollador enfocarse por separado en cada elemento. Es una de las mejores prácticas para desarrollar aplicaciones web y existen muchas aplicaciones implementadas con esta arquitectura. De otro lado ExtJS es un Framework más completo que otros Framework JavaScript; se hizo una comparación de dos aplicaciones, una utilizo ExtJS y otra cshtml y ExtJS ofrece más beneficios.

(Márquez, 2016) Realizó un análisis cuantitativo y cualitativo para determinar que Framework JavaScript tiene mayor aceptación a la hora de su elección para la construcción de una aplicación web. La Tabla 11 muestra el resultado del análisis; se revisó las estadísticas de cada JavaScript Framework en GitHub. La mayor nota obtenida fue para AngularJS. Se utilizaron los siguientes criterios:

- C1. Popularidad y tamaño de la comunidad.
- C2. Soporte y resolución de problemas.

- C3. Seguridad.
- C4. Documentación.
- C5. Licencia y restricciones de uso.
- C6. Patrones y Filosofía de diseño.
- C7. Actualizaciones.

Tabla 11
Notas de JavaScript Framework por criterio

JS Framework	C1	C2	C3	C4	C5	C6	C7	Nota
AngularJS	2	1	2	2	2	2	2	13
React	2	1	1	2	2	2	2	12
EmberJS	2	2	1	1	2	1	2	11
Marionette	1	2	1	2	2	2	1	11
Dojo	1	2	2	2	2	1	1	11
VueJS	1	2	2	2	2	1	1	11
Backbone	2	2	0	2	2	2	1	11
Ampersand	0	1	2	2	2	2	1	10
Flight	0	1	2	2	2	1	1	09
CanJS	1	2	1	2	2	1	0	09
KnockoutJS	0	1	1	1	2	2	1	08
Knockback	0	2	1	1	2	2	0	08
Mithril	0	2	1	1	2	1	0	07
Troopjs	0	0	0	0	2	1	0	03

Nota. Fuente (Márquez, 2016)

(Mora, 2016) Menciona que JSON ofrece grandes ventajas para la serialización/deserialización e intercambio de datos, puede serializar/deserializar 100 veces más rápido que XML en navegadores modernos. Se analizó JSON y otra tecnología para comparar la serialización/deserialización y transmisión de datos en un ambiente igual, un servidor receptor y un cliente emisor; JSON demostró ser más rápido en serialización/deserialización, pero requiere un poco más recursos de CPU. Se analizó también con YAML y JSON resulto mucho más rápido en serializar/deserializar. Es ideal para envió de datos mediante servicios web y el protocolo REST.

(Ochoa, 2016) En un estudio sobre tipos de serialización utilizando .NET Framework; se realizaron 03 tipos: Serialización XML, Binaria y SOAP. Se demostró que la serialización XML no guarda métodos privados, campos privados, propiedades de solo lectura o indexadores; solo se serializan los miembros públicos. Cuando la serialización fue binaria y/o SOAP, todos los miembros de la clase son serializados incluso de ámbito privado y protegido y si un atributo referencia a otra clase, ésta también es serializada. Cuando se desee implementar la serialización en necesario valorarlo por facilidad de uso y la flexibilidad; además mencionó que la serialización JSON es más ligera y rápida.

Año 2017

En un estudio realizado por (Moreno, 2017) se aplica ISO/IEC 9126 para verificar las características de calidad entre Angular 2 y ReactJS; estas características fueron:

- **Funcionalidad**: idoneidad, exactitud, interoperabilidad, seguridad.
- **Fiabilidad:** madurez, tolerancia a fallos, recuperabilidad.
- **Usabilidad:** comprensibilidad, capacidad de aprendizaje, operabilidad, estética.
- **Eficiencia:** comportamiento en el tiempo, uso de recursos.
- **Mantenibilidad:** analizibilidad, posibilidad de cambio, estabilidad, testeabilidad.
- **Portabilidad:** adaptabilidad, instalación, coexistencia, reemplazabilidad.

Se demostró que la herramienta con mejor proyección es Angular 2, teniendo una ventaja mayor en la usabilidad y mantenibilidad, seguida de la portabilidad y funcionalidad. En Fiabilidad, Eficiencia tiene desventaja con relación a ReactJS. La Tabla 12 muestra el resumen de los resultados.

Tabla 12
Resumen de evaluación Web Components

Característica	**Máximo**		**Angular 2**		**React JS**	
	%	**Puntaje**	**%**	**Puntaje**	**%**	**Puntaje**
Funcionalidad	15,66	13	13,25	11	12,05	10
Fiabilidad	20,48	17	15,66	13	18,07	15
Usabilidad	19,28	16	18,07	15	15,66	13
Eficiencia	09,64	08	07,23	06	09,64	08
Mantenibilidad	15,66	13	15,66	13	13,25	11
Portabilidad	19,28	16	18,07	15	16,87	14
Total	**100.00**	**83**	**87,95**	**73**	**85,54**	**71**

Nota. Fuente (Moreno, 2017)

(Craig, Tiffany, Norbert, Jaiden, Liqiao, & Ross, 2017) Realizaron un estudio de diversos Frameworks para analizar cuál es el apropiado en el desarrollo de una aplicación móvil llamada TacApps. Se descartó AngularJS, a pesar de ser popular para la creación de prototipos, pero presento problemas de estabilidad entre una versión y otra. La interfaz de jQuery fue lenta y no proporcionó widgets ricos en características y además no es un Framework netamente para aplicativos móviles. Ember y Meteor fueron descartados como viables para el presente desarrollo. Web Components presentó limitaciones en algunos navegadores y solo son compatibles con Google Chrome y Opera. Para el desarrollo de TacApps se decidió por ReactJS ya que es un Framework de "solo lectura", esto no enlaza al modelo y controlador en el patrón MVC, permitiendo el desarrollo de interfaz de usuario (vista). Así mismo permite interacción fluida con otros Frameworks y bibliotecas de JavaScript.

Año 2018

(Santos, 2018) En su estudio realizó un experimento entre ASP.NET MVC y Web Form, se contó con la participación de 06 desarrolladores con amplia experiencia y consistía en desarrollar un mismo prototipo de registro aplicando los siguientes criterios: tiempo de codificación, media de cantidad de errores y media de cantidad de líneas de código. MVC presenta ventajas en tiempo de codificación y Web Form en cantidad de líneas de código. La Tabla 13 muestra el resumen de los resultados obtenidos.

Tabla 13
Resultados del Experimiento segun criterios

Criterio	ASP.NET MVC	Web Form
Tiempo de codificación (minutos)	25.50	39.83
Media de cantidad de errores	0.102	0.102
Media de cantidad de líneas de código	0.935	0.902

Nota. Fuente (Santos, 2018)

(Kaluža, Troskot, & Vukelić, 2018) Realizó un estudio desde el entorno SPA y MPA dado que existen diferencias entre éstas. SPA (Single Page Application): Es un enfoque más nuevo para desarrollo de aplicaciones web más simples con menos contenido. MPA (Multi Page Application): Aplicaciones destinados a sistemas más grandes con gran cantidad de diferentes tipos de servicios y más interacciones con visitantes. Los Frameworks analizados fueron VueJS, ReactJS y AngularJS. AngularJS presenta mayor puntaje en SPA y VueJS en MPA. Los resultados se resumen en la Tabla 14. Los criterios para MPA fueron:

- ¿Sólo la importación Js es suficiente para comenzar?
- ¿El Framework incluye demasiada sobrecarga?
- ¿Es el flujo de trabajo necesario?
- Representación del servidor.

Los criterios para SPA fueron:

- La capacidad de escribir y mantener una gran cantidad de código complejo
- ¿Debería el desarrollador confiar en paquetes de terceros?
- ¿Cuál es la posibilidad de compresión de aplicación, minimizando los archivos?
- Gestión de estado de datos

Tabla 14
Resultados de valores numéricos de análisis de Framework

Framework	MPA	SPA	Total
VueJS	08.0	06.0	14
ReactJS	05.5	04.5	10
AngularJS	02.0	07.0	09

Nota. Fuente (Kaluža, Troskot, & Vukelić, 2018)

CAPÍTULO IV. DISCUSIÓN Y CONCLUSIONES

Conclusiones

El presenta trabajo científico de la revisión de la literatura fue realizado en el periodo 2015 al 2018 y se buscó información en diferentes bases de datos. Se logró analizar estudios de los frameworks JavaScript y MVC, técnicas de serialización/deserialización de datos y analizar su incidencia en el rendimiento.

El análisis de los resultados nos indica que de diversos Frameworks JavaScript analizados y comparados el más óptimo trabajando con el patrón MVC es el Framework AngularJS, este presenta mayores beneficios tanto en tiempos de respuesta, rendimiento, usabilidad, mantenibilidad y portabilidad.

Desde el entorno SPA AngularJS ofrece mayores ventajas pero contrariamente desde MPA VueJS es mejor.

Se pudo evidenciar que JavaScript presento mejor tiempo de respuesta que el framework BackboneJS y jQuery.

Otro framework que ofrece ventajas pero que es superado por AngularJS es ReactJS, este presenta beneficios como eficiencia, fiabilidad; además éste framework trabaja mejor que otros en el desarrollo de aplicaciones móviles.

Se evidencia que la arquitectura MVC ofrece grandes ventajas en comparación con otras arquitecturas en cuanto a escalabilidad, rendimiento, tratamiento de grandes volúmenes de datos, separación de componentes de una aplicación y en arquitectura propiamente dicha.

Referente a la serialización/deserialización se menciona que JSON es más óptimo para el intercambio y tratamiento de datos; éste serializa más rápido que XML y YAML. La serialización XML presenta algunas desventajas ya que solo serializa atributos públicos y no los privados.

Se identifica que la utilización de un framework JavaScript presenta beneficios en el rendimiento de aplicaciones web; estos beneficios van de la mano con el procesamiento de grandes volúmenes de datos y que cumplen estándares de calidad.

Se concluye con esta revisión sistémica que el uso de un Framework JavaScript MVC incide positivamente en el rendimiento tales como mejora el tiempo de respuesta, mayor usabilidad, mejor mantenibilidad y mayor portabilidad.

Se recomienda así mismo que en investigaciones futuras se realicen comparaciones a nivel más literario y que las investigaciones no sean tan técnicas, este ayudaría a un mejor entendimiento y conocimiento teórico del tema a estudiar. También mencionar que se puedan realizar pruebas comparativas en otros navegadores u otros escenarios, ya que se notó que solo se usaron navegadores como el Firefox y Google Chrome y puede haber un sesgo al hacer las pruebas en otros navegadores.

REFERENCIAS

Bendezú, J., & Figueroa, C. (2017). *Evaluación de la eficiencia, según la Norma ISO 9126, de un sistema web desarrollado e implementado en el área de Ventas y Servicios de la empresa Intecsh.* Obtenido de http://repositorio.udh.edu.pe/bitstream/handle/123456789/731/Bendezu%20Cabello%2c%20Jean%20Carlos.pdf?sequence=1&isAllowed=y

Correia, A. (2016). *JS Performance Certifier.* Obtenido de https://repositorio-aberto.up.pt/handle/10216/104323

Craig, K., Tiffany, R., Norbert, A., Jaiden, C., Liqiao, H., & Ross, A. (2017). *Tactical Applications (Tacapps) Javascript Framework Investigation.* Obtenido de www.dtic.mil/get-tr-doc/pdf?AD=AD1025789

Echeverria, D. (2016). *Tiempo de Respuestas y Experiencia de Usuario Estudio Experimental.* Obtenido de http://revistas.unla.edu.ar/software/article/view/1280/1115

Enache, M. (2015). *Web Application Frameworks.* Obtenido de https://doaj.org/article/46e46caf91fb415c8c837fb217b467f1

Garcia, J., López, J., Jiménez, F., Ramírez, Y., Lino, L., & Reding, A. (2014). *Metodología de la investigación, bioestadística y bioinformática en ciencias médicas y de la salud.* Obtenido de http://ebookcentral.proquest.com/lib/upnortesp/reader.action?docID=3219463&query=revisiones+sistem%C3%A1ticas

Gómez, J. (2015). *Análisis de Framework ASP.NET MVC 5. Prototipo sistema para el control interno de uso del tiempo de los recursos humanos de CRERATEC S.A.* Obtenido de http://repositorio.utn.edu.ec/handle/123456789/4637

Hans, D. (2015). *Performance of JavaScript frameworks on web single page applications(SPA).* Obtenido de https://repositorio.uc.cl/handle/11534/16587

Kaluža, M., Troskot, K., & Vukelić, B. (2018). *Comparison of Front-End Frameworks for Web Applications Development.* Obtenido de https://hrcak.srce.hr/index.php?show=clanak&id_clanak_jezik=294389

Ladan, Z. (2015). *Comparing performance between plain JavaScript and popular JavaScript frameworks.* Obtenido de http://www.diva-portal.org/smash/record.jsf?dswid=-4965&pid=diva2%3A785240&c=1&searchType=SIMPLE&language=en&query=Comparing+performance+JavaScript+frameworks&af=%5B%22personName%3A%5C%22Ladan%2C+Zlatko%5C%22%22%5D&aq=%5B%5B%5D%5D&aq2=%5B%5B%5D%5D&aqe=%5B

Liberati, A., Altman, D., Tetzlaff, J., Mulrow, C., Gotzsche, P., Ioannidis, J., y otros. (2009). *The PRISMA Statement for Reporting Systematic Reviews and Meta-Analyses of Studies That Evaluate Health Care Interventions: Explanation and Elaboration.* Obtenido de http://journals.plos.org/plosmedicine/article?id=10.1371/journal.pmed.1000100

Manish, S. (2016). *Web development using C# MVC and ExtJS.* Obtenido de http://repository.stcloudstate.edu/csit_etds/6/

Márquez, D. (2016). *Propuesta de mejora tecnológica utilizando un JavaScript Framework para la Empresa Sidekick.* Obtenido de https://repositorio.usm.cl/handle/11673/13669

Mora, J. (2016). *Serialización/deserialización de objetos y transmisión de datos con JSON: una revisión de la literatura.* Obtenido de https://dialnet.unirioja.es/servlet/articulo?codigo=5432058

Moreno, M. (2017). *Análisis comparativo de herramientas orientadas a componentes Web validado con un caso de estudio.* Obtenido de http://repositorio.espe.edu.ec/handle/21000/13539

Ochoa, R. (2016). *La serialización de datos utilizando un Framework de desarrollo integrado.* Obtenido de

http://www.ecorfan.org/spain/researchjournals/Sistemas_Computacionales_y_TICs/vol2num4/Revista_de_Sistemas_Computacionales_y_TIC%60S_V2_N4.pdf#page=65

Peña, J., & Cambisaca, M. (2015). *Análisis Comparativo entre los frameworks JavaScript MVC, AngularJs Y Ember JS para el desarrollo de aplicaciones Web. Caso Práctico: "Sistema de Control de Botiquin Veterinario para el MAGAP, Morona Santiago"*. Obtenido de http://dspace.espoch.edu.ec/handle/123456789/4583

Santos, A. (2018). *Análise Comparativa entre os modelos de desenvolvimento ASP.NET WebForms e ASP.NET MVC por meio de um experimento controlado.* Obtenido de https://periodicos.set.edu.br/index.php/index/search/search?query=ASP.NET+WEB FORMS+MVC&searchJournal=&authors=&title=&abstract=&galleyFullText=&suppFiles=&dateFromMonth=&dateFromDay=&dateFromYear=&dateToMonth=&dateToDay=&dateToYear=&dateToHour=23&dateToMi

Svensson, A. (2015). *Speed Performance Comparison of JavaScript MVC Frameworks.* Obtenido de http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A998701&dswid=1480

Zurita, M., & Zavala, C. (2016). *Análisis y evaluación del trabajo de los Framework del lado del cliente y del lado del servidor, para medir tiempos de respuesta y carga del servidor mediante la elaboración de un CRUD.* Obtenido de http://repositorio.uss.edu.pe/handle/uss/344