

# FACULTAD DE INGENIERÍA

Carrera de Ingeniería de Sistemas  
Computacionales



“ANÁLISIS DE LOS FRAMEWORKS JAVASCRIPT  
NATIVO Y ANGULAR EN LA INCIDENCIA DEL  
TIEMPO DE RESPUESTA EN UNA WEB MVC EN EL  
SECTOR COMERCIAL”

Tesis para optar el título profesional de:

INGENIERO EN SISTEMAS COMPUTACIONALES

Autor:

Juan Carlos Espinoza Pereda

Asesor:

Mg. Jean Janssen Matos Manguinuri

Lima - Perú

2020

## ACTA DE AUTORIZACIÓN PARA SUSTENTACIÓN DE TESIS

El asesor Mg. Jean Janssen Matos Manguinuri, docente de la Universidad Privada del Norte, Facultad de Ingeniería, Carrera profesional de **INGENIERÍA DE SISTEMAS COMPUTACIONALES**, ha realizado el seguimiento del proceso de formulación y desarrollo de la tesis de los estudiantes:

- Espinoza Pereda Juan Carlos

Por cuanto, **CONSIDERA** que la tesis titulada: Análisis de los Frameworks JavaScript Nativo y Angular en la incidencia del tiempo de respuesta en una Web MVC en el sector comercial para aspirar al título profesional de: Ingeniero de Sistemas Computacionales por la Universidad Privada del Norte, reúne las condiciones adecuadas, por lo cual, **AUTORIZA** al o a los interesados para su presentación.

---

Mg. Jean Janssen Matos Manguinuri

Asesor

## ACTA DE APROBACIÓN DE LA TESIS

Los miembros del jurado evaluador asignados han procedido a realizar la evaluación de la tesis de los estudiantes: *Haga clic o pulse aquí para escribir texto*, para aspirar al título profesional con la tesis denominada: Análisis de los Frameworks JavaScript Nativo y Angular en la incidencia del tiempo de respuesta en una Web MVC en el sector comercial

Luego de la revisión del trabajo, en forma y contenido, los miembros del jurado concuerdan:

**Aprobación por unanimidad**

**Aprobación por mayoría**

Calificativo:

Excelente [20 - 18]

Sobresaliente [17 - 15]

Bueno [14 - 13]

Calificativo:

Excelente [20 - 18]

Sobresaliente [17 - 15]

Bueno [14 - 13]

Desaprobado

Firman en señal de conformidad:

---

Ing./Lic./Dr./Mg. Nombre y Apellidos  
Jurado  
Presidente

---

Ing./Lic./Dr./Mg. Nombre y Apellidos  
Jurado

---

Ing./Lic./Dr./Mg. Nombre y Apellidos  
Jurado

## **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 consejos, 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.

**Juan Carlos Espinoza Pereda**

## TABLA DE CONTENIDO

<b>ACTA DE AUTORIZACIÓN PARA SUSTENTACIÓN DE TESIS .....</b>	<b>1</b>
<b>ACTA DE APROBACIÓN DE LA TESIS.....</b>	<b>2</b>
<b>DEDICATORIA .....</b>	<b>3</b>
<b>AGRADECIMIENTO.....</b>	<b>4</b>
<b>INDICE DE TABLAS .....</b>	<b>9</b>
<b>INDICE DE FIGURAS .....</b>	<b>11</b>
<b>RESUMEN.....</b>	<b>15</b>
<b>CAPÍTULO I. INTRODUCCIÓN .....</b>	<b>16</b>
1.1.    Realidad Problemática .....	16
1.1.1.    Antecedentes Internacionales .....	17
1.1.2.    Antecedentes Nacionales .....	22
1.2.    Formulación del Problema .....	23
1.2.1.    Delimitación de la Investigación.....	24
1.2.2.    Justificación e Importancia.....	24
1.2.3.    Justificación económica .....	26
1.2.4.    Justificación social .....	26
1.2.5.    Justificación académica .....	26
1.3.    Objetivos.....	26
1.3.1.    Objetivo general.....	26
1.3.2.    Objetivos específicos.....	27
1.4.    Hipótesis .....	27
1.4.1.    Hipótesis general.....	27
1.4.2.    Hipótesis específicas .....	27
1.5.    Marco Teórico .....	28
1.5.1.    Navegador Web .....	28
1.5.2.    HTTP.....	29
1.5.3.    MVC (Model-View-Controller).....	31
1.5.4.    HTML5.....	34
1.5.5.    The Document Object Model (DOM) .....	35
1.5.6.    Hojas de estilo (CSS) .....	36
1.5.7.    Framework .....	37
1.5.8.    JavaScript.....	37

1.5.9.	jQuery.....	39
1.5.10.	Angular.....	40
1.5.11.	EmberJS.....	41
1.5.12.	Aurelia.....	42
1.5.13.	BackboneJS.....	42
1.5.14.	Knockout.....	42
1.5.15.	Mithril.....	43
1.5.16.	ReactJS.....	43
1.5.17.	VueJS.....	43
1.5.18.	ExtJS.....	44
1.5.19.	Serialización/deserialización.....	44
1.5.20.	XML (eXtensive Markup Language).....	44
1.5.21.	JSON (JavaScript Object Notation).....	45
<b>CAPÍTULO II. METODOLOGÍA.....</b>		<b>46</b>
2.1.	Tipo y diseño de investigación.....	46
2.1.1.	Tipo de Investigación.....	46
2.1.2.	Diseño de Investigación.....	46
2.2.	Población y muestra.....	47
2.2.1.	Población.....	47
2.2.2.	Muestra.....	47
2.3.	Variables e indicadores.....	49
2.3.1.	Variables independientes.....	49
2.3.2.	Variables dependientes.....	50
2.4.	Técnicas de recolección de datos.....	51
2.4.1.	Observación.....	51
2.4.2.	Encuesta.....	51
2.5.	Validación y Confiabilidad del Instrumento.....	52
2.6.	Procedimiento de recolección de datos.....	58
2.6.1.	Análisis estadísticos.....	60
2.6.2.	Herramienta estadística.....	60
2.6.3.	Medida de datos estadísticos.....	60
2.7.	Criterios éticos.....	61
2.8.	Criterios de rigor científico.....	62
2.9.	Cronograma de actividades.....	63
2.10.	Presupuesto.....	64
2.11.	Financiamiento.....	64
<b>CAPÍTULO III. RESULTADOS.....</b>		<b>65</b>

3.1.	Análisis e Interpretación de los datos .....	65
3.1.1.	Selección de los Framework.....	65
3.1.2.	Selección de la Herramienta.....	66
3.2.	Resultados de las mediciones .....	68
3.2.1.	Navegador Google Chrome.....	68
3.2.2.	Promedio de datos Navegador Chrome .....	68
3.2.3.	Interpretación de Resultados en navegador Chrome .....	68
3.2.4.	Navegador Mozilla Firefox .....	70
3.2.5.	Promedio de datos Navegador Firefox .....	70
3.2.6.	Interpretación de Resultados en el navegador Firefox .....	70
3.2.7.	Combinación Navegadores Chrome y Mozilla.....	72
3.2.8.	Promedio de datos Combinación Navegador Chrome y Firefox.....	72
3.2.9.	Interpretación de Resultados combinación navegadores Chrome y Mozilla .....	73
3.2.10.	Promedio del tiempo de respuesta de cada Framework .....	74
3.2.11.	Resultado tiempo de respuesta de Frameworks Angular y JavaScript Nativo .....	74
3.2.12.	Resultado encuesta Expertos Curva Aprendizaje.....	74
3.2.13.	Promedio encuesta Expertos Curva Aprendizaje .....	78
3.2.14.	Resultado encuesta Expertos Tiempo de Desarrollo .....	79
3.2.15.	Promedio encuesta Expertos Tiempo de Desarrollo .....	82
3.3.	Contrastación de la Hipótesis .....	83
<b>CAPÍTULO IV. DISCUSIÓN Y CONCLUSIONES.....</b>		<b>85</b>
4.1.	Discusión de Resultados .....	85
4.2.	Conclusiones y Recomendaciones .....	86
4.2.1.	Conclusiones .....	86
4.2.2.	Recomendaciones .....	87
<b>REFERENCIAS .....</b>		<b>89</b>
<b>ANEXOS .....</b>		<b>94</b>
Anexo 01 – Matriz de Consistencia .....		94
Anexo 02 - Solución Tecnológica .....		95
Anexo 03 – Formato Encuesta Juicio de Experto.....		98
Anexo 04 – Resultados de Medición de Tiempos de Respuesta Angular – JavaScript Nativo Google Chrome .....		99
Anexo 05 – Resultados de Medición de Tiempos de Respuesta Angular – JavaScript Nativo Mozilla .....		111
Anexo 06 – Carta de compromiso de Expertos .....		123
Anexo 07 – Tabla de Medición de Expertos.....		126
Anexo 08 – Documentación del Framework Angular.....		132
Anexo 09 – Documentación del Framework JavaScript Nativo.....		140



Anexo 10 – Certificado de validez de contenido de los instrumentos.....	146
Anexo 11 – Entrevista a experto .....	149

## INDICE DE TABLAS

<b>Tabla 1</b> Códigos de una respuesta HTTP/1.0.....	30
<b>Tabla 2</b> Operacionalización de variables independientes.....	49
<b>Tabla 3</b> Operacionalización de variable dependiente.....	50
<b>Tabla 4</b> Instrumento Ficha de Observación Framework JavaScript.....	52
<b>Tabla 5</b> Resumen Ficha Observación Framework JavaScript.....	53
<b>Tabla 6</b> Frecuencia Ficha Observación Framework JavaScript.....	53
<b>Tabla 7</b> Resultado IBM SPSS Prueba binomial – Validador 1.....	54
<b>Tabla 8</b> Resultado IBM SPSS Prueba binomial – Validador 2.....	55
<b>Tabla 9</b> Resultado IBM SPSS Prueba binomial – Validador 3.....	56
<b>Tabla 10</b> Resumen Resultado IBM SPSS Prueba binomial – Validador 1, 2, 3.....	57
<b>Tabla 11</b> Criterios de rigor de nuestra investigación.....	62
<b>Tabla 12</b> Materiales y Servicios para el desarrollo del proyecto.....	64
<b>Tabla 13</b> Selección del Framework Angular.....	65
<b>Tabla 14</b> Selección del Framework JavaScript Nativo.....	66
<b>Tabla 15</b> Selección de la Herramienta Chrome Developer Tools.....	67
<b>Tabla 16</b> Selección de la Herramienta Mozilla Tools.....	67
<b>Tabla 17</b> Resultado medición Tiempo Respuesta Angular y JavaScript Nativo - Chrome.....	68
<b>Tabla 18</b> Resultado medición Tiempo Respuesta Angular y JavaScript Nativo - Mozilla.....	70
<b>Tabla 19</b> Resultado medición Tiempo Respuesta Angular y JavaScript Nativo - Chrome Mozilla.....	72
<b>Tabla 20</b> Curva Aprendizaje –Experto 1. Fuente: elaboración propia.....	75
<b>Tabla 21</b> Curva Aprendizaje –Experto 2. Fuente: elaboración propia.....	76
<b>Tabla 22</b> Curva Aprendizaje –Experto 3. Fuente: elaboración propia.....	77
<b>Tabla 23</b> Resultados del promedio Expertos Curva Aprendizaje.....	78

<b>Tabla 24</b>	Tiempo de Desarrollo –Experto 1. Fuente: elaboración propia .....	79
<b>Tabla 25</b>	Tiempo de Aprendizaje –Experto 2. Fuente: elaboración propia.....	80
<b>Tabla 26</b>	Tiempo de Aprendizaje –Experto 3. Fuente: elaboración propia.....	81
<b>Tabla 27</b>	Resultados del promedio Expertos Tiempo de Desarrollo .....	82

## INDICE DE FIGURAS

<i>Figura 1.</i> Navegadores más usados Mayo 2017 a Mayo 2018. Fuente (www.netmarketshare.com, s.f.) .....	29
<i>Figura 2.</i> Controller = Browser. Fuente. elaboración propia.....	32
<i>Figura 3.</i> Esquema Model Vista Controlador. Fuente. elaboración propia .....	34
<i>Figura 4.</i> Documento HTML convertido a un árbol de modelo de objetos de documento. Fuente (Svensson, 2015).....	35
<i>Figura 5.</i> Ejemplo de regla en una hoja de estilo CSS. Fuente. (Lozada, 2015) .....	36
<i>Figura 6.</i> Principales objetos accesibles desde el contexto JavaScript. Fuente (Lozada, 2015).....	39
<i>Figura 7.</i> Ejemplo Angular Utilizando CDN Google. Fuente (Peña & Cambisaca, 2015) .....	41
<i>Figura 11.</i> Proceso de recolección de datos. Fuente: elaboración propia .....	59
<i>Figura 12.</i> Gráfico porcentaje promedio de tiempo respuesta – google chrome. Fuente. Elaboración propia .....	69
<i>Figura 13.</i> Gráfico porcentaje promedio de tiempo respuesta – mozilla firefox. Fuente. Elaboración propia .....	71
<i>Figura 14.</i> Gráfico porcentaje promedio de tiempo respuesta – Chrome y firefox. Fuente. Elaboración propia .....	74
<i>Figura 15</i> Grafico Puntaje Promedio Curva Aprendizaje. Fuente. Elaboración propia .....	78
<i>Figura 16</i> Grafico Puntaje Promedio Tiempo de Desarrollo. Fuente. Elaboración propia .....	82
<i>Figura 17</i> Diseño de Mockup pantalla acceso. Fuente: información propia .....	95
<i>Figura 18</i> Diseño de Mockup pantalla mantenimiento. Fuente: información propia .....	95
<i>Figura 19</i> Diseño de Mockup pantalla mantenimiento. Fuente: información propia .....	96
<i>Figura 20.</i> Tiempo respuesta Agregar Cliente – Framework Angular – Google Chrome. Fuente elaboración propia.....	99

<i>Figura 21.</i> Tiempo respuesta Editar Cliente – Framework Angular – Google Chrome. Fuente elaboración propia.....	100
<i>Figura 22.</i> Tiempo respuesta Eliminar Cliente – Framework Angular – Google Chrome. Fuente elaboración propia.....	101
<i>Figura 23.</i> Tiempo respuesta Listar 500 – Framework Angular – Google Chrome. Fuente elaboración propia.....	102
<i>Figura 24.</i> Tiempo respuesta Listar 1000 – Framework Angular – Google Chrome. Fuente elaboración propia.....	103
<i>Figura 25.</i> Tiempo respuesta Listar 2000 – Framework Angular – Google Chrome. Fuente elaboración propia.....	104
<i>Figura 26.</i> Tiempo respuesta Agregar Cliente – Framework JavaScript Nativo – Google Chrome. Fuente elaboración propia.....	105
<i>Figura 27.</i> Tiempo respuesta Editar Cliente – Framework JavaScript Nativo – Google Chrome. Fuente elaboración propia.....	106
<i>Figura 28.</i> Tiempo respuesta Eliminar Cliente – Framework JavaScript Nativo – Google Chrome. Fuente elaboración propia.....	107
<i>Figura 29.</i> Tiempo respuesta Listar 500 – Framework JavaScript Nativo – Google Chrome. Fuente elaboración propia.....	108
<i>Figura 30.</i> Tiempo respuesta Listar 1000 – Framework JavaScript Nativo – Google Chrome. Fuente elaboración propia.....	109
<i>Figura 31.</i> Tiempo respuesta Listar 2000 – Framework JavaScript Nativo – Google Chrome. Fuente elaboración propia.....	110
<i>Figura 32.</i> Tiempo respuesta Agregar Cliente – Framework Angular – Mozilla. Fuente elaboración propia.....	111

<i>Figura 33.</i> Tiempo respuesta Editar Cliente – Framework Angular – Mozilla. Fuente elaboración propia .....	112
<i>Figura 34.</i> Tiempo respuesta Eliminar Cliente – Framework Angular – Mozilla. Fuente elaboración propia.....	113
<i>Figura 35.</i> Tiempo respuesta Listar 500 – Framework Angular – Mozilla. Fuente elaboración propia .....	114
<i>Figura 36.</i> Tiempo respuesta Listar 1000 – Framework Angular – Mozilla. Fuente elaboración propia .....	115
<i>Figura 37.</i> Tiempo respuesta Listar 2000 – Framework Angular – Mozilla. Fuente elaboración propia .....	116
<i>Figura 38.</i> Tiempo respuesta Agregar Cliente – Framework JavaScript Nativo – Mozilla. Fuente elaboración propia.....	117
<i>Figura 39.</i> Tiempo respuesta Editar Cliente – Framework JavaScript Nativo – Mozilla. Fuente elaboración propia.....	118
<i>Figura 40.</i> Tiempo respuesta Eliminar Cliente – Framework JavaScript Nativo – Mozilla. Fuente elaboración propia.....	119
<i>Figura 41.</i> Tiempo respuesta Listar 500 – Framework JavaScript Nativo – Mozilla. Fuente elaboración propia.....	120
<i>Figura 42.</i> Tiempo respuesta Listar 1000 – Framework JavaScript Nativo – Mozilla. Fuente elaboración propia.....	121
<i>Figura 43.</i> Tiempo respuesta Listar 2000 – Framework JavaScript Nativo – Mozilla. Fuente elaboración propia.....	122
<i>Figura 44.</i> Angular-module .....	132
<i>Figura 45.</i> Angular configuración .....	132
<i>Figura 46.</i> Service consulta .....	133

<i>Figura 47.</i> Service mantenimiento.....	133
<i>Figura 48.</i> Controller consulta 1 .....	134
<i>Figura 49.</i> Controller consulta 2 .....	134
<i>Figura 50.</i> Controller mantenimiento 1 .....	135
<i>Figura 51.</i> Controller mantenimiento 2 .....	135
<i>Figura 52.</i> Consulta 1 .....	136
<i>Figura 53.</i> Consulta 2 .....	136
<i>Figura 54.</i> Cliente 1 .....	137
<i>Figura 55.</i> Cliente 2 .....	137
<i>Figura 56.</i> Mvc controller 1 .....	138
<i>Figura 57.</i> Mvc controller 2.....	138
<i>Figura 58.</i> Vista add cliente.....	139
<i>Figura 59.</i> TablasGeneralesController.cs - 1 .....	140
<i>Figura 60.</i> TablasGeneralesController.cs - 2.....	140
<i>Figura 61.</i> TablasGeneralesController.cs - 3.....	141
<i>Figura 62.</i> Cliente.cshtml.....	141
<i>Figura 63.</i> Consulta.cshtml - 1.....	142
<i>Figura 64.</i> Consulta.cshtml - 2.....	142
<i>Figura 65.</i> Consulta.cshtml - 3.....	143
<i>Figura 66.</i> Consulta.js.....	143
<i>Figura 67.</i> Cliente.js - 1 .....	144
<i>Figura 68.</i> Cliente.js - 2 .....	144
<i>Figura 69.</i> Cliente.js - 3 .....	145
<i>Figura 70.</i> Cliente.js - 4 .....	145

## RESUMEN

En la actualidad el desarrollo y crecimiento tecnológico va de la mano con el uso de aplicaciones web ya que éstas utilizan frameworks robustos que aplican mejores prácticas para su construcción. Dichos frameworks existen tanto del lado del cliente como del lado de servidor. Muchas empresas desarrollan aplicaciones web con el patrón web MVC y un framework del lado del cliente basado en javascript. Estudios realizados a estos framework validan algunas ventajas notables de los mismos respecto a la robustez, escalabilidad, mantenibilidad, seguridad, etc. En éste trabajo de investigación se realizará un estudio comparativo de los frameworks Angular y Javascript Nativo para analizar la incidencia del tiempo de respuesta de cada framework. Para realizar dicha comparación se desarrollará un módulo para agregar, listar, eliminar y consultar datos utilizando para cada framework y poder tomar la información del tiempo de respuesta. Se consultará así mismo a expertos que nos brinden información adicional sobre la curva de aprendizaje y el tiempo de desarrollo. Los tiempos de respuesta de cada frameworks serán evaluados y medidos en los navegadores chrome y mozilla. La investigación tendrá enfoque cuantitativo y tipo descriptivo. Nos apoyaremos de la observación y la encuesta. Los resultados obtenidos serán promediados, graficados y nos permitirán validar que framework es más eficiente respecto al tiempo de respuesta.

**Palabras clave:** framework, angular, javascript, mvc, web



## CAPÍTULO I. INTRODUCCIÓN

### 1.1. Realidad Problemática

A lo largo de los años 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 como la utilización de algún Framework. En el mundo diferente programadores emplean para el desarrollo de aplicaciones web el lenguaje JavaScript que tiene mayor popularidad y a su vez se han creado diferentes frameworks (Cano, 2018). Un Framework se puede definir como un conjunto de aplicaciones genéricas no concretas, adaptables y configurables que nos facilitan la construcción de aplicaciones específicas; los Framework no solo sirven para construir aplicaciones web, sino nos aceleran el proceso de desarrollo optimizando el código y reutilizándolo y aplicando las mejores prácticas y técnicas (Sanchez, Tuesta, & Mejía, 2015).

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.

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.

### **1.1.1. Antecedentes Internacionales**

Según (Rico & Rondon, 2013) mencionan que la mayoría de los frameworks están basados en arquitectura y patrón MVC, son de código libre y permiten construir proyectos web y móviles escalables; además ofrecen la posibilidad de construir aplicaciones desde el Back-End: programación del lado de servidor o desde el Front-End: programación del lado del cliente.

Según (Libardo, 2015) afirma que la elección de un framework es determinante para el éxito y la reducción en el tiempo de desarrollo, éstos frameworks usan

arquitecturas y patrones que facilitan el mantenimiento del código, permiten crear aplicaciones robustas y evitan crear funcionalidades en capas incorrectas.

Según (Ladan, 2015) menciona en un estudio realizado entre JavaScript, jQuery y Backbone.js referente al tiempo de ejecución al realizar transacciones y al tamaño en Kilobytes al cargar la página; que los resultados fueron favorables para JavaScript.

Según (Peña & Cambisaca, 2015) mencionan en su estudio que 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. Realizaron estudios entre Angular y EmberJS; los criterios fueron porcentaje de consumo o utilización de recursos del CPU. Se emplearon métodos GET Y POST para las diferentes pruebas. A menor consumo o utilización de recursos, se obtiene un mayor rendimiento y viceversa; por lo que se infirió que Angular presenta mejor rendimiento.

Según (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 suministrar sistemas que incluyen un diseño de patrón óptimo de uso, conocido, apropiado, redundancia mínima, escalabilidad, fácil de mantenimiento, reducir costos y previsibilidad del desarrollo.

Según (Hans, 2015) realizó unos 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 simultáneo 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.

Según (Gómez, 2015) menciona en su estudio 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.

Según (Svensson, 2015) menciona que 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. Realizó un estudio de los Frameworks JavaScript 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. Los resultados fueron favorables para el Framework JavaScript Angular 2.0.

Según (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 ésta 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.

Según (Ochoa, 2016) menciona en un estudio realizado sobre tipos de serialización como son XML, Binaria, SOAP y JSON. La más ligera y rápida es JSON.

Según (Mora, 2016) menciona que JSON puede serializar/deserializar 100 veces más rápido que XML en navegadores modernos pero requiere un poco más de recursos de CPU. Se analizó también con YAML y JSON resulto mucho más rápido en serializar/deserializar. Es ideal para envío de datos mediante servicios web y el protocolo REST.

Según (Márquez, 2016) en su estudio 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. Se revisó las estadísticas de cada JavaScript Framework en GitHub. La mayor nota obtenida fue para Angular.

Según (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%.

Según (Craig, y otros, 2017) mencionan en un estudio de diversos frameworks para analizar cuál es el apropiado en el desarrollo de una aplicación móvil. Se descartaron JQuery, Ember, Meteor y Angular. Se decidió por ReactJS por ser de sólo lectura y permitir interacción fluida con otros frameworks y bibliotecas JavaScript.

Según (Moreno, 2017) destaca que existen diversos frameworks desarrollados en diferentes lenguajes como Java, JavaScript, entre otros; implementar algunos de éstos en algún proyecto debe cumplir criterios como mantener una codificación limpia, estructura óptima, facilidad de implementación y minimizar el tiempo de desarrollo. Realizó un estudio para verificar las características de calidad entre Angular2 y ReactJS; las características fueron: funcionalidad, fiabilidad,

usabilidad, eficiencia, mantenibilidad y portabilidad. Se demostró que el framework con mejor proyección es Angular 2, teniendo una ventaja mayor en la usabilidad, mantenibilidad, portabilidad y funcionalidad. En fiabilidad y eficiencia tiene ventaja ReactJS.

Según (Santos, 2018) evidencia en su trabajo de investigación que desarrollar en el patrón MVC aumenta la productividad de los programadores y requiere menos tiempo de codificación que realizar la misma tarea en webforms. En su estudio nos menciona que realizó un experimento entre ASP.NET MVC y webform; se desarrolló un prototipo de registro con 06 desarrolladores con amplia experiencia 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 webform en cantidad de líneas de código.

Según (Kaluža, Troskot, & Vukelić, 2018) mencionan en su investigación desde los enfoques SPA (Single Page Application: aplicaciones web simples con poco contenido) y MPA (Multi Page Application: aplicaciones con mayor cantidad de tipos de servicios e interacciones con visitantes). Los frameworks analizados fueron VueJS, ReactJS y Angular. Angular presentó mayor puntaje en SPA y VueJS en MPA.

### **1.1.2. Antecedentes Nacionales**

Según (Zurita & Zavala, 2016) en una investigación realizaron un análisis comparativo de Frameworks tanto del lado del cliente como del lado del servidor.

Se emplearon criterios como la 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 seleccionados fueron Angular y BackboneJS. Los resultados dieron mayor puntaje promedio a framework Angular.

Según (Bendezú & Figueroa, 2017) menciona en su tesis que el lenguaje interprete JavaScript podría minificarse para optimizar el tiempo de descarga de los recursos de una web así como también se podría abreviar las nombres de algunas variables y reducir el tamaño del archivo para mejorar el rendimiento de las aplicaciones web.

## **1.2. Formulación del Problema**

En las empresas del sector comercial actualmente existen algunos problemas de comunicación con las aplicaciones web, esto debido a la gran cantidad de accesos a la aplicación web, dichas aplicaciones sirven para realizar ventas de publicidad y/o productos que realizan los vendedores a nivel nacional e internacional. Nuestro problema de investigación sería:

**¿Cuál es la incidencia en el tiempo de respuesta del Framework JavaScript en un aplicativo web MVC para las empresas del sector comercial?**



### **1.2.1. Delimitación de la Investigación**

Dado que la mejora del tiempo de respuesta en las aplicaciones web es un tema relevante para poder ser competitivas en la actualidad, nuestra investigación consistirá en el análisis del tiempo de respuesta de los frameworks Angular y JavaScript Nativo en un aplicativo web MVC para empresas del sector comercial. Se seleccionó el Framework Angular ya que fue el que mejor resultados obtuvo frente a otros en base a estudios previos de revisión de literatura en un periodo del 2015 al 2018. Así mismo se optó por seleccionar JavaScript Nativo porque en un estudio realizado se obtuvo ventajas frente a otros frameworks.

### **1.2.2. Justificación e Importancia**

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 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 el 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 priorizando 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.

La investigación se basa en el uso de Frameworks JavaScript con gran aceptación en la actualidad y que fueron seleccionados mediante una investigación previa y que aportarían mejor rendimiento en el tiempo de respuesta de las aplicaciones web a desarrollar.

### **1.2.3. Justificación económica**

Si se realiza el análisis y descripción se logra identificar cual Framework del lado del cliente es el más óptimo para reducir el tiempo de respuesta de la aplicación web; esto generará que la web sea más productiva y eficiente, en consecuencia, las ventas aumentarían en un rango de 10-15%.

### **1.2.4. Justificación social**

En la actualidad la gran mayoría de empresas que cuentan con aplicaciones web presentan problemas de lentitud y tiempos de respuesta con demora exagerada. Esta investigación servirá para poder dejar un precedente y un marco de trabajo para el desarrollo de aplicaciones web utilizando Frameworks del lado del cliente y buenas prácticas que mejoran los tiempos de respuesta de los aplicativos webs.

### **1.2.5. Justificación académica**

Se pondrá en práctica nuestros conocimientos adquiridos en nuestra formación y poder aplicarlos en un trabajo de investigación acorde a nuestra carrera profesional.

## **1.3. Objetivos**

### **1.3.1. Objetivo general**

Analizar los Frameworks Angular y JavaScript Nativo en la incidencia del tiempo de respuesta en un aplicativo web MVC para empresas del sector comercial.

### 1.3.2. Objetivos específicos

- Evaluar el framework Angular y JavaScript Nativo referente al tiempo de desarrollo.
- Evaluar los frameworks Angular y JavaScript Nativo referente a la curva de aprendizaje.
- Analizar el tiempo de respuesta del framework Angular en un aplicativo web MVC.
- Analizar el tiempo de respuesta del framework JavaScript Nativo en un aplicativo web MVC.
- Implementar un mantenimiento y consulta web MVC utilizando los frameworks Angular y JavaScript Nativo para poder observar la incidencia en el tiempo de respuesta.
- Realizar los test para medir el tiempo de respuesta de los frameworks Angular y JavaScript Nativo en un aplicativo web MVC.

## 1.4. Hipótesis

### 1.4.1. Hipótesis general

Mediante el análisis de los frameworks Angular y JavaScript Nativo se podrá obtener los resultados de la incidencia en el tiempo de respuesta de un aplicativo web MVC para las empresas del sector comercial.

### 1.4.2. Hipótesis específicas

- Mediante el análisis de los frameworks Angular y JavaScript Nativo obtendremos resultados sobre el tiempo de desarrollo.

- Mediante el análisis de los frameworks Angular y JavaScript Nativo obtendremos resultados sobre la curva de aprendizaje.
- Mediante el análisis del framework Angular se podrá describir los resultados de la incidencia en el tiempo de respuesta de un aplicativo web MVC.
- Mediante el análisis del framework JavaScript Nativo se podrá describir los resultados de la incidencia en el tiempo de respuesta de un aplicativo web MVC.
- Utilizando los frameworks Angular y JavaScript Nativo en un aplicativo web MVC podremos observar la incidencia en el tiempo de respuesta.
- Mediante los test aplicados al aplicativo web MVC podremos medir los tiempos de respuesta de los frameworks Angular y JavaScript Nativo.

## 1.5. Marco Teórico

### 1.5.1. Navegador Web

Software que utiliza conexión a internet para hacer solicitudes de páginas a un servidor y nos presenta la información en cualquier dispositivo (PC, tablets, Smartphone, etc.) Para la comunicación utilizan un protocolo HTTP. El primer navegador fue Mosaic, posteriormente aparecieron Netscape Navigator, Mozilla, Internet Explorer, Chrome, Opera, Safari, etc. (Sánchez, 2014). La Figura 1 muestra los navegadores más usados. Son las aplicaciones que se ejecutan en el entorno del cliente y que permiten visualizar los documentos escritos en lenguaje HTML y capturar las interacciones del usuario. (Vara, López, & Verde, 2014)

## Browser Share by Version

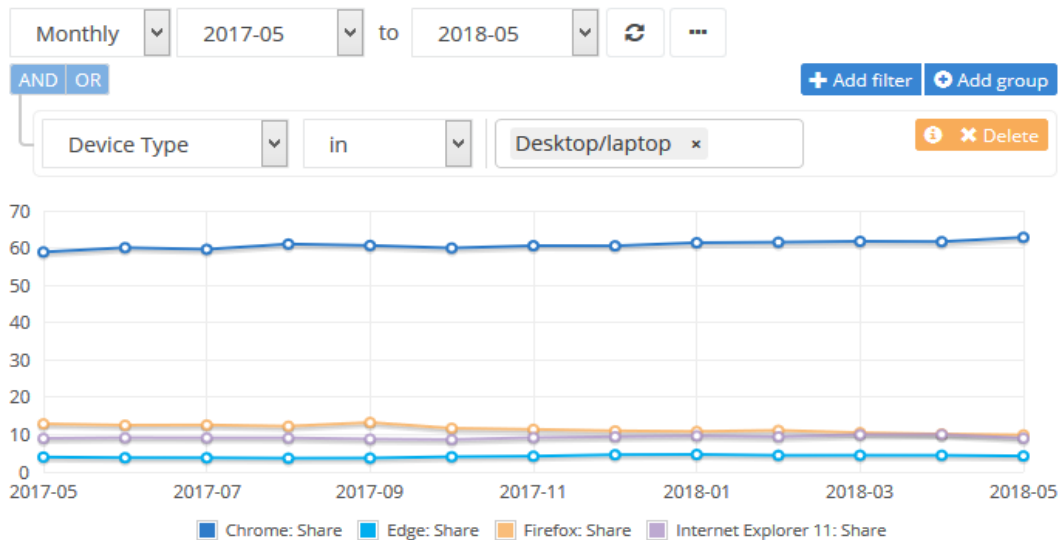


Figura 1. Navegadores más usados Mayo 2017 a Mayo 2018. Fuente (www.netmarketshare.com, s.f.)

### 1.5.2.HTTP

HTTP sigue el modelo general de peticiones y respuestas entre un cliente y un servidor. Se basa en un servicio de transporte fiable; pero, es independiente del mecanismo de transporte concreto utilizado. No obstante, lo más habitual es que cliente y servidor se comuniquen por medio del TCP. En este caso, el puerto por defecto para establecer las conexiones es el asignado oficialmente al servicio WWW, es decir, el 80. En el HTTP/1.0, el cliente establece una conexión con el servidor y le envía un mensaje HTTP con la petición; y, a continuación, el servidor envía al cliente otro mensaje HTTP con la respuesta y cierra la conexión. Si quiere efectuar más peticiones, el cliente debe establecer una nueva conexión para cada una. En el HTTP/1.1, en cambio, es posible intercambiar diferentes peticiones y respuestas en una misma conexión que se denomina conexión persistente. Éste es

el modo de funcionamiento por defecto en el HTTP/1.1 (Barceló, Griera, & Viejo, 2008, págs. 161-163). La Tabla 1 muestra algunos códigos de respuesta HTTP/1.0

**Tabla 1**

*Códigos de una respuesta HTTP/1.0.*

Código	Significado
200	Operación efectuada.
201	Se ha creado un nuevo recurso.
202	Petición aceptada.
204	La respuesta está vacía.
301	El recurso solicitado ha cambiado de dirección.
302	El recurso solicitado ha cambiado temporalmente de dirección.
304	El contenido del recurso no ha cambiado.
400	Petición incorrecta.
401	Usuario no autorizado.
403	Acceso prohibido.
404	No se ha encontrado el recurso.
500	Error interno del servidor.
501	Operación no implementada.
502	Error en otro servidor.
503	Servicio no disponible en la actualidad.

**Nota.** Fuente (Barceló, Griera, & Viejo, 2008)

Los autores Juan Manuel Vara Mesa, Marcos López Sanz y Jénifer Verde Marín describen distintos modos o métodos para intercambiar información entre cliente y servidor.

**Método GET:** es un método de invocación en el que el cliente le solicita al servidor web que le devuelva la información identificada en la propia URL. Lo más común es que las peticiones se refieran a un documento HTML o a una imagen, aunque también se puede referir a un programa de base de datos. En tal caso, el servidor ejecuta ese programa y le devuelve al cliente el resultado generado tras esa petición.

**Método POST:** mientras que el método GET se utiliza para recuperar información, el método POST se usa habitualmente para enviar información a un servidor web. Estos casos suelen darse al enviar el contenido de un formulario de autenticación, así como entradas de datos o especificar parámetros para algún tipo de componente ejecutado en el servidor (Vara, López, & Verde, 2014, pág. 20).

### 1.5.3. MVC (Model-View-Controller)

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.

**Model = HTML**

```
<div id="logo1" class="titulocss" style="height:125px;width:142px;">
  <h1>
    <a href="http://www. web.pe/" title=" Eco">Facultad </a>
  </h1>
</div>
```

Model: el HTML es el "esqueleto" del contenido y el texto que comunica información al lector.

**View = CSS**



```
<style type="text/css">
  #logo1 {
    align:center;
    background: rgba(255, 255, 255, 0)
  }
  .titulocss {
    background:url ("img/logo.png")}
</style>
```

View: el CSS agrega estilo visual al contenido. Es la "piel" que utilizamos para dar cuerpo a nuestro esqueleto; podemos intercambiar diferentes máscaras a través de CSS sin alterar el contenido original. Son relativamente, pero no completamente, independientes.

**Controller = Browser.** La Figura 2 muestra esta relación



Figura 2. Controller = Browser. Fuente. elaboración propia

**Controller:** el browser es responsable de combinar y renderizar CSS y HTML.

Recopila la información del usuario y la dirige a cualquier código JavaScript necesario para el funcionamiento de la página (Enache, 2015).

Daniel Pagés Chacón y Julio Pablo Martínez Prieto en su tesis mencionan algunos beneficios:

- Se hace más eficiente la construcción de aplicaciones que requieren diferentes representaciones de una misma información dado que permite simplemente construir vistas distintas que interactúen con el mismo controlador y la misma estructura o modelo de datos.
- De la misma forma pueden ser representadas diferentes interfaces condicionadas por distintos diseños o look and feel, a su vez establecidos por la necesidad de desarrollo para diferentes Sistemas Operativos (SO), o sea multiplataforma.
- Se hace palpable la reusabilidad de componentes de interfaz ya que estos no se encuentran integrados con ninguna información de lógica de negocio o de acceso a datos.
- El desacoplamiento con el modelo y el controlador permite que los cambios realizados en estas capas no afecten a las restantes, lo cual permite mayor eficiencia en el desarrollo y mantenimiento, dado que ya no se emplea tiempo ni esfuerzo en realizar cambios en toda la aplicación por el hecho de que sus partes se encuentren con un alto grado de acoplamiento. (Pagés & Martínez, 2010, págs. 8-9).

(Ordax & Ocaña, 2012) Definen modelo, vista y controlador:

**El Modelo (Model):** Representa los datos y cualquier lógica de negocio relacionada con ellos.

**La Vista (View):** renderiza el contenido de los modelos dependiendo de la tipología de cliente (navegador web, teléfono móvil, etc.), permitiendo su visualización.

**El Controlador (Controller):** define el comportamiento general de la aplicación coordinando a las otras dos partes (Modelo y Vista) (p. 9). La Figura 3 muestra el esquema MVC.

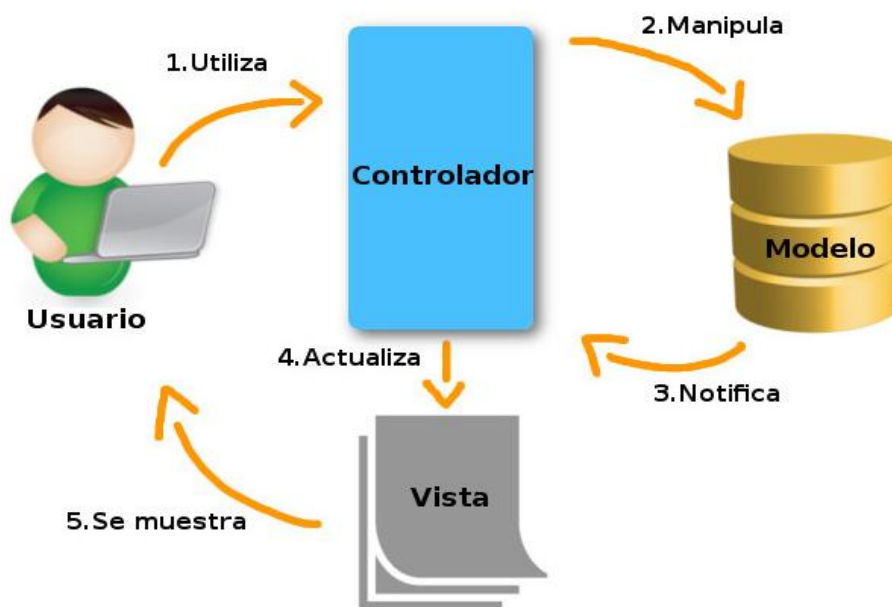


Figura 3. Esquema Model Vista Controlador. Fuente. elaboración propia

#### 1.5.4. HTML5

HyperText Mark-up Language 5 es la versión más reciente del lenguaje y empezó a ser desarrollado por el Web Hypertext Application Technology Working 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)

### 1.5.5. 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).

La Figura 4 muestra éste esquema.

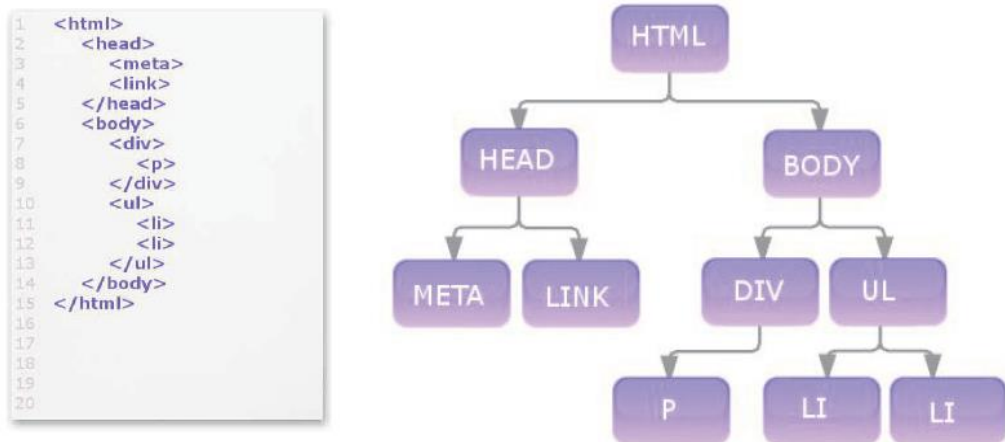


Figura 4. Documento HTML convertido a un árbol de modelo de objetos de documento. Fuente

(Svensson, 2015)

### 1.5.6. Hojas de estilo (CSS)

CSS (Cascading Style Sheet en inglés) Describe el formato y apariencia de los elementos de un documento HTML (tamaño de la fuente, el color o el espaciado entre elementos). Separa el contenido y su presentación. Contiene reglas y cada regla contiene un selector y un bloque de declaración. La Figura 6 nos muestra un ejemplo.

**Selector:** permite identificar al elemento o elementos sobre los cuales se aplicará la regla. Se identifica por el nombre de la etiqueta HTML, el atributo id (#), el atributo class.

**Declaración:** contiene una o más propiedades separadas por un punto y coma.

**Propiedad:** colores, bordes, fondo de pantalla, textos, animaciones, propiedades de las listas, efectos visuales, paginación, etc.

**Valor:** se expresan como cadenas de caracteres, valores numéricos. (Lozada, 2015).

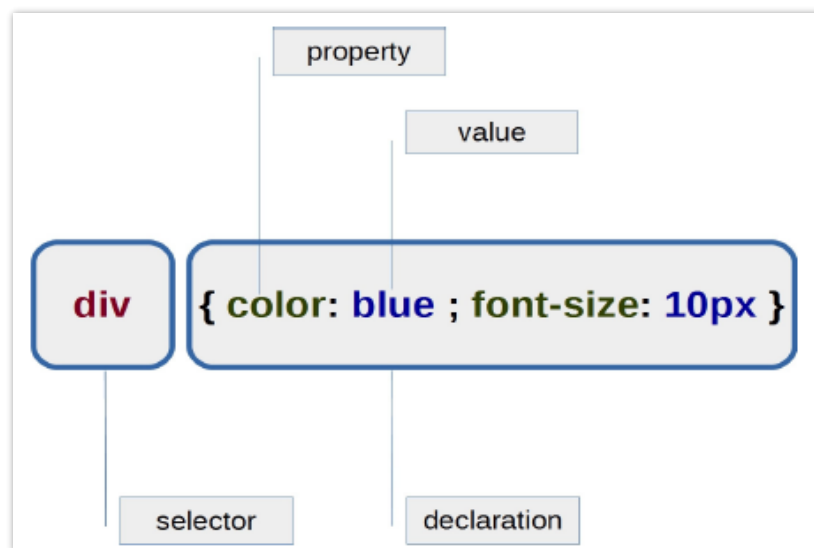


Figura 5. Ejemplo de regla en una hoja de estilo CSS. Fuente. (Lozada, 2015)

### **1.5.7. Framework**

La autora Sandra Cabello Jurado lo define como un conjunto de librerías desarrolladas en un lenguaje determinado que se utilizar para el desarrollo de páginas web con contenidos previamente programados y reutilizables para una web, como por ejemplo una galería de imágenes (Cabello, 2014, pág. 115).

### **1.5.8. JavaScript**

El autor Purificación Ribes Alba menciona que JavaScript es un lenguaje de los denominados lenguajes de scripting. Los scripts (script se traduce como guión, literalmente) son archivos de órdenes, programas por lo general simples. Es por esto que no podemos definir JavaScript como un lenguaje de programación en un sentido estricto, pero sin embargo sí nos permite crear páginas dinámicas, con algunos efectos realmente interesantes y que mejoren considerablemente su aspecto. Nos permite tener cierta interacción con el usuario de nuestras páginas, reconocer determinados eventos que se puedan producir y responder a éstos adecuadamente. Incluso podemos crear algunos programas más complejos que manejen estructuras de datos. Tratándose de un lenguaje de script, los programas que realicemos, no necesitarán ser compilado. Los lenguajes de scripting son lenguajes interpretados. Esto en la práctica significa que cuando trabajemos con JavaScript escribiremos nuestro programa y podremos ejecutarlo de forma directa, sin necesidad de hacer nada más (Ribes, 2011, pág. 15).

Basado en el estándar ECMAScript [ECMASC]. Permite acceder y manipular el árbol DOM de los documentos HTML, para generar contenidos de manera

dinámica, realizar peticiones HTTP asíncronas, etc. Soporta estructuras de programación similares a lenguajes como C o Java, con la principal diferencia de que el ámbito en el que se define una variable no es el bloque en el que ha sido definida, sino la función en la que ha sido declarada. Un documento HTML, puede incluir código ejecutable en JavaScript utilizando nodos con etiqueta script. Estos nodos pueden contener el código ejecutable de forma directa, o pueden hacer referencia a un fichero externo con el código. Los navegadores web tradicionales, incluyen dentro del contexto de ejecución de JavaScript, una serie de elementos predefinidos:

1. Tipos de datos y objetos predefinidos de uso general.
2. Objetos que permiten manipular el árbol DOM del documento HTML (por ejemplo: el objeto window o el objeto document).
3. Objetos que permiten realizar otro tipo de operaciones de más alto nivel (por ejemplo, peticiones AJAX).

Los principales objetos accesibles desde el contexto de ejecución de JavaScript en los navegadores convencionales son:

1. El objeto window: representa la ventana actual y además, también es el contexto de ejecución de más alto nivel.
2. El objeto document: representa al documento que está cargado en una ventana. Desde este objeto se puede acceder a todos los elementos del árbol DOM.

Tanto el objeto document, como los otros objetos del contexto JavaScript, proporcionan una serie de funciones y métodos que permiten manipular el árbol

DOM de la página web. Por ejemplo, el objeto document permite, entre otras muchas funcionalidades, las siguientes operaciones:

1. Acceder al nodo raíz del árbol utilizando la llamada document.documentElement.
2. Obtener los nodos de un tipo determinado utilizando la función getElementByTagName.
3. Obtener nodos por su identificador utilizando la función getElementById.

(Lozada, 2015, págs. 19-21). La Figura 6 esquematiza estos objetos.

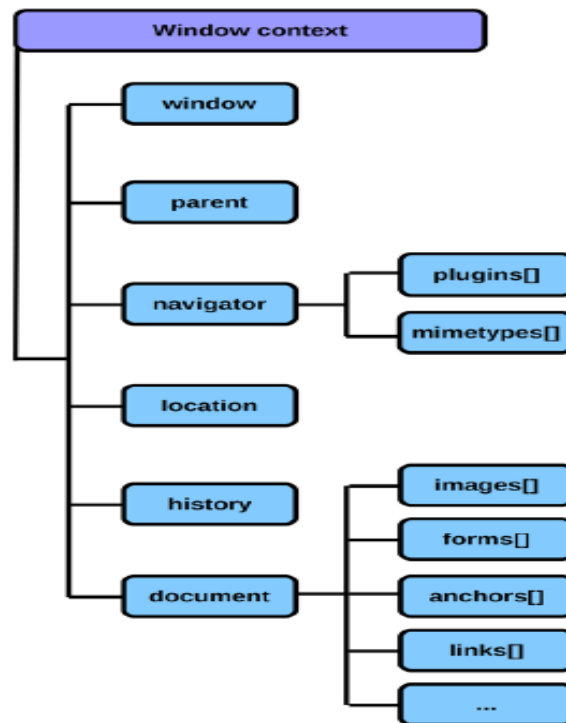


Figura 6. Principales objetos accesibles desde el contexto JavaScript. Fuente (Lozada, 2015)

### 1.5.9. jQuery

El autor Fernando Rodríguez Diéguez lo define como una de las librerías más potentes y nos permite realizar multitud de tareas de forma simple, desde efectos gráficos a comunicación AJAX con el servidor, pasando por validación de campos de formularios, etc. (Rodríguez, 2014, pág. 77).



Juan Luis Perles García lo define como una biblioteca libre y de código abierto que ofrece una serie de funcionalidades basadas en JavaScript para hacer más sencillo la interacción con los documentos HTML, objetos DOM, manejo de eventos e integración de la tecnología AJAX en la página web. (Perles, 2014, pág. 337).

### 1.5.10. Angular

Creado para aplicaciones web dinámicas, Utiliza HTML para representar la información y permite reducir líneas de código mediante enlace de datos, inyecciones de dependencia. Es compatible con Java, Python, Ruby, C# o cualquier otro lenguaje. La sincronización vista - modelo es automática y en tiempo real, usa peticiones HTTP XML o JSON (Peña & Cambisaca, 2015). La Figura 8 muestra un ejemplo.

(Ingavélez, Hilera, & Timbi, 2016, págs. 448-449) Definen a Angular como un lenguaje estructurado para aplicaciones web dinámicas JavaScript, orientado a operaciones CRUD (Create, Read, Update y Delete) mediante técnica AJAX. Es un framework de código abierto (Open-Source) mantenido por Google. La idea principal de Angular es la de sincronizar el documento HTML con las necesidades de una aplicación web dinámica mediante la creación de bloques HTML a través de las “directivas”.

#### **Características:**

La vinculación de los datos del modelo con las vistas está en constante sincronía (data binding). Si un elemento de la vista cambia, Angular se dará cuenta del

cambio y lo sincronizará con el modelo vinculado. Sucediendo exactamente lo mismo si lo que cambia es el modelo.

Desacopla el lado del cliente de una aplicación del lado del servidor favoreciendo el progreso en paralelo y la reutilización de ambos lados.

Desacopla la manipulación del árbol DOM de la lógica de la aplicación y del modelo.

Propone una alternativa para la manipulación del árbol DOM que consiste en el uso de las directivas. Angular proporciona una API de directivas para repetir, mostrar, ocultar elementos del árbol DOM.

Aporta unas pautas de creación personalizada de directivas. Lo que proporciona la posibilidad de crear estructuras HTML personalizadas y reutilizables.

Permite el uso de expresiones, filtros y conversión de datos en las propias vistas (documentos HTML) dentro de llaves dobles “{{ }}”.

```
<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 7. Ejemplo Angular Utilizando CDN Google. Fuente (Peña & Cambisaca, 2015)

### 1.5.11. 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).

#### **1.5.12. 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).

#### **1.5.13. 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 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).

#### **1.5.14. 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).

#### **1.5.15. 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 Angular. La principal diferencia es como manejan sus plantillas (Svensson, 2015, pág. 16).

#### **1.5.16. ReactJS**

Creada por Facebook, facilita la construcción de interfaces de usuarios de manera eficiente y flexible del lado del cliente; utiliza JSX del HTML y se puede complementar con otras librerías para desarrollo de aplicaciones complejas (Moreno, 2017).

#### **1.5.17. 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).

### 1.5.18. 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).

### 1.5.19. Serialización/deserialización

Se define como el proceso de convertir un objeto en un formato legible, que se pueda transportar entre aplicaciones. La deserialización convierte el formato legible 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).

### 1.5.20. XML (eXtensive Markup Language)

Lenguaje de marcado extendido se define como un formato extensible y codificación de texto mediante etiquetas y lenguajes de marcas para la transmitir datos complejos y estructurados por internet (García & Arroyo, 2007).

```
<agenda>
  <contacto>
    <nombre>José Sandro</nombre>
    <celular>956859325</celular>
  </contacto>
  <contacto>
    <nombre>María Aponte</nombre>
```

```
<celular>972536168</celular>  
</contacto>  
</agenda>
```

### 1.5.21. 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 }  
                    ]  
}
```

## CAPÍTULO II. METODOLOGÍA

### 2.1. Tipo y diseño de investigación

#### 2.1.1. Tipo de Investigación

La Investigación descriptiva representa y describe con palabras precisas e inequívocas características y la realidad de un objeto de estudio detallando aspectos, hechos, situaciones, cosas, partes, clases, categorías, personas y/o relaciones con otros objetos (Niño, 2011). Según la referencia bibliográfica nuestra investigación será de tipo descriptivo ya que se mostrará datos relevantes de los frameworks Angular y JavaScript Nativo investigados. Así mismo tendrá un enfoque cuantitativo ya que la información obtenida de los frameworks Angular y JavaScript Nativo será procesada utilizando métodos estadísticos en función de las variables de estudio y así poder validar nuestra hipótesis ya que según (Guerrero & Guerrero, 2014) menciona que la hipótesis en una investigación cuantitativa será aceptada o rechazada mediante la observación y experimentación sobre las variables investigadas.

#### 2.1.2. Diseño de Investigación

**Documental:** Se analiza información de investigaciones existentes respecto a nuestro objeto de estudio (Moreno W. , 2013) en nuestro caso los frameworks JavaScript para poder deducir cual son los mejores con relación a tiempo de respuesta, curva de aprendizaje y tiempo de desarrollo.

**No-experimental transeccional:** Hace referencia a la indagación en el que se recogen datos, las variables se observan en su contexto sin manipulación de las

variables independientes en un momento determinado (transeccional) (Sáez, 2017) que consistirá en medir los tiempos de respuesta de los frameworks Angular y JavaScript Nativo en un instante determinado en cada uno de los módulos desarrollados, dicha información será tabulada y comparada para su posterior análisis e interpretación.

## **2.2. Población y muestra**

Nuestra unidad de análisis serán los frameworks JavaScript que inciden en el tiempo de respuesta de las aplicaciones web.

### **2.2.1. Población**

La población de estudio para el presente trabajo de investigación constará de los Frameworks JavaScript que existen en la actualidad y que como referencia se estudiaron en diferentes artículos científicos en los últimos cuatro años (revistas científicas, papers, tesis, revisiones de literatura, etc). Los Framework JavaScript más importantes son:

Knockout, JavaScript, VueJS, Backbone, Angular, EmberJS, Aurelia, Mithril, ReactJS, ExtJS.

### **2.2.2. Muestra**

Para éste trabajo de investigación la muestra es no probabilística y estará conformada por 02 de los Frameworks JavaScript los cuales son Angular y JavaScript Nativo.



Los métodos de muestreo fueron por conveniencia y juicio de expertos. El muestreo por conveniencia es rápida y sin costo, ya que se selecciona la muestra de manera arbitraria ya que las unidades se autoseleccionan o se eligen de acuerdo a su fácil disponibilidad, la representación estructural es nula, es utilizada en la etapa explorativa para generar hipótesis, seleccionar el problema de estudio y conocer el objeto de investigación. El muestreo por juicio es una aproximación conceptual del universo de estudio y se selecciona unidades a partir de criterios conceptuales y las variables de la muestra de definen de manera teórica por el investigador y consta de 2 pasos: señalar características del objeto de estudio mediante criterios teóricos y elegir informantes según los tipos o niveles estructurales (Mejía, 1996).

Se seleccionó el framework Angular ya que en la mayoría de comparaciones tubo mejores resultados en rendimiento, usabilidad, mantenibilidad y portabilidad. Así mismo se seleccionó JavaScript ya que en una comparación realizada obtuvo mejor puntuación en tiempos de respuesta que BackboneJS y jQuery. Estos fueron seleccionados para poder realizar el análisis y descripción de la incidencia del tiempo de respuesta:

Angular

JavaScript Nativo

## 2.3. Variables e indicadores

### 2.3.1. Variables independientes

La variable independiente es: Framework JavaScript (Angular y JavaScript Nativo). La Tabla 1 muestra la operacionalización de la variable.

**Tabla 2**

*Operacionalización de variables independientes*

O. Específicos	V. independiente	Dimensión	Indicador	Instrumento
Evaluar el framework Angular y JavaScript Nativo referente al tiempo de desarrollo	Framework JavaScript	Tiempo de desarrollo	Nro días	Encuesta de Experto
Evaluar los frameworks Angular y JavaScript Nativo referente a la curva de aprendizaje		Curva de aprendizaje		

**Nota:** Fuente. elaboración propia

### 2.3.2. Variables dependientes

La variable dependiente es: tiempo de respuesta. La Tabla 2 muestra su operacionalización.

**Tabla 3**

*Operacionalización de variable dependiente*

O. Específicos	V. dependiente	Dimensión	Indicador	Instrumento
<p>Analizar el tiempo de respuesta del framework Angular en un aplicativo web MVC</p> <p>Analizar el tiempo de respuesta del framework JavaScript Nativo en un aplicativo web MVC</p>	Tiempo de respuesta del framework	<p>Tiempo de respuesta framework Angular</p> <p>Tiempo de respuesta framework JavaScript Nativo</p>	Tiempo (milesegundos)	Herramienta consola Google Chrome y Firefox
<p>Implementar un mantenimiento y consulta web MVC utilizando los frameworks Angular y JavaScript Nativo para poder observar la incidencia en el tiempo de respuesta</p> <p>Realizar los test para medir el tiempo de respuesta de los frameworks Angular y JavaScript Nativo en un aplicativo web MVC.</p>	Módulo desarrollado del framework	<p>Módulo framework Angular</p> <p>Módulo framework JavaScript Nativo</p>	Agregar, Editar, Eliminar, Listar	<p>IDE Desarrollo (Visual Studio)</p> <p>Ficha observación (Excel)</p>

**Nota:** Fuente. elaboración propia

## **2.4. Técnicas de recolección de datos**

### **2.4.1. Observación**

(Fresno Chavez, 2019, pág. 114). La observación se utiliza para la obtención de información inmediata del objeto investigado y comprobación empírica de las hipótesis; es sistemática, consciente y objetiva. En tal sentido utilizaremos ésta técnica para obtener la información de los diferentes valores del indicador tiempo de respuesta tomados en la pruebas de insertar, actualizar, eliminar y consultar los datos realizados a los frameworks Angular y JavaScript Nativo

### **2.4.2. Encuesta**

Cuando la información es obtenida a partir de respuestas que una o varias personas que colaboran con la investigación puedan dar a un cuestionario; es rígida, cuentan con una estructura lógica y facilita su procesamiento estadístico (León & González, 2006, pág. 68). La investigación se apoyará en la encuesta de expertos para tener una puntuación referente a los indicadores curva de aprendizaje y el tiempo de desarrollo respecto a los frameworks Angular y JavaScript nativo.

## 2.5. Validación y Confiabilidad del Instrumento

El instrumento guía de observación que utilizaremos en nuestra investigación es la que se muestra en la siguiente Tabla 4

**Tabla 4**

*Instrumento Ficha de Observación Framework JavaScript*

Framework Angular y JavaScript Nativo								
INVESTIGACION	Análisis de los frameworks JavaScript Nativo y Angular en la incidencia del tiempo de respuesta en una Web MVC en el sector comercial.							
Framework	Angular				JavaScript Nativo			
Indicador	Tiempo Respuesta				Tiempo Respuesta			
				(Listar 500+ Listar1000+ Listar2000)/3				(Listar 500+ Listar1000+ Listar2000)/3
Tiempo Ms	Agregar	Editar	Eliminar	Listar	Agregar	Editar	Eliminar	Listar

**Nota.** Fuente elaboración propia

Para validar el instrumento se realizó la certificación con la verificación de validadores ingenieros colegiados, el detalle de los resultados de la evaluación respectiva se muestran en el Anexo 10. Se tabularon los resultados y lo validaremos usando la prueba binomial que es un modelo probabilístico de utilidad en aplicaciones producidas en ensayos con dos resultados mutuamente excluyentes (Vásquez, 2012) y para realizar

los cálculos nos apoyaremos del software IBM SPSS. Los datos de los resultados resumen de la validación de los validadores expertos se muestran en la siguiente Tabla 5 y las tablas 6, 7, 8, 9 y 10 muestran la secuencia de resultados.

**Tabla 5**  
*Resumen Ficha Observación Framework JavaScript*

<b>Ficha Observación Framework JavaScript</b>	<b>Validador1</b>	<b>Validador2</b>	<b>Validador3</b>
<b>Dimensión: Tiempo de Respuesta</b>			
Tiempo al Agregar (ms)	SI	SI	SI
Tiempo al Editar (ms)	SI	SI	SI
Tiempo al Eliminar (ms)	SI	SI	SI
Tiempo al Listar (ms)	SI	SI	SI

**Nota.** Fuente elaboración propia

**Tabla 6**  
*Frecuencia Ficha Observación Framework JavaScript*

<b>Validador</b>	<b>Frecuencia</b>	<b>Frecuencia</b>	<b>Frecuencia</b>
<b>Opinión</b>	<b>Validador1</b>	<b>Validador2</b>	<b>Validador2</b>
SI	4	4	4
NO	0	0	0

**Nota.** Fuente elaboración propia

Los resultados de software IBM SPSS para cada validador fueron:

**Tabla 7**

*Resultado IBM SPSS Prueba binomial – Validador 1*

<b>Prueba binomial – Validador 1</b>						
		Categoría	N	Prop. observada	Prop. de prueba	Significación exacta (bilateral)
Validador1 Opinión	Grupo 1	SI	4	1,00	0,85	0,522
	Grupo 2	NO	0	0		
	Total		4	1,00		

**Nota.** Fuente elaboración propia

Grupo 1: El Validador acepta la validez de la pregunta del instrumento framework javascript

Grupo 2: El Validador no acepta la validez de la pregunta del instrumento framework javascript

### **Hipótesis Estadística**

H0: La proporción de preguntas aceptadas en el instrumento es igual a 85%

H1: La proporción de preguntas aceptadas en el instrumento es diferente del 85%

### **Criterio de decisión**

Se rechaza H0 si  $SIG < 0.05$ , caso contrario aceptar H0

Entonces  $SIG = 0.522 > 0.05$ , => se acepta H0

### **Conclusión**

La proporción de preguntas aceptadas por el validador para el instrumento framework javascript es igual a 85%.

**Tabla 8**

*Resultado IBM SPSS Prueba binomial – Validador 2*

<b>Prueba binomial – Validador 2</b>						
		Categoría	N	Prop. observada	Prop. de prueba	Significación exacta (bilateral)
Validador1 Opinión	Grupo 1	SI	4	1,00	0,85	0,522
	Grupo 2	NO	0	0		
	Total		4	1,00		

**Nota.** Fuente elaboración propia

Grupo 1: El Validador acepta la validez de la pregunta del instrumento framework javascript

Grupo 2: El Validador no acepta la validez de la pregunta del instrumento framework javascript

### **Hipótesis Estadística**

H0: La proporción de preguntas aceptadas en el instrumento es igual a 85%

H1: La proporción de preguntas aceptadas en el instrumento es diferente del 85%

### **Criterio de decisión**

Se rechaza H0 si  $SIG < 0.05$ , caso contrario aceptar H0

Entonces  $SIG = 0.522 > 0.05$ , => se acepta H0

### **Conclusión**

La proporción de preguntas aceptadas por el validador para el instrumento framework javascript es igual a 85%.



**Tabla 9**

*Resultado IBM SPSS Prueba binomial – Validador 3*

Prueba binomial – Validador 3						
		Categoría	N	Prop. observada	Prop. de prueba	Significación exacta (bilateral)
Validador1 Opinión	Grupo 1	SI	4	1,00	0,85	0,522
	Grupo 2	NO	0	0		
	Total		4	1,00		

**Nota.** Fuente elaboración propia

Grupo 1: El Validador acepta la validez de la pregunta del instrumento framework javascript

Grupo 2: El Validador no acepta la validez de la pregunta del instrumento framework javascript

### **Hipótesis Estadística**

H0: La proporción de preguntas aceptadas en el instrumento es igual a 85%

H1: La proporción de preguntas aceptadas en el instrumento es diferente del 85%

### **Criterio de decisión**

Se rechaza H0 si  $SIG < 0.05$ , caso contrario aceptar H0

Entonces  $SIG = 0.522 > 0.05$ , => se acepta H0

### **Conclusión**

La proporción de preguntas aceptadas por el validador para el instrumento framework javascript es igual a 85%.

**Tabla 10**

*Resumen Resultado IBM SPSS Prueba binomial – Validador 1, 2, 3*

Prueba binomial – Validador 1, 2, 3						
		Categoría	N	Prop. observada	Prop. de prueba	Significación exacta (bilateral)
Validador 1	Grupo 1	SI	4	1,00	0,85	0,522
	Grupo 2	NO	0	0		
Validador 2	Grupo 1	SI	4	1,00	0,85	0,522
	Grupo 2	NO	0	0		
Validador 3	Grupo 1	SI	4	1,00	0,85	0,522
	Grupo 2	NO	0	0		

**Nota.** Fuente elaboración propia

Grupo 1: El Validador acepta la validez de la pregunta del instrumento framework javascript

Grupo 2: El Validador no acepta la validez de la pregunta del instrumento framework javascript

### **Hipótesis Estadística**

H0: La proporción de preguntas aceptadas en el instrumento es igual a 85%

H1: La proporción de preguntas aceptadas en el instrumento es diferente del 85%

### **Criterio de decisión**

Se rechaza H0 si  $SIG < 0.05$ , caso contrario aceptar H0

Entonces  $SIG = 0.522 > 0.05$ , => se acepta H0

### **Conclusión final**

La proporción de preguntas aceptadas por el validador 1, 2 y 3 para el instrumento ficha observación framework javascript es igual a 85%.

## **2.6. Procedimiento de recolección de datos**

Se implementará un módulo de mantenimiento y consulta por cada uno de los frameworks Angular y JavaScript Nativo. Se medirá el tiempo al momento de agregar, actualizar, eliminar y consultar registros. Para las consultas de 500, 1000 y 2000 registros se realizarán cargas de datos. Los tiempos de respuesta se obtendrán utilizando las herramientas chrome-devtools y Firefox Developer Tools Depurador; dichas medidas se registrarán en la guía de test. Finalmente se procesarán las medidas obtenidas calculando la media del tiempo de respuesta de cada uno de los frameworks y se describirán los resultados. Así mismo se realizará una encuesta a especialistas en el tema para tener una puntuación de los frameworks en estudio como son JavaScript Nativo y Angular.

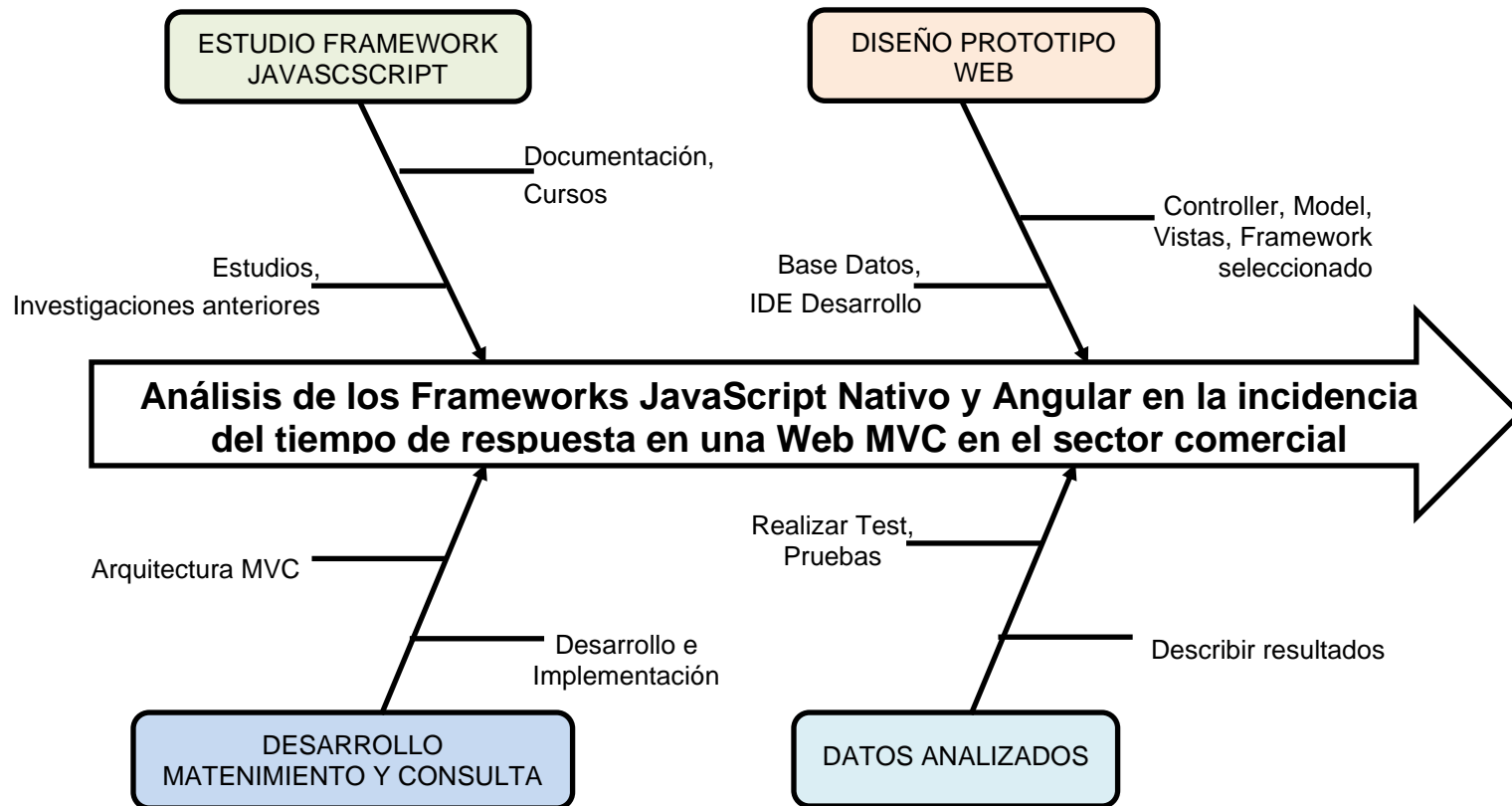


Figura 8. Proceso de recolección de datos. Fuente: elaboración propia

### 2.6.1. Análisis estadísticos

Para poder organizar los datos de forma ordenada y estadística nos apoyaremos en el uso de un aplicativo de hoja de cálculo. La forma que se presentara la información será tabular para un mejor entendimiento y poder interpretar los resultados obtenidos. Referente a la interpretación de los datos se efectuará en base al cálculo de la media de los tiempos de respuesta de los diferentes test aplicados de cada framework.

### 2.6.2. Herramienta estadística

**Microsoft Excel 2013.** Permite crear archivos .xlsx que contiene un libro de trabajo y éste contiene un número ilimitado de hojas de cálculo; cada hoja tiene una capacidad de 1'048,576 filas por 16,384 columnas. Las hojas de cálculo nos sirven para ingresar datos o números, crear fórmulas y funciones y generar gráficos (Verschuuren, 2014).

### 2.6.3. Medida de datos estadísticos

El cálculo de la media muestral se aplicará al tiempo de respuesta en base a los datos obtenidos en las diferentes pruebas de cada framework Angular y JavaScript Nativo; además se calculará la media en base a la puntuación de los expertos referente a los indicadores tiempo de desarrollo y curva de aprendizaje de cada uno de los frameworks seleccionados.

**Media aritmética.** La media aritmética, media o valor medio, es quizá la medida de posición más importante para una variable, pues proporciona una medida de la ubicación central de los datos. Si los datos son para una muestra, la media se denota  $\bar{x}$ ; si son para una población, se denota con la letra griega  $\mu$ . En las formulas estadísticas se acostumbra denotar el valor de la primera observación de la variable  $x$  mediante  $x_1$ , el valor de la segunda observación de la variable  $x$  por medio de  $x_2$ , y así sucesivamente. En general, el valor de la  $i$ -ésima observación de la variable  $x$  se representa por medio de  $x_i$ . Si se tiene una muestra con  $n$  observaciones, la fórmula para la media muestral es la siguiente: (Anderson, Sweeney, Williams, Camm, & Cochran, 2014, pág. 101).

$$\bar{x} = \frac{\sum x_i}{n}$$

## 2.7. Criterios éticos

La ética aplicada a la sociedad del conocimiento y era de globalización se denomina Ética de Investigación, que es una herramienta para estimular ingenio, nutrir o vertebrar talento para transmitir conocimiento, facilitar buenas prácticas y promover corresponsabilidad (Domingo, 2018). Los criterios aplicados a éste trabajo de investigación son:

**Honestidad:** en ésta investigación no se falsificara ni manipulara los datos.

**Objetividad:** las hipótesis serán aceptadas o rechazadas en base a hechos e información contrastable.

**Apertura:** se podrá difundir datos, resultados, herramientas, etc. y se respetará las críticas.

**Confidencialidad:** se asegurará y protegerá la información privada y confidencial, así mismo se protegerá la identificación de las personas que intervienen en la investigación.

**Respeto a la propiedad intelectual:** se hará las referencias respectivas de las fuentes y no se utilizará material o información sin permiso.

## 2.8. Criterios de rigor científico

Para garantizar la calidad del presente trabajo de investigación se establecen los siguientes criterios. La Tabla 4 muestra los criterios considerados.

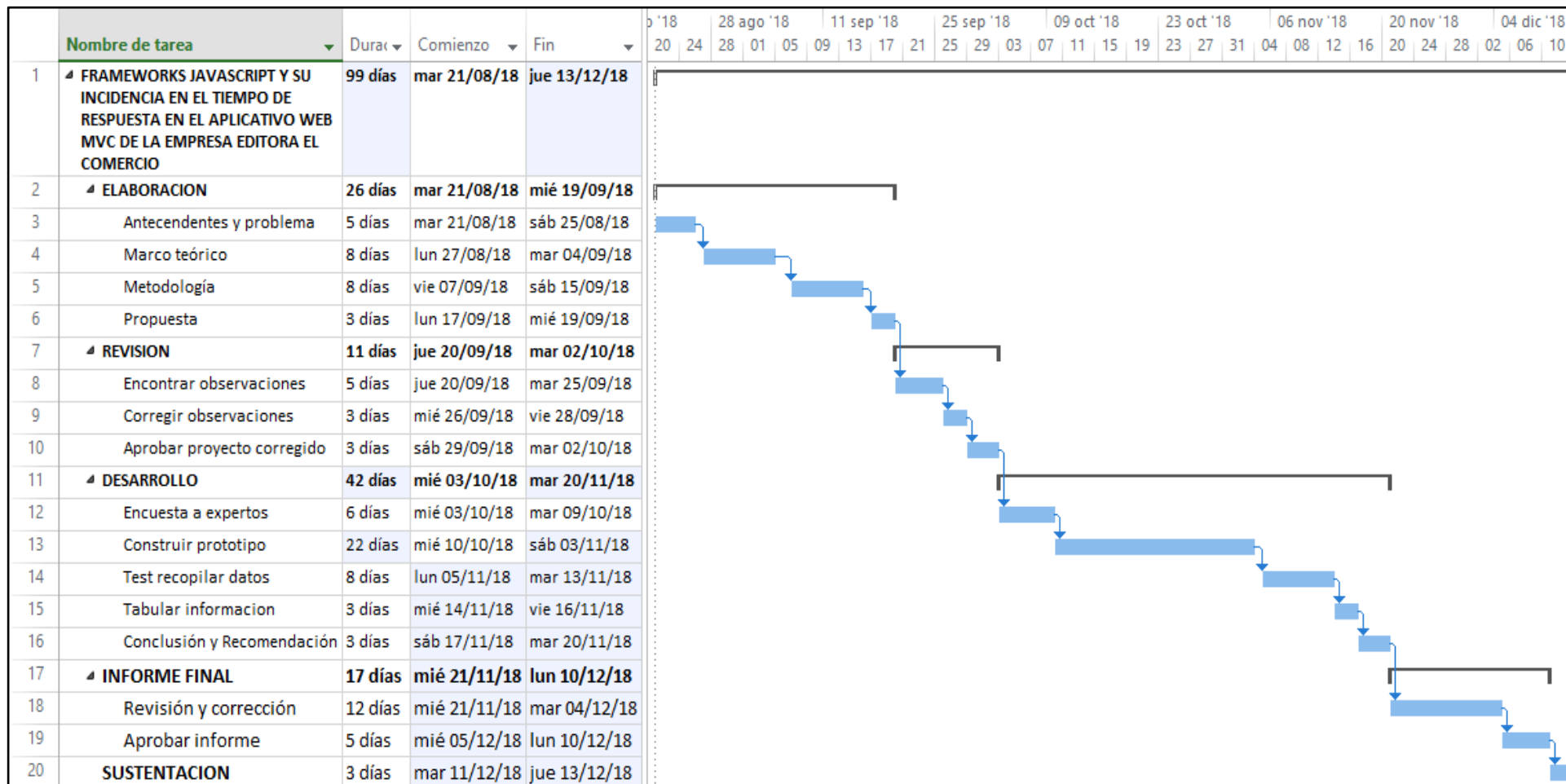
**Tabla 11**

*Criterios de rigor de nuestra investigación*

Criterio	Característica del criterio
Fiabilidad	El estudio podrá ser replicado y los resultados serán válidos e inequívocos.
Contrastación	Los resultados obtenidos son contrastables y permitirán responder a las hipótesis y preguntas de investigación

**Nota:** Fuente. elaboración propia

## 2.9. Cronograma de actividades.



Nota: Fuente. elaboración propia



## 2.10. Presupuesto

**Tabla 12**

*Materiales y Servicios para el desarrollo del proyecto*

Descripción	Cantidad	Costo Unitario	Sub Total
Laptop Lenovo	01	S/. 3500.00	S/. 4500.00
Memoria USB	01	S/. 40.00	S/. 40.00
Servicio de Internet	04	S/. 75.00	S/. 300.00
Movilidad	30	S/. 8.00	S/. 240.00
Impresiones	06	S/. 8.00	S/. 48.00
Luz	04	S/. 30.00	S/. 120.00
Telefonía Móvil	04	S/. 29.00	S/. 116.00
<b>Total</b>			<b>S/. 5364.00</b>

**Nota.** Fuente elaboración propia

## 2.11. Financiamiento

Para el desarrollo del proyecto el financiamiento será asumido únicamente por el tesista y el monto total de la inversión es la suma de S/. 5364 nuevos soles.

## CAPÍTULO III. RESULTADOS

### 3.1. Análisis e Interpretación de los datos

#### 3.1.1. Selección de los Framework

Se procedió con la selección del framework según las diferentes fuentes de investigación e información como tesis y paper. La Tabla 6 y 7 muestran información sobre la selección de los Framework Angular y JavaScript Nativo.

**Tabla 13**

*Selección del Framework Angular*

	<b>Tesis:</b>	<b>Paper:</b>
Investigación	Comparison of front-end frameworks for web applications development (Kaluža, Troškot, & Vukelić, 2018)	Utilización del modelo de referencia Web para el desarrollo de aplicaciones basadas en un Framework; evaluación establecida con indicadores (Ferreira, Ayala, & Cuin, 2018)
Justificación	Angular en un framework que nos permite escribir código completo y ordenado para su mantenimiento y está respaldado por google y Microsoft.	Uno de los mejores frameworks del lado del cliente que ofrece mayor aceptación y robustez en la creación de aplicaciones web es Angular que obtuvo una puntuación de 25 en el indicador de calidad FURPS, seguridad media, aplica a MVC, soporta distintos navegadores, Css3, html5 y JQuery.

**Nota.** Fuente elaboración propia

**Tabla 14**

*Selección del Framework JavaScript Nativo*

Investigación	<p><b>Paper:</b> Comparing performance between plain JavaScript and popular JavaScript frameworks (Ladan, 2015)</p>	<p><b>Tesis:</b> Contribución a una herramienta web de autoría de recursos de e-learning mediante la integración de un componente de realidad virtual (Gonzales, 2018)</p>
Justificación	<p>Realizó un estudio de JavaScript, jQuery y Backbone.js. Referente al tiempo de ejecución. Los resultados fueron favorables para JavaScript</p>	<p>JavaScript nos permite asignar valores, crear funciones, operadores en objetos nativos y estándares y su gran beneficio es la utilización de API (interfaz de programación en aplicaciones)</p>

**Nota.** Fuente elaboración propia

### 3.1.2. Selección de la Herramienta

La herramienta para el navegador Chrome es Chrome Developer Tools. Para Firefox la herramienta es Mozilla Tools. (Ferrer, 2014) Menciona que Mozilla cuenta con una herramienta integrada para diseñadores y programadores web, es potente, ayuda a depurador y medir el rendimiento de la web. (Atassi, 2014) Menciona que Chrome ofrece la herramienta para desarrolladores que ayuda a la depuración y administración de recursos de una web.

La Tabla 8 y 9 muestran la herramienta Chrome y Mozilla Tools respectivamente.

**Tabla 15**

*Selección de la Herramienta Chrome Developer Tools*

Investigación	<b>Paper:</b> Speed Performance Comparison of JavaScript MVC Frameworks (Svensson, 2015)	<b>Bibliografía:</b> Chrome ofrece la herramienta para desarrolladores que ayuda a la depuración y administración de recursos de una web (Atassi, 2014)
Justificación	Se utilizó la herramienta Chrome Developer Tools para medir los tiempos de respuesta	Mediante la opción red nos permite rastrear las actividades de red muestra información http, solicitudes, respuestas y tiempos

**Nota.** Fuente elaboración propia

**Tabla 16**

*Selección de la Herramienta Mozilla Tools*

Investigación	<b>Bibliografía:</b> Firefox es un navegador de comunidad de software libre con variedad de complementos y extensiones con diferentes funcionalidades (Álvarez, 2014)	<b>Bibliografía:</b> Mozilla cuenta con una herramienta integrada para diseñadores y programadores web, es potente, ayuda a depurador y medir el rendimiento de la web (Ferrer, 2014)
Justificación	Nos permite usar el analizador de conexiones de red, dónde se conecta el navegador, analizando el rendimiento	Nos muestra todas las solicitudes de red que Firefox realiza al cargar una página, muestra detalles de cada petición.

**Nota.** Fuente elaboración propia

### 3.2. Resultados de las mediciones

(Ver Anexos 04 y 05)

#### 3.2.1. Navegador Google Chrome

**Tabla 17**

*Resultado medición Tiempo Respuesta Angular y JavaScript Nativo - Chrome*

Framework Angular y JavaScript Nativo								
INVESTIGACION	Análisis de los frameworks JavaScript Nativo y Angular en la incidencia del tiempo de respuesta en una Web MVC en el sector comercial.							
Framework	Angular				JavaScript Nativo			
Criterio	Tiempo Respuesta				Tiempo Respuesta			
				(Listar 500+ Listar1000+ Listar2000)/3				(Listar 500+ Listar1000+ Listar2000)/3
Tiempo en Ms	Agregar	Editar	Eliminar	Listar	Agregar	Editar	Eliminar	Listar
	127	46	42	89.666	145	91	57	97.333

**Nota.** Fuente elaboración propia

#### 3.2.2. Promedio de datos Navegador Chrome

$$\text{Promedio Agregar} \quad 127 + 145 = \frac{272}{2} = 136 \text{ ms}$$

$$\text{Promedio Editar} \quad 46 + 91 = \frac{137}{2} = 68.50 \text{ ms}$$

$$\text{Promedio Eliminar} \quad 42 + 57 = \frac{99}{2} = 49.50 \text{ ms}$$

$$\text{Promedio Listar} \quad 89.66 + 97.33 = \frac{186.99}{2} = 93.495 \text{ ms}$$

#### 3.2.3. Interpretación de Resultados en navegador Chrome

Los resultados del Framework Angular referentes al tiempo de respuesta fueron:

**Agregar:** Resultado 127ms menor al resultado promedio cuyo valor es 136ms.

**Editar:** Resultado 46ms menor al resultado promedio cuyo valor es 68.50ms.

**Eliminar:** Resultado 42ms menor al resultado promedio cuyo valor es 49.50ms.

**Listar:** Resultado 88.66ms menor al resultado promedio cuyo valor es 93.495ms.

Los resultados del Framework JavaScript Nativo referentes al tiempo de respuesta fueron:

**Agregar:** Resultado 145ms mayor al resultado promedio cuyo valor es 136ms.

**Editar:** Resultado 91ms mayor al resultado promedio cuyo valor es 68.50 ms.

**Eliminar:** Resultado 57ms mayor al resultado promedio cuyo valor es 49.50 ms.

**Listar:** Resultado 97.33ms mayor al resultado promedio cuyo valor es 93.495ms.

### TIEMPO DE RESPUESTA PROMEDIO ANGULAR Y JAVASCRIPT NAVEGADOR GOOGLE CHROME

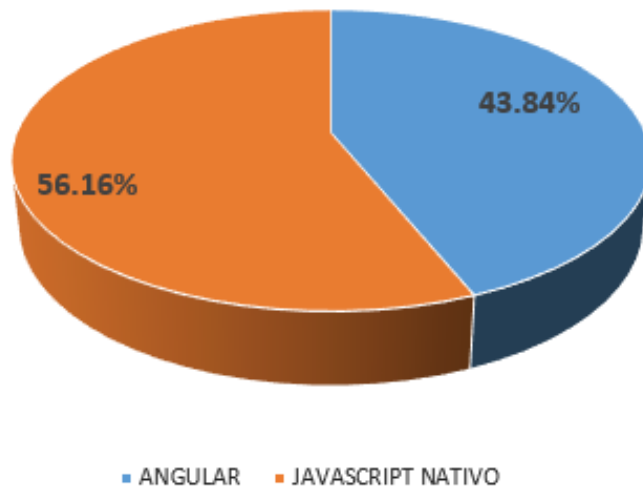


Figura 9. Gráfico porcentaje promedio de tiempo respuesta – google chrome. Fuente. Elaboración propia

### 3.2.4. Navegador Mozilla Firefox

**Tabla 18**

*Resultado medición Tiempo Respuesta Angular y JavaScript Nativo - Mozilla*

Framework Angular y JavaScript Nativo								
INVESTIGACION	Análisis de los Frameworks JavaScript Nativo y Angular en la incidencia del tiempo de respuesta en una Web MVC en el sector comercial.							
Framework	Angular				JavaScript Nativo			
Criterio	Tiempo Respuesta				Tiempo Respuesta			
				(Listar 500+ Listar1000+ Listar2000)/3				(Listar 500+ Listar1000+ Listar2000)/3
Tiempo en Ms	Agregar	Editar	Eliminar	Listar	Agregar	Editar	Eliminar	Listar
	32	17	32	40.00	39	23	48	50.333

**Nota.** Fuente elaboración propia

### 3.2.5. Promedio de datos Navegador Firefox

$$\begin{aligned} \text{Promedio Agregar} &= \frac{32 + 39}{2} = 35.5 \text{ ms} \\ \text{Promedio Editar} &= \frac{17 + 23}{2} = 20 \text{ ms} \\ \text{Promedio Eliminar} &= \frac{32 + 48}{2} = 40 \text{ ms} \\ \text{Promedio Listar} &= \frac{40.0 + 50.33}{2} = 45.16 \text{ ms} \end{aligned}$$

### 3.2.6. Interpretación de Resultados en el navegador Firefox

Los resultados del Framework Angular referentes al tiempo de respuesta fueron:

**Agregar:** Resultado 32ms menor al resultado promedio cuyo valor es 35.5ms.

**Editar:** Resultado 17ms menor al resultado promedio cuyo valor es 20ms.

**Eliminar:** Resultado 32ms menor al resultado promedio cuyo valor es 40ms.

**Listar:** Resultado 40ms menor al resultado promedio cuyo valor es 45.16ms.

Los resultados del Framework JavaScript Nativo referentes al tiempo de respuesta fueron:

**Agregar:** Resultado 39ms mayor al resultado promedio cuyo valor es 35.5ms.

**Editar:** Resultado 23ms mayor al resultado promedio cuyo valor es 20ms.

**Eliminar:** Resultado 48ms mayor al resultado promedio cuyo valor es 40ms.

**Listar:** Resultado 50.33ms mayor al resultado promedio cuyo valor es 45.16ms.

**TIEMPO DE RESPUESTA PROMEDIO  
ANGULAR Y JAVASCRIPT  
NAVEGADOR MOZILLA FIREFOX**

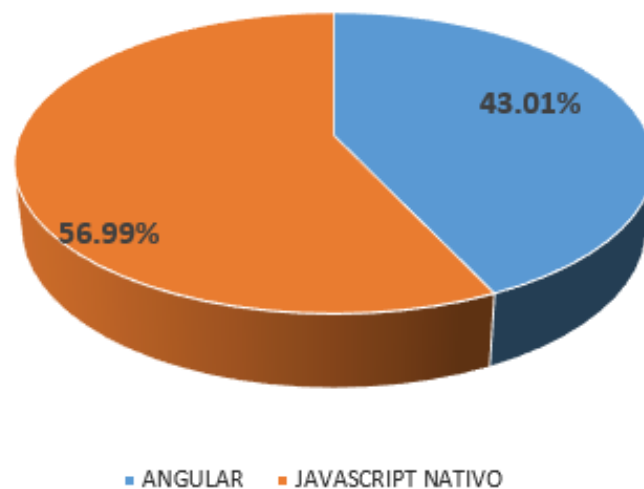


Figura 10. Gráfico porcentaje promedio de tiempo respuesta – mozilla firefox. Fuente. Elaboración propia



### 3.2.7. Combinación Navegadores Chrome y Mozilla

**Tabla 19**

*Resultado medición Tiempo Respuesta Angular y JavaScript Nativo - Chrome Mozilla*

Framework Angular y JavaScript Nativo																
INVESTIGACION	Análisis de los Frameworks JavaScript Nativo y Angular en la incidencia del tiempo de respuesta en una Web MVC en el sector comercial.															
Framework	Angular								JavaScript Nativo							
Criterio	Tiempo Respuesta								Tiempo Respuesta							
Tiempo en Ms	Agregar		Editar		Eliminar		Listar		Agregar		Editar		Eliminar		Listar	
	Chrome	Mozilla	Chrome	Mozilla	Chrome	Mozilla	Chrome	Mozilla	Chrome	Mozilla	Chrome	Mozilla	Chrome	Mozilla	Chrome	Mozilla
	127	32	46	17	42	32	89.666	40.00	145	39	91	23	57	48	97.333	50.333
	159		63		74		129.66		184		114		105		147.66	

**Nota.** Fuente elaboración propia

### 3.2.8. Promedio de datos Combinación Navegador Chrome y Firefox

Promedio Agregar	159 + 184	=	343/2	=	171.50 ms
Promedio Editar	63 + 114	=	177/2	=	88.50 ms
Promedio Eliminar	74 + 105	=	179/2	=	89.50 ms
Promedio Listar	129.66 + 147.66	=	277.32/2	=	138.66 ms

### 3.2.9. Interpretación de Resultados combinación navegadores Chrome y Mozilla

Los resultados para el Framework Angular combinando los navegadores Chrome y Mozilla fueron:

**Agregar:** El resultado fue de 159 ms y fue menor al resultado promedio de los 2 frameworks cuyo valor es 171.5 ms.

**Editar:** El resultado fue de 63 ms y fue menor al resultado promedio de los 2 frameworks cuyo valor es 88.5 ms.

**Eliminar:** El resultado fue de 74 ms y fue menor al resultado promedio de los 2 frameworks cuyo valor es 89.5 ms.

**Listar:** El resultado fue de 129.66 ms y fue menor al resultado promedio de los 2 frameworks cuyo valor es 138.66 ms.

Los resultados para el Framework JavaScript Nativo combinando los navegadores Chrome y Mozilla fueron:

**Agregar:** El resultado fue de 184 ms y fue mayor al resultado promedio de los 2 frameworks cuyo valor es 171.5 ms.

**Editar:** El resultado fue de 114 ms y fue mayor al resultado promedio de los 2 frameworks cuyo valor es 88.5 ms.

**Eliminar:** El resultado fue de 105 ms y fue mayor al resultado promedio de los 2 frameworks cuyo valor es 89.5 ms.

**Listar:** El resultado fue de 147.66 ms y fue mayor al resultado promedio de los 2 frameworks cuyo valor es 138.66 ms.

### 3.2.10. Promedio del tiempo de respuesta de cada Framework

$$\text{Promedio Angular} = \frac{159+63+74+129.66}{4} = 106.41 \text{ ms}$$

$$\text{Promedio JavaScript Nativo} = \frac{184+114+105+147.66}{4} = 137.66 \text{ ms}$$

### 3.2.11. Resultado tiempo de respuesta de Frameworks Angular y JavaScript Nativo

Los resultados nos muestran que el Framework que tiene mejor tiempo de respuesta es el Angular.

**TIEMPO DE RESPUESTA PROMEDIO  
ANGULAR Y JAVASCRIPT  
NAVEGADORES CHROME Y FIREFOX**

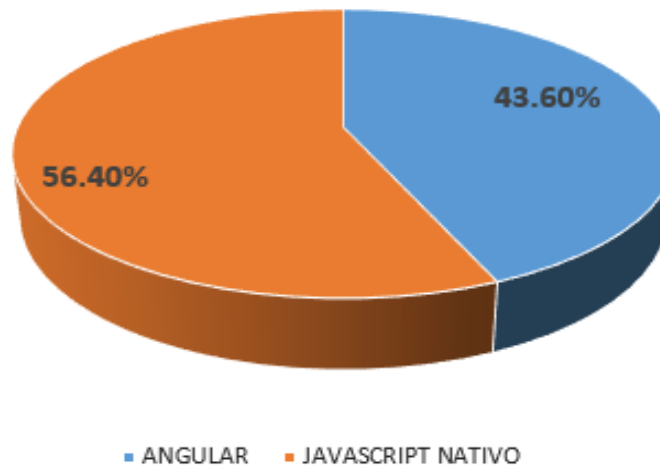


Figura 11. Gráfico porcentaje promedio de tiempo respuesta – Chrome y firefox. Fuente. Elaboración propia

### 3.2.12. Resultado encuesta Expertos Curva Aprendizaje

(Ver Anexos 06 y 07)

Las métricas de medición se clasifican en curva de aprendizaje y tiempo de desarrollo. La Tabla 13, 14 y 15 muestran los resultados de la encuesta sobre los frameworks referente a la Curva de aprendizaje. La Tabla 16 muestra el resultado promedio de los expertos.

Valores de puntuación: Excelente 10, Muy Bueno 7-9, Bueno 4-6 y Regular 1-3.



**Experto 1: Miguel Ángel Anastasio Velásquez**

**Tabla 20**

*Curva Aprendizaje –Experto 1. Fuente: elaboración propia*

---

<b>CURVA APRENDIZAJE</b>		
<b>FRAMEWORKS</b>		
<b>ATRIBUTOS</b>	<b>ANGULAR</b>	<b>JAVASCRIPT NATIVO</b>
Revisión de teoría y contenidos	9	7
Aplicar teoría en ejemplos prácticos	9	7
Operación	18/2	14/2
Promedio	9	7

---

**Nota.** Fuente elaboración propia

**Interpretación de los datos**

El resultado de la encuesta realizada al Ingeniero Miguel Ángel Anastasio Velásquez, se muestra los siguientes resultados:

**Angular:** 9 (Muy bueno)

**JavaScript Nativo:** 7 (Muy Bueno)

**Experto 2: James Huiza Pereida**

**Tabla 21**

*Curva Aprendizaje –Experto 2. Fuente: elaboración propia*

<b>CURVA APRENDIZAJE</b>		
<b>FRAMEWORKS</b>		
<b>ATRIBUTOS</b>	<b>ANGULAR</b>	<b>JAVASCRIPT NATIVO</b>
Revisión de teoría y contenidos	10	8
Aplicar teoría en ejemplos prácticos	10	8
Operación	20/2	16/2
Promedio	10	8

**Nota.** Fuente elaboración propia

**Interpretación de los datos**

El resultado de la encuesta realizada al Ingeniero James Huiza Pereira, se muestra los siguientes resultados:

**Angular:** 10 (Excelente)

**JavaScript Nativo:** 8 (Muy Bueno)

**Experto 3: Javier Azabeche Azmat**

**Tabla 22**

*Curva Aprendizaje –Experto 3. Fuente: elaboración propia*

<b>CURVA APRENDIZAJE</b>		
<b>FRAMEWORKS</b>		
<b>ATRIBUTOS</b>	<b>ANGULAR</b>	<b>JAVASCRIPT NATIVO</b>
Revisión de teoría y contenidos	9	7
Aplicar teoría en ejemplos prácticos	9	7
Operación	18/2	14/2
Promedio	9	7

**Nota.** Fuente elaboración propia

**Interpretación de los datos**

El resultado de la encuesta realizada al Ingeniero Javier Azabache Azmat, se muestra los siguientes resultados:

**Angular:** 9 (Muy Bueno)

**JavaScript Nativo:** 7 (Muy Bueno)

### 3.2.13. Promedio encuesta Expertos Curva Aprendizaje

**Tabla 23**

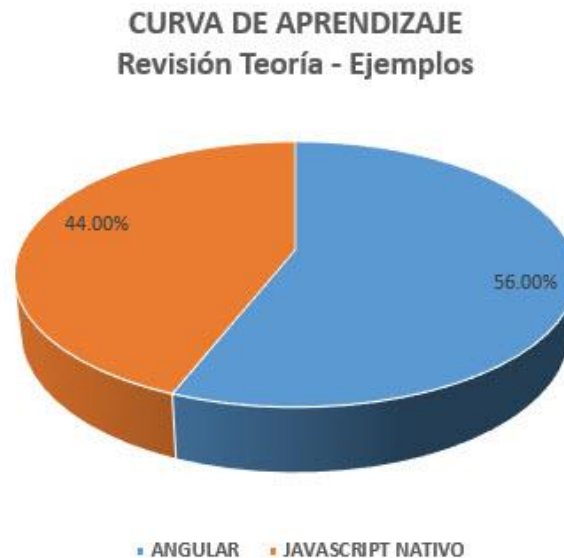
*Resultados del promedio Expertos Curva Aprendizaje*

PROMEDIO CURVA DE APRENDIZAJE		
FRAMEWORKS		
EXPERTOS	ANGULAR	JAVASCRIPT NATIVO
Experto 1	9	7
Experto 2	10	8
Experto 3	9	7
Operación	28/3	19.5/3
Promedio	9.33	7.33

**Nota.** Fuente elaboración propia

El promedio obtenido según la puntuación de expertos para el Framework Angular es de 9.33 indicando que es MUY BUENO.

El promedio obtenido según la puntuación de expertos para el Framework JavaScript Nativo es de 7.33 indicando que es MUY BUENO.



*Figura 12 Grafico Puntaje Promedio Curva Aprendizaje. Fuente. Elaboración propia*

### 3.2.14. Resultado encuesta Expertos Tiempo de Desarrollo

Las métricas de medición se clasifican en curva de aprendizaje y tiempo de desarrollo. La Tabla 17, 18 y 19 muestran los resultados de la encuesta del Framework Angular y JavaScript Nativo referente al Tiempo de Desarrollo. La Tabla 20 muestra el resultado promedio de los expertos.

Valores de puntuación: Excelente 10, Muy Bueno 7-9, Bueno 4-6 y Regular 1-3.

#### Experto 1: Miguel Ángel Anastasio Velásquez

**Tabla 24**

*Tiempo de Desarrollo –Experto 1. Fuente: elaboración propia*

TIEMPO DE DESARROLLO		
FRAMEWORKS		
ATRIBUTOS	ANGULAR	JAVASCRIPT NATIVO
Diseño y creación prototipo para uso del Framework	9	6
Programar y dar funcionalidad al prototipo	10	6
Operación	19/2	12/2
Promedio	9.5	6

**Nota.** Fuente elaboración propia

#### Interpretación de los datos

El resultado de la encuesta realizada al Ingeniero Miguel Ángel Anastasio Velásquez, se muestra los siguientes resultados:

**Angular:** 9.5 (Excelente)

**JavaScript Nativo:** 6 (Bueno)



**Experto 2: James Huiza Pereida**

**Tabla 25**

*Tiempo de Aprendizaje –Experto 2. Fuente: elaboración propia*

TIEMPO DE APRENDIZAJE		
FRAMEWORKS		
ATRIBUTOS	ANGULAR	JAVASCRIPT NATIVO
Diseño y creación prototipo para uso del Framework	9	7
Programar y dar funcionalidad al prototipo	9	7
Operación	18/2	14/2
Promedio	9	7

**Nota.** Fuente elaboración propia

**Interpretación de los datos**

El resultado de la encuesta realizada al Ingeniero James Huiza Pereira, se muestra los siguientes resultados:

**Angular:** 9 (Muy Bueno)

**JavaScript Nativo:** 7 (Muy Bueno)

**Experto 3: Javier Azabeche Azmat**

**Tabla 26**

*Tiempo de Aprendizaje –Experto 3. Fuente: elaboración propia*

TIEMPO DE APRENDIZAJE		
FRAMEWORKS		
ATRIBUTOS	ANGULAR	JAVASCRIPT NATIVO
Diseño y creación prototipo para uso del Framework	9	7
Programar y dar funcionalidad al prototipo	9	7
Operación	18/2	14/2
Promedio	9	7

**Nota.** Fuente elaboración propia

**Interpretación de los datos**

El resultado de la encuesta realizada al Ingeniero Javier Azabache Azmat, se muestra los siguientes resultados:

**Angular:** 9 (Muy Bueno)

**JavaScript Nativo:** 7 (Muy Bueno)

### 3.2.15. Promedio encuesta Expertos Tiempo de Desarrollo

**Tabla 27**

*Resultados del promedio Expertos Tiempo de Desarrollo*

PROMEDIO TIEMPO DE DESARROLLO		
EXPERTOS	FRAMEWORKS	
	ANGULAR	JAVASCRIPT NATIVO
Experto 1	9.5	6
Experto 2	9	7
Experto 3	9	7
Operación	27.5/3	23/3
Promedio	9.16	6.66

El promedio obtenido según la puntuación de expertos para el Framework Angular es de 9.16 indicando que es MUY BUENO.

El promedio obtenido según la puntuación de expertos para el Framework JavaScript Nativo es de 6.66 indicando que es MUY BUENO.



*Figura 13 Grafico Puntaje Promedio Tiempo de Desarrollo. Fuente. Elaboración propia*

### 3.3. Contrastación de la Hipótesis

Realizando las pruebas y verificaciones de cada framework seleccionado como es Angular y JavaScript Nativo para validar cual es más óptimo en el tiempo de respuesta, Los resultados muestran que el framework Angular presenta mejor tiempo de respuesta con un valor de 106.41ms y el framework JavaScript Nativo presenta un valor de 137.66 ms. Para contrastar nuestra hipótesis tomaremos como valor promedio de tiempo de respuesta 222ms sin aplicar los frameworks.

En nuestra investigación el tamaño de la muestra es  $n = 2$  que es menor a  $n < 30$ ; así mismo no se conoce el valor de la varianza poblacional  $\sigma^2$  para lo cual la calcularemos en base a la varianza de la muestra  $S$  de esta manera utilizaremos el estadístico  $t_0$  representado por: (Morán, 2003)

$$t_0 = \frac{\bar{x} - \mu_0}{\frac{S}{\sqrt{n}}}$$

Ecuación 01

La distribución T de Student será con  $(n-1)$  grados de libertad (gl), con un nivel de significancia de  $\alpha = 0.05$  y nuestra Hipótesis Nula  $H_0$  y la Hipótesis Alternativa  $H_1$ :

$$H_0: \mu = 223$$

$$H_1: \mu \neq 223$$

Datos:

$\alpha = 0.05$ ,  $gl = 1$ , en base a los datos obtenemos nuestro valor de dos colas que es 6.31

Procedemos a calcular nuestro error estándar que será calculado en base a la desviación

estándar de la muestra  $S = 22.09$ . El valor de la media de la muestra es  $\bar{X} = 122.04$

Reemplazando valores en la ecuación 01

$$t_0 = \frac{122.04 - 222}{\frac{22.09}{\sqrt{2}}} = -6.39$$

Verificamos que el resultado del estadístico muestral es -6.39 y es menor que el valor crítico de -6.31 lo cual indica que está en la zona crítica y la hipótesis nula  $H_0$  se rechaza y nos hace concluir que aplicando un framework javascript incide en un menor tiempo de respuesta con una significación del 5%.

## CAPÍTULO IV. DISCUSIÓN Y CONCLUSIONES

### 4.1. Discusión de Resultados

Los resultados obtenidos en éste trabajo de investigación arrojan un puntaje favorable para el framework Angular respecto al tiempo de respuesta. En la comparación realizada sobre las operaciones de agregar, editar, eliminar y listar en el navegador google chrome el framework Angular representa el 43.84% respecto al tiempo de respuesta; mientras que el framework JavaScript Nativo representa el 56.16% respecto al tiempo de respuesta. En el navegador mozilla firefox el framework Angular representa el 43.01% respecto al tiempo de respuesta; mientras que el framework JavaScript Nativo representa el 56.99% respecto al tiempo de respuesta. Al promediar el resultado de ambos navegadores Angular representa el 43.60% y JavaScript Nativo representa el 56.40% respecto al tiempo de respuesta respectivamente. Contrastando con la investigación “Análisis Comparativo entre los frameworks JavaScript MVC, Angular y Ember para el desarrollo de aplicaciones Web” también demuestran que Angular tiene mayor rendimiento (Peña & Cambisaca, 2015). Así mismo los resultados de la investigación “Speed Performance Comparison of JavaScript MVC Frameworks” dan mejor puntuación al framework Angular (Svensson, 2015).

Sobre la puntuación de la curva de aprendizaje la encuesta realizada a 03 expertos en la materia nos ilustra que hay un esfuerzo mayor para el aprendizaje del framework Angular que representa el 56% versus el framework JavaScript Nativo que representa el 44% del esfuerzo respecto al aprendizaje. Al contrastar con la investigación “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” (Zurita & Zavala, 2016) los expertos dieron un puntaje similar para diferentes frameworks incluyendo Angular. Referente a la curva de aprendizaje del Framework JavaScript Nativo no se encontraron estudios en donde se muestren datos referente a la curva de aprendizaje por lo cual no es posible contrastar nuestra investigación.

Respecto al tiempo de desarrollo los resultados de la encuesta realizada a los 03 expertos en la materia nos indican que el framework Angular representa un 57.90% respecto al tiempo de desarrollo; mientras que el framework JavaScript Nativo representa el 42.10% de la misma. La misma investigación que contrasta la curva de aprendizaje da como resultado un valor simétrico referente al tiempo de desarrollo para frameworks en estudio incluido Angular. En la entrevista realizada a un experto en desarrollo menciona que el framework angular gestiona mejor la velocidad de carga mediante utilizando patrones propios de su arquitectura, reduciendo el tiempo de respuesta a la primera carga y respecto al javascript nativo menciona que es usado en proyectos pequeños que no tienen muchos requerimientos funcionales; además menciona que angular permite organizar el código por componentes, plantillas y pruebas unitarias, lo que no ofrece javascript al momento de implementar aplicaciones web.

## **4.2. Conclusiones y Recomendaciones**

### **4.2.1. Conclusiones**

El mejor tiempo de desarrollo entre los frameworks Angular y JavaScript Nativo lo obtuvo el framework JavaScript Nativo representando el 42.10% del total del tiempo de desarrollo promedio.

La mejor puntuación respecto a la curva de aprendizaje entre los frameworks Angular y JavaScript Nativo la obtuvo el framework JavaScript Nativo representando el 40.00% del total de la curva de aprendizaje promedio.

Respecto al tiempo de respuesta entre los frameworks Angular y JavaScript Nativo ésta investigación demostró que el más óptimo fue para Angular representando el 43.60% del total de tiempo de respuesta promedio.

Implementar aplicaciones web con arquitectura MVC combinándolas con un framework del lado del cliente nos permite obtener productos con gran rendimiento en entornos web respecto al tiempo de respuesta.

La elaboración de un buen diseño de prototipo de una web, ayuda a agilizar la construcción e implementación de dicha aplicación.

Al realizar las test en los navegadores google chrome y mozilla se validó que los mismos poseen herramientas flexibles que facilitar a los desarrolladores realizar distintos tipos de medidas que en nuestro caso fue el tiempo de respuesta.

#### **4.2.2.Recomendaciones**

Se recomienda que se realicen otras investigaciones con diferentes framework JavaScript y que puedan combinarse dichos frameworks para poder obtener diferentes resultados y publicar cuales son los más óptimos para aplicarlos en el desarrollo de aplicaciones web.



Se recomienda que se realicen pruebas de medición de tiempos de respuesta con otras herramientas y con otros navegadores que cuenten con herramientas integradas para poder obtener datos válidos.

Realizar investigaciones para medir el tamaño de carga, recursos usados, etc.

## REFERENCIAS

- Álvarez, J. A. (2014). *Entornos personales de aprendizaje (PLE): aprendizaje conectado en red*. Obtenido de <https://ebookcentral.proquest.com/lib/upnortesp/detail.action?docID=4824648>
- Anderson, D., Sweeney, D., Williams, T., Camm, J., & Cochran, J. (2014). *Estadística para Negocios y Economía 12a. ed.* Obtenido de [https://bibliotecavirtual.cengage.com/online\\_viewer/pdf](https://bibliotecavirtual.cengage.com/online_viewer/pdf)
- Atassi, S. (2014). *Publicación de páginas web (MF0952\_2)*. Obtenido de <http://ebookcentral.proquest.com/lib/upnortesp/detail.action?docID=4184142>
- Barceló, J., Griera, J., & Viejo, S. (2008). *Protocolos y aplicaciones Internet*. Obtenido de <http://ebookcentral.proquest.com/lib/upnortesp/reader.action?docID=3206994>
- 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>
- Cabello, S. (2014). *Elaboración de plantillas y formularios (UF1304)*. Obtenido de <http://ebookcentral.proquest.com/lib/upnortesp/detail.action?docID=4184035>
- Cacharreros, w. (s.f.). *Cacharreros de la Web*. Obtenido de <https://cacharrerodelaweb.com/2018/02/que-es-javascript-y-para-que-sirve.html>
- Cano, C. (2018). *Mejores frameworks 2018 en JavaScript*. Obtenido de <https://comenzandodecero.com/mejores-frameworks-2018-en-javascript/>
- 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](http://www.dtic.mil/get-tr-doc/pdf?AD=AD1025789)
- Domingo, A. (2018). *Ética de la investigación: ingenio, talento y responsabilidad*. Obtenido de <http://ebookcentral.proquest.com/lib/upnortesp/detail.action?docID=5426388>
- 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>
- Ferreira, H., Ayala, K., & Cuin, J. (2018). *Utilización del modelo de referencia Web para el desarrollo de aplicaciones basadas en un framework; evaluación establecida con indicadores*. Obtenido de [https://www.researchgate.net/profile/Juan\\_Ruiz\\_Lagunas/publication/327602243\\_Utilizacion\\_del\\_modelo\\_de\\_referencia\\_Web\\_para\\_el\\_desarrollo\\_de\\_aplicaciones\\_basadas\\_en\\_un\\_framework\\_evaluacion\\_establecida\\_con\\_indicadores/links/5b992606a6fdcc59bf8bb5d3/](https://www.researchgate.net/profile/Juan_Ruiz_Lagunas/publication/327602243_Utilizacion_del_modelo_de_referencia_Web_para_el_desarrollo_de_aplicaciones_basadas_en_un_framework_evaluacion_establecida_con_indicadores/links/5b992606a6fdcc59bf8bb5d3/)
- Ferrer, J. (2014). *Publicación de páginas web*. Obtenido de <http://ebookcentral.proquest.com/lib/upnortesp/detail.action?docID=5759029>
- Fresno Chavez, C. (2019). *Metodología de la investigación: así de fácil*. Obtenido de <http://ebookcentral.proquest.com/lib/upnortesp/detail.action?docID=5635734>
- García, C., & Arroyo, D. (2007). *Biblioteca digital y web semántica*. Obtenido de <http://ebookcentral.proquest.com/lib/upnortesp/detail.action?docID=3174761>
- 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>
- Gonzales, P. (2018). *Contribución a una Herramienta Web de autoría de recursos de E-Learning mediante la Integración de un componente de realidad virtual*. Obtenido de [http://oa.upm.es/52001/1/PFC\\_PAULA\\_GONZALEZ\\_GOMEZ\\_2018.pdf](http://oa.upm.es/52001/1/PFC_PAULA_GONZALEZ_GOMEZ_2018.pdf)
- Google, D. (2018). *Chrome DevTools*. Obtenido de [https://developers.google.com/web/tools/chrome-devtools/?utm\\_source=dcc&utm\\_medium=redirect&utm\\_campaign=2018Q2](https://developers.google.com/web/tools/chrome-devtools/?utm_source=dcc&utm_medium=redirect&utm_campaign=2018Q2)
- Guerrero, G., & Guerrero, C. (2014). *Metología de la Investigación*. Obtenido de <http://ebookcentral.proquest.com/lib/upnortesp/reader.action?docID=3228613>
- Hans, D. (2015). *Performance of JavaScript frameworks on web single page applications(SPA)*. Obtenido de <https://repositorio.uc.cl/handle/11534/16587>
- Ingavélez, P., Hilerá, J., & Timbi, C. (2016). *Ática 2016. Tecnología y accesibilidad*. Obtenido de <http://ebookcentral.proquest.com/lib/upnortesp/detail.action?docID=5308458>
- 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](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>
- León, R., & González, S. (2006). *El proceso de investigación científica*. Obtenido de <http://ebookcentral.proquest.com/lib/upnortesp/detail.action?docID=3192826>
- Libardo, C. (2015). *Evaluando la Facilidad de Aprendizaje de Frameworks MVC en el Desarrollo de Aplicaciones Web*. Obtenido de <http://hemeroteca.unad.edu.co/index.php/publicaciones-e-investigacion/article/view/1592/1924>
- Lozada, J. (2015). *Ejecución eficiente de secuencias de navegación web*. Obtenido de <http://ruc.udc.es/dspace/handle/2183/16579>
- Manish, S. (2016). *Web development using C# MVC and ExtJS*. Obtenido de [http://repository.stcloudstate.edu/csit\\_etds/6/](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>
- Mejía, J. (1996). *El Muestreo en la Investigación Cualitativa*. Obtenido de <http://revistasinvestigacion.unmsm.edu.pe/index.php/sociales/article/viewFile/6851/6062>
- 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>
- Morán, S. (2003). *Uso de Minitab Statistica y Excel para contrastar hipótesis estadísticas paramétricas*. Obtenido de <https://cdigital.uv.mx/bitstream/handle/123456789/47497/MoranOlguinSevero.pdf?sequence=1&isAllowed=y>
- 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>
- Moreno, W. (2013). *Ocho pasos para el desarrollo de una investigación*. Obtenido de <http://ebookcentral.proquest.com/lib/upnortesp/detail.action?docID=3225679>
- Niño, V. (2011). *Metodología de la Investigación*. Obtenido de <http://ebookcentral.proquest.com/lib/upnortesp/reader.action?docID=3198784&query=dise%C3%B1o+investigaci%C3%B3n>

- 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](http://www.ecorfan.org/spain/researchjournals/Sistemas_Computacionales_y_TICs/vol2num4/Revista_de_Sistemas_Computacionales_y_TIC%60S_V2_N4.pdf#page=65)
- Ordax, J., & Ocaña, P. (2012). *Programación web en java*. Obtenido de <http://ebookcentral.proquest.com/lib/upnortesp/detail.action?docID=3214540>
- Pagés, D., & Martínez, J. (2010). *Cartucho de AndroMDA para JSF interpretando nueva estrategia de modelado*. Obtenido de <http://ebookcentral.proquest.com/lib/upnortesp/detail.action?docID=3202659>
- 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>
- Perles, J. (2014). *Juan Luis Perles García*. Obtenido de <http://ebookcentral.proquest.com/lib/upnortesp/detail.action?docID=5214331>
- Ribes, P. (2011). *Manual de Javascript*. Obtenido de <http://ebookcentral.proquest.com/lib/upnortesp/detail.action?docID=3217353>
- Rico, J., & Rondon, J. (2013). *Desarrollo de un Framework orientado a la web basado en los patrones de diseño MVC y DAO en el lenguaje de Programación PHP*. Obtenido de [http://repository.uniminuto.edu:8080/xmlui/bitstream/handle/10656/2897/TTI\\_RicoGuevaraJaime\\_2013.pdf?sequence=1&isAllowed=y](http://repository.uniminuto.edu:8080/xmlui/bitstream/handle/10656/2897/TTI_RicoGuevaraJaime_2013.pdf?sequence=1&isAllowed=y)
- Rodríguez, F. (2014). *Integración de componentes software en páginas web*. Obtenido de <http://ebookcentral.proquest.com/lib/upnortesp/detail.action?docID=3229028>
- Sáez, J. (2017). *Investigación Educativa. Fundamentos teóricos, procesos y elementos prácticos*. Obtenido de <https://ebookcentral.proquest.com/lib/upnortesp/reader.action?docID=4870247&ppg=1>
- Sánchez, C. (2014). *Creación de Páginas Web con el lenguaje de marcas*. Obtenido de <http://ebookcentral.proquest.com/lib/upnortesp/reader.action?docID=4422020>
- Sanchez, C., Tuesta, V., & Mejía, I. (2015). *Análisis Comparativo de Frameworks para el desarrollo de aplicaciones Web en Java*. Obtenido de <http://revistas.uss.edu.pe/index.php/ING/article/view/101>

- 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+WEBFORMS+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>
- Torres, J. (2017). *Las 7 razones para utilizar Angular 4 en tus proyectos de desarrollo web*. Obtenido de <https://www.offing.es/7-razones-para-utilizar-angular-4/>
- Vara, J., López, M., & Verde, J. (2014). *Desarrollo web en entorno servidor*. Obtenido de <http://ebookcentral.proquest.com/lib/upnortesp/reader.action?docID=3229695>
- Vásquez, E. (2012). *Contribución al tratamiento estadístico de datos con distribución binomial en el modelo de análisis de varianza*. Obtenido de <http://ebookcentral.proquest.com/lib/upnortesp/detail.action?docID=3201533>
- Verschuuren, G. (2014). *Excel 2013 for Scientists*. Obtenido de <https://ebookcentral.proquest.com/lib/upnpe/reader.action?docID=1652816&query=excel+2013>
- www.netmarketshare.com. (s.f.). <https://netmarketshare.com>. Obtenido de <https://netmarketshare.com>
- 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>

## ANEXOS

### Anexo 01 – Matriz de Consistencia

PROBLEMA	OBJETIVOS	HIPOTESIS	VARIABLES	INDICADORES	INSTRUMENTOS	TECNICAS
<b>GENERAL</b>	<b>GENERAL</b>	<b>GENERAL</b>				
¿Cuál es la incidencia en el tiempo de respuesta del Framework JavaScript en un aplicativo web MVC de empresas del sector comercial?	Analizar los Frameworks Angular y JavaScript Nativo en la incidencia en el tiempo de respuesta en un aplicativo web MVC de empresas del sector comercial.	Mediante el análisis y evaluación de los Frameworks Angular y JavaScript Nativo se podrá obtener los resultados de la incidencia en el tiempo de respuesta en un aplicativo web MVC de empresas del sector comercial.	<b>Variable Independiente</b> Framework JavaScript	Tiempo de desarrollo Curva de aprendizaje	Planilla de Juicio de Expertos	Encuesta
<b>ESPECIFICOS</b>	<b>ESPECIFICOS</b>	<b>ESPECIFICAS</b>				
¿Cuál es la análisis de los frameworks JavaScript Nativo y angular en tiempo de desarrollo y curva de aprendizaje? ¿Cuál es el tiempo de respuesta del framework Angular en una aplicación MVC? ¿Cuál es el tiempo de respuesta del framework JavaScript Nativo en una aplicación MVC?.	Analizar los Frameworks Angular y JavaScript Nativo referente al tiempo de desarrollo y curva de aprendizaje. Analizar el tiempo de respuesta del Framework Angular en un aplicativo web MVC. Analizar el tiempo de respuesta utilizando JavaScript Nativo en un aplicativo web MVC	Mediante el análisis de los Frameworks Angular y JavaScript Nativo obtendremos resultados de tiempo de desarrollo y curva de aprendizaje. Mediante el análisis del Framework Angular se podrá describir los resultados de la incidencia en el tiempo de respuesta en un aplicativo web MVC. Mediante el análisis del Framework JavaScript Nativo se podrá describir los resultados de la incidencia en el tiempo de respuesta en un aplicativo web MVC.	<b>Variable dependiente</b> Tiempos de respuesta de cada Framework	Tiempo de respuesta	Guías de Test Herramienta chrome-devtools Firefox Developer Tools Depurador	Observación
	Implementar un mantenimiento y consulta utilizando los Frameworks Angular y JavaScript Nativo para poder observar la incidencia en el tiempo de respuesta. Realizar los test para medir el tiempo de respuesta de los Frameworks Angular y JavaScript Nativo en un aplicativo web MVC.	Utilizando los Frameworks Angular y JavaScript Nativo en la implementación de mantenimiento y consulta podremos observar la incidencia en el tiempo de respuesta. Mediante los test aplicados al aplicativo web MVC podremos medir los tiempos de respuesta de los Frameworks Angular y JavaScript Nativo.	<b>Población</b> Knockout, JavaScript, VueJS, Backbone, Angular, EmberJS, Aurelia, Mithril, ReactJS, ExtJS <b>Muestra</b> Angular JavaScript Nativo			

**Nota:** Fuente. elaboración propia

## Anexo 02 - Solución Tecnológica

Maquetación de acceso, mantenimiento y consulta (Angular y JavaScript Nativo)

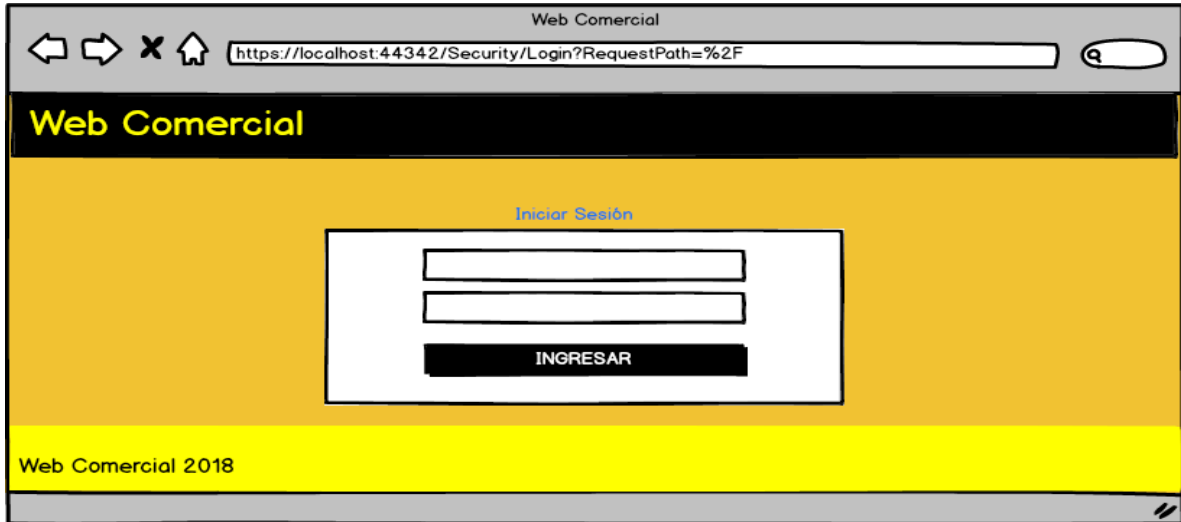


Figura 14 Diseño de Mockup pantalla acceso. Fuente: información propia

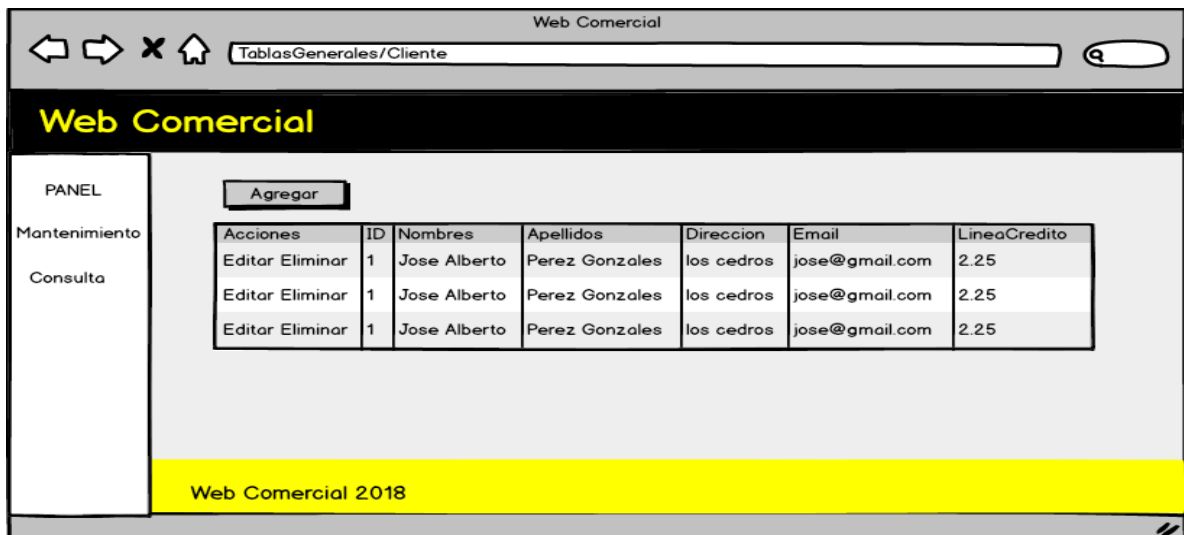


Figura 15 Diseño de Mockup pantalla mantenimiento. Fuente: información propia



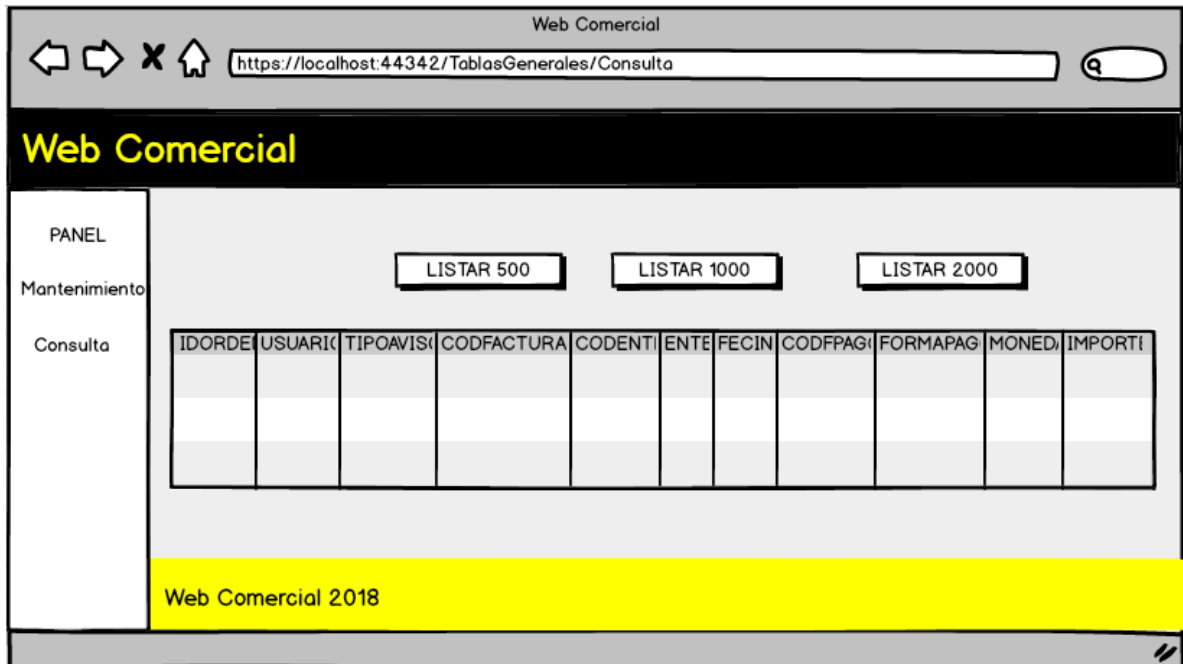


Figura 16 Diseño de Mockup pantalla mantenimiento. Fuente: información propia

### Carga Masiva.

Se harán cargas masivas a la base de datos con diferentes cantidades de registros para poder realizar las consultas y obtener los tiempos de respuesta en cada prueba.

### Test.

Se aplicaran diferentes test para poder capturar y obtener los tiempos de respuesta de cada uno de los Frameworks seleccionados. Las pruebas se realizaran con mantenimiento de registros y con diferentes cargas masivas de data para las consultas y se emplearan herramientas de los navegadores para medir el tiempo (Google Chrome y Mozilla Firefox).

### **Encuesta a Expertos.**

Se realizará una encuesta a profesionales expertos referente a indicadores como el tiempo de desarrollo y curva de aprendizaje de cada uno de los Frameworks Angular y JavaScript Nativo.

### **Guías de Test**

Documentos con formato tabulado diseñado para una mejor organización de la información recolectada en los procesos de la test de los tiempos de respuesta de los Frameworks.

### **Formato Encuesta Juicio de Experto**

Formato que nos servirá para plasmar los resultados de las encuestas realizadas a los profesionales expertos sobre los indicadores de los Framework Angular y JavaScript Nativo. (Ver Anexo 03).

### **Herramienta chrome-devtools**

Esta herramienta viene integrada en el navegador google Chrome y nos servirá para medir los tiempos de respuesta del aplicativo web MVC.

### **Firefox Developer Tools Depurador**

Herramienta para medir los tiempos de respuesta en el navegador Mozilla Firefox.

### Anexo 03 – Formato Encuesta Juicio de Experto

<b>ENCUESTA JUICIO DE EXPERTO</b>							
<b>EXPERTO</b>							
<b>FRAMEWORK</b>							
En este documento se recopilará información fidedigna que cada experto a juicio personal emitirá de cada Framework en estudio. La puntuación será asignada según los valores a continuación.							
Excelente	10	Muy bueno	7 - 9	Bueno	4 - 6	Regular	1 – 3
	<b>INDICADOR DEL FRAMEWORK</b>						
<b>A</b>	<b>CURVA APRENDIZAJE</b>			<b>Nº DIAS</b>	<b>PUNTUACION</b>		
	Revisión de teoría y contenidos						
	Aplicar teoría en ejemplos prácticos						
	<b>PROMEDIO PUNTUACION</b>						
<b>B</b>	<b>TIEMPO DE DESARROLLO</b>			<b>Nº DIAS</b>	<b>PUNTUACION</b>		
	Diseño y creación prototipo para uso del Framework						
	Programar y dar funcionalidad al prototipo						
	<b>PROMEDIO PUNTUACION</b>						

**Nota.** Fuente elaboración propia

### Anexo 04 – Resultados de Medición de Tiempos de Respuesta Angular – JavaScript Nativo Google Chrome

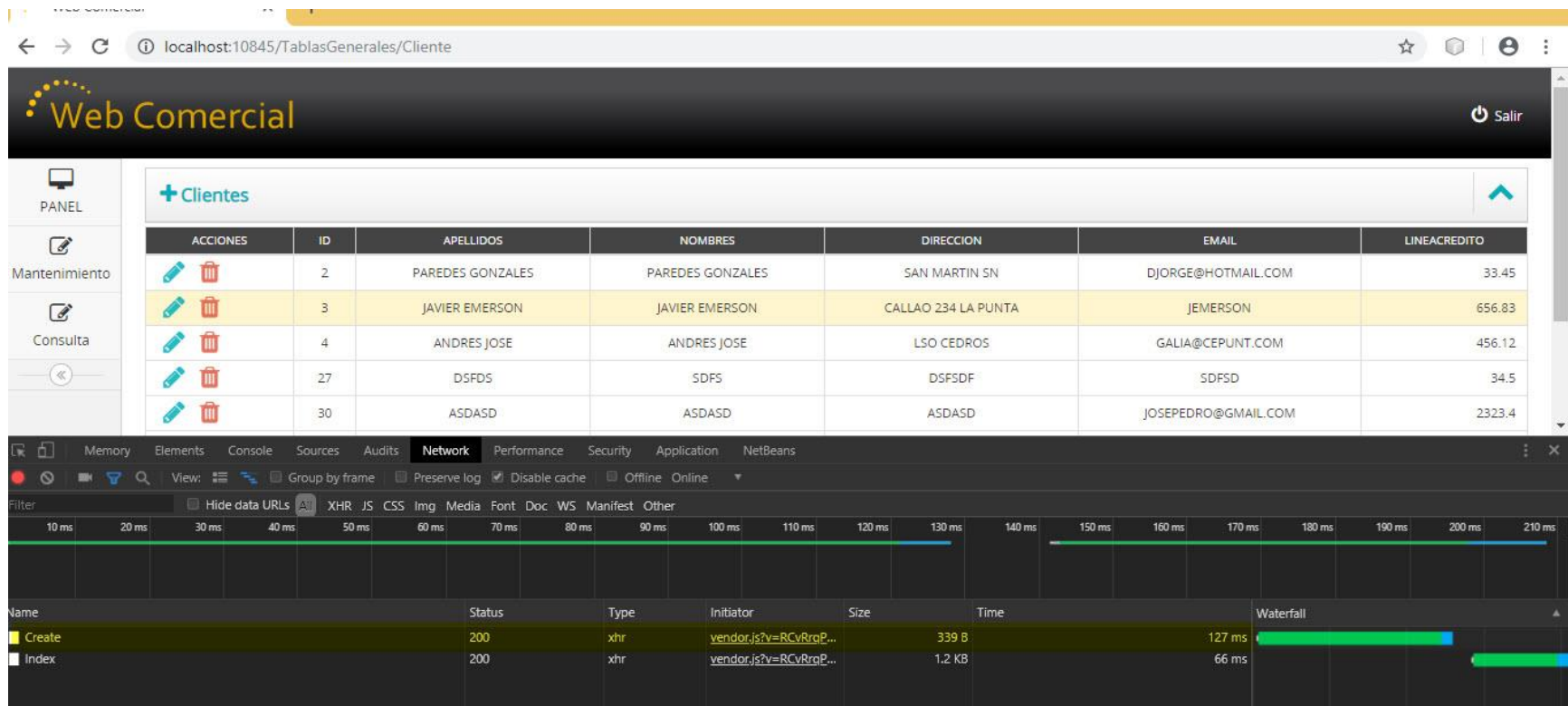


Figura 17. Tiempo respuesta Agregar Cliente – Framework Angular – Google Chrome. Fuente elaboración propia

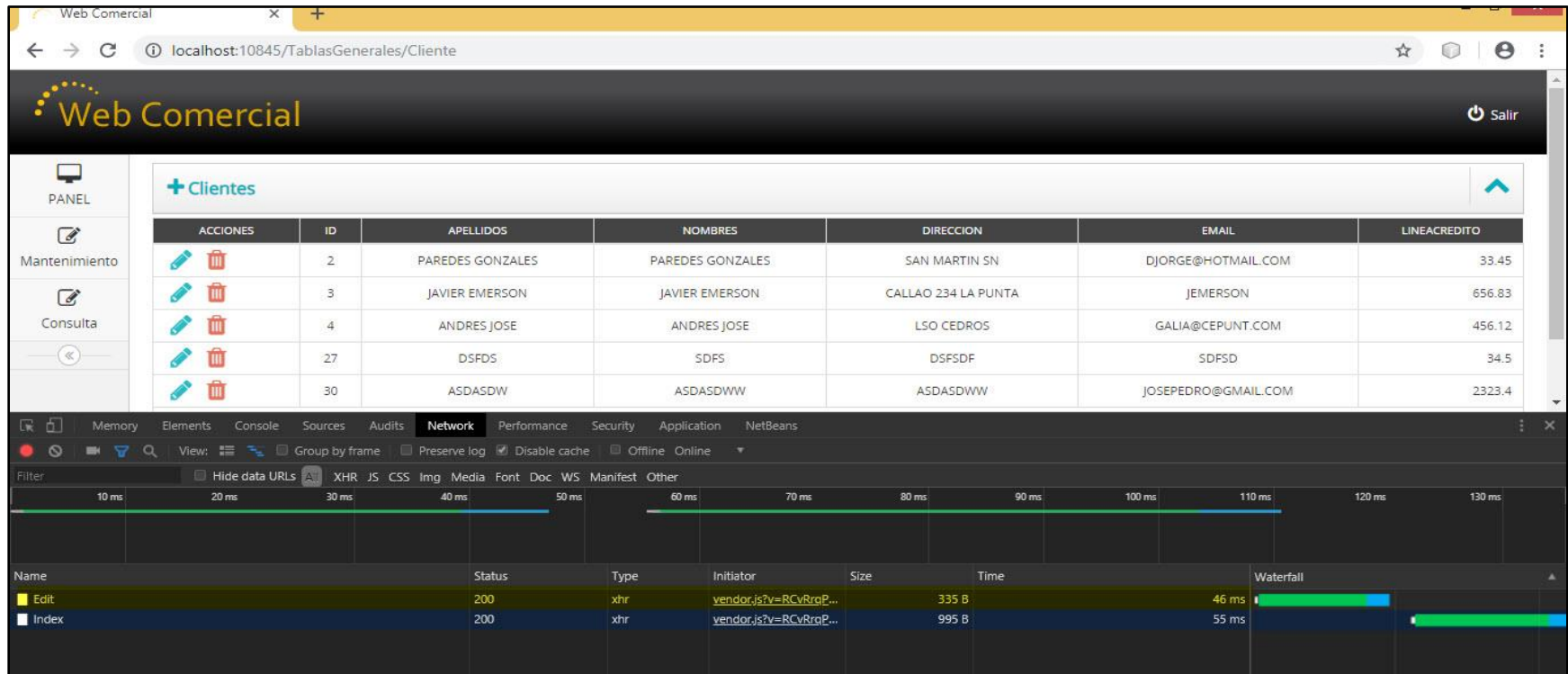


Figura 18. Tiempo respuesta Editar Cliente – Framework Angular – Google Chrome. Fuente elaboración propia

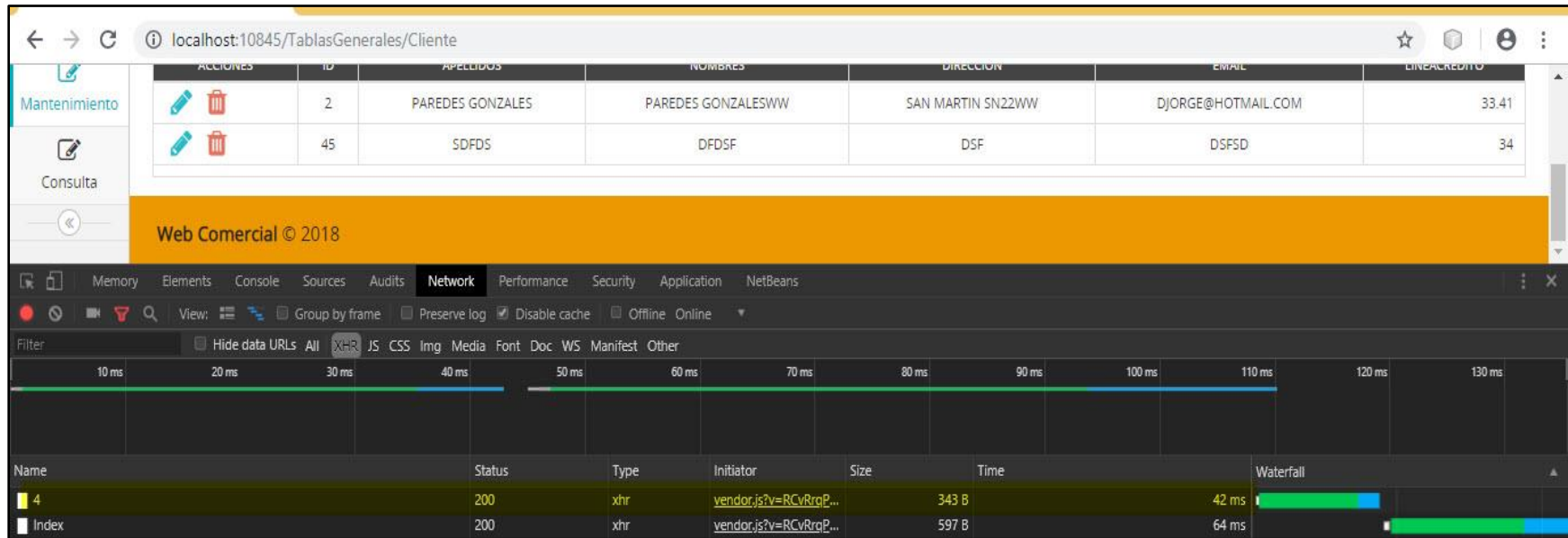


Figura 19. Tiempo respuesta Eliminar Cliente – Framework Angular – Google Chrome. Fuente elaboración propia

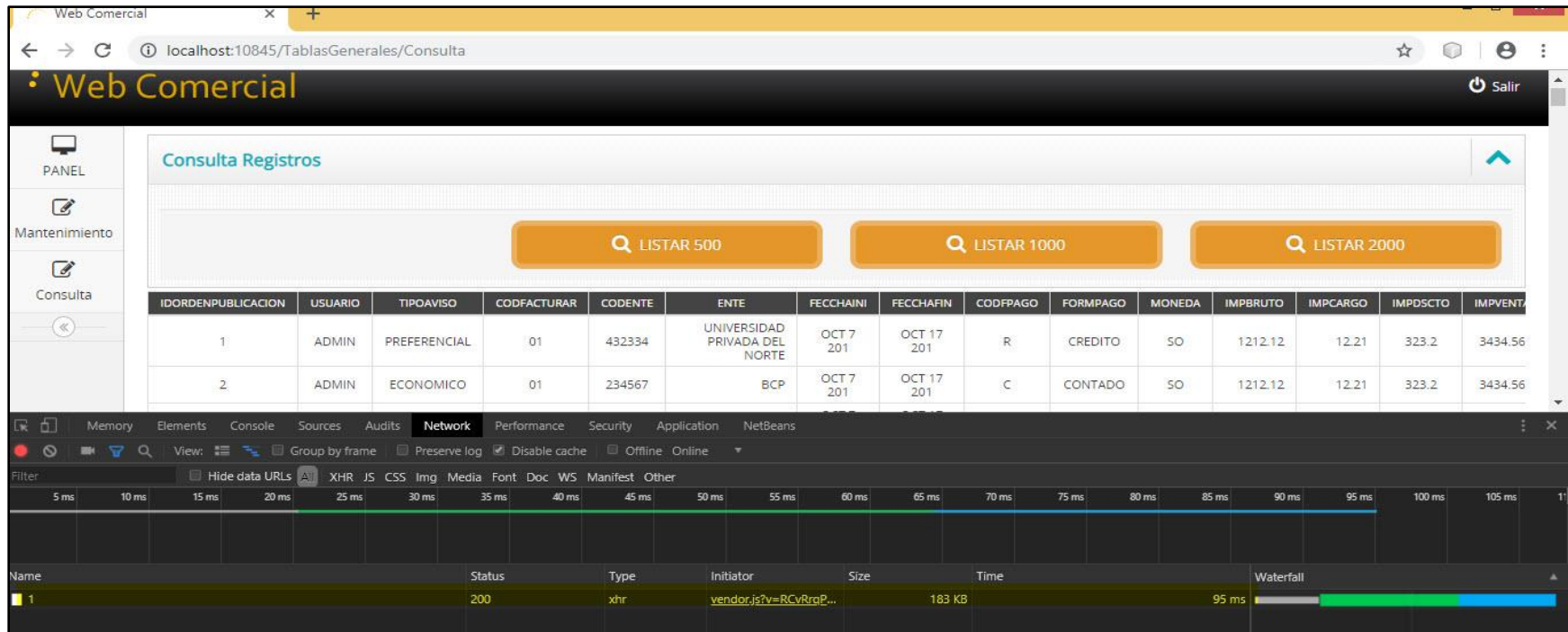


Figura 20. Tiempo respuesta Listar 500 – Framework Angular – Google Chrome. Fuente elaboración propia

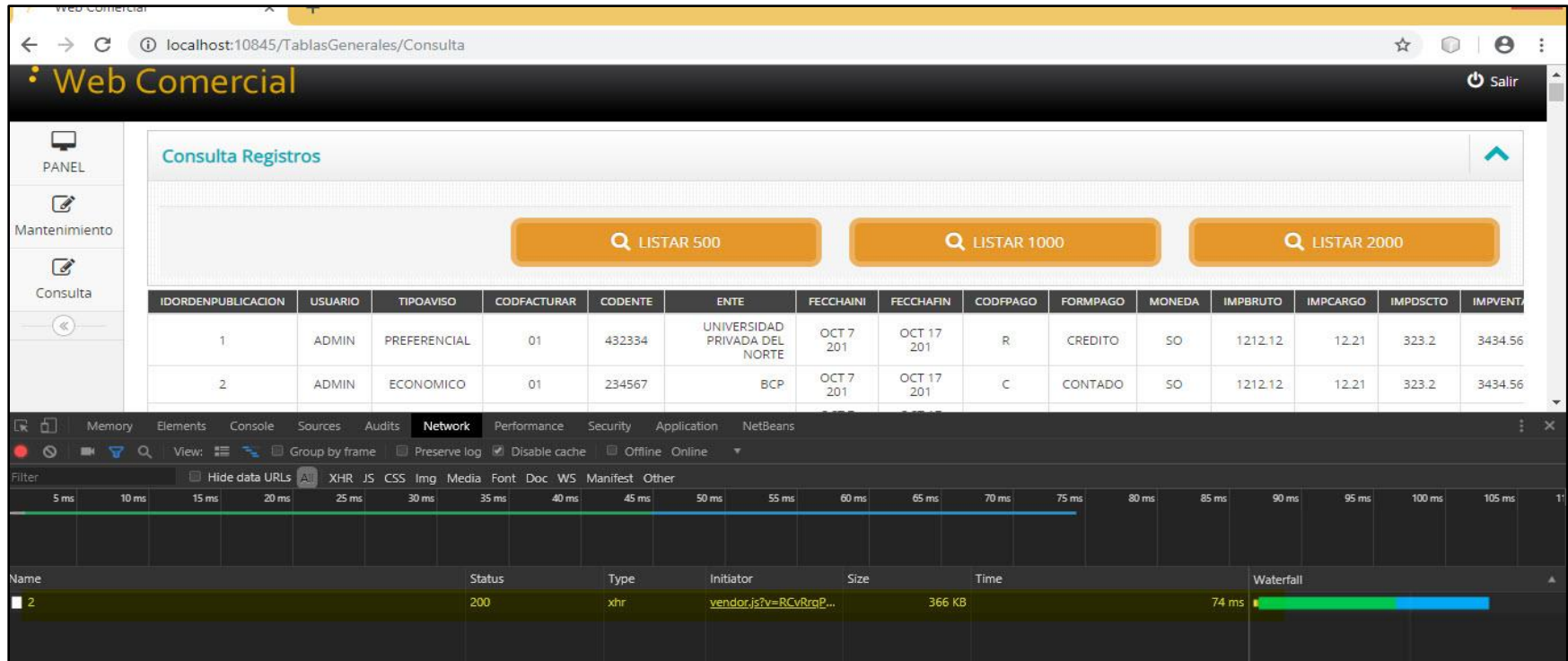


Figura 21. Tiempo respuesta Listar 1000 – Framework Angular – Google Chrome. Fuente elaboración propia



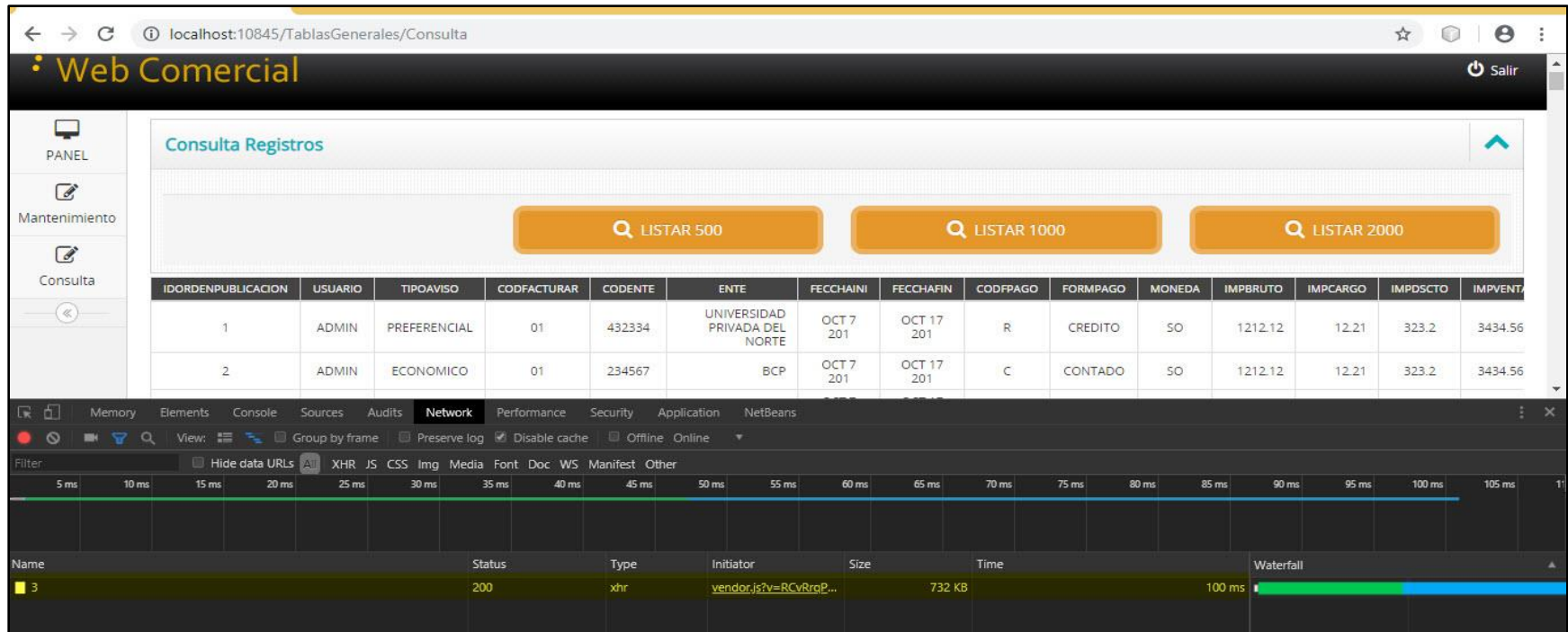


Figura 22. Tiempo respuesta Listar 2000 – Framework Angular – Google Chrome. Fuente elaboración propia

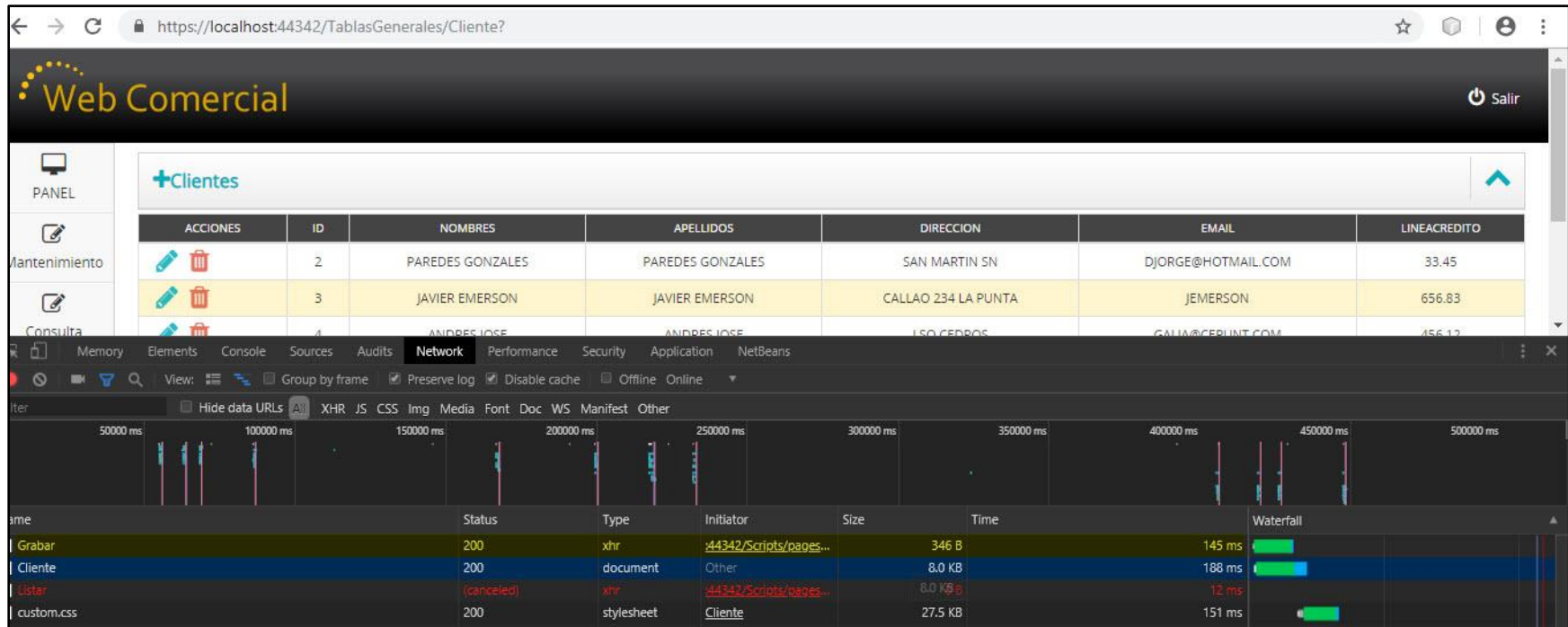


Figura 23. Tiempo respuesta Agregar Cliente – Framework JavaScript Nativo – Google Chrome. Fuente elaboración propia

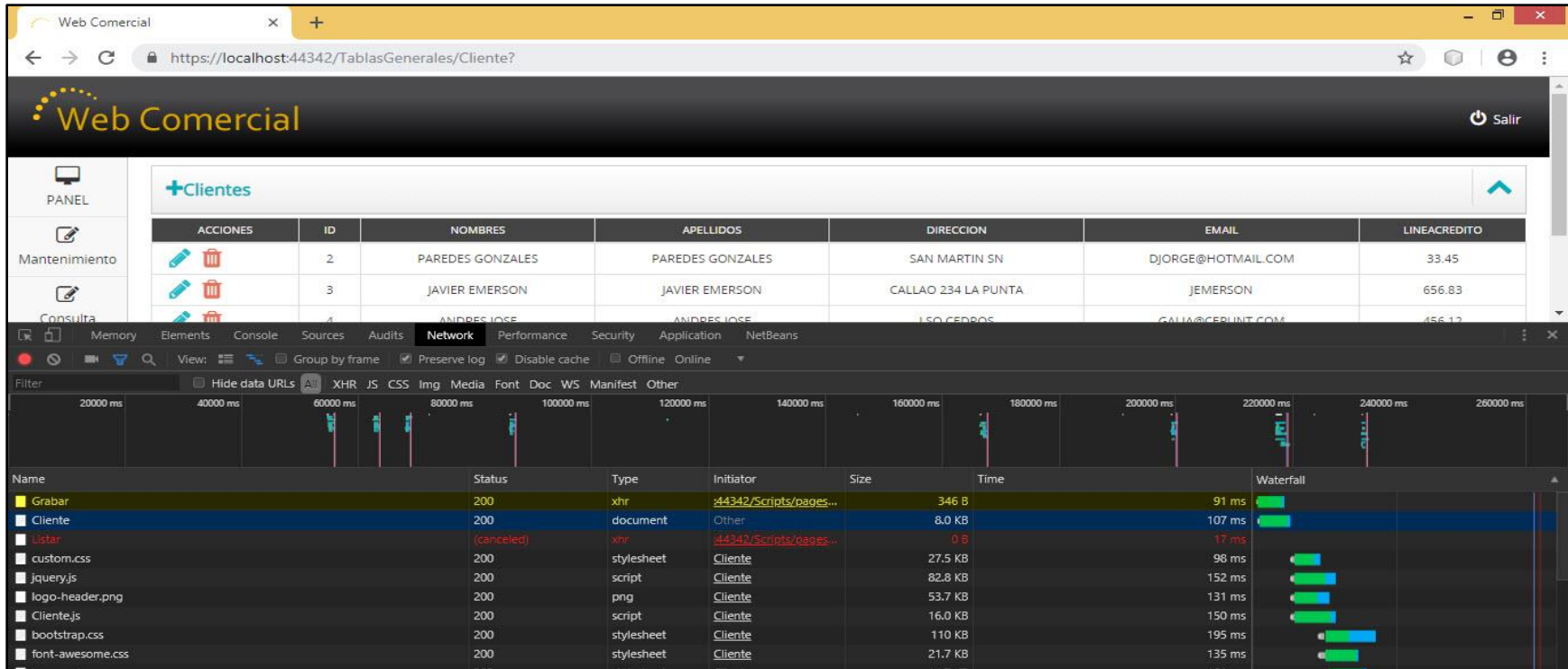


Figura 24. Tiempo respuesta Editar Cliente – Framework JavaScript Nativo – Google Chrome. Fuente elaboración propia

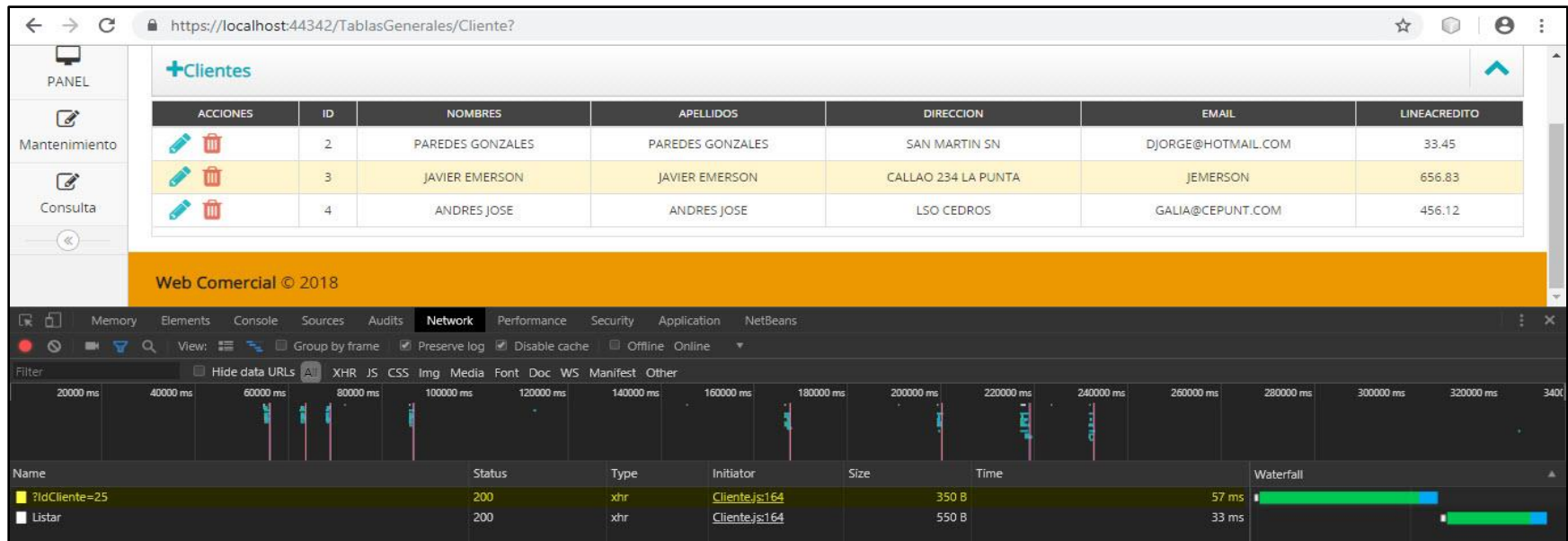


Figura 25. Tiempo respuesta Eliminar Cliente – Framework JavaScript Nativo – Google Chrome. Fuente elaboración propia

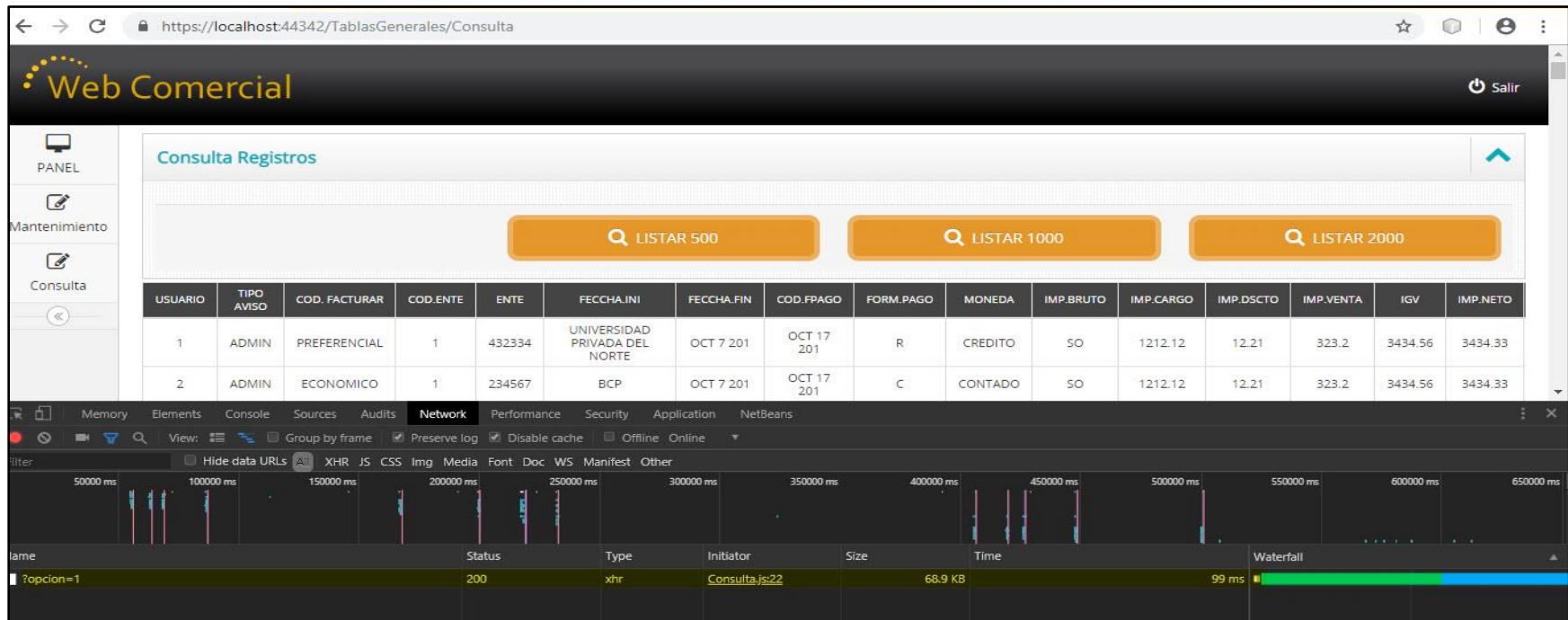


Figura 26. Tiempo respuesta Listar 500 – Framework JavaScript Nativo – Google Chrome. Fuente elaboración propia

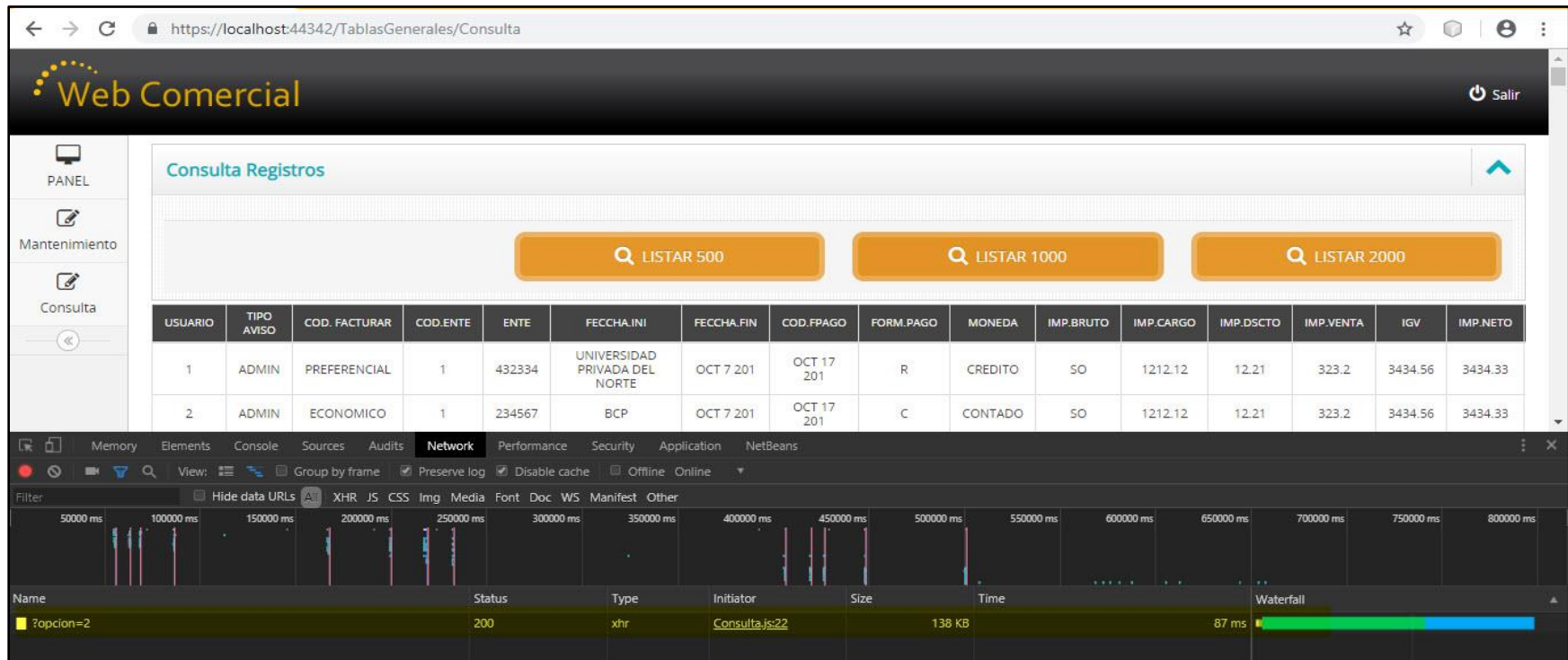


Figura 27. Tiempo respuesta Listar 1000 – Framework JavaScript Nativo – Google Chrome. Fuente elaboración propia

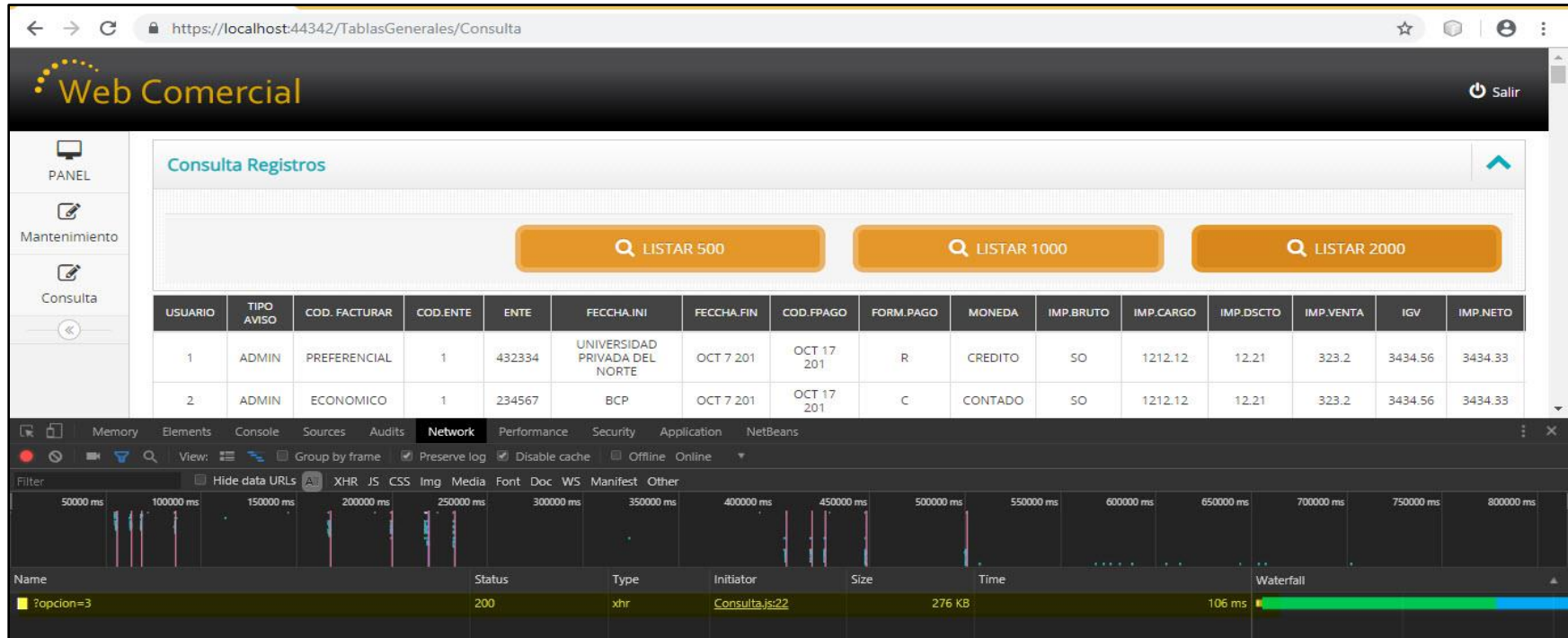


Figura 28. Tiempo respuesta Listar 2000 – Framework JavaScript Nativo – Google Chrome. Fuente elaboración propia

**Anexo 05 – Resultados de Medición de Tiempos de Respuesta Angular – JavaScript Nativo Mozilla**

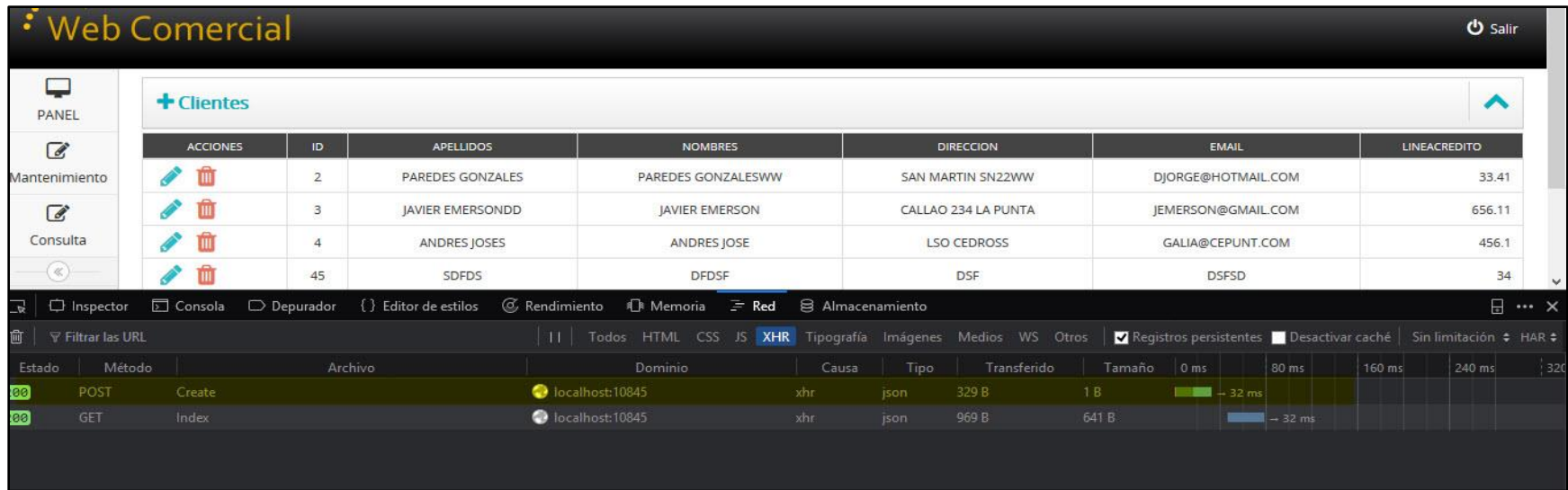


Figura 29. Tiempo respuesta Agregar Cliente – Framework Angular – Mozilla. Fuente elaboración propia



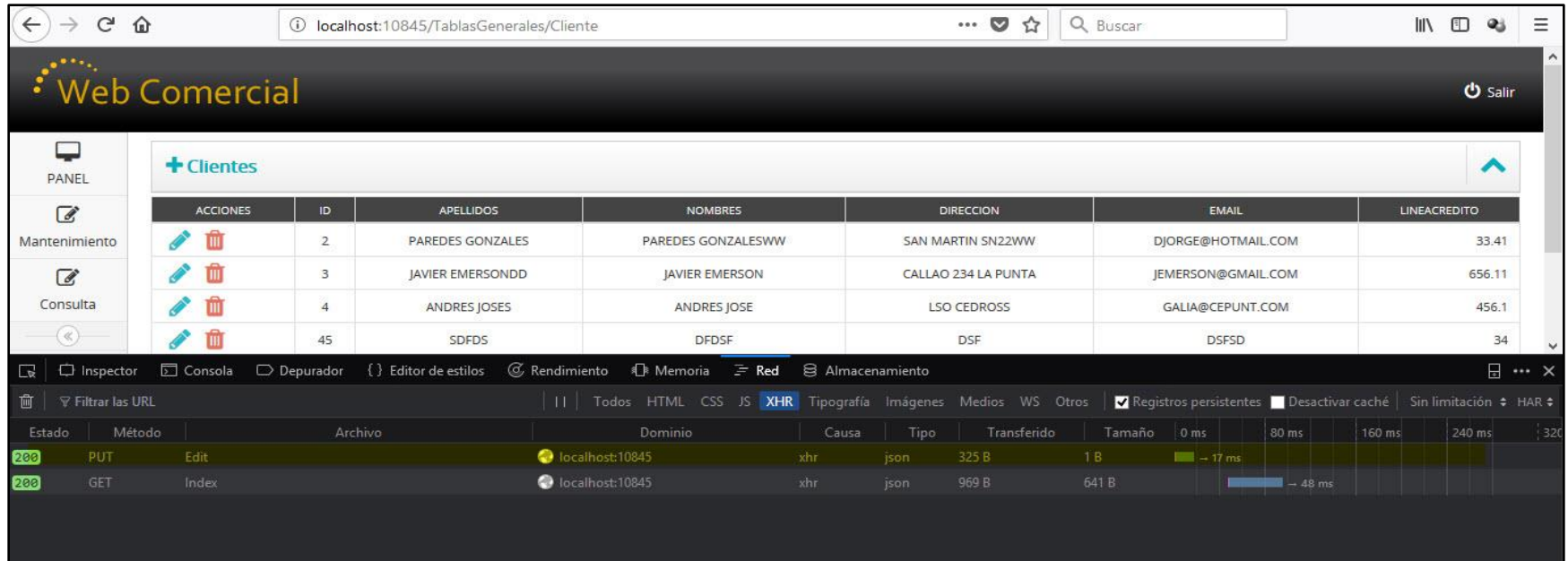
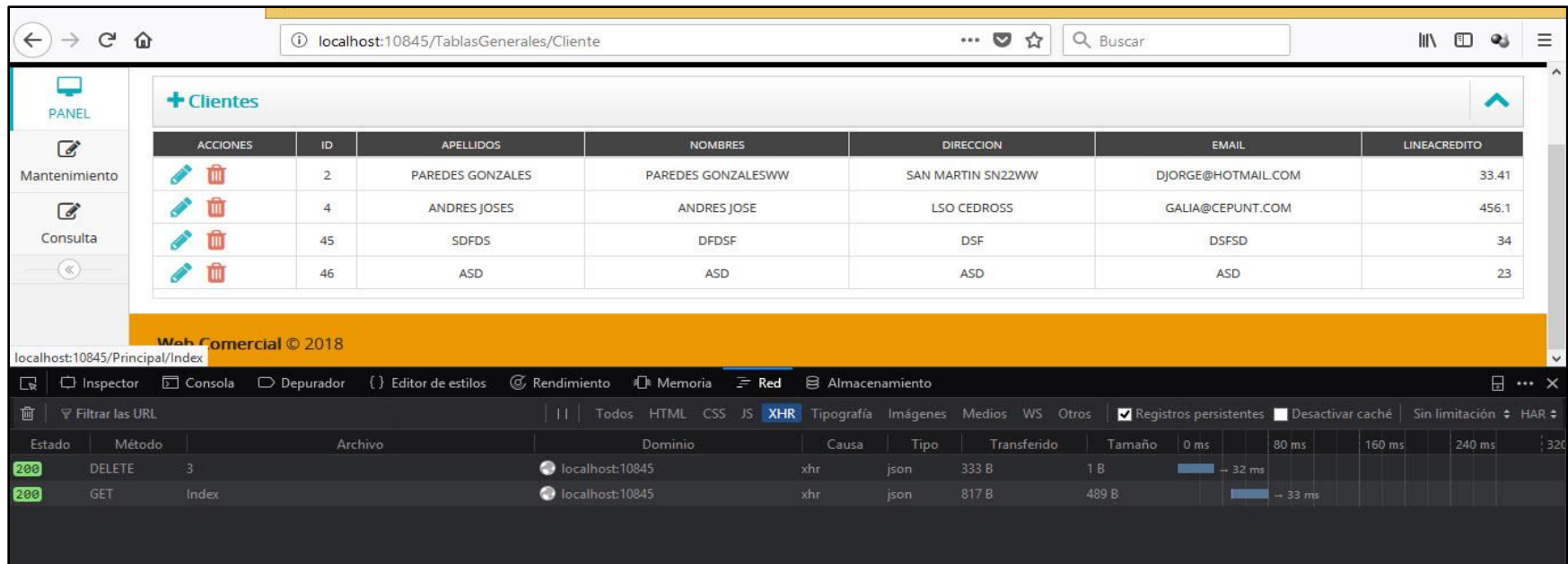


Figura 30. Tiempo respuesta Editar Cliente – Framework Angular – Mozilla. Fuente elaboración propia



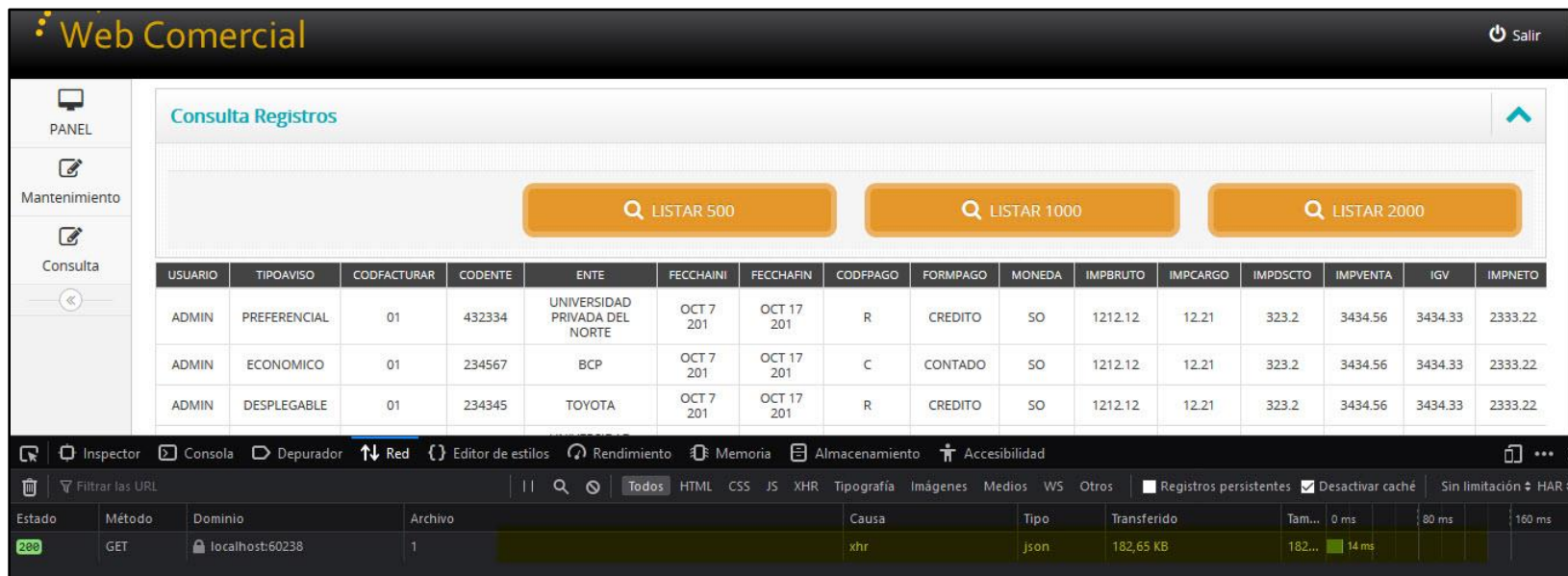
The screenshot shows a web application interface with a table of clients and a network inspector. The table has the following data:

ACCIONES	ID	APELLIDOS	NOMBRES	DIRECCION	EMAIL	LINEACREDITO
	2	PAREDES GONZALES	PAREDES GONZALESWW	SAN MARTIN SN22WW	DJORGE@HOTMAIL.COM	33.41
	4	ANDRES JOSES	ANDRES JOSE	LSO CEDROSS	GALIA@CEPUNT.COM	456.1
	45	SDFDS	DFDSF	DSF	DSFSD	34
	46	ASD	ASD	ASD	ASD	23

The network inspector shows the following request details:

Estado	Método	Archivo	Dominio	Causa	Tipo	Transferido	Tamaño	0 ms	80 ms	160 ms	240 ms	320 ms
200	DELETE	3	localhost:10845	xhr	json	333 B	1 B	→	32 ms			
200	GET	Index	localhost:10845	xhr	json	817 B	489 B	→	33 ms			

Figura 31. Tiempo respuesta Eliminar Cliente – Framework Angular – Mozilla. Fuente elaboración propia

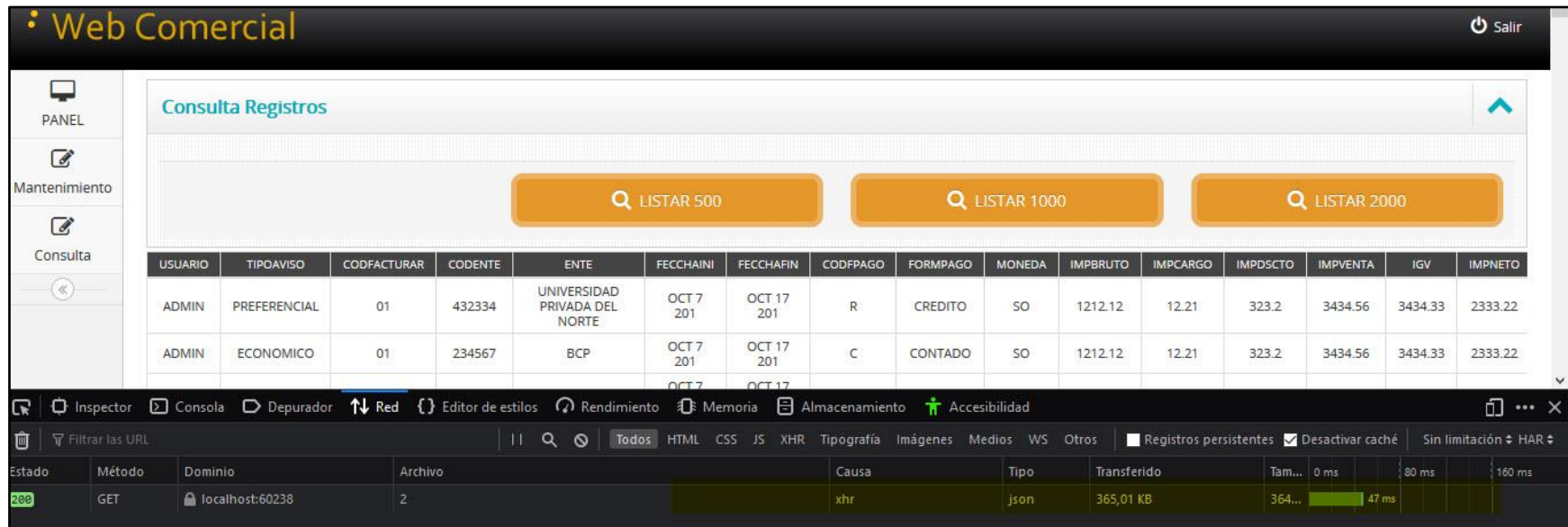


The screenshot shows a web application titled 'Web Comercial' with a sidebar containing 'PANEL', 'Mantenimiento', and 'Consulta'. The main content area is 'Consulta Registros', featuring three orange buttons: 'LISTAR 500', 'LISTAR 1000', and 'LISTAR 2000'. Below these buttons is a table with the following data:

USUARIO	TIPOAVISO	CODFACTURAR	CODENTE	ENTE	FECCHAINI	FECCHAFIN	CODFPAGO	FORMPAGO	MONEDA	IMPBRUTO	IMPCARGO	IMPDSCTO	IMPVENTA	IGV	IMPNETO
ADMIN	PREFERENCIAL	01	432334	UNIVERSIDAD PRIVADA DEL NORTE	OCT 7 201	OCT 17 201	R	CREDITO	SO	1212.12	12.21	323.2	3434.56	3434.33	2333.22
ADMIN	ECONOMICO	01	234567	BCP	OCT 7 201	OCT 17 201	C	CONTADO	SO	1212.12	12.21	323.2	3434.56	3434.33	2333.22
ADMIN	DESPLEGABLE	01	234345	TOYOTA	OCT 7 201	OCT 17 201	R	CREDITO	SO	1212.12	12.21	323.2	3434.56	3434.33	2333.22

At the bottom, the browser's developer console shows a network log entry for a GET request to localhost:60238, with a response time of 14 ms and a size of 182.65 KB.

Figura 32. Tiempo respuesta Listar 500 – Framework Angular – Mozilla. Fuente elaboración propia

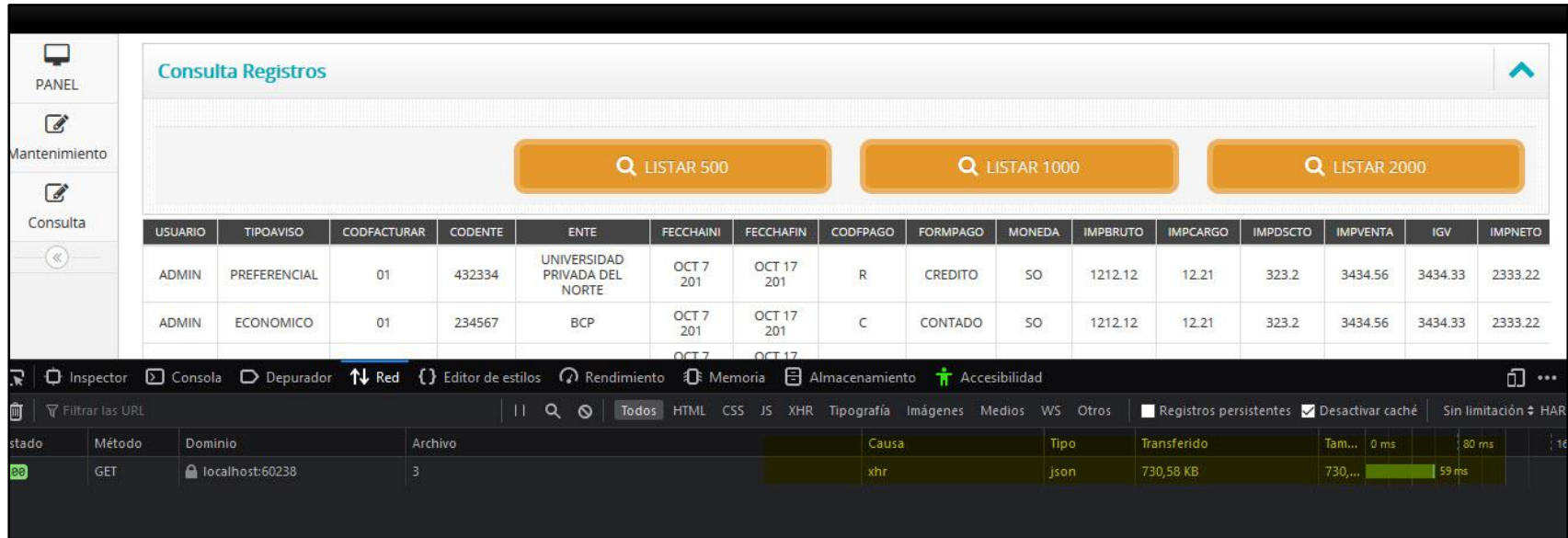


The screenshot shows a web application titled 'Web Comercial' with a sidebar containing 'PANEL', 'Mantenimiento', and 'Consulta'. The main content area is titled 'Consulta Registros' and features three buttons: 'LISTAR 500', 'LISTAR 1000', and 'LISTAR 2000'. Below these buttons is a table with the following data:

USUARIO	TIPOAVISO	CODFACTURAR	CODENTE	ENTE	FECCHAINI	FECCHAFIN	CODFPAGO	FORMPAGO	MONEDA	IMPBRUTO	IMPCARGO	IMPDSTO	IMPVENTA	IGV	IMPNETO
ADMIN	PREFERENCIAL	01	432334	UNIVERSIDAD PRIVADA DEL NORTE	OCT 7 201	OCT 17 201	R	CREDITO	SO	1212.12	12.21	323.2	3434.56	3434.33	2333.22
ADMIN	ECONOMICO	01	234567	BCP	OCT 7 201	OCT 17 201	C	CONTADO	SO	1212.12	12.21	323.2	3434.56	3434.33	2333.22

At the bottom, the browser's developer tools are open, showing a network log with a 200 status, GET method, and a response time of 47 ms for an XHR request.

Figura 33. Tiempo respuesta Listar 1000 – Framework Angular – Mozilla. Fuente elaboración propia



The screenshot displays a web application interface with a sidebar on the left containing navigation options: PANEL, Mantenimiento, and Consulta. The main content area is titled 'Consulta Registros' and features three orange buttons labeled 'LISTAR 500', 'LISTAR 1000', and 'LISTAR 2000'. Below these buttons is a table with the following data:

USUARIO	TIPOAVISO	CODFACTURAR	CODENTE	ENTE	FECCHAINI	FECCHAFIN	CODFPAGO	FORMPAGO	MONEDA	IMPBRUTO	IMPCARGO	IMPDSCTO	IMPVENTA	IGV	IMPNETO
ADMIN	PREFERENCIAL	01	432334	UNIVERSIDAD PRIVADA DEL NORTE	OCT 7 201	OCT 17 201	R	CREDITO	SO	1212.12	12.21	323.2	3434.56	3434.33	2333.22
ADMIN	ECONOMICO	01	234567	BCP	OCT 7 201	OCT 17 201	C	CONTADO	SO	1212.12	12.21	323.2	3434.56	3434.33	2333.22

At the bottom of the screenshot, the browser's developer tools are open, showing the 'Network' tab. A request is selected, and the 'Performance' sub-tab is active, displaying a green bar representing the response time of 59 ms. The 'Request' tab shows the request details: GET method, localhost:60238 domain, and a 730,58 KB response.

Figura 34. Tiempo respuesta Listar 2000 – Framework Angular – Mozilla. Fuente elaboración propia

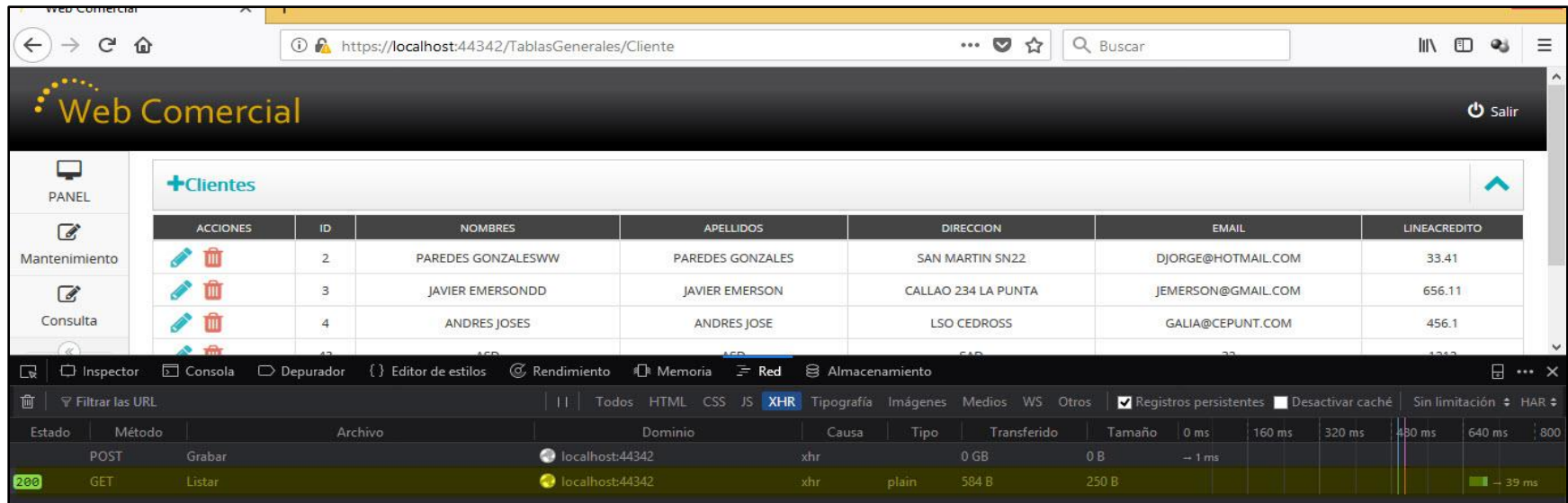


Figura 35. Tiempo respuesta Agregar Cliente – Framework JavaScript Nativo – Mozilla. Fuente elaboración propia

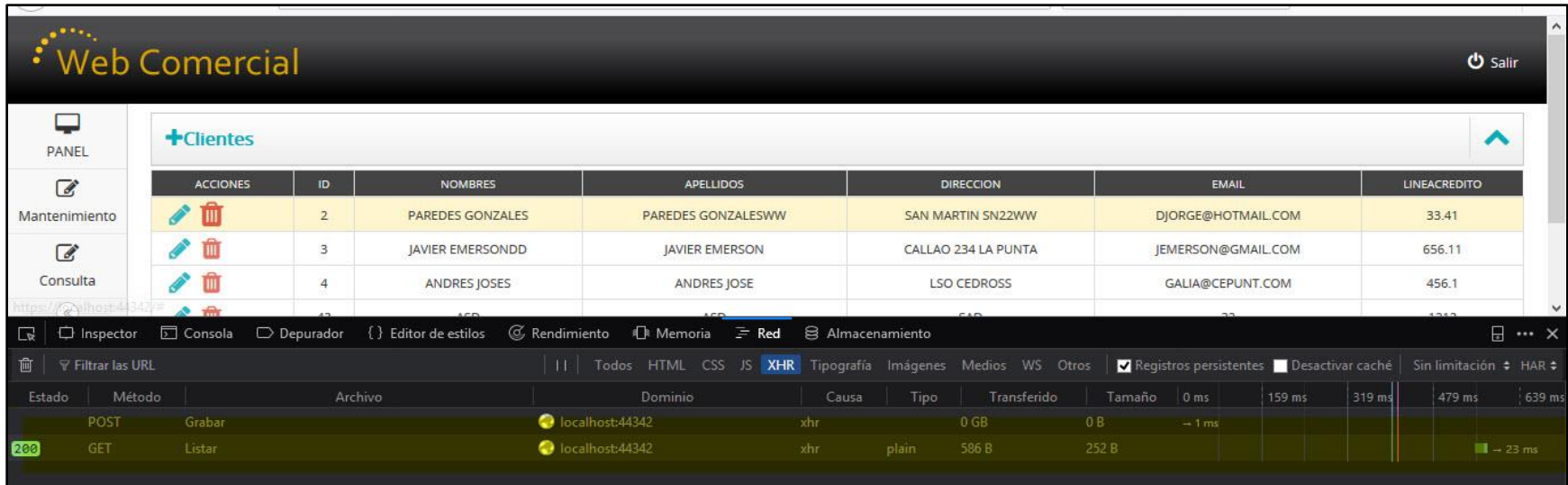


Figura 36. Tiempo respuesta Editar Cliente – Framework JavaScript Nativo – Mozilla. Fuente elaboración propia

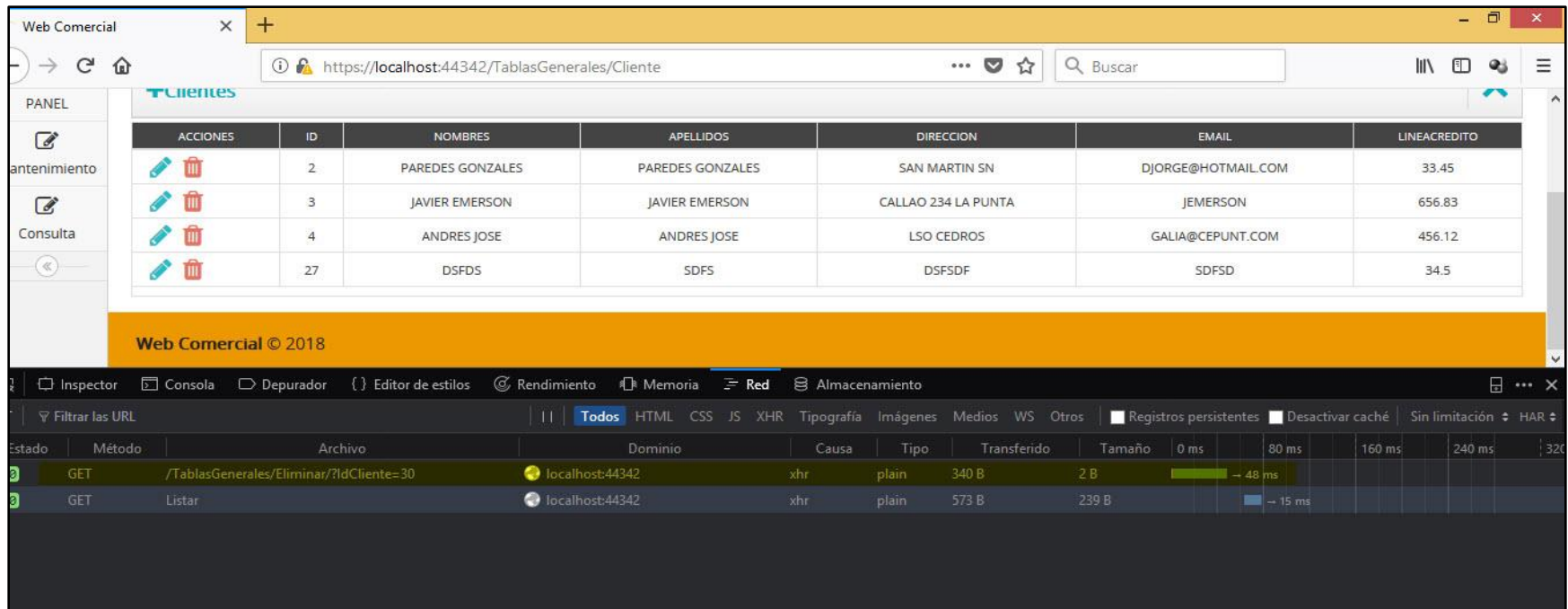


Figura 37. Tiempo respuesta Eliminar Cliente – Framework JavaScript Nativo – Mozilla. Fuente elaboración propia



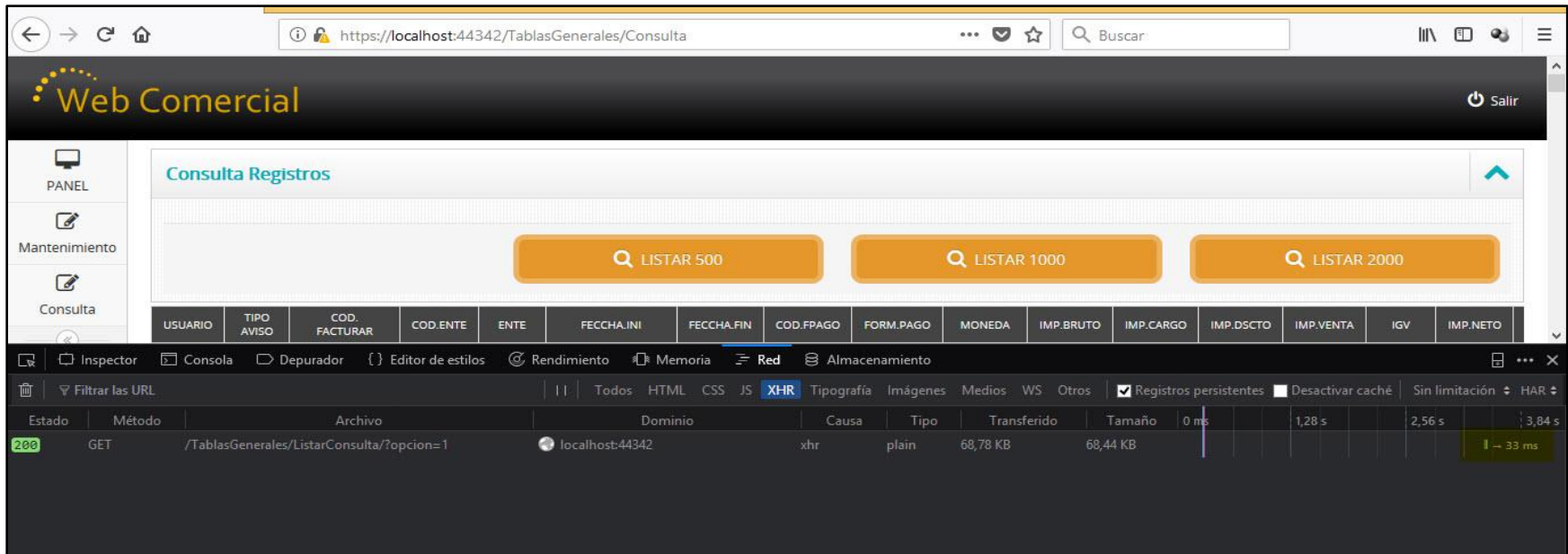


Figura 38. Tiempo respuesta Listar 500 – Framework JavaScript Nativo – Mozilla. Fuente elaboración propia

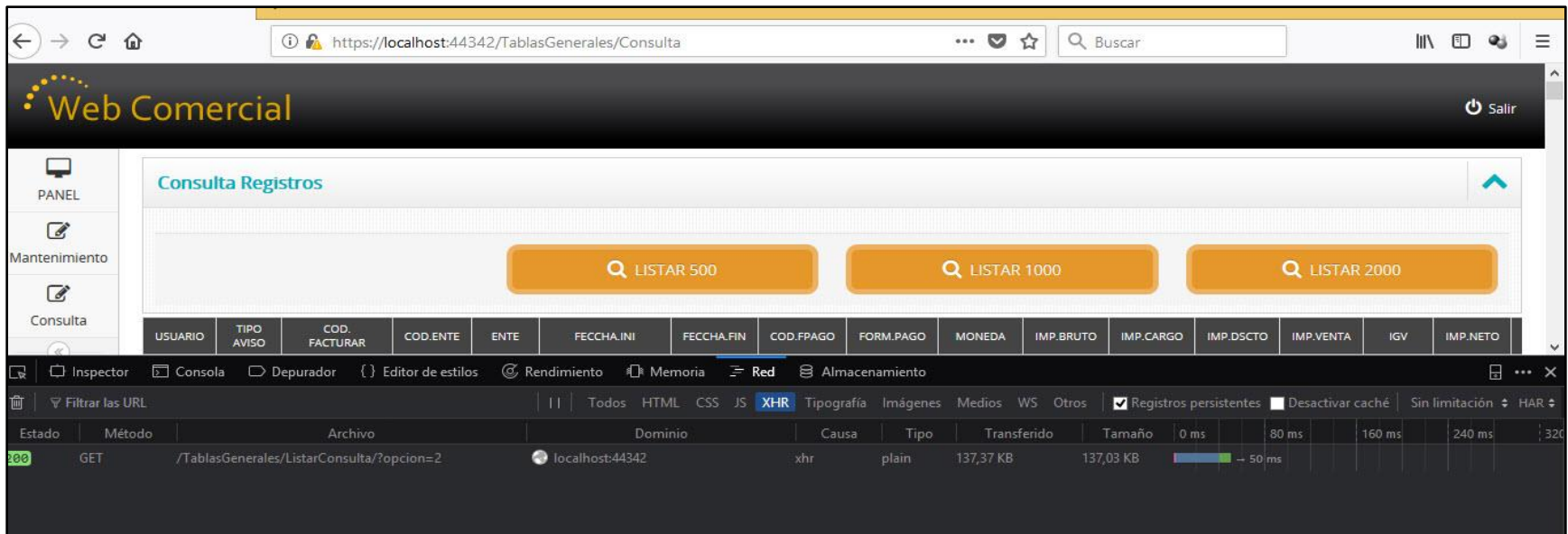


Figura 39. Tiempo respuesta Listar 1000 – Framework JavaScript Nativo – Mozilla. Fuente elaboración propia

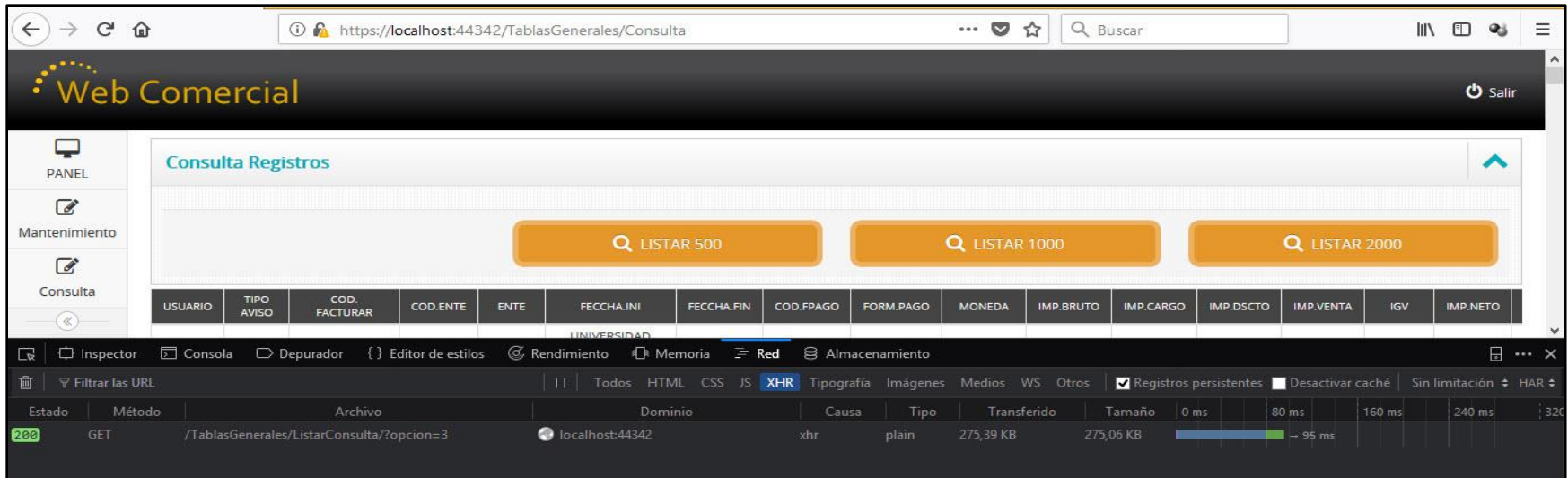


Figura 40. Tiempo respuesta Listar 2000 – Framework JavaScript Nativo – Mozilla. Fuente elaboración propia


## Anexo 06 – Carta de compromiso de Expertos

**EXPERTO FRAMEWORK JAVASCRIPT**

**Nombre:** ING. MIGUEL ANASTACIO VELASQUEZ

**Cargo:** JEFE DE TECNOLOGIA

Me comprometo a evaluar con completa ética y honestidad como aporte para el presente trabajo de investigación “Análisis de los Frameworks JavaScript Nativo y Angular en la incidencia del tiempo de respuesta en una Web MVC en el sector comercial”, donde emitiré mi puntuación según mi experiencia para los fines esperados.

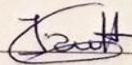
  
Firma

**EXPERTO FRAMEWORK JAVASCRIPT**

**Nombre: ING. JAMES HUIZA PEREYRA**

**Cargo: JEFE DE IMPLEMENTACION**

Me comprometo a evaluar con completa ética y honestidad como aporte para el presente trabajo de investigación "Análisis de los Frameworks JavaScript Nativo y Angular en la incidencia del tiempo de respuesta en una Web MVC en el sector comercial", donde emitiré mi puntuación según mi experiencia para los fines esperados.



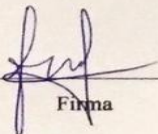
Firma

**EXPERTO FRAMEWORK JAVASCRIPT**

**Nombre: ING. JAVIER AZABACHE ASMAT**

**Cargo: JEFE DE PROYECTOS**

Me comprometo a evaluar con completa ética y honestidad como aporte para el presente trabajo de investigación "Análisis de los Frameworks JavaScript Nativo y Angular en la incidencia del tiempo de respuesta en una Web MVC en el sector comercial", donde emitiré mi puntuación según mi experiencia para los fines esperados.

  
Firma

**Anexo 07 – Tabla de Medición de Expertos**

ENCUESTA JUICIO DE EXPERTO							
EXPERTO		MIGUEL ANASTACIO VELASQUEZ					
FRAMEWORK		ANGULAR					
En este documento se recopilará información fidedigna que cada experto a juicio personal emitirá de cada Framework en estudio. La puntuación será asignada según los valores a continuación.							
Excelente	10	Muy bueno	7 - 9	Bueno	4 - 6	Regular	1 - 3
INDICADOR DEL FRAMEWORK							
<b>A</b>	<b>CURVA APRENDIZAJE</b>			<b>Nº DIAS</b>	<b>PUNTUACION</b>		
	Revisión de teoría y contenidos			5	9		
	Aplicar teoría en ejemplos prácticos			5	9		
	<b>PROMEDIO PUNTUACION</b>					9	
<b>B</b>	<b>TIEMPO DE DESARROLLO</b>			<b>Nº DIAS</b>	<b>PUNTUACION</b>		
	Diseño y creación prototipo para uso del Framework			6	9		
	Programar y dar funcionalidad al prototipo			6	10		
	<b>PROMEDIO PUNTUACION</b>					95	

ENCUESTA JUICIO DE EXPERTO							
EXPERTO		MIGUEL ANASTACIO VELASQUEZ					
FRAMEWORK		JAVASCRIPT NATIVO					
En este documento se recopilará información fidedigna que cada experto a juicio personal emitirá de cada Framework en estudio. La puntuación será asignada según los valores a continuación.							
Excelente	10	Muy bueno	7 - 9	Bueno	4 - 6	Regular	1 - 3
INDICADOR DEL FRAMEWORK							
<b>A</b>	CURVA APRENDIZAJE			Nº DIAS	PUNTUACION		
	Revisión de teoría y contenidos			10	7		
	Aplicar teoría en ejemplos prácticos			10	7		
	PROMEDIO PUNTUACION					7	
<b>B</b>	TIEMPO DE DESARROLLO			Nº DIAS	PUNTUACION		
	Diseño y creación prototipo para uso del Framework			10	6		
	Programar y dar funcionalidad al prototipo			10	6		
	PROMEDIO PUNTUACION					6	



ENCUESTA JUICIO DE EXPERTO							
EXPERTO		JAMES HUIZA PEREYRA					
FRAMEWORK		ANGULAR					
En este documento se recopilará información fidedigna que cada experto a juicio personal emitirá de cada Framework en estudio. La puntuación será asignada según los valores a continuación.							
Excelente	10	Muy bueno	7 - 9	Bueno	4 - 6	Regular	1 - 3
INDICADOR DEL FRAMEWORK							
<b>A</b>	<b>CURVA APRENDIZAJE</b>			<b>Nº DIAS</b>	<b>PUNTUACION</b>		
	Revisión de teoría y contenidos			6	10		
	Aplicar teoría en ejemplos prácticos			6	10		
	<b>PROMEDIO PUNTUACION</b>					10	
<b>B</b>	<b>TIEMPO DE DESARROLLO</b>			<b>Nº DIAS</b>	<b>PUNTUACION</b>		
	Diseño y creación prototipo para uso del Framework			6	9		
	Programar y dar funcionalidad al prototipo			6	9		
	<b>PROMEDIO PUNTUACION</b>					9	

— 1

ENCUESTA JUICIO DE EXPERTO							
EXPERTO		JAMES HUIZA PEREYRA					
FRAMEWORK		JAVASCRIPT NATIVO					
En este documento se recopilará información fidedigna que cada experto a juicio personal emitirá de cada Framework en estudio. La puntuación será asignada según los valores a continuación.							
Excelente	10	Muy bueno	7 - 9	Bueno	4 - 6	Regular	1 - 3
INDICADOR DEL FRAMEWORK							
<b>A</b>	<b>CURVA APRENDIZAJE</b>				<b>Nº DIAS</b>	<b>PUNTUACION</b>	
	Revisión de teoría y contenidos				8	8	
	Aplicar teoría en ejemplos prácticos				7	8	
	<b>PROMEDIO PUNTUACION</b>					8	
<b>B</b>	<b>TIEMPO DE DESARROLLO</b>				<b>Nº DIAS</b>	<b>PUNTUACION</b>	
	Diseño y creación prototipo para uso del Framework				10	7	
	Programar y dar funcionalidad al prototipo				9	7	
	<b>PROMEDIO PUNTUACION</b>					7	

— 2

ENCUESTA JUICIO DE EXPERTO							
EXPERTO		JAVIER AZABACHE ASMAT					
FRAMEWORK		ANGULAR					
En este documento se recopilará información fidedigna que cada experto a juicio personal emitirá de cada Framework en estudio. La puntuación será asignada según los valores a continuación.							
Excelente	10	Muy bueno	7 - 9	Bueno	4 - 6	Regular	1 - 3
INDICADOR DEL FRAMEWORK							
<b>A</b>	<b>CURVA APRENDIZAJE</b>				<b>Nº DIAS</b>	<b>PUNTUACION</b>	
	Revisión de teoría y contenidos				5	9	
	Aplicar teoría en ejemplos prácticos				6	9	
	<b>PROMEDIO PUNTUACION</b>					9	
<b>B</b>	<b>TIEMPO DE DESARROLLO</b>				<b>Nº DIAS</b>	<b>PUNTUACION</b>	
	Diseño y creación prototipo para uso del Framework				5	9	
	Programar y dar funcionalidad al prototipo				6	9	
	<b>PROMEDIO PUNTUACION</b>					9	

P2

ENCUESTA JUICIO DE EXPERTO							
EXPERTO		JAVIER AZABACHE ASMAT					
FRAMEWORK		JAVASCRIPT NATIVO					
En este documento se recopilará información fidedigna que cada experto a juicio personal emitirá de cada Framework en estudio. La puntuación será asignada según los valores a continuación.							
Excelente	10	Muy bueno	7 - 9	Bueno	4 - 6	Regular	1 - 3
INDICADOR DEL FRAMEWORK							
A	CURVA APRENDIZAJE			Nº DIAS	PUNTUACION		
	Revisión de teoría y contenidos			8	7		
	Aplicar teoría en ejemplos prácticos			7	7		
PROMEDIO PUNTUACION						7	
B	TIEMPO DE DESARROLLO			Nº DIAS	PUNTUACION		
	Diseño y creación prototipo para uso del Framework			7	7		
	Programar y dar funcionalidad al prototipo			8	7		
PROMEDIO PUNTUACION						7	

— P1

## Anexo 08 – Documentación del Framework Angular

```
GP_MVC_Angular JavaScript Content Files
1   var app;
2   var _ID;
3   (function () {
4       app = angular.module("crudModule", []);
5       _ID = 0;
6   })();
7
```

Figura 41. Angular-module

```
@model UsuariosBE
@{
    string cPathApplication = Url.Content("~/");
    ViewBag.Title = "Web Comercial";
}
using BusinessEntity;
<!DOCTYPE html>
<html lang="es-ES" xmlns:og="http://ogp.me/ns#" xmlns:fb="http://ogp.me/ns/fb#" ng-app="crudModule">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta name="language" content="es">
    <title>@ViewBag.Title</title>
    <meta name="title" content="Web Comercial">

    <base href="~/>
    <link rel="shortcut icon" href="@Url.Content("~/Content/static/gp/favicon.ico")" type="image/x-icon">
    <link rel="stylesheet" href="@Url.Content("~/Content/static/gp/css/custom.css")" />
    <script type="text/javascript" src="@Url.Content("~/Content/static/gp/js/jquery.js")"></script>
</head>
<body class="no-skin" ng-controller="crudController" >
```

Figura 42. Angular configuración

```
app.service('crudService', function ($http) {  
  
    var baseURL = 'http://localhost:60238/';  
  
    this.getAllOrdenPublicacion = function (opcion) {  
        return $http.get(baseURL + "api/OrdenPublicacion/Consulta/"+opcion);  
    };  
  
});
```

Figura 43. Service consulta

```
app.service('crudService2', function ($http) {  
    var baseURL = 'http://localhost:60238/';  
    this.getClientes = function () {  
        return $http.get(baseURL + "api/Cliente");  
    };  
    this.post = function (Cliente) {  
        var request = $http({  
            method: "post",  
            url: baseURL + "api/Cliente",  
            data: Cliente  
        });  
        return request;  
    };  
    this.get = function (ID) {  
        return $http.get(baseURL + "api/Cliente/" + ID);  
    };  
    this.put = function (ID, Cliente) {  
        var request = $http({  
            method: "put",  
            url: baseURL + "api/Cliente/" + ID,  
            data: Cliente  
        });  
        return request;  
    };  
    this.delete = function (ID) {  
        var request = $http({  
            method: "delete",  
            url: baseURL + "api/Cliente/" + ID  
        });  
        return request;  
    };  
});
```

Figura 44. Service mantenimiento

```

app.controller('crudController', function ($scope, crudService) {
    $scope.opList = null;
    loadRecords();
    function loadRecords() {
        $scope.opList = null;
        var tabla = document.getElementById("tbConsulta");
        if (tabla != null) {
            document.getElementById("tbConsulta").innerHTML = "";
            var loadingDiv = $('#loading-body-all');
            loadingDiv.show();
            var promiseGet = crudService.getAllOrdenPublicacion(0);
            promiseGet.then(function (p1) {
                $scope.opList = p1.data;
                loadingDiv.hide();
            });
            function (errorP1) {
                console.log('failure loading Orden Publicacion', errorP1);
                loadingDiv.hide();
            };
        }
    };
    $scope.getOrdenPublicaciones = function (opcion) {
        $scope.opList = null;
        var loadingDiv = $('#loading-body-all');
        loadingDiv.show();
        document.getElementById("tbConsulta").innerHTML = '';
        var promiseGet = crudService.getAllOrdenPublicacion(opcion);
        promiseGet.then(function (p1) {
            $scope.opList = p1.data;
            var tbConsulta = document.getElementById("tbConsulta");
            for (var i = 0; i < $scope.opList.length; i++) {
                var tr = document.createElement('tr');
                tr.innerHTML = '<td class="vert-align text-center">' + $scope.opList[i].Usuario + '</td>' +

```

Figura 45. Controller consulta 1

```

var promiseGet = crudService.getAllOrdenPublicacion(opcion);
promiseGet.then(function (p1) {
    $scope.opList = p1.data;
    var tbConsulta = document.getElementById("tbConsulta");
    for (var i = 0; i < $scope.opList.length; i++) {
        var tr = document.createElement('tr');
        tr.innerHTML = '<td class="vert-align text-center">' + $scope.opList[i].Usuario + '</td>' +
            '<td class="vert-align text-center">' + $scope.opList[i].Tipoaviso + '</td>' +
            '<td class="vert-align text-center">' + $scope.opList[i].Codfacturar + '</td>' +
            '<td class="vert-align text-center">' + $scope.opList[i].Codente + '</td>' +
            '<td class="vert-align text-center">' + $scope.opList[i].Ente + '</td>' +
            '<td class="vert-align text-center">' + $scope.opList[i].Fecchaini + '</td>' +
            '<td class="vert-align text-center">' + $scope.opList[i].Fecchafin + '</td>' +
            '<td class="vert-align text-center">' + $scope.opList[i].Codfpago + '</td>' +
            '<td class="vert-align text-center">' + $scope.opList[i].Formpago + '</td>' +
            '<td class="vert-align text-center">' + $scope.opList[i].Moneda + '</td>' +
            '<td class="vert-align text-center">' + $scope.opList[i].Impbruto + '</td>' +
            '<td class="vert-align text-center">' + $scope.opList[i].Impcarga + '</td>' +
            '<td class="vert-align text-center">' + $scope.opList[i].Impdscto + '</td>' +
            '<td class="vert-align text-center">' + $scope.opList[i].Impventa + '</td>' +
            '<td class="vert-align text-center">' + $scope.opList[i].Igv + '</td>' +
            '<td class="vert-align text-center">' + $scope.opList[i].Impneto + '</td>' +
            '<td class="vert-align text-center">' + $scope.opList[i].Impnetobase + '</td>' +
            '<td class="vert-align text-center">' + $scope.opList[i].Estado + '</td>';
        tbConsulta.appendChild(tr);
    }
    loadingDiv.hide();
},
function (errorP1) {
    console.log('failure loading Orden Publicacion', errorP1);
    loadingDiv.hide();
});
});

```

Figura 46. Controller consulta 2

```

app.controller('crudController2', function ($scope, crudService2, sharedContext) {
    $scope.cliList = null;
    $scope.listado = false;
    $('#paginaClienteAgregar').hide();
    $('#paginaClienteEditar').hide();
    $('#paginaCliente').show();

    if (parseInt(sharedContext.getData()) === 0) {
        loadRecords();
    }

    function loadRecords() {
        if (!$scope.listado) {
            $scope.cliList = null;
            var tbConsulta = document.getElementById("tbCliente");
            if (tbConsulta != null) {
                document.getElementById("tbCliente").innerHTML = "";
                var loadingDiv = $("#Loading-body-all");
                loadingDiv.show();
                var promiseGet = crudService2.getClientes();
                promiseGet.then(function (pl) {
                    $scope.cliList = pl.data;

                    for (var i = 0; i < $scope.cliList.length; i++) {
                        var tr = document.createElement("tr");
                        tr.innerHTML = "td class=vert-align"> +
                            'div class=action-buttons' +
                            'ca href=javascript: onclick=Edit(' + $scope.cliList[i].ID + ') class=tooltip blue title=Modificar' + 'i class=ace-icon fa fa-pencil bigger-200' + '</i></a' +
                            'ca href=javascript: onclick=Showconfirm(' + $scope.cliList[i].ID + ') class=tooltip-error data-rel=tooltip title=Eliminar' +
                            'span class=red' +
                            'i class=ace-icon fa fa-trash bigger-200' + '</i>' +
                            '</span' +
                            '</a' +
                            '</div></td' +
                            'td class=vert-align text-center' + $scope.cliList[i].ID + '</td' +
                            'td class=vert-align text-center' + $scope.cliList[i].Apellidos + '</td' +
                            'td class=vert-align text-center' + $scope.cliList[i].Nombres + '</td' +
                            'td class=vert-align text-center' + $scope.cliList[i].Direccion + '</td' +
                            'td class=vert-align text-center' + $scope.cliList[i].Email + '</td' +
                            'td class=vert-align text-right' + $scope.cliList[i].LineaCredito + '</td' +
                        tbConsulta.appendChild(tr);
                    }
                    loadingDiv.hide();
                });
            }
        }
    }

    function (errorPl) {
        console.log('failure loading Clientes', errorPl);
        loadingDiv.hide();
    };
    });
}

$scope.edit = function (ID) {
    var promiseGetSingle = crudService2.get(ID);
    promiseGetSingle.then(function (pl) {
        $scope.editMode = true;
        $scope.cliente = pl.data;
        sharedContext.addData(ID);
        $('#paginaCliente').hide();
        $('#paginaClienteAgregar').hide();
        $('#paginaClienteEditar').show();
    });

    function (errorPl) {
        console.log('failure loading Cliente', errorPl);
    };
};

$scope.saveAdd = function () {
    var Cliente = $scope.cliente;
    var promisePost = crudService2.post(Cliente);
    promisePost.then(function (pl) {
        $scope.clear();
        $scope.listado = false;
        $('#paginaCliente').show();
        $('#paginaClienteAgregar').hide();
        $('#paginaClienteEditar').hide();
        sharedContext.addData(0);
        loadRecords();
    }, function (err) {
        console.log("Err" + err);
    });
};

$scope.saveEdit = function () {
    var Cliente = $scope.cliente;
    var promisePut = crudService2.put(Cliente.ID, Cliente);
    promisePut.then(function (pl) {
        $scope.clear();
        $scope.listado = false;
        $('#paginaCliente').show();
        $('#paginaClienteAgregar').hide();
        $('#paginaClienteEditar').hide();
        sharedContext.addData(0);
        loadRecords();
    }, function (err) {
        console.log("Err" + err);
    });
};
}

```

Figura 47. Controller mantenimiento 1

```

        loadingDiv.hide();
    },
    function (errorPl) {
        console.log('failure loading Clientes', errorPl);
        loadingDiv.hide();
    });
    });
}

$scope.edit = function (ID) {
    var promiseGetSingle = crudService2.get(ID);
    promiseGetSingle.then(function (pl) {
        $scope.editMode = true;
        $scope.cliente = pl.data;
        sharedContext.addData(ID);
        $('#paginaCliente').hide();
        $('#paginaClienteAgregar').hide();
        $('#paginaClienteEditar').show();
    });

    function (errorPl) {
        console.log('failure loading Cliente', errorPl);
    };
};

$scope.saveAdd = function () {
    var Cliente = $scope.cliente;
    var promisePost = crudService2.post(Cliente);
    promisePost.then(function (pl) {
        $scope.clear();
        $scope.listado = false;
        $('#paginaCliente').show();
        $('#paginaClienteAgregar').hide();
        $('#paginaClienteEditar').hide();
        sharedContext.addData(0);
        loadRecords();
    }, function (err) {
        console.log("Err" + err);
    });
};

$scope.saveEdit = function () {
    var Cliente = $scope.cliente;
    var promisePut = crudService2.put(Cliente.ID, Cliente);
    promisePut.then(function (pl) {
        $scope.clear();
        $scope.listado = false;
        $('#paginaCliente').show();
        $('#paginaClienteAgregar').hide();
        $('#paginaClienteEditar').hide();
        sharedContext.addData(0);
        loadRecords();
    }, function (err) {
        console.log("Err" + err);
    });
};
}

```

Figura 48. Controller mantenimiento 2



```

ViewBag.Title = "Index";
Layout = "~/Views/Shared/_Layout.cshtml";

<div class="main-content">
  <div class="main-content-inner">
    <div class="page-content">
      <div class="page-content-area">
        <!-- inicio div contenedor -->
        <div class="row">
          <div class="col-xs-12">
            <div class="row">
              <div class="col-xs-12 col-sm-12">
                <div class="widget-box">
                  <div class="widget-header">
                    <h4 class="widget-title lighter"><b>Consulta Registros</b></h4>
                    <div class="widget-toolbars">
                      <a href="#" data-action="collapse">
                        <i class="ace-icon fa fa-chevron-up blue"></i>
                      </a>
                    </div>
                  </div>
                  <div class="widget-body collapse in id="ocultarcontainer" aria-expanded="true">
                    <div class="widget-main">
                      <div class="space space-8"> </div>
                      <div class="clearfix form-actions">
                        <div class="row">
                          <div class="espacio hidden-md visible-xs"></div>
                          <div class="col-md-3 col-sm-4 col-xs-12 pull-right">
                            <button class="btn btn-warning btn-block btn-xl" id="btnBuscar2000" ng-click="getOrdenPublicaciones(3)">
                              <i class="ace-icon fa fa-search bigger-125"></i>
                              LISTAR 2000
                            </button>
                          </div>
                          <div class="espacio hidden-md visible-xs"></div>
                          <div class="col-md-3 col-sm-4 col-xs-12 pull-right">
                            <button class="btn btn-warning btn-block btn-xl" id="btnBuscar1000" ng-click="getOrdenPublicaciones(2)">
                              <i class="ace-icon fa fa-search bigger-125"></i>
                              LISTAR 1000
                            </button>
                          </div>
                        </div>
                      <div class="espacio hidden-md visible-xs"></div>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

Figura 49. Consulta 1

```

<div class="col-xs-12">
  <div>
    <div class="clearfix">
      <div class="pull-right tableTools-container"></div>
    </div>
    <div id="dvData">
      <div class="table-responsive">
        <table class="table table-bordered table-hover table-consulta">
          <thead>
            <tr>
              <th>usuario</th>
              <th>tipoaviso</th>
              <th>codfacturar</th>
              <th>codente</th>
              <th>ente</th>
              <th>fecchaini</th>
              <th>fecchafin</th>
              <th>codfpago</th>
              <th>formpago</th>
              <th>moneda</th>
              <th>impbuto</th>
              <th>impcargo</th>
              <th>impdcto</th>
              <th>impventa</th>
              <th>igw</th>
              <th>impneto</th>
              <th>impnetobase</th>
              <th>estado</th>
            </tr>
          </thead>
          <tbody id="tbConsulta">
            <tbody>
              <tfoot><tr><td id="tdPaginacion" style="text-align:center" colspan="19"></td></tr></tfoot>
            </tbody>
          </table>
        </div>
      </div>
    </div>
  </div>
</div>

```

Figura 50. Consulta 2

```

ViewBag.Title = "Index";
Layout = "~/Views/Shared/_Layout2.cshtml";

<div class="main-content">
  <div class="main-content-inner">
    <div class="page-content">
      <div class="page-content-area" id="paginaCliente">
        <!-- inicio div contenedor -->
        <div class="row">
          <div class="col-xs-12">
            <div class="row">
              <div class="col-xs-12 col-sm-12">
                <div class="widget-box">
                  <div class="widget-header">
                    <a href="#" ng-click="showadd($event)" class='blue'>
                      <i class='ace-icon fa fa-plus bigger-160' title='Agregar'></i>
                    </a>
                    <h4 class="widget-title lighter"><b>Clientes</b></h4>
                    <div class="widget-toolbar">
                      <a href="#" data-action="collapse">
                        <i class="ace-icon fa fa-chevron-up blue"></i>
                      </a>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
          <div id="divCliente">
            <div class='row'>
              <div class='col-xs-12'>
                <div>
                  <div class='clearfix'>
                    <div class='pull-right tableTools-container'></div>
                  </div>
                  <div id='dvData'>
                    <div class='table-responsive'>

```

Figura 51. Cliente 1

```

<div id='dvData'>
  <div class='table-responsive'>
    <table class='table table-bordered table-hover table-consulta'>
      <thead>
        <tr>
          <th class='col-acciones-md'>Acciones</th>
          <th id</th>
          <th Apellidos</th>
          <th Nombres</th>
          <th Direccion</th>
          <th Email</th>
          <th LineaCredito</th>
        </tr>
      </thead>
      <tbody id='tbcliente'>
      </tbody>
      <tfoot><tr><td id='tdPaginacion' style='text-align:center' colspan='7'></td></tr></tfoot>
    </table>
  </div>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- fin div contenedor -->
<div id="paginaClienteEditar" class="row">...</div> <!-- fin pagina cliente -->
<div id="paginaClienteAgregar" class="row">...</div> <!-- fin pagina Agregar -->
</div>
</div>
</div>

```

Figura 52. Cliente 2

```
[Authorize]
1 reference
public class TablasGeneralesController : Controller
{
    private IPrincipal principalRepos;
    0 references
    public TablasGeneralesController()
    {
        principalRepos = new PrincipalRepository(System.Configuration.ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString);
    }
    0 references
    public ActionResult Consulta()
    {
        UsuariosBE usuarios = new UsuariosBE();
        usuarios = (UsuariosBE)Session["Usuarioweb"];
        bool lError = (usuarios == null ? true : false);
        string xRespuesta = (lError ? Respuestas.ERROR_SESION : string.Empty);
        if (!lError)
        {
            return PartialView(usuarios);
        }
        return RedirectToAction("Login", "Security");
    }

    0 references
    public ActionResult Cliente()
    {
        UsuariosBE usuarios = new UsuariosBE();
        usuarios = (UsuariosBE)Session["Usuarioweb"];
        bool lError = (usuarios == null ? true : false);
        string xRespuesta = (lError ? Respuestas.ERROR_SESION : string.Empty);
        if (!lError)
        {
            return PartialView(usuarios);
        }
        return RedirectToAction("Login", "Security");
    }
}
```

Figura 53. Mvc controller 1

```

}
0 references
public ActionResult Menus()
{
    bool lError = (Session["Usuarioweb"] == null ? true : false);
    int iPerfil = 0;
    Int64 id = 0;
    if (!lError)
    {
        iPerfil = ((UsuariosBE)Session["Usuarioweb"]).iPerfil;
        id = ((UsuariosBE)Session["Usuarioweb"]).iIdUsuario;
        return PartialView(_Menus(id, iPerfil));
    }
    return RedirectToAction("Login", "Account");
}
1 reference
private MenuViewModel _Menus(Int64 id, int iPerfil)
{
    IEnumerable<MenuViewModel> menus = principalRepos.arbMenus(id, iPerfil);
    var model = principalRepos.PadresMenus(0, menus);
    return model;
}
}
```

Figura 54. Mvc controller 2

```
<div class="row" id="paginaClienteAgregar" style="display:none;">
  <!--Inicio pagina Agregar-->
  <div class="col-xs-12 col-sm-12">
    <div class="widget-box form-horizontal">
      <div class="widget-header">
        <h4 id="idTituloOrden" class="widget-title lighter"><i class="menu-icon fa fa-edit bigger-140"></i>Nuevo Cliente</h4>
      </div>
      <div class="widget-body" id="containerReg">
        <div class="widget-main">
          <div class="row">
            <form class="form-horizontal" [formGroup]="clienteForm" ng-submit="saveAdd()" #formDir="ngForm" novalidate>
              <div class="row">
                <div class="col-xs-10">
                  <div class="form-group">
                    <label class="col-sm-2 control-label no-padding-right" for="txtId">ID</label>
                    <div class="col-sm-8">
                      <input type="text" class="form-control" readonly value="0" ng-model="cliente.ID" id="title" required title="ID requerido" />
                    </div>
                  </div>
                </div>
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Figura 55. Vista add cliente

## Anexo 09 – Documentación del Framework JavaScript Nativo

```

using BusinessEntity;
using GP_MVC_JavaScript.Models.Abstract;
using GP_MVC_JavaScript.Models.Repository;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Options;
using System.Collections.Generic;

namespace GP_MVC_JavaScript.Controllers
{
    1 reference
    public class TablasGeneralesController : Controller
    {
        private readonly AppSettings _appSettings;
        private ICliente _clienteRepos;
        private IOrdenPublicacion _ordenPublicacionRepos;
        0 references | 0 exceptions
        public TablasGeneralesController(IOptions<AppSettings> appSettings)
        {
            _appSettings = appSettings.Value;
            _clienteRepos = new ClienteRepository(_appSettings.Connection.ConnectionString);
            _ordenPublicacionRepos = new OrdenPublicacionRepository(_appSettings.Connection.ConnectionString);
        }
        [Authorize]
        0 references | 0 requests | 0 exceptions
        public IActionResult Cliente()
        {
            UsuariosBE sUsuarios = new UsuariosBE();
            sUsuarios = (UsuariosBE)HttpContext.Session.GetObjectFromJson<UsuariosBE>("UsuarioWeb");
            bool lError = (sUsuarios == null ? true : false);
            string xRespuesta = (lError ? Respuestas.ERROR_SESION : string.Empty);
            if (!lError)
            {
                return PartialView(sUsuarios);
            }
            return RedirectToAction("Login", "Security");
        }
        0 references | 0 requests | 0 exceptions
        public JsonResult LeerCliente(int IdCliente)
    }
}

```

Figura 56. TablasGeneralesController.cs - 1

```

0 references | 0 requests | 0 exceptions
public JsonResult LeerCliente(int IdCliente)
{
    ClienteBE oclienteBE = new ClienteBE();
    oclienteBE = _clienteRepos.GetCliente(IdCliente);
    if (oclienteBE != null)
    {
        return Json(new { success = true, data = oclienteBE });
    }
    return Json(new { success = false, data = oclienteBE });
}
0 references | 0 requests | 0 exceptions
public JsonResult Listar()
{
    List<ClienteBE> oListClienteBE = _clienteRepos.GetAllClientes();
    if (oListClienteBE != null && oListClienteBE.Count > 0)
    {
        return Json(new { success = true, data = oListClienteBE });
    }
    return Json(new { success = false, data = oListClienteBE });
}
0 references | 0 requests | 0 exceptions
public JsonResult grabar(ClienteBE oclienteBE)
{
    int response = 0;
    if (oclienteBE.IdCliente == 0) response = _clienteRepos.AddCliente(oclienteBE);
    else response = _clienteRepos.UpdateCliente(oclienteBE);
    if (response > 0)
    {
        return Json(new { success = true });
    }
    return Json(new { success = false });
}
}

```

Figura 57. TablasGeneralesController.cs - 2

```

public JsonResult Eliminar(int IdCliente)
{
    int response = 0;
    response = _clienteRepos.DeleteCliente(IdCliente);
    if (response > 0)
    {
        return Json(new { success = true });
    }
    return Json(new { success = false });
}

[Authorize]
0 references | 0 requests | 0 exceptions
public IActionResult Consulta()
{
    UsuariosBE sUsuarios = new UsuariosBE();
    sUsuarios = (UsuariosBE)HttpContext.Session.GetObjectFromJson<UsuariosBE>("UsuarioWeb");
    bool lError = (sUsuarios == null ? true : false);
    if (!lError)
    {
        return PartialView(sUsuarios);
    }
    return RedirectToAction("Login", "Security");
}

[Authorize]
0 references | 0 requests | 0 exceptions
public JsonResult ListarConsulta(int opcion)
{
    List<OrdenPublicacionBE> oOrdenPublicacionBE = _ordenPublicacionRepos.GetAllOrdenPublicacion(opcion);
    if (oOrdenPublicacionBE != null && oOrdenPublicacionBE.Count > 0)
    {
        return Json(new { success = true, data = oOrdenPublicacionBE });
    }
    return Json(new { success = false, data = oOrdenPublicacionBE });
}

```

Figura 58. TablasGeneralesController.cs - 3

```

Cliente.cshtml  TablasGeneralesController.cs  HomeController.cs  PrincipalCon
1  @model BusinessEntity.UsuariosBE
2  @{
3      Layout = "~/Views/Shared/_Layout.cshtml";
4  }
5  <div class="main-content">
6      <div class="main-content-inner">
7          <div class="page-content">
8              <div class="row" id="page_current">
9
10                 </div>
11             </div>
12         </div>
13     </div>
14     <script src="@Url.Content("~/Scripts/pages/Cliente.js")"></script>
15     <script type="text/javascript">
16         $(window).load(function () {
17
18             $("#status").fadeOut();
19             $("#preloader").delay(1000).fadeOut("slow");
20         });
21     </script>
22

```

Figura 59. Cliente.cshtml

```

Consulta.cshtml  Cliente.cshtml  TablasGeneralesController.cs  HomeController.cs  PrincipalController.cs  Menus.cshtml
1
2  Layout = "~/Views/Shared/_Layout.cshtml";
3
4  <div class="main-content">
5    <div class="main-content-inner">
6      <div class="page-content">
7        <div class="row">
8          <div class="col-xs-12">
9            <div class="row">
10             <div class="col-xs-12 col-sm-12">
11               <div class="widget-box">
12                 <div class="widget-header">
13                   <h4 class="widget-title lighter"><b>Consulta Registros</b></h4>
14                   <div class="widget-toolbar">
15                     <a href="#" data-action="collapse">
16                       <i class="ace-icon fa fa-chevron-up blue"></i>
17                     </a>
18                   </div>
19                 </div>
20                 <div class="widget-body collapse in" id="ocultarcontainer" aria-expanded="true">
21                   <div class="widget-main">
22                     <div class="space space-8"> </div>
23                     <div class="clearfix form-actions">
24                       <div class="row">
25                         <div class="espacio hidden-md visible-xs"></div>
26                         <div class="col-md-3 col-sm-4 col-xs-12 pull-right">
27                           <button class="btn btn-warning btn-block btn-xxl" id="btnBuscar2000">
28                             <i class="ace-icon fa fa-search bigger-125"></i>
29                             LISTAR 2000
30                           </button>
31                         </div>
32                         <div class="espacio hidden-md visible-xs"></div>
33                         <div class="col-md-3 col-sm-4 col-xs-12 pull-right">
34                           <button class="btn btn-warning btn-block btn-xxl" id="btnBuscar1000">
35                             <i class="ace-icon fa fa-search bigger-125"></i>
36                             LISTAR 1000

```

Figura 60. Consulta.cshtml - 1

```

Consulta.cshtml  Cliente.cshtml  TablasGeneralesController.cs  HomeController.cs  PrincipalController.cs  Menus.cshtml
36
37   LISTAR 1000
38   </button>
39   </div>
40   <div class="espacio hidden-md visible-xs"></div>
41   <div class="col-md-3 col-sm-4 col-xs-12 pull-right">
42     <button class="btn btn-warning btn-block btn-xxl" id="btnBuscar500">
43       <i class="ace-icon fa fa-search bigger-125"></i>
44       LISTAR 500
45     </button>
46   </div>
47 </div>
48 </div>
49 </div>
50 </div>
51 </div>
52 </div>
53 <div id="divPartial_Consulta">
54   <div class="row">
55     <div class="col-xs-12">
56       <div>
57         <div class="clearfix">
58           <div class="pull-right tableTools-container"></div>
59         </div>
60         <div id="dvData">
61           <div class="table-responsive">
62             <table class="table table-bordered table-hover table-consulta">
63               <thead>
64                 <tr>
65                   <th>idordenpublicacion</th>
66                   <th>usuario</th>
67                   <th>tipoaviso</th>
68                   <th>codfacturar</th>
69                   <th>codente</th>
70                   <th>ente</th>
71                   <th>fecchaini</th>

```

Figura 61. Consulta.cshtml - 2





```

var lista;

window.onload = function () {
    EncabezadoPagina();
    document.getElementById("page1").style.display = 'block';
    document.getElementById("page2").style.display = 'none';
    var url = "TablasGenerales/Listar";
    Get(url, mostrarLista);
    configurarMantenimiento();
};

function EncabezadoPagina() {
    var contenido = "";
    contenido += "<div class='col-xs-12' id='page1'>";
    contenido += "<div class='row'>";
    contenido += "<div class='col-xs-12 col-sm-12'>";
    contenido += "<div class='widget-box'>";
    contenido += "<div class='widget-header'>";
    contenido += "<a class='blue' href='#' onclick='return mostrarDetalle();'>";
    contenido += "<i class='ace-icon fa fa-plus bigger-160' title='Agregar'></i>";
    contenido += "</a>";
    contenido += "<h4 class='widget-title lighter'><b>Clientes</b></h4>";

    contenido += "<div class='widget-toolbar'>";
    contenido += "<a href='#' data-action='collapse'>";
    contenido += "<i class='ace-icon fa fa-chevron-up blue'></i>";
    contenido += "</a>";
    contenido += "</div>";
    contenido += "</div>";
    contenido += "</div>";
    contenido += "</div>";
    contenido += "</div>";
    contenido += "<div id='divCliente'></div>";
    contenido += "</div>";

    contenido += "<div class='col-xs-12 col-sm-12' id='page2'>";
    contenido += "<div class='widget-box form-horizontal'>";

```

Figura 64. Cliente.js - 1

```

    contenido += "</div>";
    contenido += "<div class='widget-body' id='containerReg'>";
    contenido += "<div class='widget-main'>";
    contenido += "<div class='row'>";
    contenido += "<form class='form-horizontal' id = 'validation-form' method = 'get'>";
    contenido += "<div class='row'>";
    contenido += "<div class='col-xs-10'>";
    contenido += "<div class='form-group'>";
    contenido += "<label class='col-sm-2 control-label no-padding-right'>ID</label>";
    contenido += "<div class='col-sm-8'>";
    contenido += "<input type='text' id='txtId' class='form-control' readonly value='' />";
    contenido += "</div>";
    contenido += "</div>";
    contenido += "</div>";
    contenido += "</div><br>";
    contenido += "<div class='row'>";
    contenido += "<div class='col-xs-10'>";
    contenido += "<div class='form-group'>";
    contenido += "<label class='col-sm-2 control-label no-padding-right' for= 'txtNombres' >Nombres</label>";
    contenido += "<div class='col-sm-10'>";
    contenido += "<div class='clearfix'>";
    contenido += "<input type='text' value='' id= 'txtNombres' class='form-control' placeholder='Nombres' />";
    contenido += "</div>";
    contenido += "</div>";
    contenido += "</div>";
    contenido += "</div>";
    contenido += "</div>";
    contenido += "<br>";
    contenido += "<div class='row'>";
    contenido += "<div class='col-xs-10'>";
    contenido += "<div class='form-group'>";

```

Figura 65. Cliente.js - 2

```

        contenido += "</div>";
        contenido += "</div>";

        var div = document.getElementById("page_current");
        if (div != null) div.innerHTML = contenido;
    }

    var Get = function (url, callBackMethod) {
        $('#loading-body-all').show();
        requestServer("get", urlBase + url, callBackMethod, null);
    };
    var PostUrl = function (url, callBackMethod) {
        $('#loading-body-all').show();
        requestServer("post", urlBase + url, callBackMethod, null);
    };
    var PostForm = function (url, callBackMethod, form) {
        $('#loading-body-all').show();
        requestServer("post", urlBase + url, callBackMethod, form);
    };
    var PostText = function (url, callBackMethod, text) {
        $('#loading-body-all').show();
        requestServer("post", urlBase + url, callBackMethod, text);
    };

    function requestServer(httpMethod,url, callBackMethod,data) {
        var xhr = new XMLHttpRequest();
        xhr.open(httpMethod, url, false);
        xhr.onreadystatechange = function () {
            if (xhr.status == 200 && xhr.readyState == 4) {
                callBackMethod(xhr.responseText);
            }
        };
    };

```

Figura 66. Cliente.js - 3

```

function crearTabla() {
    var contenido = "";
    if (lista != null && lista.length > 0) {
        contenido += "<div class='row'>";
        contenido += "<div class='col-xs-12'>";
        contenido += "<div>";
        contenido += "<div class='clearfix'>";
        contenido += "<div class='pull-right tableTools-container'></div>";
        contenido += "</div>";
        contenido += "<div id='dvData'>";
        contenido += "<div class='table-responsive'>";
        contenido += "<table id='dynamic-table' class='table table-bordered table-hover table-consulta' name='tabladinamica'>";
        contenido += "<thead><tr><th class='col-acciones-md'>Acciones</th>";
        var Campos = ["Id", "Nombres", "Apellidos", "Direccion", "Email", "LineaCredito"];
        var nCampos = Campos.length;
        for (j = 0; j < nCampos; j++) {
            contenido += "<th>";
            contenido += Campos[j];
            contenido += "</th>";
        }
        contenido += "</tr>";
        contenido += "</thead><tbody id='tbCliente'>";
        contenido += "</tbody><tfoot><tr><td id='tdPaginacion' style='text-align:center' colspan='";
        contenido += (nCampos+1).toString();
        contenido += "'></td></tr></tfoot>";
        contenido += "</table>";
        contenido += "</div>";
        contenido += "</div>";
        contenido += "</div>";
        contenido += "</div>";
        contenido += "</div>";
        var div = document.getElementById("divCliente");
        if (div != null) div.innerHTML = contenido;
    }
}

```

Figura 67. Cliente.js - 4

**Anexo 10 – Certificado de validez de contenido de los instrumentos**

**CERTIFICADO DE VALIDEZ DE CONTENIDO DE LOS INSTRUMENTOS**

Variable: **Framework JavaScript**

Nº	DIMENSIONES/Indicadores	Pertinencia		Relevancia		Claridad		Observaciones
		SI	NO	SI	NO	SI	NO	
	<b>Dimensión: Tiempo de Respuesta</b>							
1	Tiempo al Agregar (ms)	X		X		X		
2	Tiempo al Editar (ms)	X		X		X		
3	Tiempo al Eliminar (ms)	X		X		X		
4	Tiempo al Listar (ms)	X		X		X		

Observaciones (precisar si hay suficiencia): \_\_\_\_\_

Opinión de aplicabilidad:  Aplicable (X)  Aplicable después de corregir ( )  No Aplicable ( )

Apellidos y Nombres del Validador: Ing. Johnny Alberto Fernández Chero

Nº DNI: 43122871 CIP: 132923

Especialidad del Validador: Ing. Informático

Grado Académico:  Magister ( )  Doctor ( )

  
 Johnny A. Fernández Chero  
 ING. INFORMÁTICO  
 R. CIP. 132923  
 \_\_\_\_\_  
 Firma del Validador

22 de Julio de 2020

<sup>1</sup>Pertinencia: El ítem corresponde al concepto teórico formulado.  
<sup>2</sup>Relevancia: El ítem es apropiado para representar al componente o dimensión específica del constructo  
<sup>3</sup>Claridad: Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión

CERTIFICADO DE VALIDEZ DE CONTENIDO DE LOS INSTRUMENTOS

Variable: Framework JavaScript

Nº	DIMENSIONES/Indicadores	Pertinencia		Relevancia		Claridad		Observaciones
		SI	NO	SI	NO	SI	NO	
	<b>Dimensión: Tiempo de Respuesta</b>							
1	Tiempo al Agregar (ms)	✓		✓		✓		
2	Tiempo al Editar (ms)	✓		✓		✓		
3	Tiempo al Eliminar (ms)	✓		✓		✓		
4	Tiempo al Listar (ms)	✓		✓		✓		

Observaciones (precisar si hay suficiencia): Si hay suficiencia

Opinión de aplicabilidad: Aplicable (X) Aplicable después de corregir ( ) No Aplicable ( )

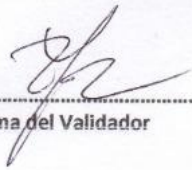
Apeellidos y Nombres del Validador: Muñoz Alegre Jesús Armando

Nº DNI: 42323928 CIP: 159984

Especialidad del Validador: Ingeniero de Sistemas

Grado Académico: Magister (X) Doctor ( )

16 de Julio de 2020



Firma del Validador

<sup>1</sup>Pertinencia: El ítem corresponde al concepto teórico formulado.

<sup>2</sup>Relevancia: El ítem es apropiado para representar al componente o dimensión específica del constructo

<sup>3</sup>Claridad: Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión

**CERTIFICADO DE VALIDEZ DE CONTENIDO DE LOS INSTRUMENTOS**

**Variable: Framework JavaScript**

Nº	DIMENSIONES/Indicadores	Pertinencia		Relevancia		Claridad		Observaciones
		SI	NO	SI	NO	SI	NO	
	<b>Dimensión: Tiempo de Respuesta</b>							
1	Tiempo al Agregar (ms)	X		X		X		
2	Tiempo al Editar (ms)	X		X		X		
3	Tiempo al Eliminar (ms)	X		X		X		
4	Tiempo al Listar (ms)	X		X		X		

Observaciones (precisar si hay suficiencia): \_\_\_\_\_

Opinión de aplicabilidad: Aplicable (X) Aplicable después de corregir ( ) No Aplicable ( )

Apellidos y Nombres del Validador: Ing. Espinoza Bravo Wilder Julio

Nº DNI: 46635819 CIP: 231046

Especialidad del Validador: Ingeniero en Informática

Grado Académico: Magister (X) Doctor ( )

20 de Julio de 2020



Firma del Validador

\***Pertinencia:** El ítem corresponde al concepto teórico formulado.

?**Relevancia:** El ítem es apropiado para representar al componente o dimensión específica del constructo

\***Claridad:** Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo

**Nota:** Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión

## Anexo 11 – Entrevista a experto

### ENTREVISTA EXPERTO EN DESARROLLO DE APLICACIONES

Apellidos y Nombre:

Profesión:

Años de experiencia:

1. **¿Qué opinión le merece el uso de un framework JavaScript en el desarrollo de aplicaciones web?. ¿Tiene alguna ventaja significativa su uso frente a otras aplicaciones que no usan? ¿Inciden en el tiempo de respuesta?**

La adopción de frameworks o conjunto de buenas prácticas en el desarrollo de software se encuentra establecido en el mundo hace muchos años porque ofrece soluciones a problemas comunes, buscando mejorar el rendimiento, escalabilidad y mantenibilidad. Javascript ha tenido un gran impulso desde sus inicios, a través de librerías y frameworks.

Uno de los principales problemas que se enfrenta al desarrollar sistemas empresariales es el uso de un número creciente de librerías, además de los componentes construidos por el equipo de desarrollo, los cuales generan demoras en la carga del sistema afectando la experiencia del usuario. Manejar cargas asíncronas combinando HTML y javascript no ha sido una solución elegante. El framework Angular permite controlar la velocidad de carga a través del patrón "lazy loading", que le permite procesar componentes cuando se necesite, reduciendo el tiempo de respuesta en la primera carga.

2. **En su experiencia, ¿Cuál es su opinión del framework Angular? ¿Podría comentar las principales ventajas que se obtiene al emplear éste framework en el desarrollo de aplicaciones web? ¿Ud. recomendaría trabajar con éste framework?**

Angular es un framework popular en nuestro país dentro del ambiente frontend. Tiene muchas ventajas entre las cuales encontramos la posibilidad de dividir el código a través de componentes, el uso de plantillas, pruebas unitarias, etc. Además que por ser popular, las empresas tienen la oportunidad de encontrar profesionales con ese perfil.

3. **En su experiencia, ¿Cuál es su opinión del framework JavaScript Nativo?. ¿Podría comentar las principales ventajas que se obtiene al emplear este framework en el desarrollo de aplicaciones web? ¿Ud. recomendaría trabajar con éste framework?**

El uso de javascript nativo ( no de framework ) tiene ventajas cuando se trabaja en pequeños proyectos o en la implementación de sitios webs comerciales donde interesa una carga muy rápida para no afectar el SEO y no existen muchos requerimientos funcionales para construir.

4. ¿En todos los diferentes proyectos de desarrollo que Ud. ha participado ha tenido algunos inconvenientes al usar los framework Angular y/o JavaScript Nativo? ¿Podría mencionar

No he tenido inconvenientes. Los sistemas realizados con angular han tenido la respuesta esperada por parte de los stakeholders.

5. En base a su amplio expertise, ¿Cuáles serían sus recomendaciones finales a la hora de seleccionar un framework al momento de construir aplicaciones web?

Cuando se decide desarrollar un sistema empresarial a nivel de frontend se busca un framework que permita utilizar componentes y templates para asegurar la mantenibilidad. El rendimiento también es un factor importante por su impacto en la experiencia del usuario que es determinante más allá del aspecto funcional.



**Mariano Alonso Alcántara Francia**  
Ing. de Computación y Sistemas