



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y
URBANISMO**

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

**EVALUACIÓN DE RENDIMIENTO DE UN SISTEMA
WEB DESARROLLADA MEDIANTE LA
TECNOLOGÍA DE APLICACIONES WEB
PROGRESIVAS**

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO
DE SISTEMAS**

Autor(a) (es):

Bach. Tello Peña Cristhian

ORCID: <https://orcid.org/0000-0001-9121-8432>

Asesor(a):

Mg. Tuesta Monteza Víctor

ORCID: <https://orcid.org/0000-0002-5913-990X>

Línea de Investigación:

Infraestructura, Tecnología y Medio Ambiente

Pimentel – Perú 2021

APROBACIÓN DEL JURADO

EVALUACIÓN DE RENDIMIENTO DE UN SISTEMA WEB DESARROLLADA MEDIANTE LA TECNOLOGÍA DE APLICACIONES WEB PROGRESIVAS

Bach. Tello Peña Cristhian

Autor

Mg. Tuesta Monteza Víctor Asesor

Dr. Mario Fernando Ramos Moscol
Presidente de Jurado

Mg. Jaime Arturo Bravo Ruiz
Secretario de Jurado

Mg. David Enrique Bances Saavedra
Vocal de Jurado

Dedicatoria

A mi familia especialmente a mis padres por su apoyo incondicional, su esfuerzo económico y motivacional, dedicación y ejemplo para seguir adelante.

Agradecimientos

Quiero extender un profundo agradecimiento, a quienes hicieron posible esta meta, aquellos que me acompañaron en todo momento y que siempre fueron mi inspiración, fortaleza y apoyo. En primer lugar, a Dios por nacer de buenos padres y pertenecer a una amorosa familia. A la plana docente por el conocimiento brindado durante todo mi tiempo universitario.

Resumen

Desde sus inicios la web, ha sido construida con tecnología tradicional (HTML, CSS y JS), y pensadas para usarse en las grandes pantallas y no en función de la usabilidad del usuario. Así mismo se observa que en los últimos años aún se pierden millones de dólares en el mundo, por la razón que aún se encuentra páginas web mal diseñadas o son lentas al momento de cargarse.

Como objetivo del presente trabajo de investigación es la evaluación del rendimiento de un Sistema web mediante la tecnología de aplicaciones web progresivas. La investigación empieza haciendo un estudio de los Frameworks más utilizados tanto para el desarrollo como para el diseño web, de esta manera se seleccionó por conveniencia los Frameworks Laravel y Vue por la razón que ambas manejan la arquitectura MVC y se encuentran entre los más usados. Luego se seleccionó como caso de estudio el desarrollo de un sistema web de reservas de ambientes el cual se desarrolló con la ayuda de los frameworks anteriormente seleccionados. De esta manera se obtuvo los dos sistemas, el sistema tradicional y el sistema con la tecnología de aplicaciones web progresivas. Para luego de esto se realizó los casos de pruebas para cada uno de los sistemas desarrollados y fueron evaluados por 5 herramientas GTmetrix, Pagetestweb, Lighthouse, Pingdom, PageSpeed Insights las cuales fueron promediadas para el resultado de los indicadores. Como resultado con una concurrencia alta de 30 usuarios en MITR, MTR, MATR, Índice de velocidad, tamaño de la web y consumo de RAM para el sistema tradicional es de (7208, 10033, 16463, 2.70Seg, 690.97Kb, 168.8 Mb) y para el sistema con tecnología de aplicaciones web progresivas (5506, 10779, 19761, 2,62 Seg, 729,47Kb, 306 Mb).

Finalmente se llegó a la conclusión que el sistema web con la tecnología de aplicaciones web progresivas si tuvo una leve mejora de rendimiento en casi todos sus indicadores menos en el tamaño de la web como en el consumo de memoria.

Palabras Clave: *Aplicaciones web progresivas, frameworks, trabajador de servicio, Sistema web, rendimiento*

Abstract

Since its inception, the web has been built with traditional technology (HTML, CSS and JS), and designed to be used on large screens and not based on user usability. Likewise, it is observed that in recent years millions of dollars are still lost in the world, for the reason that there are still poorly designed web pages or they are slow to load.

The objective of this research work is the evaluation of the performance of a web system using progressive web application technology. The research begins by doing a study of the most used Frameworks for both development and web design, in this way the Laravel and Vue Frameworks were selected for convenience for the reason that both handle the MVC architecture and are among the most used. Then, the development of a web environment reservation system was selected as a case study, which was developed with the help of the previously selected frameworks. In this way, the two systems were obtained, the traditional system and the system with progressive web applications technology. After this, the test cases were carried out for each of the developed systems and they were evaluated by 5 tools GTmetrix, Pagetestweb, Lighthouse, Pingdom, PageSpeed Insights which were averaged for the result of the indicators. As a result with a high attendance of 30 users in MITR, MTR, MATR, Speed index, web size and RAM consumption for the traditional system is (7208, 10033, 16463, 2.70Seg, 690.97Kb, 168.8 Mb) and for the system with progressive web applications technology (5506, 10779, 19761, 2.62 Sec, 729.47Kb, 306 Mb).

Finally, it was concluded that the web system with progressive web applications technology did have a slight performance improvement in almost all its indicators except in the size of the web and in memory consumption.

Keywords: *Frameworks, service worker, web system, progressive web applications, performance*

Índice

I. INTRODUCCIÓN	viii
1.1. Realidad Problemática	11
1.2. Trabajos previos.	13
1.3. Teorías relacionadas al tema	19
1.4. Formulación del Problema.	54
1.5. Justificación e importancia del estudio	54
1.6. Hipótesis.	55
1.7. Objetivos.	55
1.7.1. Objetivo general	55
1.7.2. Objetivos específicos	55
II. MATERIAL Y MÉTODO	55
2.1. Tipo y Diseño de Investigación	55
2.2. Población y muestra.	56
2.3. Variables, Operacionalización	57
2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad.	59
2.5. Procedimiento de análisis de datos	59
2.6. Criterios éticos.	61
2.7. Criterios de Rigor Científico.	61
III. RESULTADOS	61
3.1. Resultados en Tablas y Figuras.	62
3.2. Discusión de resultados.	67
3.3. Aporte práctico.	69
IV. CONCLUSIONES Y RECOMENDACIONES	108
4.1. Conclusiones.	108
4.2. Recomendaciones.	109
REFERENCIAS	110
ANEXOS.	115

Índice de Figuras

Figura 1 Las URL especifican el protocolo, el servidor y el recurso local. Fuente: (Gourley & Totty, 2002).....	20
Figura 2 Protocolo HTTP (Cliente. - Servidor). Fuente: (Gourley & Totty, 2002)	21
Figura 3 Desarrollo de la web social. Fuente: (Mika, 2007).	24
Figura 4 Evolución de la Web. Fuente: (Lerma-Blasco et al., 2013).....	25
Figura 5 Dimensiones según ISO / IEC 9126-1 para la categorización de características de aplicaciones web. Fuente: (Kappel G., 2006).	27
Figura 6 Páginas web de los principales navegadores. Fuente: Lerma, Murcia, & Mifsud Talón, (2013).....	28
Figura 7 Principales sistemas operativos (Lerma, Murcia, & Mifsud, 2013)	30
Figura 8 Máquina virtual con el emulador del navegador Opera TV. Fuente: (Lerma, Murcia, & Mifsud, 2013).....	31
Figura 9 Escenario de un Framework. Fuente: (Lerma, Murcia, & Mifsud, 2013).....	32
Figura 10 Características comunes en los frameworks existentes. Fuente: (Gutiérrez, 2014)	33
Figura 11 Configuración para indicar aVue donde montara la aplicación. Fuente: (Gore, 2015)	36
Figura 12 Usando sintaxis de bigote. Fuente: (Gore, 2015).....	36
Figura 13 Directiva v-for. Fuente: (Gore, 2015).	37
Figura 14 Representación v-for Fuente: (Gore, 2015).	37
Figura 15 Creación de función. Fuente: (Gore, 2015).....	37
Figura 16Cambio de función inicial. Fuente: (Gore, 2015)	38
Figura 17 Creación de elemento grocery-item. Fuente: (Gore, 2015)	38
Figura 18 Uso del componente. Fuente: (Gore, 2015).....	38
Figura 19Arquitectura del Framework Angular. Fuente: (Seshadri, 2018).	40
Figura 20 Sintaxis de Importación y Exportación. Fuente: (Dayley, Dayley, & Dayley, 2018)	41
Figura 21 Ejemplo de Organización del código. Fuente: (Zaninotto & Potencier, 2009)....	45
Figura 22 Proceso de instalación del services workes. Fuente: (Lepage, sf).....	49
Figura 23 Especificaciones del W3C. Fuente: (Hajian, 2019).....	50
Figura 24 Proceso de notificaciones Push. Fuente: (Hajian, 2019)	52
Figura 25 Ilustración del proceso del uso de la cache. Fuente: (whatwg, 2018).....	53
Figura 26 Estados y sistema. Fuente: (Sukoco, Marzuki, & Cucus, 2012).	54
Figura 27 Tamaño de los sistemas web. Fuente: Elaboración propia	64
Figura 28 Comparación de resultados de consumo de memoria. Fuente: Elaboración propia	65
Figura 29 Índice de velocidad de los Sistemas web. Fuente: Elaboración propia	66
Figura 30 Método Propuesto. Fuente: Elaboración propia	70
Figura 31 Descargas por paquetes npm en el último año. Fuente: (npm trends, 2020). ...	71
Figura 32 Frameworks con mejor puntuación. npm. Fuente: (npm trends, 2020).....	71
Figura 33 Vista de la ventana Login o inicio de sesión. Fuente: elaboración propia	80
Figura 34Vista reserva. Fuente: elaboración propia.....	81
Figura 35 Vista Usuario ambiente-usuario final. Fuente: Elaboración propia	81
Figura 36 Vista de la ventana ambiente-administrado. Fuente: Elaboración propia.....	82
Figura 37 Vista mantenimiento usuarios. Fuente: Elaboración propia.....	82

Figura 38 Vista de la ventana Nuevo usuario. Fuente: Elaboración propia	83
Figura 39 Vista de la ventana nuevo ambiente. Fuente: Elaboración propia.....	83
Figura 40 Modelo de base datos inicial. Fuente: Elaboración propia.....	84
Figura 41 Vista de creación de base de datos. Fuente: Elaboración propia.....	84
Figura 42Figura N° 35 Base de datos del sistema. Fuente: Elaboración propia.....	85
Figura 43 Velocidad de Internet utilizado. Fuente: (testdevelocidad.es,2020)	89
Figura 44 Directorio de JMeter. Fuente: Elaboración propia.....	89
Figura 45 Archivo ejecutable ApacheJMeter. Fuente: Elaboración propia.....	89
Figura 46 Grupo de Hilos. Fuente: Elaboración propia.....	90
Figura 47 Controladores para las funcionalidades y el informe agregado. Fuente: Elaboración propia	90
Figura 48 Servidor proxy en la herramienta JMeter. Fuente: Elaboración propia.....	91
Figura 49 Configuración del proxy en el navegador Firefox. Fuente: Elaboración propia .	91
Figura 50 Grupo de hilos que simulan la concurrencia de usuarios. Fuente: elaboración propia	92
Figura 51 Concurrencia baja - funcionalidades del Sistema Tradicional. Fuente: Elaboración propia	92
Figura 52 Concurrencia media - funcionalidades del Sistema Tradicional. Fuente: Elaboración propia.	93
Figura 53 Concurrencia alta - funcionalidades del Sistema Tradicional. Fuente: Elaboración propia.	93
Figura 54Concurrencia baja - funcionalidades del Sistema con PWA. Fuente: Elaboración propia.	96
Figura 55Figura N° 48 Concurrencia media - funcionalidades del Sistema PWA. Fuente: Elaboración propia.	96
Figura 56Figura N° 49 Concurrencia alta - funcionalidades del Sistema PWA. Fuente: Elaboración propia.	97
Figura 57 Uso de memoria RAM de Inicio de Sesión. Fuente: Elaboración propia	100
Figura 58 Uso de memoria RAM de vistas ambientes. Fuente: Elaboración propia.....	100
Figura 59 Uso de memoria RAM de Nuevo Ambiente. Fuente: Elaboración propia.....	100
Figura 60 Uso de memoria RAM de Nueva Reserva. Fuente: Elaboración propia.....	101
Figura 61 Uso de memoria RAM de Inicio de sesión. Fuente: Elaboración propia.....	102
Figura 62 Uso de memoria RAM de Nuevo ambiente. Fuente: Elaboración propia	102
Figura 63 Uso de memoria RAM de Nueva reserva. Fuente: Elaboración propia	102
Figura 64Uso de memoria RAM de Vista de Ambientes. Fuente: Elaboración propia	103
Figura 65 Índice de velocidad con la herramienta PageSpeed Insights. Fuente: Elaboración propia	104
Figura 66 Índice de velocidad con la herramienta webpagetest. Fuente: Elaboración propia	104
Figura 67 Índice de velocidad con la herramienta Lighthouse. Fuente: Elaboración propia	105
Figura 68 Índice de velocidad con la herramienta PageSpeed Insights. Fuente: Elaboración propia	106
Figura 69 Índice de velocidad con la herramienta Pagewebtest. Fuente: Elaboración propia	106
Figura 70 Índice de velocidad con la herramienta Lighthouse. Fuente: Elaboración propia	107

Índice de Tablas

Tabla 1 Principales códigos de respuesta HTTP	22
Tabla 4 Técnicas, Instrumentos	59
Tabla 5 Resultados de tiempo de respuesta para funcionalidad Inicio de sesión	62
Tabla 6 Resultados de tiempo de respuesta para funcionalidad Nuevo ambiente	62
Tabla 7 Resultados de tiempo de respuesta para funcionalidad Nuevo reserva	63
Tabla 8 Resultados de tiempo de respuesta para funcionalidad Vista ambientes	63
Tabla 9 Tamaño de los sistemas web	63
Tabla 10 Resultados de consumo de memoria por funcionalidad	64
Tabla 11 Índice de velocidad de los Sistemas web	65
Tabla 12 Resumen de resultados con relación al comportamiento en el tiempo	66
Tabla 13 Resumen de resultados respecto al rendimiento	67
Tabla 14 Comparación de Frameworks Front-end	72
Tabla 15 Comparación de Frameworks Back-end.....	72
Tabla 16 Requerimientos para usuarios	74
Tabla 17 Requerimientos para Reservas.....	78
Tabla 18 Requerimientos no funcionales para el sistema	79
Tabla 19 Hardware a utilizar para el desarrollo de los sistemas propuestos.	86
Tabla 20 Herramientas a utilizar para el comportamiento en el tiempo.....	86
Tabla 21 Herramientas a utilizar para calcular el tamaño de los sistemas	87
Tabla 22 Herramientas a utilizar para calcular el uso de la memoria RAM.....	87
Tabla 23 Herramientas a utilizar para calcular el índice de velocidad	88
Tabla 24 Funcionalidad Inicio de sesión	94
Tabla 25 Funcionalidad Nuevo ambiente.....	94
Tabla 26 Funcionalidad Nueva reserva.....	94
Tabla 27 Funcionalidad Vista ambientes	95
Tabla 28 Funcionalidad Inicio de sesión	97
Tabla 29 Resultado del tamaño de la web del sistema tradicional.....	99
Tabla 30 Resultado del tamaño de la web con tecnología de aplicaciones web progresivas.....	99
Tabla 31 Consumo de memoria RAM por funcionalidad	101
Tabla 32 Consumo de memoria RAM por funcionalidad	103
Tabla 33 Índice de velocidad del Sistemas con 3 herramientas diferentes	105
Tabla 34 Índice de velocidad del Sistema con 3 herramientas diferentes	107
Tabla 35 Índice de velocidad promedio de ambos sistemas.....	107

I. INTRODUCCIÓN

1.1. Realidad Problemática.

Desde sus inicios la web, ha sido construida con tecnología tradicional (HTML, CSS y JS), y pensadas para usarse en las grandes pantallas y no en función de la usabilidad del usuario (Rieger & Majchrzak, 2019) . De la misma manera en la actualidad se ha dado un continuo crecimiento en los dispositivos móviles en la última década, esto ha permitido que las aplicaciones tengan una tendencia a ser interoperables orientado a tareas y obligando a los diseñadores y desarrolladores a crear y mantener aplicaciones que puedan ejecutarse en múltiples sistemas operativos (Rieger & Majchrzak, 2019).

En los últimos años en el mundo aún se pierde demasiado dinero, por la razón que aún se encuentran páginas web móviles que están mal diseñadas o son lentas al momento de cargarse, las razones más usuales de estas son: las redes de datos, falta de flexibilidad en plataformas de desarrollo, comportamientos web replicados en móviles, altos precios para crear soluciones efectivas y lo más importante, el usuario de móviles que demanda una experiencia más profesional. (Palatnik, 2017).

An, (2018) también afirma que los intereses y necesidades de los clientes están variando y cambiando, por lo tanto, las empresas tienen el deber de transformarse e innovar para poder dar mejores experiencias en dispositivos móviles y en la web, más rápidas e inmersivas.

Según la gran compañía de iPhone supuso un antes y un después en el desarrollo de aplicaciones web. (López, 2017). Hasta ese momento no se tenía aplicaciones multiplataforma, es así que aparecen las apps web móviles desarrolladas únicamente con tecnología web, el cual se accede mediante un localizador de recursos uniformes (URL) a través del navegador del dispositivo, ofrece rápido desarrollo, simple mantenimiento y acceso multiplataforma, pero el acceso a los recursos del dispositivo móvil es muy limitado (Malavolta, 2016).

En un estudio realizado por (Spero, 2017) de la empresa Google, describe que el 53% de usuarios abandonará un sitio web si no carga dentro de 3 segundos en un dispositivo móvil.

En la investigación realizada a los países de Latinoamérica (Google and Nodus, 2018) se obtiene que 6 de 10 latinoamericanos son usuarios de Internet y el 90% lo hacen a través de móviles. También se afirma que el consumo de internet seguirá aumentando en un 35% más en Latinoamérica impulsado por el uso de smartphones. Las búsquedas en Mobile cada vez son mayores a desktop (6 de cada 10 son Mobile), el 52% de los usuarios dicen que la velocidad es importante, por último, se asegura que 1 segundo de retraso en el tiempo de carga puede impactar sobre el 8% de aumento en el porcentaje de rebote y una disminución del 16% en satisfacción de los usuarios, en conversiones se obtiene un 7% de disminución y un 11% menos de visitas al sitio y en ingresos se obtiene un 2,1% de disminución en el tamaño de la cesta.

Coe, (2019) menciona que si la experiencia en un dispositivo móvil es lenta puede causar la pérdida de los clientes, en cambio una experiencia rápida puede beneficiar mucho a conservar e incluso a traer más clientes.

Los mayores retos que tienen las empresas en el proceso de la digitalización de su negocio, es poder ofrecer una experiencia de calidad a sus clientes en la web como en móviles (Baca, 2018).

Como soluciones a estos problemas se crean las aplicaciones nativas, para el lado móvil, estas se crean como una capa sobre los servicios proporcionados por una plataforma móvil específica. Por ejemplo, Java se utiliza para desarrollar aplicaciones Android, mientras que las aplicaciones iOS de Apple se desarrollan con Swift, Objective -C. Con el desarrollo viene una gran responsabilidad de tiempo, pruebas, mantenimiento y portabilidad. Desafortunadamente, las aplicaciones nativas no están al día debido a su conocido desafío de la fragmentación de la plataforma móvil.(Gambhir & Raj, 2018).

Mientras que las Aplicaciones web, no es necesario adaptarse a ningún sistema operativo específico. Solo se requiere un navegador. Por otro lado, esto podría reducir la velocidad de ejecución y dar lugar a una experiencia de usuario más pobre con interfaces que son más limitadas que las ofrecidas por aplicaciones nativas. Además, el rendimiento puede verse afectado por problemas de conectividad. (Delia, Galdamez, Corbalan, Pesado, & Thomas, 2017)

En cambio, las aplicaciones híbridas nos dan diferentes ventajas porque permiten la reutilización de código para las diversas plataformas, pero la experiencia del usuario sufre de no utilizar los componentes nativos en la interfaz, la ejecución podría ser más lenta debido a la carga adicional asociada al contenedor web. (Delia et al., 2017)

Teniendo en cuenta estos resultados, en el año 2013, se denominó "Año de diseño web responsivo" y el anunciado Mobilegeddon de Google en el 2015, propició el desarrollo de las aplicaciones web progresivas (PWA), una tecnología que combina lo mejor de las aplicaciones web y las aplicaciones nativas, lo que permitió el crecimiento de manera espectacular de las tecnologías web y también lo hicieron las empresas que sabían cómo aprovechar las nuevas tendencias (Frankston, 2018).

1.2. Trabajos previos.

Al describir el trabajo Enhanced Web Application and Browsing performance through Service-worker Infusion Framework desarrollado por (Pande, Somani, Samal, & Kakkirala, 2018) proponen un método para inyectar un service worker en una página web en los servidores proxy. El service worker este empaquetado junto con una página web que no es PWA como respuesta a las solicitudes del navegador web. A partir de ello, el motor web ejecuta el service worker para verificar cuales son los recursos web obsoletos que al mismo tiempo serán útiles para evitar la transferencia de recursos web. Para validar el método propuesto se ha evaluado un conjunto de 25 páginas web que no son PWA durante un mes. Se logro demostrar un ahorro promedio de 25% del tráfico de datos.

Según (de Andrade, Cardieri & Zaina, 2018) en su investigación *Analyzing User Experience in Mobile Web, Native and Progressive Web Applications: A User and HCI Specialist Perspectives*, tiene como objetivo principal analizar la experiencia de usuario al momento de utilizar una aplicación móvil nativa, web y una aplicación web progresiva (PWA), con este propósito se desarrolló un análisis cualitativo en dos etapas. Primero, se consideró los comentarios y puntos de vista por parte del usuario. Y por último se analizó el comportamiento hombre-maquina-interacción (HCI) examinando las expresiones faciales de los usuarios con la finalidad de ver que emociones predominaron al interactuar con las aplicaciones. Se han considerado ocho personas entre 18 y 27 años y se ha demostrado que los principales estados de ánimo presentes es sorpresa e incertidumbre al utilizar una aplicación web y nativa. El análisis HCI fue relevante, pues las PWA es una tecnología reciente y teniendo como base los comentarios de los usuarios permite que este enfoque sea más maduro.

En el trabajo de investigación *Pride and Prejudice in Progressive Web Apps: Abusing Native App-like Features in Web Applications* propuesto por (Lee, Kim & Park, 2018) realiza un estudio sistemático de la seguridad y privacidad de los PWA. Se desarrolla dos modelos de ataques: PWA atacante y atacante de red. El primero modelo de ataque controla su propio sitio web y una vez que el usuario visita el sitio web se registra el service worker automáticamente en el navegador de la víctima, así como todos los permisos para recibir notificaciones. Cuando se realiza el atacante de red se establece un vigilante activo con la finalidad de monitorear, interceptar y modificar el tráfico de red a través del protocolo HTTP utilizando biblioteca de inserción de terceros. Para poner a prueba los PWA, se ha analizado 100 000 dominios de Alexa, de los cuales el 56% de los sitios PWA utilizan sus logotipos de sus marcas como notificaciones. Esta tendencia abre las puertas para que los atacantes de sitios web imiten los logotipos de las marcas para suplantar la identidad mediante notificaciones automáticas. Cuando se realiza los ataques offline de cache se obtiene que el 36.5% de sitios web es vulnerable. Esto demuestra que las notificaciones push en un PWA debe ser tratado con mucho recelo, y sobre todo establecer nuevos

mecanismos de protección con estas nuevas tecnologías que tratan de mejorar el uso de una aplicación nativa.

En la investigación “Development of Monitoring System for Smart Farming Using Progressive Web App” realizado por (Nugroho, Pratama, Mustika, & Ferdiana, 2017), establece como principal objetivo desarrollar un sistema para monitorear la plantación de palma de aceite. Se ha integrado interfaces de programación de aplicaciones (API) para conectar diferentes sensores, el cual ha sido integrado una aplicación web progresiva (PWA). Como punto de partida se ha obtenido las historias de usuario teniendo en cuenta la información obtenida por los empleados que realizan las plantaciones de palma de aceite. Basado en las historias de usuario, se ha realizado el prototipado para luego ser implementado con diferentes tecnologías web incluyendo la implementación de una aplicación web progresiva. Se ha realizado un test de caja negra, y también se ha evaluado diferentes métricas con las herramientas Lighthouse y PageSpeed. Los resultados obtenidos con PageSpeed alcanza una puntuación de 87/100 mientras que con LightHouse se obtiene una puntuación de 94/100. Estos resultados demuestran que el enfoque de una PWA proporciona una experiencia de usuario muy parecido al uso de las aplicaciones nativas.

Según (Malavolta, Procaccianti, Noorland, & Vukmirovi, 2017) en su trabajo *Assessing the Impact of Service Workers on the Energy Efficiency of Progressive Web Apps*, tiene como objetivo principal realizar un estudio empírico de los service worker en la eficiencia energética de los PWA. Con este propósito se ha realizado un experimento con dos principales factores: el uso de los service workers y el tipo de red disponible. Se ha utilizado un total de 7 PWA que ha sido utilizado en dispositivos LG G2 y un NEXUS 6P. Los resultados muestran que el service worker no tienen un impacto significativo sobre el consumo de energía de los dos dispositivos utilizados, independientemente de las condiciones de la red. Además, no se detectó interacción entre los dos factores. Sin embargo, algunos patrones en los datos muestran diferentes comportamientos entre los PWA. Por esta razón se

concluye que esta tecnología es prometedora en términos de eficiencia energética.

En la investigación *Analysis of Cache in Service Worker and Performance Scoring of Progressive Web Application* realizada por (Gambhi, Raj, 2018), tiene como fin el superar el problema de la fragmentación en plataformas de desarrollo y los comportamientos web replicados en móviles. Por ello se proponen las aplicaciones web progresivas, para de esta manera cerrar la brecha entre aplicaciones móviles web y las nativas. Teniendo en cuenta estos puntos se analizó la memoria cache en el service worker, también se evaluó el rendimiento de las PWAs mediante la herramienta lighthouse. Por último, se hizo la comparación de una web con tecnologías tradicionales vs las PWAs en la web de Pinterest. En los resultados obtenidos se acata que a medida que crece el número de usuarios, el tiempo medio de respuesta de PWA disminuye drásticamente, es decir, la aplicación se carga rápidamente. Por otro lado, el tiempo de respuesta de la aplicación Android aumenta. El motivo de esto es el almacenamiento en caché previo realizado por el Service Worker en el back-end. Esto demuestra que el rendimiento de PWA de la misma aplicación es mejor que la versión de Android y también el proceso de almacenamiento en caché aumenta el rendimiento de una PWA.

Según Delia, (2017) en su trabajo *Approaches to mobile application development: Comparative performance analysis*, tienen como principal fin hacer un estudio de rendimiento de los enfoques utilizados para desarrollar software para dispositivos móviles. Debido a que la diversidad de plataformas, lenguajes de programación, herramientas de desarrollo, estándares, protocolos y tecnologías de red, capacidad es limitada por los dispositivos. Se propuso comparar el tiempo de procesamiento de las aplicaciones de software para dispositivos móviles. Teniendo en cuenta los escenarios de prueba utilizados que comprendían los dos más grandes sistemas operativos que dominan el mercado de dispositivos móviles, Android e iOS. Se utilizaron diferentes dispositivos móviles para comparar el rendimiento de las aplicaciones creadas con enfoques de desarrollo nativos y multiplataforma. Los resultados obtenidos

permiten concluir que el enfoque nativo utilizado en iOS es mucho más eficiente que el utilizado en Android. También se concluyó que, el enfoque de desarrollo web multiplataforma sería una opción conveniente para lograr un buen rendimiento en todos los dispositivos móviles, independientemente de su sistema operativo.

En su trabajo de investigación *Progressive web apps: An alternative to the native mobile Apps* realizado por (Fortunato & Bernardino, 2018) describen un estudio de las PWAs como tipo de solución, también estudian su evolución y analizan los próximos avances de esta tecnología, tratando de entender si este tipo de soluciones pueden incluso terminar con los desarrollos nativos, también se presentan las ventajas de las tecnologías web con el fin de entender la mejor solución para cada tipo de proyecto. Afirman que el desarrollo de apps puede ser demasiado costosos, particularmente en el caso de desarrollar de forma nativa para cada plataforma. Se llegó a la conclusión de que, siempre que se requiere una interacción con el hardware (GPS o cámara), la tecnología WEB sigue sin ser factible. Sin embargo, una aplicación móvil que sólo necesita presentar una interfaz nativa y presentación de contenido, las soluciones WEB son soluciones viables. También es posible proporcionar una aplicación de muy buena calidad y optimizada para cualquier dispositivo que desarrolle una sola aplicación web progresiva (PWA).

En su trabajo de investigación titulado *A Progressive Web Application Based on Microservices Combining Geospatial Data and the Internet of Things* realizado por (Mena, Corral, Iribarne, Criado, 2019) inicialmente nos dicen que la información relacionada con los dispositivos IoT se muestra comúnmente en paneles para fines de monitoreo y control, y esto lleva que a menudo no se combinen con otros tipos de datos. Además, se menciona que es importante basar la información en la ubicación que se muestra en el contexto del usuario.

Por esta razón proponen el uso de una arquitectura de software basada en microservicios y micro interfaces para ayudar al usuario en la adquisición amigable y sin problemas de datos e información geoespaciales relacionados con el IoT. La solución organiza esos microservicios y una aplicación web

progresiva (PWA) basada en componentes. En resumen, las principales contribuciones son: Arquitectura basada en microservicios, integrando diferentes tecnologías y un método de orquestación para combinar IoT e información georreferenciada. Integración de dispositivos IoT en una interfaz de usuario basada en componentes. El emplear un modelo de datos establecido en gráficos con un algoritmo de selección capaz de reunir componentes basados en información de contexto del usuario. Una interfaz de usuario basada en componentes de micro frontend junto con PWA. Por último, como conclusión se obtuvo que se pudo superar varios problemas, como desarrollar una arquitectura basada en microservicios, elegir el número óptimo de esos microservicios, integrar las diferentes tecnologías. Al mismo tiempo, mostramos datos en tiempo real basados en dispositivos IoT. Además, tuvimos que construir las bases sobre las cuales los componentes podrían agregarse y expandirse según las nuevas categorías o regiones que se admitirían. Al mismo tiempo, el uso basado en este modelo hace que nuestro algoritmo de selección de componentes sea bastante rápido, ya que, en este contexto, las relaciones entre las entidades son más importantes que las propias entidades. Con los PWA, la barrera tecnológica de la compatibilidad de aplicaciones front-end con dispositivos móviles se supera fácilmente simplemente ajustando una aplicación web y haciéndola híbrida, por lo que el alcance podría ampliarse, llegando así a más usuarios.

En su trabajo de investigación titulado Progressive Web Application Assessment Using AHP realizado por (Khan, Al-Badi, & Al-Kindi, 2019) Afirman que los desarrolladores de aplicaciones móviles han estado enfrentando restricciones de configuración de hardware y comunicación de red en la arquitectura de software basada en web nativa, híbrida y móvil durante mucho tiempo. Se propuso la idea de la arquitectura de aplicaciones web progresivas. Mediante un estudio se compara la arquitectura nativa, híbrida, web móvil y PWA basada en tres factores importantes de aceptación del usuario del tamaño de la aplicación, el soporte multiplataforma y la disponibilidad fuera de línea utilizando Proceso de jerarquía analítica (AHP). De la evaluación general de los

cuatro tipos de arquitectura de desarrollo de aplicaciones móviles utilizando la técnica AHP dio como resultado en el criterio de tamaño de software, soporte multiplataforma y accesibilidad fuera de línea que los pesos finales de los cuatro tipos de la arquitectura de desarrollo de aplicaciones móviles es la siguiente; nativo (0.12), híbrido (0.16), web móvil (0.17) y PWA (0.26). Por lo tanto, se llega a la conclusión que mediante la evaluación AHP, la arquitectura PWA debido a la característica única del modo fuera de línea, accesibilidad, tamaño pequeño y soporte multiplataforma es superior las demás arquitecturas de desarrollo.

1.3. Teorías relacionadas al tema.

1.3.1. Fundamentos de la Web

Los navegadores web, servidores y aplicaciones web relacionadas del mundo se comunican entre sí a través de HTTP, el Protocolo de transferencia de hipertexto. HTTP es el lenguaje común de la Internet global moderna. Las aplicaciones web usan HTTP para comunicarse y tendrá una idea aproximada de cómo HTTP hace su trabajo. (Gourley & Totty, 2002). Los tres puntos principales sobre los que se basa la arquitectura lógica de la web observada desde una forma técnica son:

A. Identificadores únicos (URI)

Cada recurso del servidor web tiene un nombre, por lo que los clientes pueden señalar qué recursos les interesan. El nombre del recurso del servidor se denomina identificador uniforme de recursos o URI. Los URI son como las direcciones postales de Internet, identificando y localizando recursos de información en todo el mundo. Aquí hay un URI para un recurso de imagen en el servidor web de la tienda de hardware de Joe: <http://www.joes-hardware.com/specials/saw-blade.gif>. La Figura 1 muestra cómo el URI especifica el protocolo HTTP para acceder al recurso GIF de hoja de sierra en el servidor de la tienda de Joe. Dado el URI, HTTP puede recuperar el objeto.

Los URI vienen en dos tipos, llamados URL y URN. Echemos un vistazo a cada uno de estos tipos de identificadores de recursos ahora (Gourley & Totty, 2002).



*Figura 1 Las URL especifican el protocolo, el servidor y el recurso local.
Fuente: (Gourley & Totty, 2002)*

B. Protocolo de transmisión de datos HTTP

Existe una enorme cantidad de imágenes JPEG, páginas HTML, archivos de texto, películas MPEG, archivos de audio WAV, applets de Java y más que navegan por Internet a cada momento. HTTP distribuye gran parte de esta información de manera rápida, conveniente y confiable desde servidores web de todo el mundo a navegadores web en los escritorios de las personas. Debido a que HTTP utiliza protocolos confiables de transmisión de datos, asegura que su información no se dañará ni se mezclarán en tránsito, incluso cuando provengan del otro lado del mundo. Esto es bueno para usted como usuario, ya que puede acceder a la información sin preocuparse por su integridad (Gourley & Totty, 2002).

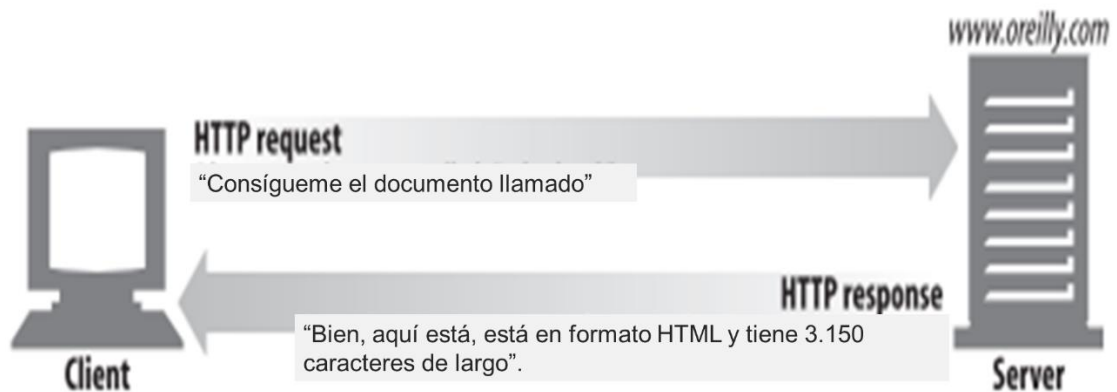


Figura 2 Protocolo HTTP (Cliente. - Servidor). Fuente: (Gourley & Totty, 2002)

Gourley & Totty, (2002) HTTP admite varios comandos de solicitud diferentes, llamados métodos HTTP. Cada mensaje de solicitud HTTP tiene un método. El método le dice al servidor qué acción realizar (buscar una página web, ejecutar un programa de puerta de enlace, eliminar un archivo, etc.). Las siguientes peticiones de información que define HTTP son:

GET Enviar recurso nombrado desde el servidor al cliente. POST Enviar datos del cliente a una aplicación de puerta de enlace del servidor. HEAD Envíe solo los encabezados HTTP de la respuesta para el recurso nombrado. PUT Almacenar datos del cliente en un recurso de servidor con nombre. El mensaje TRACE ofrece a los clientes una forma de verificar la ruta de red a un servidor. DELETE Eliminar el recurso nombrado de un servidor. Los clientes pueden usar un mensaje de OPTIONS para descubrir qué capacidades admite un servidor. CONNECT El estándar no define cómo funciona, excepto para indicar que está destinado a soportar túneles (Gourley & Totty, 2002).

Los códigos esenciales de respuesta para una petición HTTP son:

Tabla 1 Principales códigos de respuesta HTTP

200	La solicitud se ha efectuado adecuadamente.
201	Recurso creado.
301	Recurso se trasladó a otra dirección
401	No autorizado
403	Prohibido
404	Recurso no encontrado

Fuente: Sansano M., (2017)

C. Lenguaje universal para describir HTML

Lo fascinante de HTML es que se basa en un archivo de texto simple. No es magia, es un marcado de hipertexto. Las etiquetas de marcado le indican a su navegador cómo mostrar la página. Parece que el mundo de Internet cambia completamente cada pocos años. Estamos al borde del cambio ahora. Desde principios de la década de 1990, HTML ha evolucionado a través de varias especificaciones de lenguaje. La última especificación HTML, HTML 4.01, le permitió utilizar etiquetas en mayúsculas y minúsculas. La próxima generación de HTML es XHTML, que significa lenguaje de marcado de hipertexto extensible. La primera especificación XHTML, XHTML 1.0, fue recomendada como estándar por el Consorcio W3C World Wide Web en 2000 (Baker, 2003).

HTML, como se escribió originalmente, nunca tuvo la intención de permitir el formateo. HTML estaba destinado a definir el contenido de un documento. Cuando se aprobó HTML 3.2, los atributos y las etiquetas de formato de repente se volvieron utilizables. Ya no se vio obligado a utilizar texto predeterminado normal en un color negro predeterminado. Sin embargo, el desarrollo de sitios que utilizan fuentes e información de color para cada texto que aparece en una página es un proceso largo y tedioso. Piense en el proyecto de Generador de confianza que acaba de completar. Cada vez que agrega texto a la página, también agrega atributos para las fuentes y los colores. De acuerdo, puedes copiar y pegar esta información de línea a línea (¡y espero que lo hayas hecho!),

Pero sigue siendo un proceso lento que está sujeto a muchos errores simples (Baker, 2003).

La descripción repetitiva de los atributos cambió con HTML 4.0. Toda la información de formato requerida para un documento podría moverse del documento HTML a una hoja de estilo separada. El contenido del documento permanece en la página HTML; la información de estilo se mueve a un archivo de hoja de estilo en cascada (CSS) (Baker, 2003).

1.3.1.1 El surgimiento de la web social.

Aunque Tim Berners-Lee imaginó una web de lectura / escritura (el primer navegador también funcionaba como editor de HTML), la Web era un espacio donde la mayoría de cibernautas tenían el permiso de solo lectura. La web de la década de 1990 se parecía mucho a la mezcla de las páginas amarillas y una guía telefónica (publicaciones individuales y listas corporativas) y no obstante el gran dominio de conexión de los hipervínculos infundió limitado aceptación de comunidad entre sus usuarios. Esta actitud pasiva hacia la Web se vio interrumpida por una serie de cambios en el uso de patrones y tecnología que ahora se conocen como Web 2.0, una palabra de moda tomada por Tim O'Reilly. La primera ola de socialización en la Web se debió a la aparición de blogs, wikis y otras formas de comunicación y colaboración basadas en la web. Blogs y Los wikis atrajeron popularidad masiva alrededor de 2003 (ver Figura N°3) (Mika, 2007).

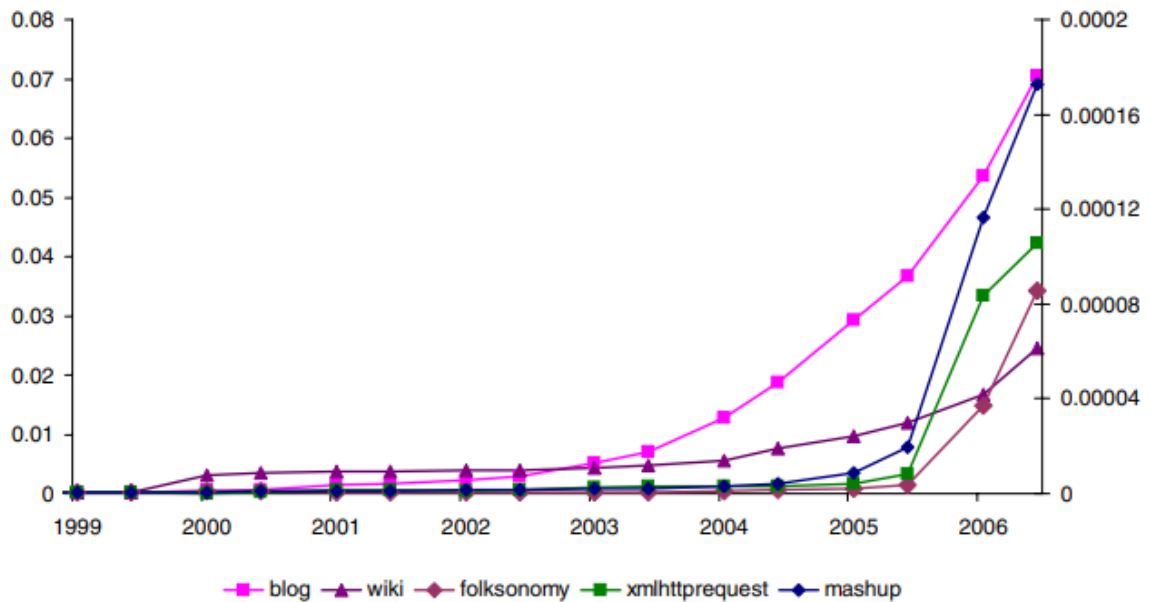


Figura 3 Desarrollo de la web social. Fuente: (Mika, 2007).

Lo que tienen en común es que ambos reducen significativamente los requisitos para agregar contenido a la Web: la edición de blogs y wikis no requirió ningún conocimiento de HTML. Más los blogs y wikis permitieron a individuos y grupos reclamar su espacio personal en la Web y llenarlo de contenido con relativa facilidad. Aún más importante, a pesar de que los blogs han sido evaluados por primera vez como publicaciones puramente personales (similares a los diarios), hoy en día la blogósfera es largamente reconocida como una densamente interconectada red social mediante de la cual las noticias, ideas e influencias viajan rápidamente como bloggers hacer referencia y reflexionar sobre las publicaciones de cada uno (Mika, 2007).

1.3.1.2 Web 1.0

Se inició como una agrupación de documentos inmóviles relacionados para su descarga o consulta, donde sus usuarios solo tenían como único permiso la lectura de los contenidos dados por el editor o autor (Lerma-Blasco et al., 2013).

1.3.1.3 Web 1.5

Ya por el año de 1997 aparecieron así los llamados CGI que significa interfaz de entrada común, estas eran aplicaciones fueron escritas para ser ejecutadas en el servidor, y de esta manera podían recibir datos requeridos de los clientes. Su esfuerzo fue de gran provecho, pero tendían a acumular el servidor, de ahí que surgió la idea de usar sistemas de ejecución de módulos más integrados en el servidor y lenguajes de programación interpretados que permitían incluir código en las páginas HTML (Lerma-Blasco et al., 2013).

1.3.1.4 Web 2.0.

En el 2003 las tecnologías ya mencionadas dieron paso a las aplicaciones Web 2.0, aquí los usuarios comienzan a actuar de una forma colaborativa entre ellos, el consumo pasa a poder ser push, que quiere decir, con tecnologías RSS (Really Simple Syndication. Esto conlleva que no necesariamente los usuarios tienen que buscar los nuevos y más recientes contenidos e incluso las actualizaciones. Ahora el sentido de los recursos , empezaría a ser bidireccional y estos también serían más fáciles de consumir y producir (Lerma-Blasco et al., 2013)

O'Reilly (2007) también da a conocer que con la llega de la Web 2.0 el usuario podría generar información para los sitios Web.

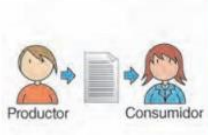
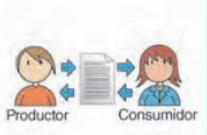
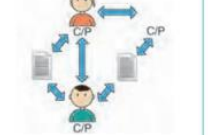


Web 1.0	Web 1.5	Web 2.0	Social media	Web semántica
				
estática	dinámica	El C/P es el Prosumidor colaborativa	social	inteligente
1989 – 1997	1997 – 2003	2003 – 2008	2008 – Actual	En un futuro cercano

Figura 4 Evolución de la Web. Fuente: (Lerma-Blasco et al., 2013)

1.3.2 Aplicaciones web

Es un sistema de software centrado en tecnologías y estándares del Consorcio Mundial de la red (W3C) que facilita los recursos característicos de la Web, como contenido y servicios mediante de una interfaz de usuario, el navegador web. Esta definición incluye explícitamente tecnologías, así como la interacción del usuario. De esto podemos concluir que las tecnologías por sí solas, como los servicios web, no son aplicaciones web, pero pueden ser parte de una. Además, esta definición implica que los sitios web sin componentes de software, como las páginas HTML estáticas, tampoco son aplicaciones web. Por supuesto, se pueden concebir definiciones más amplias que podrían incluir servicios web y sitios web (Kappel G., 2006).

Las aplicaciones web usan herramientas o tecnologías habilitadoras para disminuir su contenido y permitir que los usuarios del sistema afecten la lógica empresarial en el servidor. Para todas las aplicaciones web, excepto las más simples, el usuario necesita impartir más que solo información de solicitud de navegación. Por lo general, los usuarios de aplicaciones web ingresan un rango variado de datos de entrada: texto simple, selecciones de casillas de verificación o incluso información binaria y de archivo (Conallen, 2002).

1.3.2.1 Características de las aplicaciones web

Las aplicaciones web difieren de las aplicaciones tradicionales no basadas en la web en una variedad de características que vale la pena considerar. Estas son características que las aplicaciones tradicionales carecen por completo (por ejemplo, navegación no lineal) por un lado y características que son de particular importancia en aplicaciones web por otro lado (por ejemplo, frecuencia de actualizaciones). Si una determinada característica está presente y hasta qué punto depende en parte del tipo de aplicación web: el desarrollo de aplicaciones web transaccionales como los sistemas de comercio electrónico requiere un mayor enfoque en el contenido actualizado y consistente en comparación con los sistemas de suministro de información pura por ejemplos

las exposiciones virtuales. Estas características son la razón por la cual muchos métodos, herramientas, conceptos y técnicas de la Ingeniería de Software tradicional tienen que adaptarse a las necesidades de la Ingeniería Web o incluso pueden ser totalmente inadecuados (Kappel G., 2006).



Figura 5 Dimensiones según ISO / IEC 9126-1 para la categorización de características de aplicaciones web. Fuente: (Kappel G., 2006).

En la Figura N° 5 se observa una visión general de estas características y organiza el dobladillo a lo largo de las tres dimensiones: "uso", "desarrollo" y "producto" con su "evolución" como una dimensión global. Estas dimensiones se basan en el estándar ISO / IEC 9126-1 para la evaluación de las características de calidad del software. Al asignar las diferentes características de las aplicaciones web a estas dimensiones, también podemos ver su influencia en la calidad de las aplicaciones y, por lo tanto, tomar las características se empieza definiendo los requisitos de ingeniería web. Además de las características relacionadas con el producto, el uso y el desarrollo, tenemos la evolución como una cuarta dimensión que rige las otras tres dimensiones. Los productos deben ser adaptables, se debe considerar la nueva

información contextual durante el uso, y el desarrollo enfrenta condiciones continuamente cambiantes (Kappel G., 2006).

1.3.2.2 El navegador web

Conocido también como explorador web, el cual permite al usuario el acceso a una página web, siempre y cuando conozca su dirección exacta llamada URL. También se afirma que un navegador web es la puerta de acceso a algunos de los servicios que puede ofrecer la web. Es importante recordar que la web trabaja en una arquitectura de Cliente – Servidor, este servidor es donde se atiende a las solicitudes de los clientes a través de los navegadores, facilitando el uso de los recursos mediante el protocolo HTTP (Lerma, Murcia, & Mifsud, 2013).

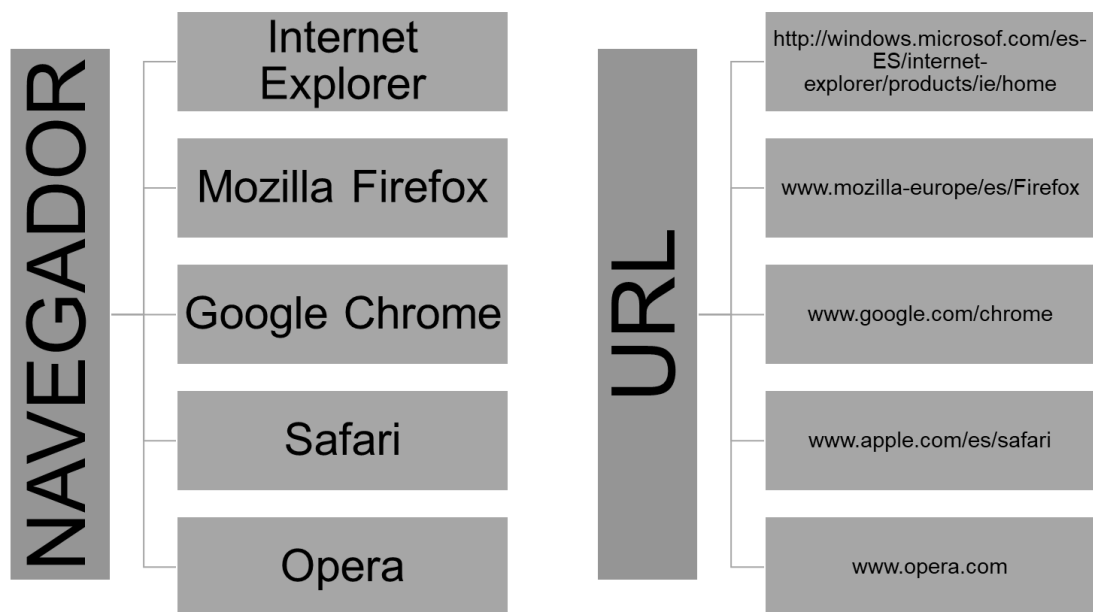


Figura 6 Páginas web de los principales navegadores. Fuente: Lerma, Murcia, & Mifsud Talón, (2013)

A. Navegadores web de escritorio

Según Lerma, Murcia, & Mifsud, (2013) afirman que los principales y navegadores más usados son los siguientes:

Internet Explorer (IE). Uno de los primeros navegadores web creados en el año de 1995. Este navegador era solo de uso exclusivo para los equipos con sistema operativo Microsoft Windows. Cabe mencionar que en su tiempo fue el más utilizado, aunque por el pasar del tiempo ha perdido su popularidad (Lerma, Murcia, & Mifsud, 2013).

Mozilla Firefox. Navegador web libre y de código abierto, creado por el personal programador de la Fundación Mozilla. Es innovador ya que adapta inmediatamente los estándares que van apareciendo, también permite estar operativos en diferentes sistemas operativos como Windows, Mac y GNU/Linux. (Lerma, Murcia, & Mifsud, 2013)

Google Chrome. En cambio, este navegador posee un software privado o de código cerrado el cual es perteneciente a la Compañía de Google. A pesar de ser privado este se basó en componentes y proyectos de código abierto. Es importante mencionar que su interfaz de usuario es muy eficiente y sencilla, permitiendo también una mejor experiencia: más rápida, estable y segura. (Lerma, Murcia, & Mifsud, 2013)

Safari. La compañía Apple no quiso quedarse atrás así que lanzó su propio navegador que inicialmente era para sus equipos, ahora existe una versión para Windows. Esta se distingue por su diseño cómodo, atractivo e intuitivo, en la que se destaca el reproductor de Apple QuickTime. (Lerma, Murcia, & Mifsud, 2013)

Opera. Y por último entre las más destacadas esta la creada por Opera Software, la cual ha tenido una gran aceptación esto se debe a su seguridad, velocidad, soporte de estándares y reducido tamaño (Lerma, Murcia, & Mifsud, 2013).

B. Navegadores webs móviles

Con el gran creciente uso de los dispositivos inteligentes (SMART), las empresas se han hallado con la necesidad de adaptar sus navegadores a estos nuevos dispositivos, siendo este el medio más utilizado para acceder a las diferentes aplicaciones web, como también navegar por la red. Muchos de ellos ya vienen incluidos con sus propios sistemas operativos móviles. (Lerma, Murcia, & Mifsud, 2013)

Sistemas operativos móviles	 android	 Apple	 BlackBerry	 Windows phone
Repositorios oficiales de aplicaciones	https://play.google.com	http://itunes.apple.com/es/genre/ios/id36?mt=8	http://appworld.blackberry.com	http://www.windowsphone.com/es-ES/marketplace
Cantidad de apps disponibles en 2012	+ de 350.000	+ de 500.000	+ de 100.000	+ de 100.000

Figura 7 Principales sistemas operativos (Lerma, Murcia, & Mifsud, 2013)

C. Otros navegadores web

Y por último es importante mencionar que están los dispositivos externos llamados Set-top Box (STB) o internos para los televisores, denominados Internet TV o Smart TV, de manera que estos cumplen la misma función de conectarse a Internet y utilizar la infinidad de aplicaciones compatibles. En este ámbito, la compañía Opera fue la que empezó con la idea de adaptar su navegador a consolas de videojuegos como Nintendo DS y Wii, de esta manera pudo encontrar hoy en día la unión con los televisores de Sony Bravia, entre otros. (Lerma, Murcia, & Mifsud, 2013)



Figura 8 Máquina virtual con el emulador del navegador Opera TV. Fuente: (Lerma, Murcia, & Mifsud, 2013).

1.3.3 Los frameworks

Reduce el desarrollo de aplicaciones al automatizar muchos de los patrones empleados para un propósito determinado. Un framework también agrega estructura al código, lo que hace que el desarrollador escriba un código mejor, más legible y más fácil de mantener. Finalmente, un framework facilita la programación, ya que empaqueta operaciones complejas en declaraciones simples (Zaninotto & Potencier, 2007) .

Según (Kappel, Pröll, Reich, & Retschitzegger, 2006) menciona que los frameworks representan otra opción para reutilizar el conocimiento arquitectónico existente. Un framework es un sistema de software reutilizable con funcionalidad general ya implementada. El marco se puede especializar en una aplicación lista para usar. El marco sirve como modelo para la arquitectura básica y las funcionalidades básicas para un campo específico de aplicación. Esto significa que el conocimiento arquitectónico contenido en un marco puede ser totalmente adoptado en la aplicación.

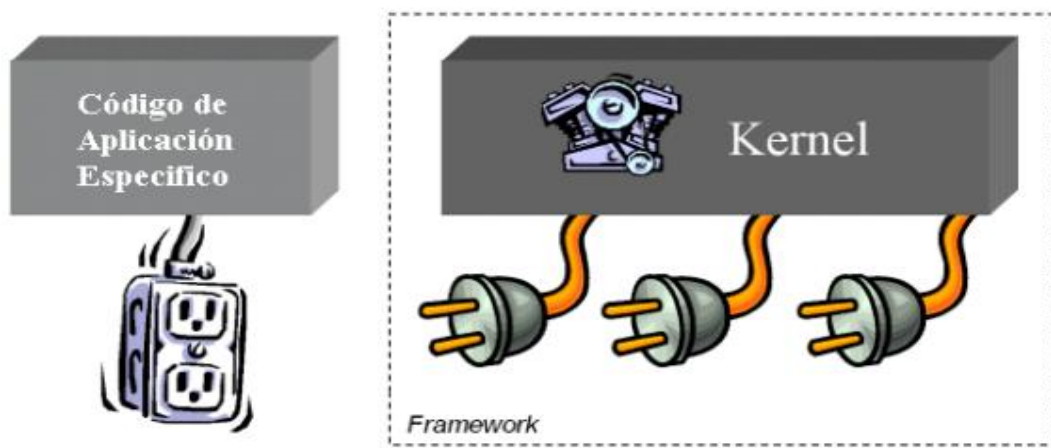


Figura 9 Escenario de un Framework. Fuente: (Lerma, Murcia, & Mifsud, 2013).

Sin embargo, los beneficios de un marco, es decir, la simple reutilización de la arquitectura y la funcionalidad, tiene que sopesarse con sus inconvenientes, es decir, un alto grado de esfuerzo de capacitación, falta de estándares para la integración de diferentes marcos y la consiguiente dependencia de los fabricantes (Kappel, Pröll, Reich, & Retschitzegger, 2006).

1.3.3.1 Características

La mayoría de frameworks existentes poseen algunas características en común los cuales se presenta en la siguiente imagen:

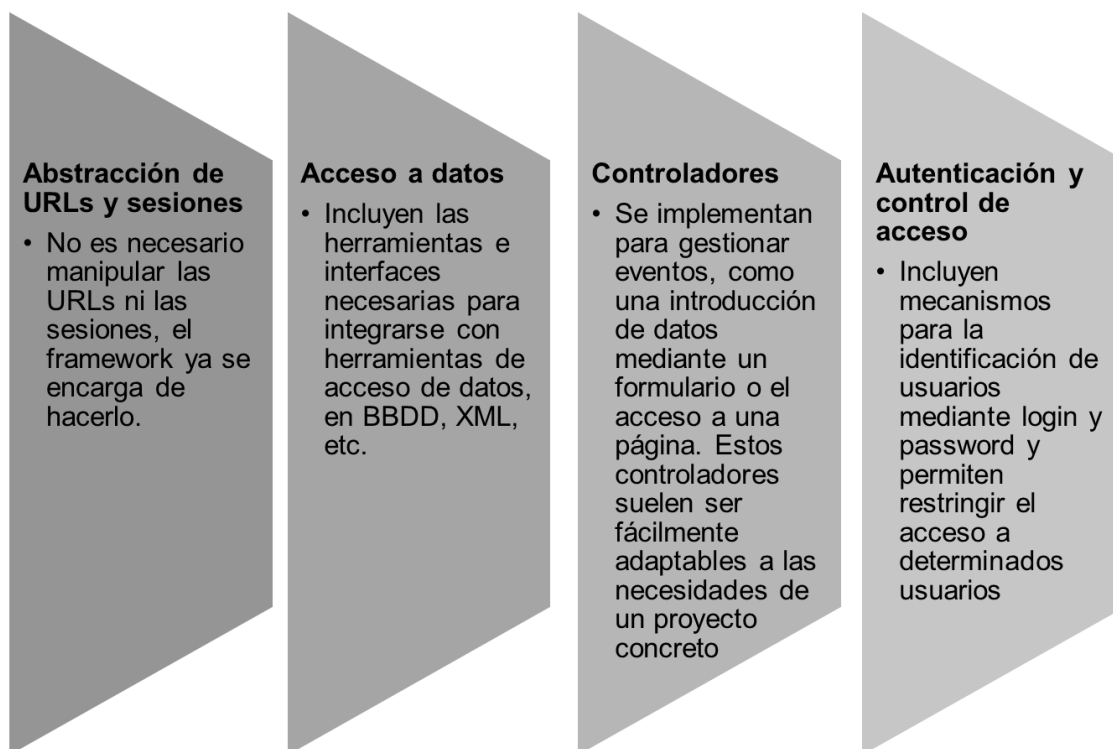


Figura 10 Características comunes en los frameworks existentes. Fuente: (Gutiérrez, 2014)

1.3.3.2 Un catálogo de Frameworks de aplicación

Una gran cantidad de frameworks son proyectos de código abierto y están activamente en desarrollo, por lo que es importante ver qué ha cambiado y qué hay de nuevo cuando revisa el marco usted mismo (Nash, 2003).

A. Frameworks de aplicación completos

Los marcos de aplicaciones completos brindan servicios para todas las partes del desarrollo de una aplicación típica. Normalmente, esto incluye al menos presentación, persistencia, configuración, lógica de la aplicación y registro. A menudo incluye mucho más. A menudo es difícil trazar la línea entre un marco de aplicación completo y uno que se concentra en un área específica de la aplicación, ya que depende en gran medida de la aplicación. Si una aplicación tiene muy poca o ninguna necesidad de persistencia o acceso a la base de datos, por ejemplo, un marco como Struts, que se concentra bastante en el lado

de la presentación del proceso de desarrollo de la aplicación, podría considerarse un marco de aplicación completo en ese contexto (Nash, 2003).

B. Frameworks de presentación

¿Qué es un Framework de presentación? En la definición más básica, un Framework de presentación es cualquier biblioteca o marco que proporciona herramientas que hacen que sea más fácil para un desarrollador construir la interfaz de usuario o el lado de presentación de una aplicación web. En su desarrollo, probablemente ya haya encontrado el problema: la forma más fácil de resolver los problemas de presentación a menudo resulta ser la peor, a largo plazo. Cuando Java Servlets apareció por primera vez, fue lo suficientemente simple como para incrustar un estado de "impresión" que escribió de nuevo en la salida de respuesta HTTP con HTML (o WML) directamente (Nash, 2003).

C. Otros frameworks de aplicación específicos

Ahora veremos algunos de los marcos específicos de aplicaciones más populares: marcos diseñados para manejar un tipo particular de dominio de problemas (como la contabilidad) o para tratar una parte específica del proceso de desarrollo, como la presentación, la base de datos persistencia o manejo de XML. Algunos de estos marcos limitan con las API, ya que proporcionan una paleta de servicios que requiere otros elementos del marco antes de que puedan ensamblarse en una aplicación completa: la línea entre una biblioteca y un marco se desdibuja un poco con los proyectos específicos de la aplicación (Nash, 2003).

1.3.3.3 Frameworks Front-end

A. Vue

En menos de cuatro años desde su primer lanzamiento, Vue se ha convertido en uno de los proyectos de código abierto más populares en GitHub. Esta popularidad se debe en parte a sus potentes características, pero también a su énfasis sobre la experiencia del desarrollador y la facilidad de adopción.

La biblioteca principal de Vue.js, como React, es solo para manipular la capa de vista desde el patrón arquitectónico MVC. Sin embargo, Vue cuenta con dos bibliotecas oficiales de soporte, Vue Router y Vuex, responsables de enrutamiento y gestión de datos respectivamente. Vue no es apoyado por un gigante tecnológico en la forma en que React y Angular son y se basa en donaciones de un pequeño número de clientes corporativos y usuarios dedicados de Vue. Aún más impresionante, Evan You es actualmente el único desarrollador de Vue a tiempo completo, aunque un equipo central de 20 desarrolladores más de todo el mundo ayuda con el desarrollo, el mantenimiento y la documentación (Gore, 2015).

Los principios clave de diseño de Vue son los siguientes:

Enfoque: Vue ha optado por una API pequeña y enfocada, y su único propósito es la creación de IE

Simplicidad: La sintaxis de Vue es escasa y fácil de seguir

Compactibilidad: El script de la biblioteca principal es de 25 KB minificado, lo que lo hace más pequeño que React e incluso jQuery

Velocidad: Los puntos de referencia de renderización superan a muchos de los principales marcos, incluyendo React

Versatilidad: Vue funciona bien para trabajos pequeños donde normalmente podría usar jQuery, pero puede escalar verticalmente como una solución SPA legítima

Plantillas

De forma predeterminada, Vue utilizará un archivo HTML para su plantilla. Un script incluido declarará una instancia de Vue y utilizará la propiedad `el` en el objeto de configuración para indicar a Vue dónde se montará la aplicación en la plantilla (Gore, 2015).

```

<div id="app">
  <!--Vue has dominion within this node-->
</div>
<script>
  new Vue({
    el: '#app'
  });
</script>

```

*Figura 11 Configuración para indicar aVue donde montara la aplicación.
Fuente: (Gore, 2015)*

Podemos enlazar datos a nuestra plantilla creándolos como una propiedad de datos y utilizando la sintaxis de bigote para imprimirlo en la página:

```

<div id="app">
  {{ message }}
  <!--Renders as "Hello World"-->
</div>
<script>
  new Vue({
    el: '#app',
    data: {
      message: 'Hello World'
    }
  });
</script>

```

Figura 12 Usando sintaxis de bigote. Fuente: (Gore, 2015)

Directivas

De forma similar a Angular, podemos agregar funcionalidad a nuestras plantillas mediante directivas. Estas son propiedades especiales que agregamos a las etiquetas HTML a partir del prefijo v-(Gore, 2015). Digamos que tenemos una serie de datos. Podemos representar estos datos en la página como elementos HTML secuenciales mediante la directiva v-for:

```

<div id="app">
  <h3>Grocery list</h3>
  <ul>
    <li v-for="grocery in groceries">{{ grocery }}</li>
  </ul>
</div>
<script>
var app = new Vue({
  el: '#app',
  data: {
    groceries: [ 'Bread', 'Milk' ]
  }
});
</script>

```

Figura 13 Directiva v-for. Fuente: (Gore, 2015).

El código anterior se representa de la siguiente manera:

```

<div id="app">
  <h3>Grocery list</h3>
  <ul>
    <li>Bread</li>
    <li>Milk</li>
  </ul>
</div>

```

Figura 14 Representación v-for Fuente: (Gore, 2015).

Reactividad

Una característica clave del diseño de Vue es su sistema de reactividad. Al modificar datos, la vista se actualiza automáticamente para reflejar ese cambio(Gore, 2015).

Por ejemplo, si creamos una función que empuja otro elemento a nuestra matriz de artículos de comestibles después de que la página ya ha sido representada, la página se volverá a representar automáticamente para reflejar ese cambio:

```

setTimeout(function() {
  app.groceries.push('Apples');
}, 2000);

```

Figura 15 Creación de función. Fuente: (Gore, 2015).

Dos segundos después de la representación inicial, vemos esto:

```

<div id="app">
  <h3>Grocery list</h3>
  <ul>
    <li>Bread</li>
    <li>Milk</li>
    <li>Apples</li>
  </ul>
</div>

```

Figura 16 Cambio de función inicial. Fuente: (Gore, 2015)

Componentes

Los componentes amplían los elementos HTML básicos y le permiten crear sus propios elementos personalizados reutilizables (Gore, 2015). Por ejemplo, aquí he creado un elemento personalizado, <grocery-item>, que se representa como un . El elemento secundario de texto de ese nodo se obtiene desde una propiedad HTML personalizada, <title>, a la que se puede acceder desde el código del componente:

```

<div id="app">
  <h3>Grocery list</h3>
  <ul>
    <grocery-item title="Bread"></grocery-item>
    <grocery-item title="Milk"></grocery-item>
  </ul>
</div>
<script>
  Vue.component( 'grocery-item', {
    props: [ 'title' ],
    template: '<li>{{ title }}</li>'
  });

  new Vue({
    el: '#app'
  });
</script>

```

Figura 17 Creación de elemento grocery-item. Fuente: (Gore, 2015)

Esto se representa de la siguiente manera:

```

<div id="app">
  <h3>Grocery list</h3>
  <ul>
    <li>Bread</li>
    <li>Milk</li>
  </ul>
</div>

```

Figura 18 Uso del componente. Fuente: (Gore, 2015)

Pero probablemente la razón principal para usar componentes es que hace que sea más fácil diseñar una aplicación más grande. La funcionalidad se puede dividir en componentes reutilizables e independientes (Gore, 2015).

B. AngularJS

AngularJS comenzó a aportar primero estructura y consistencia a la web de una sola página desarrollo de aplicaciones, al tiempo que proporciona una forma de desarrollar rápidamente aplicaciones web mantenibles. En el tiempo transcurrido desde su lanzamiento, la web y los navegadores las personas han avanzado a pasos agigantados y algunos de los problemas que AngularJS estaba resolviendo que ya no eran tan relevantes (Seshadri, 2018).

Angular era básicamente una versión reescrita completamente nueva del marco, construido para la web de la nueva era. Aprovechó muchos de los avances más recientes, desde módulos hasta componentes web, al tiempo que mejora las características existentes de AngularJS, como depend-Incy inyección y plantillas. Angular como marco proporciona algunas ventajas significativas al tiempo que proporciona una estructura común para los desarrolladores en un equipo para trabajar. Nos permite desarrollar grandes aplicaciones de manera mantenible (Seshadri, 2018).

Angular tiene el mejor diseño, conjunto de características, y trayectoria en este escrito. Estos son algunos de los beneficios de Angular:

- Enlace de datos: Angular tiene un método limpio para enlazar datos a elementos HTML utilizando su poderoso mecanismo de alcance.
- Extensibilidad: la arquitectura angular le permite extender fácilmente casi cada aspecto del lenguaje para proporcionar sus propias implementaciones personalizadas.
- Limpio: Angular te obliga a escribir código limpio y lógico.

- Código reutilizable: la combinación de extensibilidad y código limpio lo hace fácil escribir código reutilizable en Angular. De hecho, el idioma a menudo te obliga a hacerlo así que al crear servicios personalizados.
- Soporte: Google está invirtiendo mucho en este proyecto, lo que le da una ventaja sobre otras iniciativas similares.
- Compatibilidad: Angular se basa en TypeScript, lo que hace que sea más fácil comenzar integrando Angular en su entorno y reutilizando piezas de su existente código dentro de la estructura del marco angular.

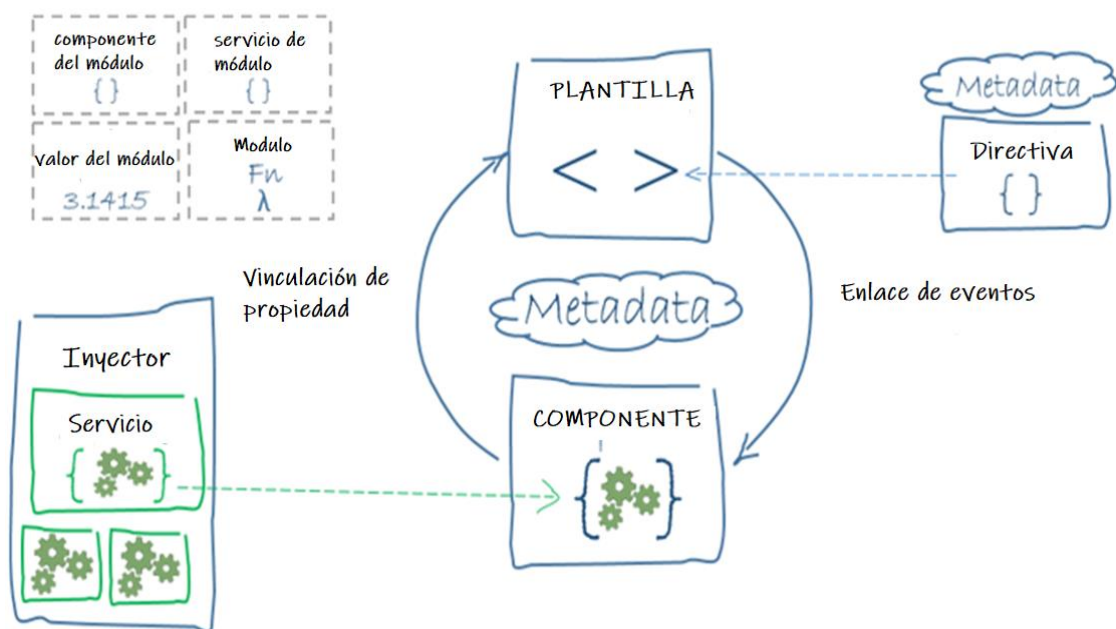


Figura 19 Arquitectura del Framework Angular. Fuente: (Seshadri, 2018).

Módulos

En general, las aplicaciones angulares utilizan un diseño modular. Si bien no es obligatorio, los módulos son muy recomendable porque te permiten separar tu código en archivos. Esto le ayuda a mantener sus archivos de código cortos y manejables mientras permite para acceder a la funcionalidad de cada uno. A diferencia de cómo usa módulos con TypeScript, con Angular usted importa externos módulos en la parte superior de un archivo y exportar la funcionalidad que necesita en la parte inferior (Dayley, Dayley, & Dayley, 2018) .

Tú Para ello, utilice los términos clave importación y exportación, con la siguiente sintaxis:

```
Import {Component} from 'angular2/core';  
Export class App{}
```

Figura 20 Sintaxis de Importación y Exportación. Fuente: (Dayley, Dayley, & Dayley, 2018)

Directivas

Dayley, Dayley, & Dayley, (2018) mencionan que las directivas son clases de JavaScript con metadatos que definen la estructura y el comportamiento. Las directivas proporcionan la mayoría de la funcionalidad de la interfaz de usuario para aplicaciones angulares. Existen tres tipos principales de directivas:

Componentes: una directiva de componentes es una directiva que incorpora una plantilla HTML con funcionalidad JavaScript para crear un elemento de IU autónomo que se puede agregar a una aplicación angular como un elemento HTML personalizado. Es probable que los componentes sean las directivas que más usa en Angular (Dayley, Dayley, & Dayley, 2018).

Estructural: utiliza directivas estructurales cuando necesita manipular el DOM. Las directivas estructurales le permiten crear y destruir elementos y componentes desde una vista HTML (Dayley, Dayley, & Dayley, 2018).

Atributo: una directiva de atributo cambia la apariencia y el comportamiento de los elementos HTML mediante el uso de atributos HTML (Dayley, Dayley, & Dayley, 2018).

El enlace de datos

Una de las mejores características de Angular es el enlace de datos incorporado: el proceso de vincular datos de un componente con lo que se muestra en una página web. Angular proporciona una interfaz muy limpia para vincular datos del modelo a elementos en una página web. Cuando los datos

se cambian en una página web, el modelo se actualiza y cuando los datos son cambiados en el modelo, la página web se actualiza automáticamente. De esta manera, el modelo es siempre es la única fuente de datos representada por el usuario, y la vista es solo una Proyección del modelo (Dayley, Dayley, & Dayley, 2018).

Inyección de dependencia

La inyección de dependencias es un proceso en el que un componente define dependencias de otros componentes. Cuando se inicializa el código, el componente dependiente está disponible para acceder dentro del componente. Las aplicaciones angulares hacen un uso intensivo de la inyección de dependencia (Dayley, Dayley, & Dayley, 2018).

Servicios

Los servicios son los caballos de batalla más importantes en el entorno angular. Los servicios son clases singleton que proporcionan funcionalidad para una aplicación web. Por ejemplo, una tarea común de las aplicaciones web es realizar solicitudes AJAX a un servidor web. Angular proporciona un servicio HTTP que alberga toda la funcionalidad para acceder a un servidor web. La funcionalidad del servicio es completamente independiente del contexto o estado, por lo que se puede consumir fácilmente desde los componentes de una aplicación. Angular proporciona muchos componentes de servicio integrados para usos básicos, como solicitudes HTTP, registro, análisis y animación. También puede crear sus propios servicios y reutilizarlos en todo su código (Dayley, Dayley, & Dayley, 2018).

1.3.3.4 Frameworks Back- end

1.3.3.4.1 Laravel

Su patrón de arquitectura es MVC de código abierto para PHP que se utiliza para crear aplicaciones web robustas. Es uno de los frameworks PHP más populares, queridos por su elegante sintaxis y potentes características (Gore, 2015).

También se menciona que es un marco de aplicación web que utilizó algunas de las mejores características de otras populares soluciones marco, entre ellas Ruby on Rails y ASP.NET MVC. Por esta razón, si tiene alguna experiencia trabajando con otros frameworks, entonces imagino que harás una transición bastante elegante a Laravel. Los recién llegados al desarrollo impulsado por el marco tendrán una curva de aprendizaje ligeramente más pronunciada debido a la introducción de nuevos conceptos. Prometo que las características prácticas y fáciles de usar de Laravel Haz que tu viaje sea agradable (Gilmore, 2018).

Laravel es adecuado para crear una variedad de proyectos basados en la web, como los siguientes:

- Sitios web con autenticación de usuario, como un portal de clientes o una red social
- Aplicaciones web, como un recortador de imágenes o un panel de supervisión
- Servicios web, como las API RESTful.

1.3.3.4.2 Symfony

Es un marco completo diseñado para optimizar el desarrollo de aplicaciones web a través de varias características clave. Para empezar, separa las reglas de negocio, la lógica del servidor y las vistas de presentación de una aplicación web. Contiene numerosas herramientas y clases destinadas a acortar el tiempo de desarrollo de una aplicación web compleja. Además, automatiza tareas comunes para que el desarrollador pueda centrarse por completo en los detalles de una aplicación (Zaninotto & Potencier, 2009).

El resultado final de estas ventajas significa que no hay necesidad de reinventar la rueda cada vez que se crea una nueva aplicación web. Symfony fue escrito completamente en PHP 5. Ha sido probado exhaustivamente en varios proyectos del mundo real, y en realidad está en uso para sitios web de

comercio electrónico de alta demanda. Es compatible con la mayoría de los motores de bases de datos disponibles, incluidos MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se ejecuta en plataformas * nix y Windows. Comencemos con un vistazo más de cerca a sus características (Zaninotto & Potencier, 2009).

Clases principales de Symfony

La implementación de MVC en Symfony utiliza varias clases que encontrarás con bastante frecuencia en este libro:

- `sfController` es la clase de controlador. Decodifica la solicitud y la entrega a la acción.
- `sfRequest` almacena todos los elementos de solicitud (parámetros, cookies, encabezados, etc.).
- `sfResponse` contiene los encabezados y contenidos de respuesta. Este es el objeto que eventualmente se convertirá en una respuesta HTML y se enviará al usuario.

El singleton de contexto (recuperado por `sfContext::getInstance()`) almacena una referencia a todos los objetos centrales y la configuración actual; Es accesible desde cualquier lugar (Zaninotto & Potencier, 2009).

Organización del código

Symfony organiza el código en una estructura de proyecto y pone el proyectar archivos en una estructura de árbol estándar. Cada aplicación es un conjunto de uno o más módulos. Un módulo generalmente representa una página o un grupo de páginas con un propósito similar. Por ejemplo, puede tener los módulos inicio, artículos, ayuda, shoppingCart, cuenta, etc. Los módulos contienen acciones, que representan las diversas acciones que se pueden realizar en un módulo. Por ejemplo, un módulo shoppingCart puede tener acciones de agregar, mostrar y actualizar. En general, las acciones pueden ser descritas por un verbo. Tratar con acciones es casi como tratar con páginas en una aplicación web clásica, aunque dos acciones pueden dar como resultado la misma página (por ejemplo, agregar un comentario a una publicación en un

weblog volverá a mostrar la publicación con el nuevo comentario) (Zaninotto & Potencier, 2009).

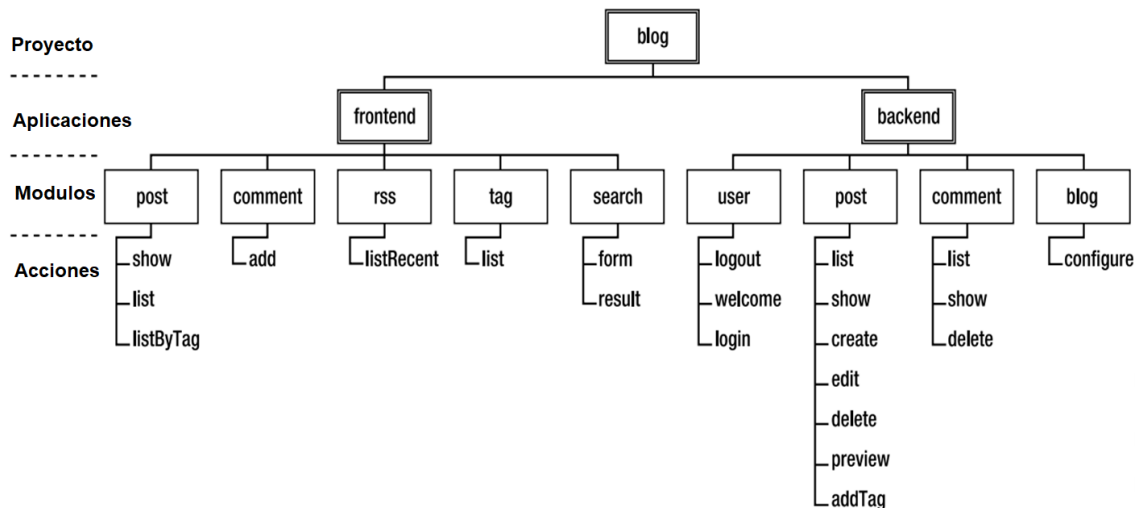


Figura 21 Ejemplo de Organización del código. Fuente: (Zaninotto & Potencier, 2009)

1.3.4 Soluciones móviles

Según Griffith, (2017) existen tres soluciones primarias que están disponibles: aplicaciones web, aplicaciones nativas y aplicaciones híbridas. A continuación, se describen cada una de ellas.

a) Aplicaciones móviles nativas

Se utiliza código nativo según la plataforma en que se quiere desarrollar la aplicación. Se puede utilizar Objective-C o Swift para dispositivos IOS, Java o Kotlin para Android, y C# o XAML para Windows. Las interfaces de programación de aplicaciones (API) y herramientas también son independientes de la plataforma a trabajar. La principal ventaja que estas aplicaciones, es que no utiliza capas intermedias de código que puedan afectar el rendimiento de la aplicación. Sin embargo, el desarrollo es centralizado de acuerdo al sistema operativo (SO), requiere de permisos para el buen funcionamiento y espacio de memoria para instalar la aplicación y actualizaciones posteriores (Griffith, 2017).

b) Aplicaciones web móviles

Estas aplicaciones son desarrolladas utilizando exclusivamente tecnología web ejecutándose mediante el navegador de cualquier dispositivo. Esto permite tener un alcance mucho más amplio, facilita el desarrollo de las actualizaciones automáticamente, sin tener la necesidad de solicitar al usuario permisos para actualizar la aplicación. Sin embargo, por el simple hecho de utilizar el navegador incorporado en un dispositivo móvil trae consigo limitaciones como no poder acceder a todas las funcionalidades del dispositivo; los usuarios están acostumbrados a descargar las aplicaciones desde una tienda de aplicaciones (Griffith, 2017).

c) Aplicaciones híbridas

Son aplicaciones nativas web que usa las chromeless web del navegador para ejecutar la aplicación en cualquier dispositivo. El código puede ser desarrollado para ejecutarse en múltiples plataformas. No es necesario utilizar un nuevo lenguaje de programación, sino más bien se utiliza algún complemento adicional para acceder a las funcionalidades del dispositivo móvil. Frameworks como Ionic o React, tratan de suplir esta brecha que aun que mucho por hacer (Griffith, 2017).

1.3.5 Fundamento de una aplicación web progresiva (PWA)

Los PWA son aplicaciones web que tratan de mejorar paulatinamente la vivencia del usuario, independientemente de sus navegadores, plataformas o dispositivos aprovechando las API modernas de los navegadores, con la finalidad de mejorar la experiencia de usuario, independientemente del marco en el que elija trabajar (Hajian, 2019). Estas aplicaciones presentan las siguientes características:

- Carga instantánea: la aplicación debe cargarse rápidamente y debe ser interactivo.

- Conectividad independiente: Puede funcionar con redes móviles desde 2G en adelante, en un red lenta o conexiones inestables.
- Diseño adaptativo: El diseño de la aplicación debe ser adaptado a las diferentes dimensiones de las pantallas que posee los dispositivos móviles, de escritorio, tablets, etc.
- Re-engaging: Utilizan notificaciones Push para alertar a los usuarios.
- Características similares a las nativas: Utiliza funciones propias del dispositivo haciendo que una aplicación web sea lo más parecido a una aplicación nativa.
- Seguridad: La característica principal de la PWA, solo funcionan bajo el protocolo HTTPS.
- Instalable: Cuando se accede a una PWA desde un dispositivo móvil, se instala automáticamente un icono en la pantalla del dispositivo.
- Progresivo: Independiente de los navegadores y dispositivos la aplicación deberá evolucionar y adoptar nuevas funciones para brindar mejores experiencias de usuario.

1.3.5.1 Arquitectura de un PWA

Es necesario tener en cuenta diferentes elementos para construir una aplicación web progresiva. Estos elementos se describen a continuación:

a) Service worker

Es considerado como el cerebro de toda la arquitectura de una PWA, desarrollado mediante un conjunto de código de JavaScript (.JS) que se ejecutan en segundo plano. Cuando es habilitado un services worker (SW), el desarrollador puede interceptar solicitudes de red, notificaciones push, administrar caches y otras tareas (Hajian, 2019).

“Un service worker es una serie de comandos que tu navegador ejecuta en segundo plano, separado de una página web, abriéndoles la puerta a funciones que no necesitan una página web ni interacción de usuario” (Gaunt, sf).

El funcionamiento de un SW actúa como un proxy entre el sitio web y el servidor teniendo en cuenta las siguientes características:

- Se ejecuta en diferentes subprocesos de JS, diseñado para ser totalmente asíncrono, no tiene acceso a XHR asíncrono y localStorage.
- Se ejecuta únicamente en protocolo HTTPS cuando esta fase de producción. Sin embargo, también se puede ejecutar en Localhost cuando está en fase de desarrollo.
- Cuando se deja de usar se detiene y cuando se le necesita otra vez se reinicia.

Ciclo de vida de Service Worker

Según Hajian, (2019) el ciclo de vida de un SW tiene tres etapas que se describen a continuación:

- Cuando el usuario visita un sitio web, se llama a la función *register()*, el navegador detecta el .js del SW y comienza con la descarga e instalación en segundo plano de todos los archivos estáticos en cache, en caso de no cumplir las directivas establecidas se devuelve un estado de error.
- Después de la instalación comienza la etapa de activación en el cual se administra la cache actual y anteriores.
- Tan pronto como la instalación se haya completado con éxito, Service Worker se activa y tiene el control total bajo su propio alcance. Al igual que en el evento de instalación, la activación solo ocurre por primera vez después del registro y una vez que se ha completado la instalación (Hajian, 2019).

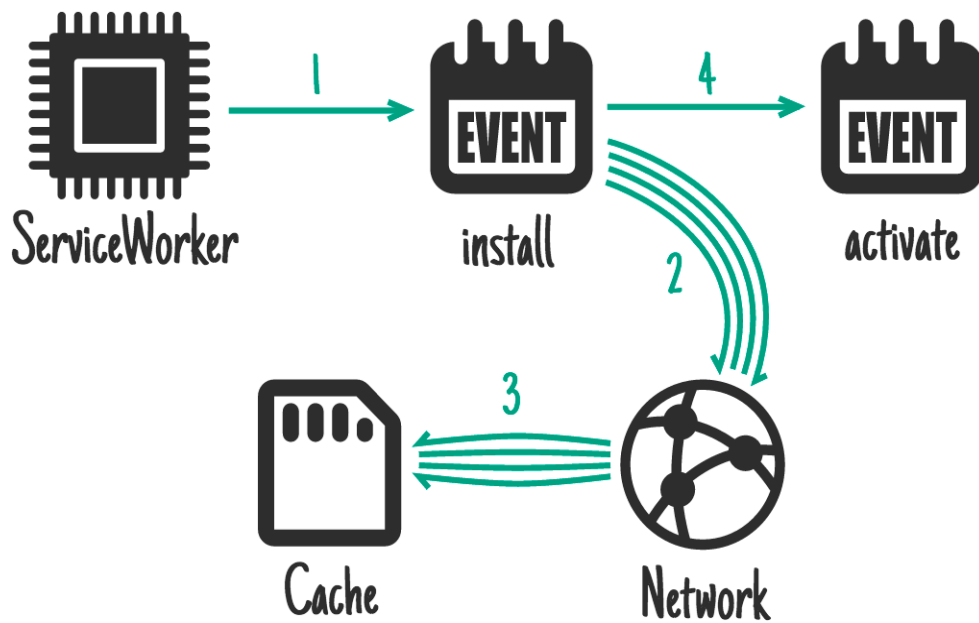


Figura 22 Proceso de instalación del services workes. Fuente: (Lepage, sf)

b) El archivo Manifest.json

Es un archivo en el cual se especifican diferentes propiedades con la finalidad que la aplicación web tenga una apariencia más parecida a una aplicación nativa. Este archivo sigue las especificaciones de W3C para aplicaciones web. Tiene la siguiente estructura tal como se muestra en el siguiente código. (Hajian, 2019)

```

{
  "name": "App Prueba",
  "short_name": "APP V.1.0",
  "description": "Mi primera APP progresiva",
  "icons": [{
    "src": "/imágenes/icons/icon-128x128.png",
    "sizes": "128x128",
  }
  ]
}

```

```

        "type": "image/png"
    }],
    "start_url": "/index.html",
    "display": "standalone",
    "background_color": "#3E4EB8",
    "theme_color": "#2F3BA2"
    "lang": "es"
    "scope": "./"
}

```

Figura 23 Especificaciones del W3C. Fuente: (Hajian, 2019)

Ahora veamos en que consiste cada una de estas propiedades:

- Name: Es el nombre que utilizará el icono de la aplicación cuando se instala en el dispositivo.
- Short_name: Nombre corto que reemplazara el nombre de la app si no se ajusta al espacio disponible.
- Description: Descripción adicional que puede llevar la aplicación.
- Icons: Es un array que define iconos de diferentes tamaños (48, 96, 144, 192, 256, 512), normalmente en formato .png y .jpg, así como también la ubicación de cada uno de ellos.
- Start_url: El punto de entrada de sus aplicaciones, generalmente es el archivo index.html.
- Display: Define como el navegador debería mostrar la información de la aplicación web. Estas opciones son: fullscreen, standalone, minimal-ui o browser.
- Background-color: Es un valor hexadecimal que define el color de fondo de la aplicación web.
- Theme-color: Es un valor hexadecimal que define el color de la barra de estado en un dispositivo móvil.
- Scope: Se define como el alcance de navegación del contexto del sitio web. En el caso de una URL relativa, la URL base será la URL del manifiesto.

c) Notificaciones PUSH

Una notificación PUSH es un mensaje que se puede mostrar en el dispositivo del usuario y se activa localmente mediante API de notificaciones o se puede enviar desde el servidor al usuario cuando la aplicación ni siquiera se está ejecutando gracias a SW (Hajian, 2019).

Hay diferentes elementos que se juntan para hacer funcionar estas notificaciones. El principal elemento es la API PUSH, que permite a los desarrolladores web hacer esto de manera similar a la tecnología de las aplicaciones nativas que se llama Push messaging. (Hajian, 2019) , describe en de una manera simplificada la arquitectura de las notificaciones Push que se describe a continuación (Hajian, 2019).

- Cuando el usuario concede permisos, la aplicación solicita un servicio web push mediante el objeto PushSubscription, según las restricciones de cada navegador.
- El servicio web push devuelve el objeto Pushsubscription y se guarda este objeto en la base de datos para reutilizarlo con la finalidad de enviar notificaciones.
- En la aplicación, se define la acción que define una notificación, el cual se enviará la notificación push basado en los detalles de suscripción.
- Finalmente, una vez que el servicio web push envíe la notificación, el SW recibirá la notificación y lo mostrará al usuario.

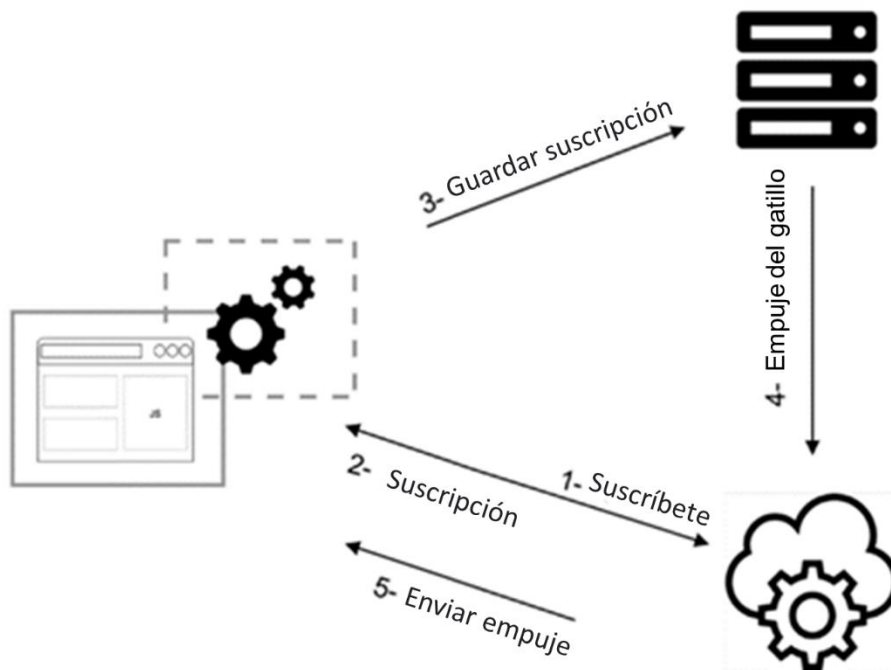


Figura 24 Proceso de notificaciones Push. Fuente: (Hajian, 2019)

d) Cache

En una aplicación web, la función de la cache funciona mejor si la lógica de la aplicación está separada de la aplicación y los datos del usuario, que se encuentran en el archivo del manifiesto y se almacenan en cache con una cantidad limitada de recursos (whatwg, 2018).

La dependencia de la red de las aplicaciones web ha obstaculizado las experiencias de navegación. La aplicación web sin conexión (o AppCache) es uno de los intentos de liberar aplicaciones web de la dependencia inherente de la red. AppCache (whatwg, 2018) permite que una aplicación web almacene en caché los recursos en el almacenamiento local para el acceso sin conexión. Sin embargo, es propenso a errores y también es difícil proporcionar una experiencia fuera de línea completa debido a la sobrecarga de administrar numerosos recursos de tipo manifiesto.

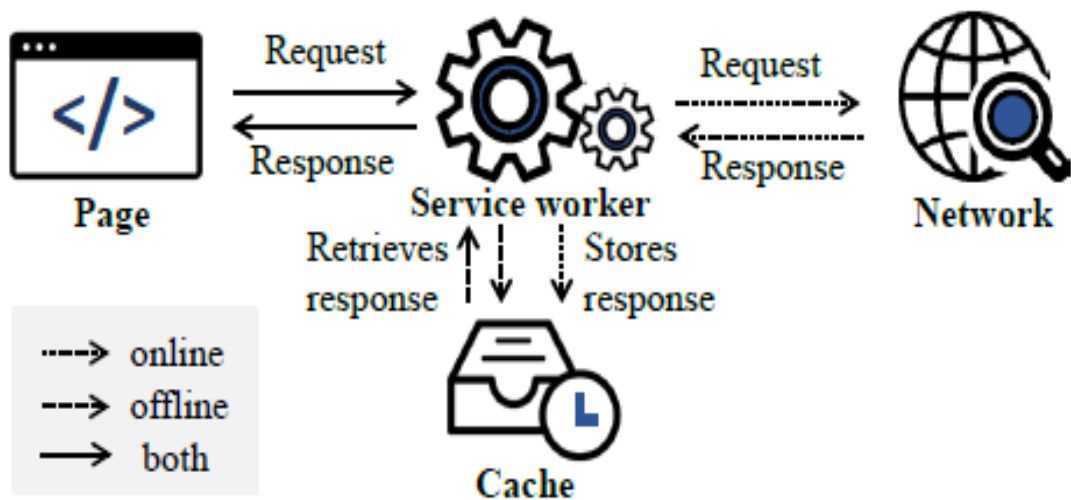


Figura 25 Ilustración del proceso del uso de la cache. Fuente: (whatwg, 2018)

La interfaz de la cache proporciona un mecanismo de almacenamiento para los pares de objetos Request/Response que almacenan en caché, como parte del ciclo de vida del Services Worker. Hay que tener en cuenta que la interfaz de la cache está expuesta a los ámbitos de ventana, así como a los trabajadores (Mozilla, 2019).

1.3.6 Norma ISO 9126

Esta norma internacional para la evaluación de la calidad de productos de software para describir un modelo se separa en dos puntos esenciales, primero sería la calidad interna y externa, y segundo sería la calidad en uso. Así vemos que la primera parte del modelo especifica seis características para la calidad interna y externa, las cuales se subdividen en sub-características y estas sub-características que se manifiestan externamente. Para la calidad interna del modelo y la calidad externa, ISO / IEC 9126 define el mismo modelo. Un sistema de información consiste en dispositivos informáticos y de hardware de comunicación y productos de software. Un sistema externo consta de sistemas de información, personas, máquinas, edificios y otros artefactos. (Sukoco, Marzuki, & Cucus, 2012)

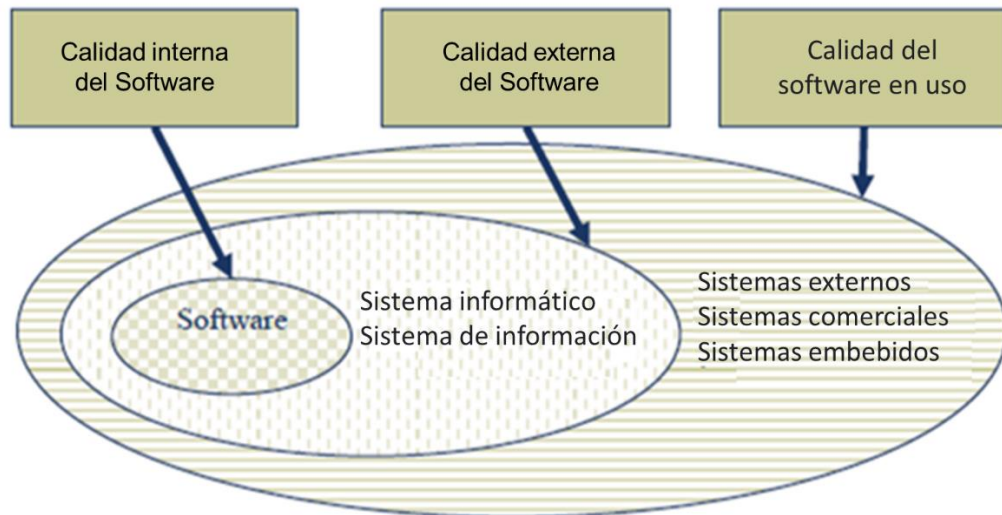


Figura 26 Estados y sistema. Fuente: (Sukoco, Marzuki, & Cucus, 2012).

1.4. Formulación del Problema.

¿De qué manera mejora el rendimiento de un sistema web mediante la tecnología de aplicaciones web progresiva?

1.5. Justificación e importancia del estudio.

Contribuir con la mejora de calidad de la experiencia del usuario al usar un sistema web ese es el principal factor que ha impulsado esta investigación.

En los últimos años el desarrollo web está avanzando a un ritmo vertiginoso, todo el tiempo impulsado por los diferentes framework tanto en el desarrollo como el diseño. El acceso a las aplicaciones de la web y móviles han superado considerablemente el acceso a las aplicaciones de escritorio, por lo que construir aplicaciones web que funcionan a la perfección en dispositivos móviles y que se puedan acceder en diferentes sistemas operativos son más importantes que nunca.

Es por ello que en el presente trabajo se pretende hacer uso de las tecnologías de PWA que combina las funcionalidades de las aplicaciones web y las aplicaciones nativas, con la finalidad de evitar el desarrollo costoso y mejorar la calidad de experiencia de la misma aplicación web para diferentes plataformas.

1.6. Hipótesis.

Al utilizar la tecnología de aplicaciones web progresivas en el desarrollo de un sistema web se mejorará en los indicadores de rendimiento.

1.7. Objetivos.

1.7.1. Objetivo general.

Evaluar el rendimiento de un Sistema web mediante la tecnología de aplicaciones web progresivas.

1.7.2. Objetivos específicos.

- a. Seleccionar los frameworks con el enfoque ligero para la implementación del sistema web.
- b. Seleccionar el caso de estudio para el desarrollo del sistema web.
- c. Realizar Pruebas para la evaluación de rendimiento del sistema web
- d. Evaluar resultados del sistema web tradicional con el sistema web usando PWA.

II. MATERIAL Y MÉTODO

2.1. Tipo y Diseño de Investigación.

El trabajo de investigación corresponde a una investigación de tipo Tecnológica - Aplicada, por la razón que se utiliza los conocimientos en la práctica, pues se demuestra los progresos considerables de un sistema web.

El diseño para esta Investigación es Cuasi-Experimental, debido a que no se utiliza ningún tipo de selección aleatoria, para las pruebas de rendimiento de una variable.

2.2. Población y muestra.

La población para el presente trabajo de investigación se definió 2 Sistemas web: Un Sistema web desarrollada con tecnología tradicional y otro sistema web con la tecnología de aplicaciones web progresivas

La muestra está constituida por el mismo tamaño de la población que son sistema web tradicional y el sistema web con la tecnología de aplicaciones web progresivas

2.3. Variables, Operacionalización.

Variables	Dimensión	Indicador	Ítem	Técnica e instrumentos de recolección de datos
DEPENDIENTE Sistema web Desarrollado	Comportamiento en el tiempo	Media del tiempo de respuesta (ms)	$MTR = \sum TR/N$	Análisis documentario Observación Ficha de Observación
		Tiempo mínimo de respuesta (ms)	$MITR = Max \sum TR$	Análisis documentario Observación Ficha de Observación
		Tiempo máximo de respuesta (ms)	$MATR = Min \sum TR$	Análisis documentario Observación Ficha de Observación
INDEPENDIENTE	Rendimiento			

**Tecnología
de PWA**

Índice de velocidad	de	$Speed\ Index = \int_0^{end} 1 - \frac{VC}{100}$ <p>Final = tiempo de finalización en milisegundos VC=% visualmente completo</p>	Análisis documentario Observación Ficha de Observación
Tamaño de la web		$TW = \sum N$ <p>N=número de archivos TW= tamaño de la web</p>	Análisis documentario Observación Ficha de Observación
Consumo de Memoria	de	$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ <p>X_i = Cada lectura parcial tomada del consumo de memoria RAM del sistema web.</p>	Análisis documentario Observación Ficha de Observación

2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad.

Las técnicas a utilizar hacen referencia a la forma de cómo se van a obtener los datos, mientras que los instrumentos representan los medios materiales, a través de los cuales se hace posible la obtención de datos. Estas técnicas e instrumentos se presentan en la siguiente tabla.

Tabla 2 Técnicas, Instrumentos

TECNICAS		
Descripción	Forma de aplicación	Forma de obtención
Análisis documentario	Personal	Corresponde al análisis de la información obtenida de papers, informes, tesis y revistas científicas.
Observación	Personal	Para verificar el rendimiento de los experimentos realizados con los sistemas a desarrollar.
INSTRUMENTOS		
Ficha de Observación	Personal	Los datos que se obtiene después de la evaluación para medir los diferentes indicadores se registran para luego ser analizados.

Fuente: Elaboración propia

2.5. Procedimiento de análisis de datos.

El punto de partida de esta investigación ha sido con la revisión documentaria de artículos, tesis, revistas, e informes referentes al tema de las aplicaciones web progresivas. Para analizar los datos obtenidos que servirán para contrastar los indicadores establecidos, se realizarán utilizando la técnica de observación y medición de tiempos para el sistema web establecido en la muestra. Estos datos obtenidos son analizados siguiendo buenas prácticas según la ISO 9126 y si es necesario hacer uso de un software para el análisis estadístico. Los datos obtenidos y analizados serán mostrados en gráficos y tablas para

visualizar el rendimiento del Sistema web con la tecnología de aplicación web progresiva en comparación con un sistema tradicional.

En este apartado los datos se han medido con base a la investigación de (González Ortiz, 2016), en donde se utilizó las 2 sub-características de eficiencia que establece el estándar ISO 9126.

A. Comportamiento en el tiempo

Media del tiempo de respuesta

$$MTR = \sum TR/N$$

Donde:

MTR = Media del tiempo de respuesta

n = número de iteraciones

Tiempo mínimo de respuesta

$$MITR = \text{Min} \sum TR$$

Donde:

MITR = Tiempo mínimo de respuesta

TR = Total de tiempos de respuesta

Tiempo máximo de respuesta

$$MATR = \text{Max} \sum TR$$

Donde:

TMAR = Tiempo mínimo de respuesta

TR = Total de tiempos de respuesta

B. Cumplimiento de la eficiencia(rendimiento)

Índice de Velocidad

El índice de velocidad es el "área por encima de la curva" calculada en ms y usando 0.0-1.0 para el rango de visualmente completo. El cálculo analiza cada intervalo de 0.1s y calcula:

$$Puntacion\ de\ intervalo = Intervalo \times \left(1.0 - \left(\frac{lo\ completo}{100} \right) \right)$$

Donde:

Lo completo es el% visualmente completo para ese cuadro e Intervalo es el tiempo transcurrido para ese cuadro de video en ms (100 en este

caso). La puntuación general es solo una suma de los intervalos individuales: SUMA (Puntuación de intervalo) (WebPagetest, 2012).

2.6. Criterios éticos.

Derechos de Autor: Todas las investigaciones, libros usados en este proyecto están citadas y referenciados en la que se indica el título, año y publicación y los autores de la investigación.

Confidencialidad: La información que sea proporcionada por las entidades de centros médicos serán manejadas bajo la normas y valores que un profesional, los datos confidenciales de la institución no serán revelados por ningún medio.

2.7. Criterios de Rigor Científico.

- a. Validez: La paralización de las variables tanto dependientes como independientes y sus dimensiones se evaluaron mediante los indicadores establecidos y las técnicas de recolección de datos.
- b. Confidencialidad: Los códigos de ética recalcan ante la seguridad y la protección de los pacientes como la identidad y como también los informantes de la investigación. También expresa como un anonimato en la identidad de los pacientes detallando en un seudónimo.
- c. Fiabilidad: para la presente investigación el tamaño de la muestra será seleccionada por frameworks con enfoque ligero.
- d. Consistencia: la investigación presentara material certificado y consistente por la comunidad científica.

III.RESULTADOS.

En esta sección se muestran y se explican los resultados obtenidos luego de medir los indicadores establecidos con el fin de determinar el rendimiento del sistema web tradicional con el sistema utilizando la tecnología de aplicaciones web progresivas. Así mismo, se muestran gráficamente valores cuantitativos que ayudaron a concluir la investigación.

3.1. Resultados en Tablas y Figuras.

En las tablas se muestra los resultados de las funcionalidades Inicio de sesión nuevo ambiente Nueva Reserva y Vista de ambientes tomados de la herramienta JMeter con diferentes tipos concurrencia de usuarios.

3.1.1. Respecto al comportamiento en el tiempo

Tabla 3 Resultados de tiempo de respuesta para funcionalidad Inicio de sesión

Inicio sesión	Sistema tradicional			Sistema con PWA		
	MITR	MTR	MATR	MITR	MTR	MATR
BAJA (ms)	1042	1475	1627	1124	1631	1041
MEDIA (ms)	3895	4489	5315	3852	4915	6878
ALTA (ms)	5463	6963	10030	3745	5969	8634

Fuente: Elaboración propia

Tabla 4 Resultados de tiempo de respuesta para funcionalidad Nuevo ambiente

Nuevo ambiente	Sistema tradicional			Sistema con PWA		
	MITR	MTR	MATR	MITR	MTR	MATR
BAJA (ms)	492	602	738	507	640	797
MEDIA (ms)	766	1119	1634	517	1321	1915
ALTA (ms)	497	1131	2766	486	2121	4506

Fuente: Elaboración propia

Tabla 5 Resultados de tiempo de respuesta para funcionalidad Nuevo reserva

Nuevo Reserva	Sistema tradicional			Sistema con PWA		
	MITR	MTR	MATR	MITR	MTR	MATR
BAJA (ms)	1005	1037	1071	1098	1027	1461
MEDIA (ms)	1058	1565	2176	1039	1862	2956
ALTA(ms)	1012	1649	3177	1041	2305	5325

Fuente: Elaboración propia

Tabla 6 Resultados de tiempo de respuesta para funcionalidad Vista ambientes

Vista ambientes	Sistema tradicional			Sistema con PWA		
	MITR	MTR	MATR	MITR	MTR	MATR
BAJA (ms)	242	256	265	231	257	281
MEDIA (ms)	238	261	344	237	281	409
ALTA(ms)	236	290	490	234	384	1296

Fuente: Elaboración propia

3.1.2. Respecto al tamaño de la web

En la tabla se muestra el resultado promedio del tamaño de la web, tomado de tres herramientas diferentes para cada sistema.

Tabla 7 Tamaño de los sistemas web

	Sistema tradicional	Sistema con PWA
Tamaño de la web (Kb)	690,97	729,47

Fuente: Elaboración propia

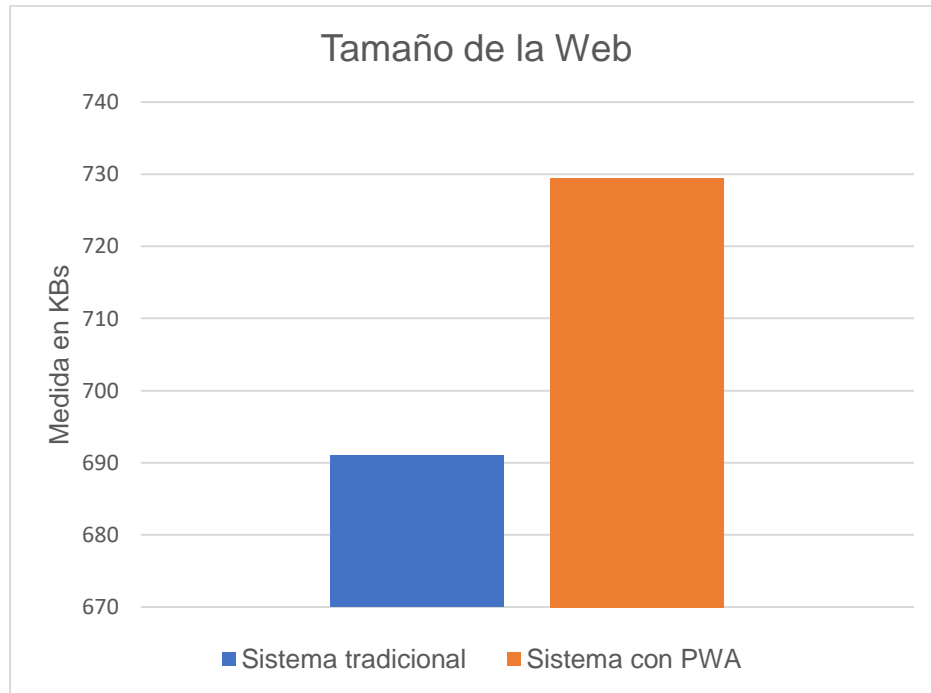


Figura 27 Tamaño de los sistemas web. Fuente: Elaboración propia

3.1.3. Respecto al consumo de memoria

En la tabla se muestra el resultado promedio del consumo de memoria RAM de la web en tiempo real, tomado de tres herramientas diferentes para cada sistema.

Tabla 8 Resultados de consumo de memoria por funcionalidad

Funcionalidad	Sistema web Tradicional	Sistema web con PWA
Inicio de sesión (mb)	41.3	91.9
Nuevo ambiente (mb)	41.7	74.3
Nueva reserva (mb)	46.7	70.5
Vista de ambientes (mb)	39.1	69.3
TOTAL	168.7	306

Fuente: Elaboración propia

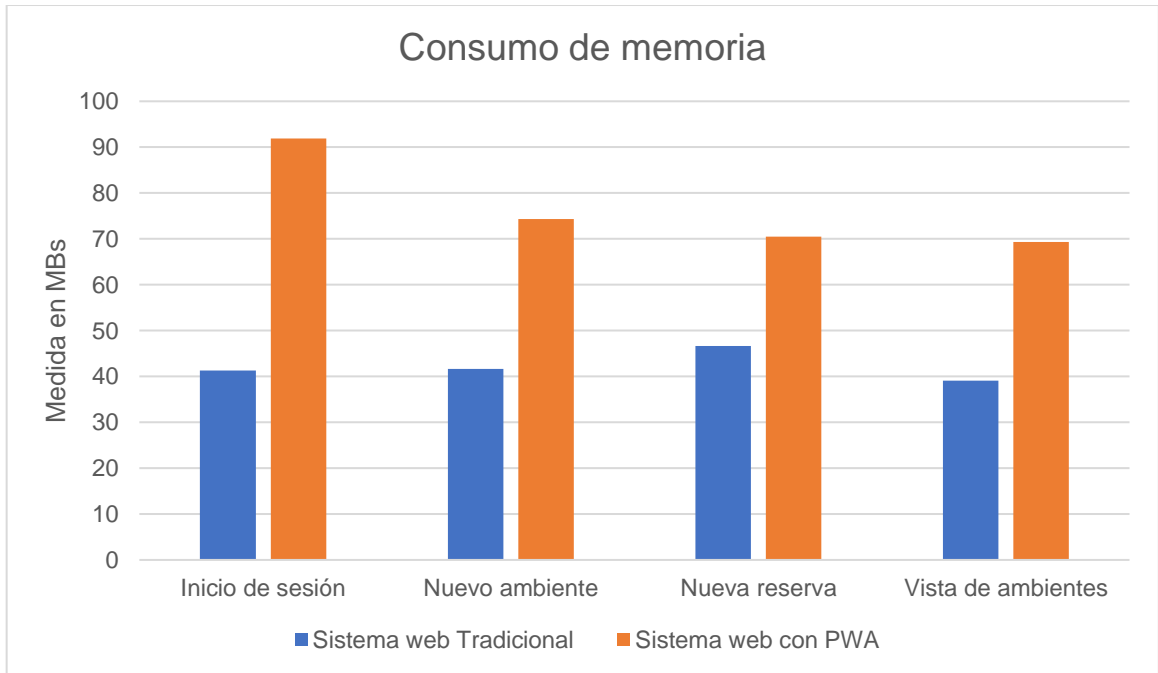


Figura 28 Comparación de resultados de consumo de memoria. Fuente: Elaboración propia

3.1.4. Respecto al índice de velocidad

En la tabla y gráfico se muestra el resultado promedio del índice de velocidad de la web en tiempo real, tomado de tres herramientas y realizado con 10 iteraciones. Eso se realizó para cada sistema.

Tabla 9 Índice de velocidad de los Sistemas web

Indicador	Sistema tradicional	Sistema con PWA
Índice de velocidad (Seg)	2,70	2,62

Fuente: Elaboración propia

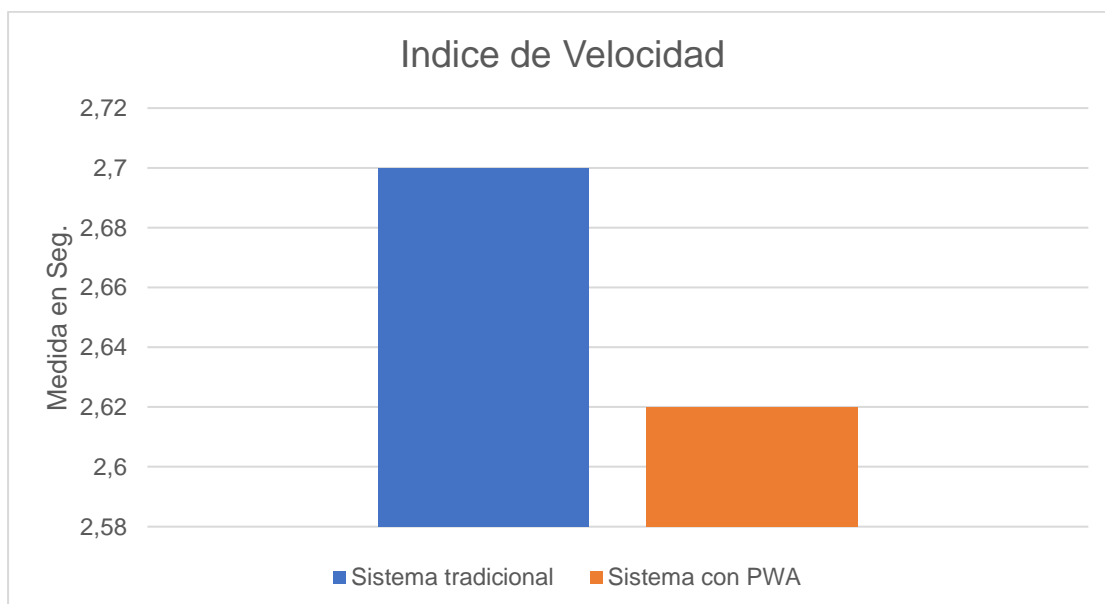


Figura 29 Índice de velocidad de los Sistemas web. Fuente: Elaboración propia

3.1.5. Resumen de los resultados

A continuación, se muestra el resultado de los indicadores para el sistema web tradicional y el sistema web aplicando la tecnología de aplicaciones web progresivas.

Tabla 10 Resumen de resultados con relación al comportamiento en el tiempo

Funcionalidad	Concurrencia	Sistema tradicional			Sistema con PWA		
		MITR	MTR	MATR	MITR	MTR	MATR
Inicio de sesión	BAJA (ms)	1042	1475	1627	1124	1631	1041
	MEDIA (ms)	3895	4489	5315	3852	4915	6878
	ALTA (ms)	5463	6963	10030	3745	5969	8634
Nuevo ambiente	BAJA (ms)	492	602	738	507	640	797
	MEDIA (ms)	766	1119	1634	517	1321	1915
	ALTA (ms)	497	1131	2766	486	2121	4506
Nueva reserva	BAJA (ms)	1005	1037	1071	1098	1027	1461
	MEDIA (ms)	1058	1565	2176	1039	1862	2956
	ALTA (ms)	1012	1649	3177	1041	2305	5325
Vista Ambientes	BAJA (ms)	242	256	265	231	257	281
	MEDIA (ms)	238	261	344	237	281	409

ALTA (ms)	236	290	490	234	384	1296
------------------	-----	-----	-----	-----	-----	------

Nota: Se utilizo un grado de concurrencia alta, media y baja. Fuente: Elaboración propia

Tabla 11 Resumen de resultados respecto al rendimiento

Indicador	Sistema tradicional	Sistema con PWA
Tamaño de la web (Kb)	690,97	729,47
Índice de velocidad (Seg)	2,70	2,62
Consumo de RAM (mb)	168.7	306

Fuente: Elaboración propia

3.2. Discusión de resultados.

a) Respecto al comportamiento al tiempo

Después del desarrollo de ambos sistemas web y analizar los resultados obtenido por funcionalidad se observa en la table N°07 el sistema con tradicional con una concurrencia alta de 30 usuarios para la funcionalidad Inicio de Sesión, se obtuvo 5463ms en el tiempo mínimo de respuesta, 6963ms en el tiempo medio de respuesta ms y en el tiempo máximo de respuesta 10030ms; mientras que el sistema con la tecnología de aplicaciones web progresivas obtuvo 3745ms en el tiempo mínimo de respuesta, 5969ms en el tiempo medio de respuesta y 8634ms en el tiempo máximo de respuesta.

En la table N°08 el sistema con tradicional con una concurrencia alta de 30 usuarios para la funcionalidad Nuevo Ambiente, se obtuvo 497ms en el tiempo mínimo de respuesta, 1131ms en el tiempo medio de respuesta ms y en el tiempo máximo de respuesta 2766ms; mientras que el sistema con la tecnología de aplicaciones web progresivas obtuvo 486ms en el tiempo mínimo de respuesta, 2121ms en el tiempo medio de respuesta y 4106ms en el tiempo máximo de respuesta

En la table N°09 el sistema con tradicional con una concurrencia alta de 30 usuarios para la funcionalidad Nueva Reserva, se obtuvo 1012ms en el tiempo mínimo de respuesta, 1649ms en el tiempo medio de respuesta ms y en el tiempo máximo de respuesta 3177ms; mientras que el sistema con la tecnología de aplicaciones web progresivas obtuvo 1041ms en el tiempo mínimo de respuesta, 2305ms en el tiempo medio de respuesta y 5325ms en el tiempo máximo de respuesta.

En la table N°10 el sistema con tradicional con una concurrencia alta de 30 usuarios para la funcionalidad Vista Ambientes, se obtuvo 236ms en el tiempo mínimo de respuesta, 290ms en el tiempo medio de respuesta ms y en el tiempo máximo de respuesta 490ms; mientras que el sistema con la tecnología de aplicaciones web progresivas obtuvo 234ms en el tiempo mínimo de respuesta, 384ms en el tiempo medio de respuesta y 1296ms en el tiempo máximo de respuesta.

b) Respecto al consumo de Memoria

En el análisis comparativo en la Grafico N° 02 del consumo de memoria RAM se obtuvo como resultado que el sistema tradicional en la funcionalidad Inicio de Sesión consume 41.3MB de la memoria, mientras que el sistema con la tecnología de aplicaciones web progresivas consume 91.9MB de la memoria. En la funcionalidad Nuevo Ambiente consume 41.7MB de la memoria, mientras que el sistema con la tecnología de aplicaciones web progresivas consume 74.3 MB de la memoria. En la funcionalidad Nueva Reserva consume 46.7MB de la memoria, mientras que el sistema con la tecnología de aplicaciones web progresivas consume 70.5MB de la memoria. Y por último en la funcionalidad Vista de Ambientes consume 39.1MB de la memoria, mientras que el sistema con la tecnología de aplicaciones web progresivas consume 69.3MB de la memoria.

Por lo tanto, el sistema tradicional consume menos memoria que un sistema con la tecnología PWA, por la razón que la tecnología PWA usa los Services workers que ocupan un lugar adicional de memoria ya que es un script que permite interceptar y controlar los requests de la red y el almacenamiento del cache del browser.

c) Respecto al tamaño de la web

En el Grafico N°2 que se refiere al tamaño de la web se obtuvo como resultado que el Sistema tradicional tiene un total 690.97 KB de tamaño de todos sus archivos y que el sistema con la tecnología de aplicaciones web progresivas obtuvo 729.47KB de tamaño de todos sus archivos. Por lo tanto, el tamaño de sus archivos del sistema es menor al tamaño de los archivos del sistema web con la tecnología PWA, por la misma razón que la tecnología PWA usa los Services workers.

d) Respecto al índice de velocidad

En cuanto al índice de velocidad que se muestra en el Grafico N°4, se obtiene que el sistema web tradicional tiene un índice de velocidad promedio en el que se muestran las partes visibles de la página de 2.70 segundos y el sistema web con tecnología PWA tiene un índice de velocidad promedio en el que se muestran las partes visibles de la página de 2.62 segundos. Siendo superior esta vez, el sistema web con la tecnología PWA solo por milésimas de segundos que el sistema tradicional.

3.3. Aporte práctico.

La propuesta de investigación se empezó con la selección de los frameworks con el enfoque ligero, para ello se obtuvieron estadísticas de los frameworks más utilizados de los últimos años de cada framework, esta información se recopiló de sitios web oficiales y confiables. Se seleccionaron por conveniencia y por su mayor utilidad, característica que los convierte en los más utilizados por los desarrolladores. Se continuó con la selección del caso de estudio para el desarrollo del sistema web tradicional, luego de implementar el sistema web tradicional y con PWA, se definieron los casos de prueba que consecuentemente brindaron las pautas y el protocolo a seguir para la medición y evaluación cuantitativa del rendimiento de dichas aplicaciones. En la figura siguiente, se representa de forma esquemática todo el proceso concerniente a la investigación realizada, así mismo se explica detalladamente cada paso de esta propuesta.

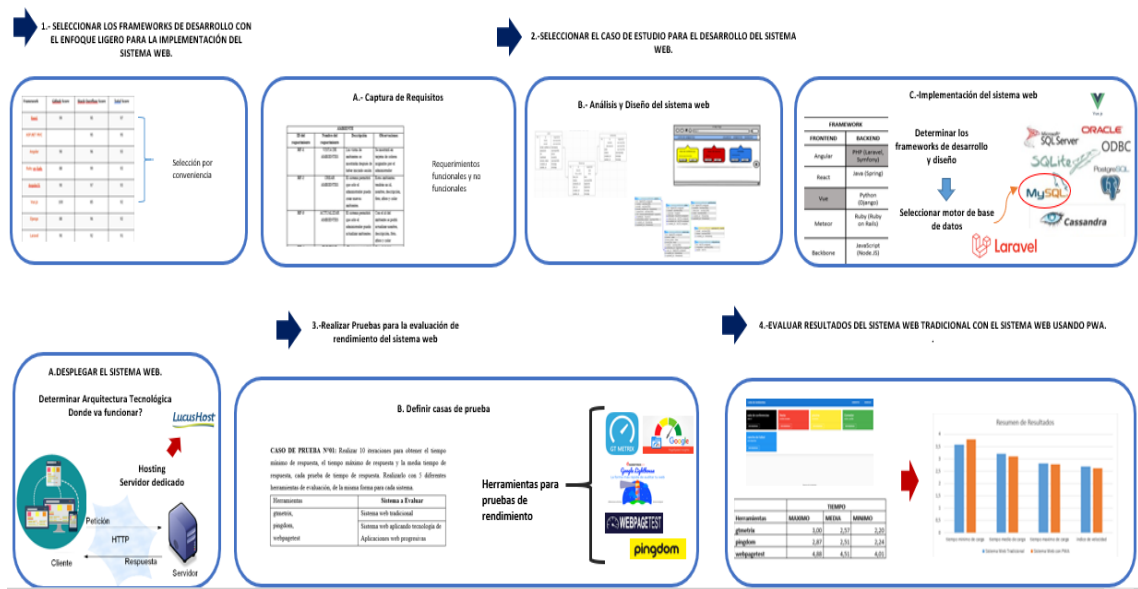


Figura 30 Método Propuesto. Fuente: Elaboración propia

3.3.1 Seleccionar los frameworks con el enfoque ligero para la implementación del sistema web.

Según la encuesta de StackOverflow de 2018, Angular es amado por el 36,9% y React es aceptado por el 27,8% de los desarrolladores encuestados, ya que JavaScript es líder en general por sexto año consecutivo como el lenguaje de programación más utilizado, total del 69,8% en la categoría "Tecnología más popular". Aunque React y Angular son votados y comparten resultados cercanos en la categoría de populares "Frameworks, bibliotecas y herramientas", Vue no ha ocupado un lugar en ninguno de los anteriores, aunque su potencial está presente (Ivanova & Georgiev, 2019).

Otra estadística interesante y es aún más útil, en términos de que de ella podemos concluir el interés real en una tecnología específica, es el número de descargas para los paquetes npm dados, como se muestra en las siguientes figuras.

Downloads in past 1 Year ▾

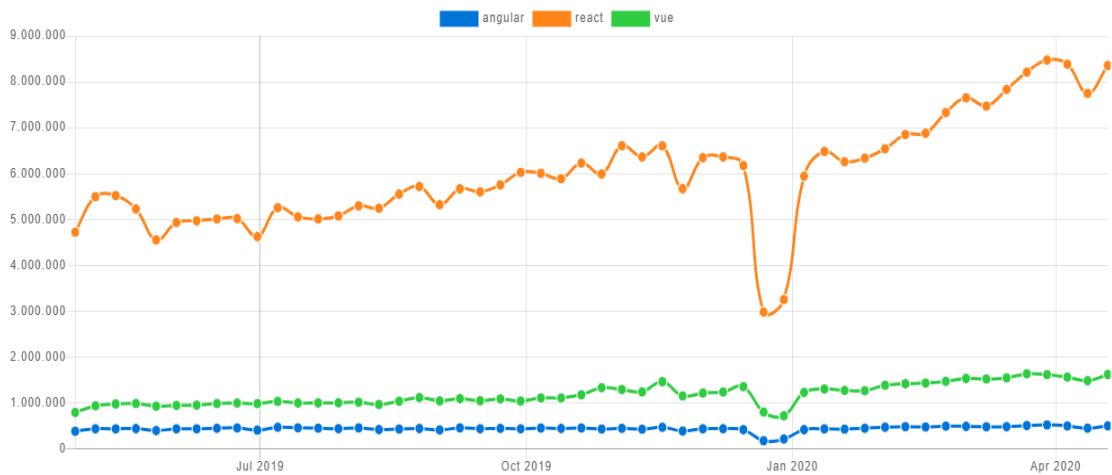


Figura 31 Descargas por paquetes npm en el último año. Fuente: (npm trends, 2020).

Estadísticas

	estrellas 🌟	problemas ⚠️	actualizado 🔄	creado 🔥	tamaño 📦
angular	59,503	467	23 de junio de 2020	5 de enero de 2010	minzipped size 62.2 KB
reaccionar	151,311	615	1 de julio de 2020	24 de mayo de 2013	minzipped size 2.6 KB
vue	166,903	519	29 de junio de 2020	28 de julio de 2013	minzipped size 22.8 KB

Figura 32 Frameworks con mejor puntuación. npm. Fuente: (npm trends, 2020).

Dada las investigaciones se observa que son muchos los Frameworks utilizados para la implementación de un sistema web, pero gracias a la puntuación que hace Github que es un portal creado para alojar el código de las aplicaciones de cualquier desarrollador, Stack OverFlow que es un sitio de preguntas y respuestas para programadores. Podemos obtener un listado del ranking de frameworks más utilizados el cual se encuentra en el ANEXO N°01. Se ha tomado los frameworks frontend y backend más usados y se analizado algunas de sus características principales.

Tabla 12 Comparación de Frameworks Front-end

FRAMEWORK FRONTEND	REACT	ANGULAR	VUE
Descripción	eficiente y flexible para construir interfaces de usuario	Para Mobile y desktop	Progresivo, incrementable para construir UI en la web
Tamaño	109,7Kib producción 774,7Kib desarrollo	167kb producción 1,2mb desarrollo	30,67Kb producción 279 Kb desarrollo
Fácil de aprender	MEDIO	Aprender TypeScript	SIMPLE
Velocidad de codificación	Normal	Lenta	Rápida
Modelo - Patrón	Virtual DOM	Virtual DOM	MVC
Reutilización de código	No, solo CSS	SI	SI, HTML Y CSS
Lenguaje	javascript	Javascript	javascript

Fuente: Elaboración propia

Tabla 13 Comparación de Frameworks Back-end

FRAMEWORK BACKEND	RUBY ON RAILS	Django	LARAVEL
Descripción	marco web del lado del servidor escrito en el lenguaje de programación Ruby.	marco de aplicaciones web de pila completa escrito en Python, normalmente hecha para aplicaciones grandes.	gratuito y totalmente compatible para crear aplicaciones web
Arquitectura	MVC (modelo,	MVT (Modelo,	MVC (modelo,

Curva de aprendizaje	vista, controlador) sintaxis muy clara	vista plantilla) Medio	vista, controlador) Fácil
Soporte Front-end	Frameworks .js	bastante complejo para vincular el framework js front-end con django	admite Vue.js fuera de la caja
mantenimiento	mantenido por el propio desarrollador y su comunidad masiva	mantenido por la fundación de software Django	mantenido por el propio desarrollador y su comunidad bajo licencia MIT
Rendimiento	normal	Rápida	normal
Lenguaje	Ruby	Python	php

Fuente: Elaboración propia

3.3.1. Seleccionar el caso de estudio para el desarrollo del sistema web.

Es indispensable para alcanzar los objetivos de la investigación se necesita de un sistema web sobre el cual podremos medir los indicadores establecidos, el caso de estudio a realizar es la creación de un sistema web que permita hacer reservar de ambientes para un determinado local, así mismo se establecido 3 puntos para describir detalladamente el sistema, así como la metodología seguida para su desarrollo.

3.3.1.1.1. Captura de Requisitos

Para el desarrollo del sistema web es necesario conocer una descripción del sistema, así como las partes que lo conforman y su funcionamiento dentro del mismo. También se dará a conocer la arquitectura del sistema y los diferentes

actores que forman parte del mismo, así como el análisis de requerimientos funcionales y no funcionales necesarios.

Descripción general del sistema

El Sistema web de Reservas de Ambientes facilita la organización, búsqueda y reserva de ambientes. Para esto cuenta con diferentes opciones y pantallas, que permiten realizar la gestión de los ambientes y reservas. Para realización de este trabajo se decidió que tuvieran 2 tipos de usuarios con privilegios y roles particulares:

Administrador:

Este usuario será la persona encargada de darle mantenimiento al sistema, en la cuestión de administración de las diferentes reservas de los ambientes que existen, así como el mantenimiento de usuarios.

Usuario final:

Cualquier persona que se registre en el sistema para poder acceder, visualizar y reservar un ambiente, que se encuentre disponible en el panel de control, así como también podrá editar su reservar y eliminarla. No podrá crear nuevos usuarios ni nuevos ambientes.

Requerimientos funcionales

Tabla 14 Requerimientos para usuarios

USUARIOS			
ID del requerimiento	Nombre del requerimiento	Descripción	Observaciones
RF-1	INICIO DE SESION ADMINISTRADOR	El sistema debe permitir la autenticación del usuario administrador	El sistema autentica rá las credenciales dadas al usuario administrador

RF-2	CAMBIAR DE CONTRASEÑA	El sistema debe permitirle al Administrador cambiar la contraseña de acceso	
RF-3	PRIVILEGIOS DEL ADMINISTRADOR	El sistema permitirá que el administrador pueda acceder a todas las pantallas del sistema, también podrá administrar los ambiente, reservas y usuarios	como excepción en el sistema no le permitirá modificar ni eliminar la reserva hecha por el usuario final, pero si crear una reserva.
RF-4	INICIO DE SESION USUARIO FINAL	El sistema debe permitir la autenticación del usuario final	El sistema autentica rá las credenciales dadas al usuario final.

RF-5	REGISTRO DE USUARIO FINAL	Solo el administrador podrá registrar usuarios finales	En un futuro se piensa hacer que el usuario final se loguee con sus redes sociales o un correo.
RF-6	PRIVILEGIOS DEL USUARIO FINAL	El sistema permitirá que solo pueda acceder algunas pantallas	Solo le es permitido ver, realizar y eliminar una reserva. Por ahora no podrá editar su perfil

Fuente elaboración propia

Tabla N° 1 Requerimientos para Ambientes

AMBIENTE			
ID del requerimiento	Nombre del requerimiento	Descripción	Observaciones
RF-1	VISTA DE AMBIENTES	Las vistas de ambientes se mostrarán después de haber iniciado sesión	Se mostrará en tarjetas de colores asignados por el administrador
RF-2	CREAR AMBIENTES	El sistema permitirá que solo el administrador pueda crear nuevos ambientes.	Estos ambientes tendrán un id, nombre, descripción, foto, aforo y color
RF-3	ACTUALIZAR AMBIENTES	El sistema permitirá que solo el administrador pueda actualizar ambientes.	Con el id del ambiente se podrá actualizar nombre, descripción, foto, aforo y color
RF-4	ELIMINAR AMBIENTES	El sistema permitirá que solo el administrador pueda eliminar ambientes.	Con el id del ambiente se podrá eliminar todo el ambiente.
RF-5	MOSTRAR AMBIENTE	Encargado de mostrar todos	Los datos de un ambiente se

		los datos de un ambiente específico.	mostrar en una tarjeta de un color específico.
RF-6	REPORTE DE AMBIENTES	El reporte de ambientes solo podrá generar el administrador	Este reporte obtendrá todos los datos de todos los ambientes creados

Fuente: elaboración propia

Tabla 15 Requerimientos para Reservas

RESERVA			
ID del requerimiento	Nombre del requerimiento	Descripción	Observaciones
RF-1	VISTA DE RESERVAS	El sistema permitirá al usuario ver una reserva solo al seleccionarla.	
RF-2	CREAR RESERVAS	El sistema permitirá ambos usuarios crear reservas.	Para hacer una reserva tendrá que seleccionar día, fecha, hora y motivo
RF-3	ACTUALIZAR RESERVAS	El sistema no permitirá que ningún usuario pueda editar la reserva.	Solo podrá eliminarse y volverse a crear.
RF-4	ELIMINAR RESERVAS	El sistema permitirá que	Al eliminarse una reserva,

		solo el creador de la reserva pueda eliminarla.	nuevamente queda disponible esa fecha y hora.
RF-5	MOSTRAR RESERVAS	Encargado de mostrar todos los datos de una reserva especifica.	Al hacer click sobre el botón reserva de un ambiente se mostrará detalle de las reservas hechas.
RF-6	REPORTE DE RESERVAS	El reporte de reservas solo podrá generar el administrador	Este reporte obtendrá todos los datos de todas las reservas creadas.

Fuente: elaboración propia

Requerimientos no funcionales

Tabla 16 Requerimientos no funcionales para el sistema

ID del requerimiento	Descripción
RNF – 001	Para el desarrollo del sistema se utilizará como framework back-end Laravel
RNF – 002	Para el diseño del sistema se utilizará como framework front end Vue
RNF - 003	La aplicación se desplegará en una red Local y en la nube
RNF - 004	Cada ruta debe estar protegida por autenticación token de Laravel.

RNF - 005	En el fron-end, el token de usuario se ingresará de manera predeterminada desde el código de la aplicación.
RNF - 006	La aplicación estará disponible a través de navegadores de internet. Google, Mozilla Firefox y Safari
RNF - 007	En la implementación de la aplicación se hará un comentario cada método o función con un mensaje corto que haga referencia clara a la funcionalidad del método.
RNF - 008	Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso a datos.
RNF - 009	El sistema debe poseer interfaces gráficas muy sencillas.

Fuente: elaboración propia

3.3.1.1.2. Análisis y Diseño del sistema web

Según los requerimientos establecidos se procedió a realizar los mockups en la herramienta Balsamiq Mockups 3 y el diseño de base de datos para el sistema de reservas de ambientes.



Figura 33 Vista de la ventana Login o inicio de sesión. Fuente: elaboración propia

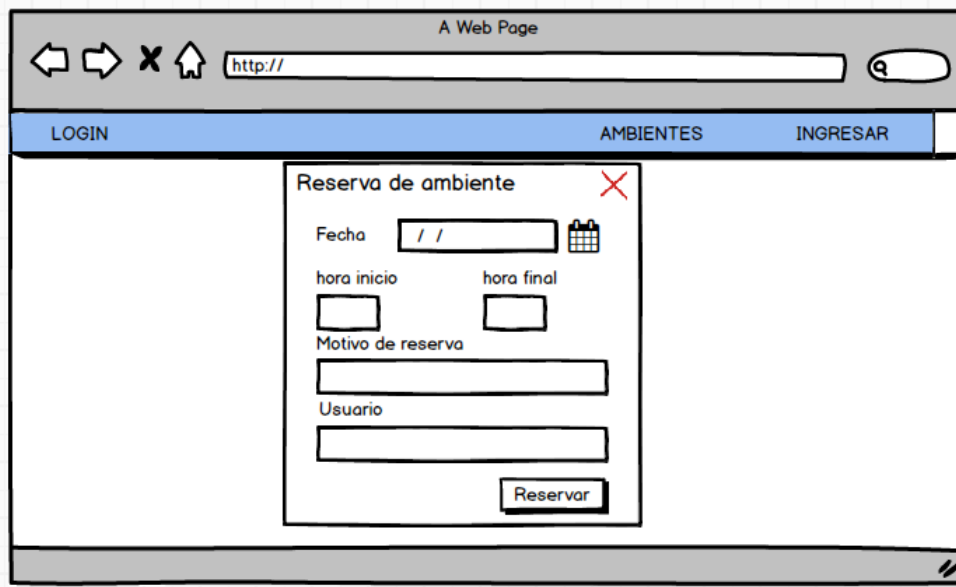


Figura 34 Vista reserva. Fuente: elaboración propia

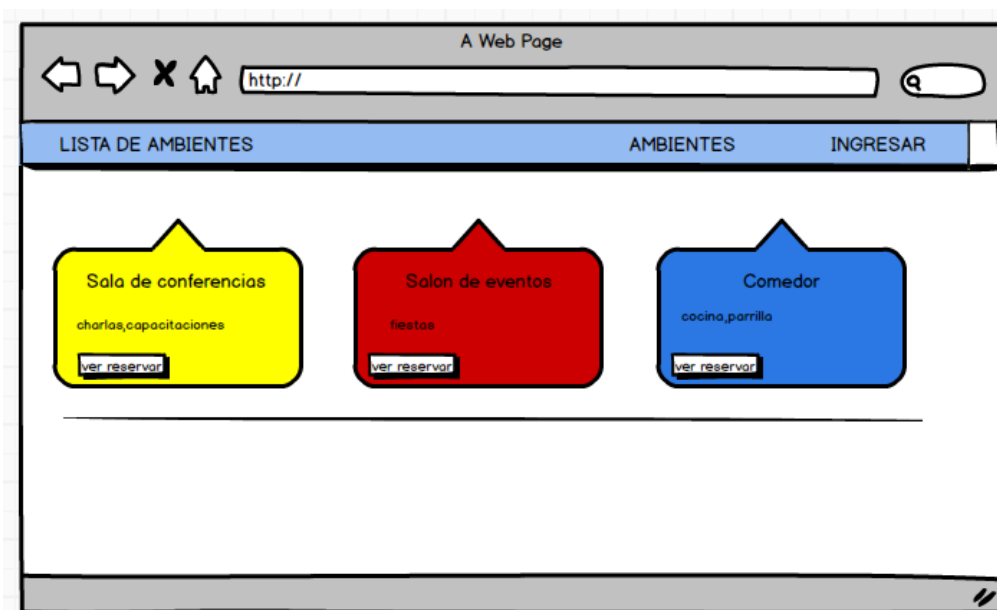


Figura 35 Vista Usuario ambiente-usuario final. Fuente: Elaboración propia

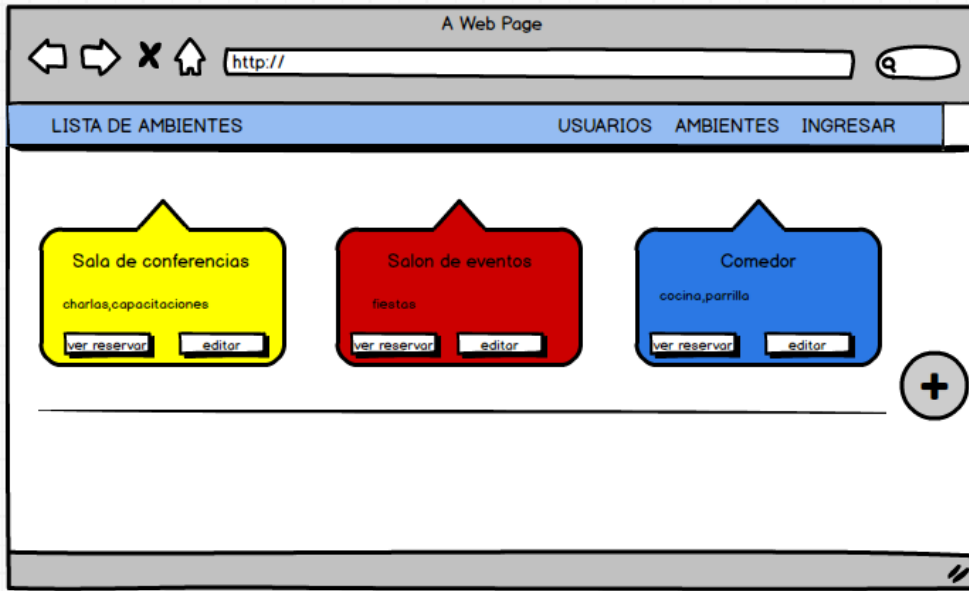


Figura 36 Vista de la ventana ambiente-administrado. Fuente: Elaboración propia

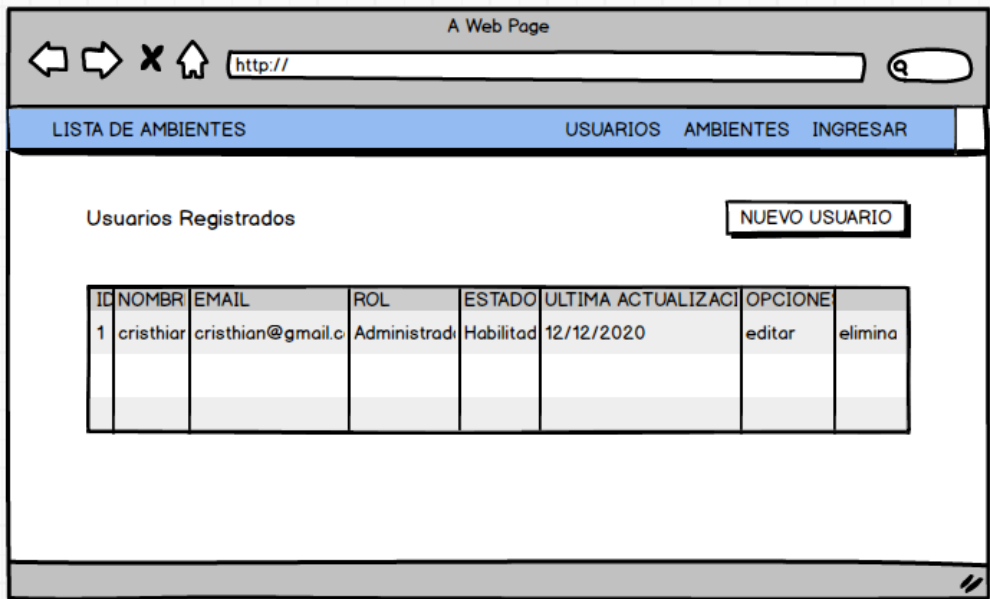


Figura 37 Vista mantenimiento usuarios. Fuente: Elaboración propia



Figura 38 Vista de la ventana Nuevo usuario. Fuente: Elaboración propia



Figura 39 Vista de la ventana nuevo ambiente. Fuente: Elaboración propia

Diseño de la base de datos

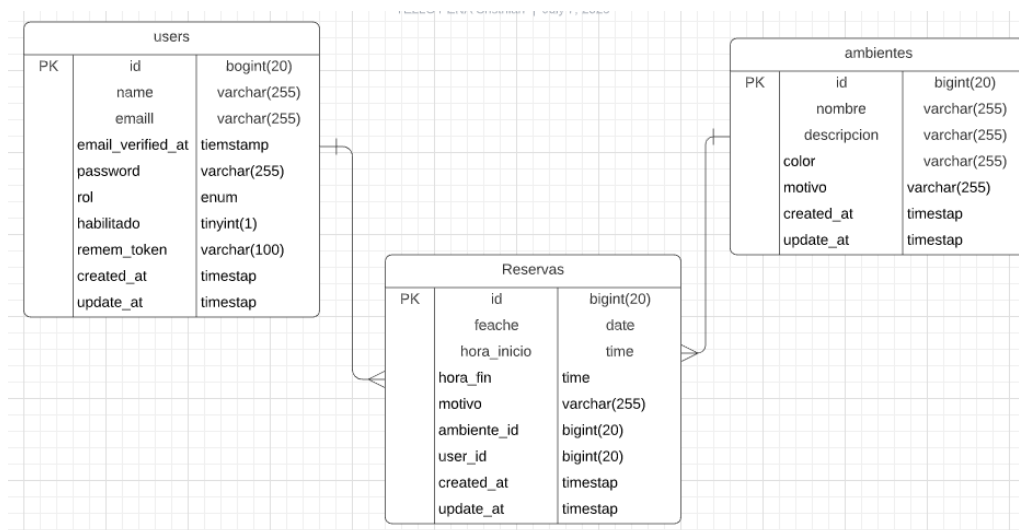


Figura 40 Modelo de base de datos inicial. Fuente: Elaboración propia

3.3.1.1.3. Implementación del sistema web

Para empezar con la implementación del sistema primero se realizó la creación de la base de datos con la ayuda del paquete Xampp, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl.

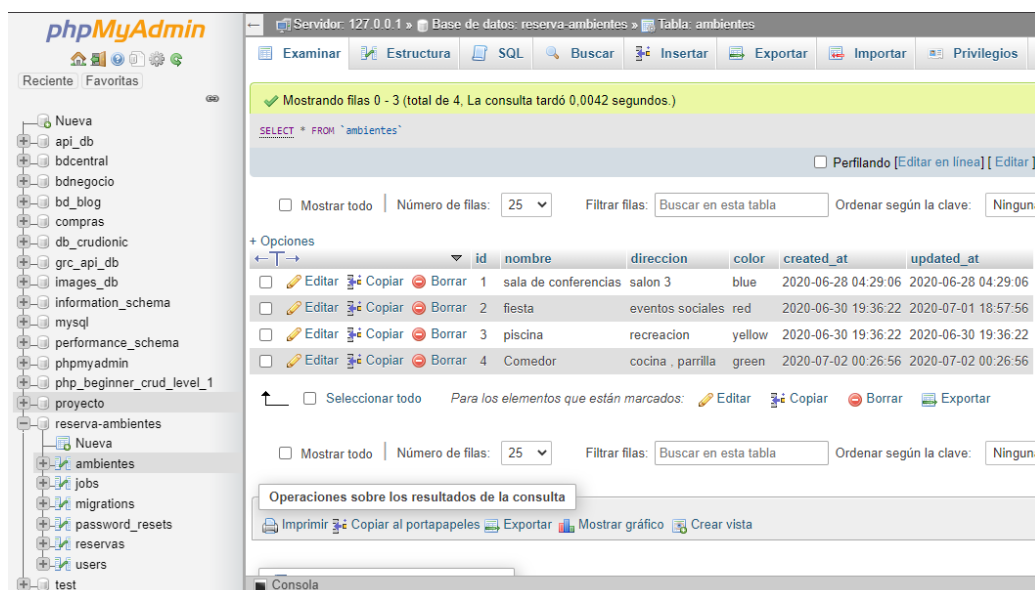


Figura 41 Vista de creación de base de datos. Fuente: Elaboración propia

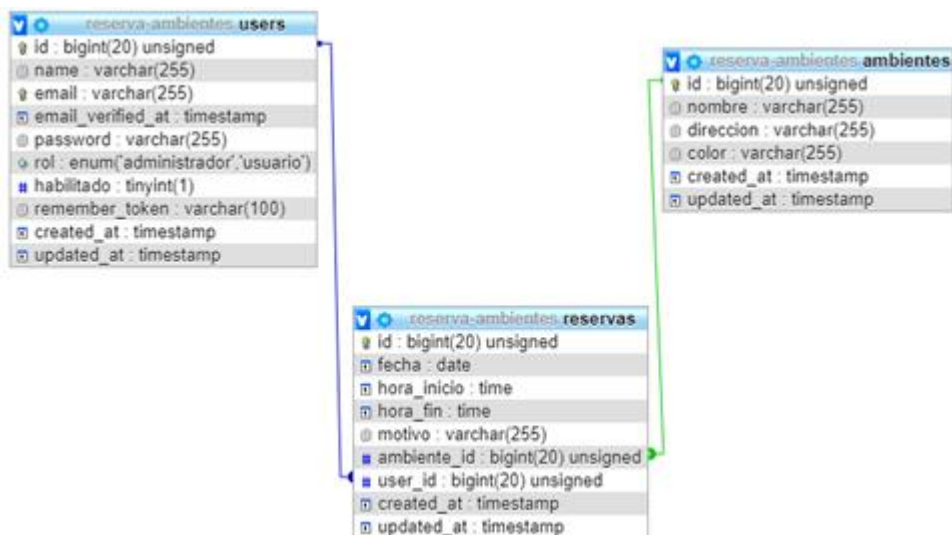


Figura 42 Figura N° 35 Base de datos del sistema. Fuente: Elaboración propia

Luego de la creación de la base de datos se realizó el diseño del sistema con el framework Fron-End Vue y para el desarrollo de este mismo se utilizó el framework Back-End Laravel, se utilizaron estos frameworks por conveniencia, ya que ambos son muy compatibles, tienen una arquitectura MVC y también se encontraban entre el ranking de los frameworks más usados. De esta manera se cumple el uso de los frameworks con el enfoque ligero. Las vistas respectivas del sistema se mostrarán en el Anexo N°02.

3.3.2. Realizar Pruebas para la evaluación de rendimiento del sistema web

3.3.2.1.1. Desplegar el sistema web.

En este punto el sistema web está apto para ser desplegado en la web para eso requirió un Dominio y Hosting, las cuales nos proveyó la empresa LucusHost, ubicada en España. También se obtuvo unas licencias y certificados para el mejor funcionamiento del sistema, tales como SSL Y HTTPS, además por la mayor razón que las aplicaciones web progresivas refiriéndonos al sistema con la tecnología PWA requieren HTTPS a menos que sean servidas desde localhost.

3.3.2.1.2. Definir casos de prueba

En este apartado se especifica todos los elementos necesarios para realizar las pruebas como son el hardware necesario, las herramientas(software) open source, el tipo de pruebas y las repeticiones necesarias para realizar el análisis comparativo de los sistemas. Además, se menciona algunos factores externos para las pruebas realizadas.

Tabla 17 Hardware a utilizar para el desarrollo de los sistemas propuestos.

Descripción	Modelo	Procesador	Memoria RAM	Sistema Operativo
Laptop LENOVO ideapad 320	LAPTOP-T70MLPA4	AMD A12-9720P Radeon R7,12 Compute cores 4C+8G 2.70GHz	16,0 GB	Windows 10 home de 64 bits

Fuente: Elaboración propia

CASO DE PRUEBA N°01: Realizar las pruebas para medir el tiempo mínimo, medio y máximo de respuesta para las funcionalidades: Inicio de Sesión, Nuevo Ambiente, Nueva Reserva y Vista de Ambientes del sistema con base a concurrencia de usuarios.

Tabla 18 Herramientas a utilizar para el comportamiento en el tiempo

Herramientas	Sistema a Evaluar	Factores Externos
JMeter de Apache	Sistema web tradicional	Describir
Navegador Firefox	Sistema web aplicando tecnología de Aplicaciones web progresivas	Velocidad de Internet

Fuente: Elaboración propia

CASO DE PRUEBA N°02: Realizar la sumatoria del tamaño del contenido (script, Font, css, etc.) del Sistema tradicional y el Sistema web aplicando tecnología de Aplicaciones web progresivas.

Tabla 19 Herramientas a utilizar para calcular el tamaño de los sistemas

Herramientas	Sistema a Evaluar	Contenido
	Sistema web tradicional	Script
gtmetrix,	Sistema web aplicando	Font
pingdom,	tecnología de	CSS
webpagetest	Aplicaciones web	HTML
	progresivas	XHR
		etc.

Fuente: Elaboración propia

CASO DE PRUEBA N°03: Medir la utilización de memoria RAM para las siguientes funcionalidades: Inicio de Sesión, Nuevo Ambiente, Nueva Reserva y Vista de Ambientes, mientras se carga en el navegador todos los elementos del DOM y se renderizan todos los recursos contenidos en el código JS. Para cada sistema respectivamente.

Tabla 20 Herramientas a utilizar para calcular el uso de la memoria RAM

Herramientas	Sistema a Evaluar
Administrador	Sistema web tradicional
de tareas de	Sistema web aplicando tecnología de
Windows	Aplicaciones web progresivas
Administrador	
de tareas de	
Microsoft	

Fuente: Elaboración propia

CASO DE PRUEBA N°04: Realizar 10 iteraciones para obtener el índice de velocidad. Realizarlo con 3 diferentes herramientas de evaluación, de la misma forma para cada sistema.

Tabla 21 Herramientas a utilizar para calcular el índice de velocidad

Herramientas	Sistema a Evaluar
Webpagetest	Sistema web tradicional
PageSpeed Insights	Sistema web aplicando tecnología de Aplicaciones web progresivas
Lighthouse	

Fuente: Elaboración propia

3.3.3. Evaluar resultados del sistema web tradicional con el sistema web usando PWA.

Se realiza la comparación de los sistemas con la finalidad de medir los indicadores de rendimiento para luego hacer una comparativa de lo mejorado al usar la tecnología de aplicaciones web progresivas.

La evaluación de los sistemas web se realizó utilizando dos sub características de eficiencia de la norma ISO 2196, comportamiento con respecto al Tiempo, Utilización de recursos. Las herramientas anteriormente seleccionadas, que permitieron realizar las pruebas de nuestros indicadores mediante peticiones http, obteniendo datos de las métricas externas. Cabe mencionar que las pruebas se realizaron con el uso de una conexión de internet domiciliario de 2.62 Mb/s de bajada y 2.82 Mb/s de subida, como se muestra en la siguiente figura:



Figura 43 Velocidad de Internet utilizado. Fuente: (testdevelocidad.es,2020)

CASO DE PRUEBA N° 01

Para realizar las mediciones de tiempo de mínimo, medio y máximo de respuesta para cada una de los sistemas. Como primer paso se descargó la herramienta JMeter versión 5.3 de su página oficial: <http://jmeter.apache.org/> con el requerimiento de tener instalado una versión superior a Java 8, luego se procedió a descomprimir el archivo .zip en el escritorio.

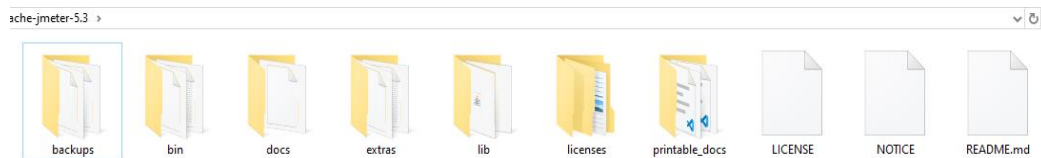


Figura 44 Directorio de JMeter. Fuente: Elaboración propia

El archivo ejecutable a utilizar se encuentra dentro de la carpeta “bin” de nombre “ApacheJmeter.jar” que nos permitirá tener acceso a la interfaz gráfica JMeter

Nombre	Fecha de modificación	Tipo
examples	01/02/1980 12:00	Carpeta de archivo
report-template	01/02/1980 12:00	Carpeta de archivo
templates	01/02/1980 12:00	Carpeta de archivo
ApacheJMeter	01/02/1980 12:00	Executable Jar File
ApacheJMeterTemporaryRootCA	13/07/2020 11:11	Certificado de seg.

Figura 45 Archivo ejecutable ApacheJMeter. Fuente: Elaboración propia

Luego como segundo paso se procedió a configurar el escenario de las pruebas en la herramienta JMeter, para eso se creó un grupo de hilos con 4 controladores para grabar cada una de las funcionalidades del sistema y un receptor llamado informe agregado donde nos mostrar los tiempos de repuesta.



The image shows the 'Grupo de Hilos' (Group of Threads) configuration window in JMeter. It includes the following fields and options:

- Nombre:** Grupo de Hilos
- Comentarios:** (empty text box)
- Acción a tomar después de un error de Muestreador:** Continuar (selected), Comenzar siguiente iteración, Parar Hilo, Parar Test, Parar test ahora
- Propiedades de Hilo:**
 - Número de Hilos:** 1
 - Periodo de Subida (en segundos):** 1
 - Contador del bucle:** Sin fin, 1

Figura 46 Grupo de Hilos. Fuente: Elaboración propia



Figura 47 Controladores para las funcionalidades y el informe agregado. Fuente: Elaboración propia

También se creó un servidor proxy para grabar las peticiones solicitadas y se configuro el servidor proxy en el navegador, cabe mencionar que se usó el puerto predefinido 8888

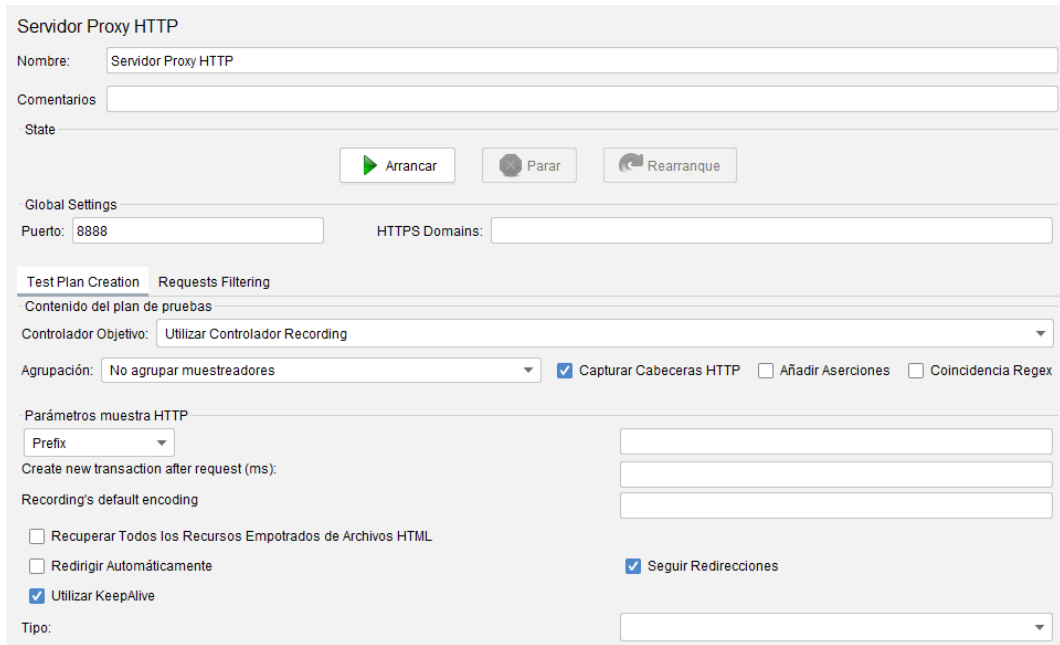


Figura 48 Servidor proxy en la herramienta JMeter. Fuente: Elaboración propia

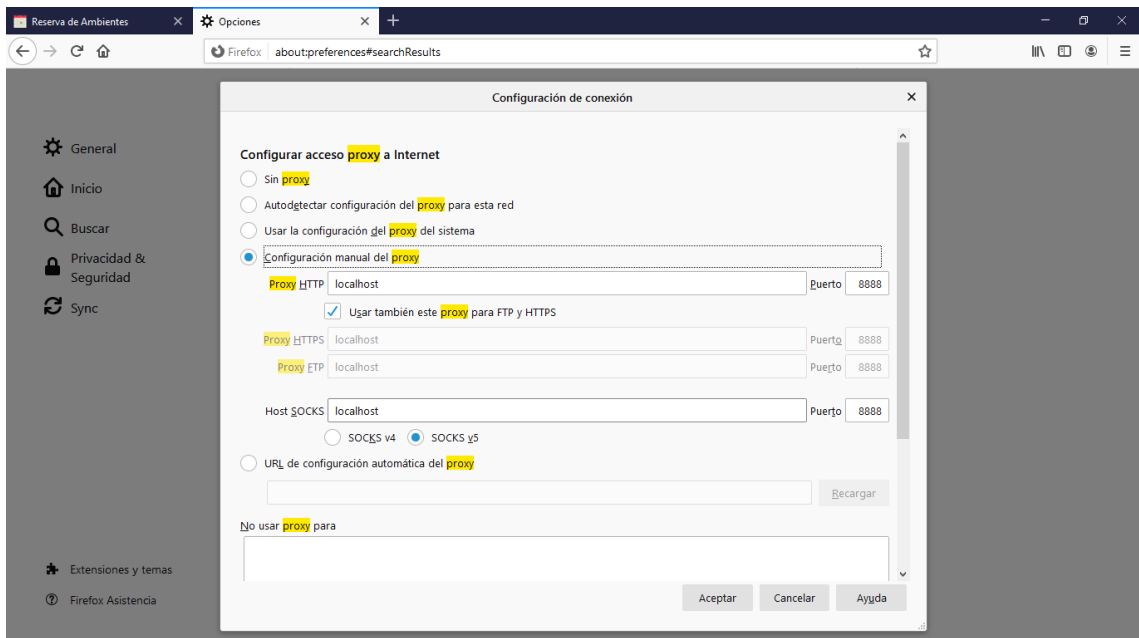


Figura 49 Configuración del proxy en el navegador Firefox. Fuente: Elaboración propia

Como último paso se realizó la carga(concurrencia) baja, media y alta que simulan 10,20,30 usuarios respectivamente en 1seg esto se realizó con base a la investigación de (González Ortiz, 2016) en donde se menciona que las

pruebas de carga(concurrencia) alta media y baja deben simular un escenario esperado de visitantes en las funcionalidades del sistema.

Figura 50 Grupo de hilos que simulan la concurrencia de usuarios. Fuente: elaboración propia

Sistema Tradicional

En las funcionalidades del sistema Tradicional con un grado concurrencia de 10,20,30 usuarios se obtuvieron como resultados

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx
/api/auth/login-110	10	1184	1104	1307	1307	1378	1042	1378
/api/ambiente-111	10	291	302	326	326	332	247	332
InicioSesion	10	1475	1396	1609	1609	1627	1343	1627
/api/ambiente-112	10	323	312	387	387	407	243	407
/api/ambiente-113	10	279	260	307	307	391	243	391
NuevoAmbiente	10	602	610	650	650	738	492	738
/api/reserva/1/ambi...	10	262	256	278	278	289	250	289
/api/reserva/1/ambi...	10	261	262	266	266	269	251	269
/api/reserva-116	10	257	255	266	266	272	247	272
/api/reserva/1/ambi...	10	255	255	261	261	266	249	266
NuevaReserva	10	1037	1032	1069	1069	1071	1005	1071
/api/ambiente-118	10	256	254	264	264	265	242	265
VistaAmbientes	10	256	254	264	264	265	242	265
Total	130	518	266	1291	1383	1609	242	1627

Figura 51 Concurrencia baja - funcionalidades del Sistema Tradicional. Fuente: Elaboración propia

Informe Agregado

Nombre: Informe Agregado

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo: Escribir en L

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx
/api/auth/login-110	20	4107	3925	4896	4913	5019	3493	5019
/api/ambiente-111	20	382	386	494	595	619	249	619
InicioSesion	20	4489	4311	5162	5166	5315	3895	5315
/api/ambiente-112	20	384	352	537	572	616	298	616
/api/ambiente-113	20	734	553	1289	1291	1323	404	1323
NuevoAmbiente	20	1119	976	1601	1602	1634	766	1634
/api/reserva/1/ambi...	20	433	403	503	848	859	251	859
/api/reserva/1/ambi...	20	399	404	493	560	639	243	639
/api/reserva-116	20	413	325	621	700	814	252	814
/api/reserva/1/ambi...	20	318	260	326	737	822	238	822
NuevaReserva	20	1565	1574	1860	1990	2176	1058	2176
/api/ambiente-118	20	261	250	313	318	344	238	344
VistaAmbientes	20	261	250	313	318	344	238	344
Total	260	1143	434	3952	4409	5148	238	5315

*Figura 52 Concurrencia media - funcionalidades del Sistema Tradicional.
Fuente: Elaboración propia.*

Informe Agregado

Nombre: Informe Agregado

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo: Escribir en L

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx
/api/auth/login-110	30	6086	5605	7688	7721	8310	4954	8310
/api/ambiente-111	30	876	519	1204	1914	4193	305	4193
InicioSesion	30	6963	6315	8709	9258	10030	5453	10030
/api/ambiente-112	30	544	393	698	698	2405	254	2405
/api/ambiente-113	30	586	413	1199	1300	2474	243	2474
NuevoAmbiente	30	1131	881	2463	2598	2766	497	2766
/api/reserva/1/ambi...	30	462	387	583	594	2386	246	2386
/api/reserva/1/ambi...	30	403	404	503	596	696	244	696
/api/reserva-116	30	370	399	501	501	597	243	597
/api/reserva/1/ambi...	30	412	394	577	601	1303	243	1303
NuevaReserva	30	1649	1800	2002	2094	3177	1012	3177
/api/ambiente-118	30	290	260	392	395	490	236	490
VistaAmbientes	30	290	260	392	395	490	236	490
Total	390	1543	491	5773	6671	8310	236	10030

*Figura 53 Concurrencia alta - funcionalidades del Sistema Tradicional.
Fuente: Elaboración propia.*

Tabla 22 Funcionalidad Inicio de sesión

SISTEMA TRADICIONAL	TIEMPO DE RESPUESTA		
	MINIMA (ms)	MEDIA (ms)	MAXIMA (ms)
CONCURRENCIA BAJA (10)	1042	1475	1627
CONCURRENCIA MEDIA (20)	3895	4489	5315
CONCURRENCIA ALTA (30)	5463	6963	1030

Fuente: Elaboración propia

Tabla 23 Funcionalidad Nuevo ambiente

SISTEMA TRADICIONAL	TIEMPO DE RESPUESTA		
	MINIMA (ms)	MEDIA (ms)	MAXIMA (ms)
CONCURRENCIA BAJA (10)	492	602	738
CONCURRENCIA MEDIA (20)	766	1119	1634
CONCURRENCIA ALTA (30)	497	1131	2766

Fuente: Elaboración propia

Tabla 24 Funcionalidad Nueva reserva

SISTEMA TRADICIONAL	TIEMPO DE RESPUESTA		
	MINIMA (ms)	MEDIA (ms)	MAXIMA (ms)
CONCURRENCIA BAJA (10)	1005	1037	1071
CONCURRENCIA MEDIA (20)	1058	1565	2176

CONCURRENCIA ALTA (30)	1012	1649	3177
-----------------------------------	------	------	------

Fuente: Elaboración propia

Tabla 25 Funcionalidad Vista ambientes

SISTEMA TRADICIONAL	TIEMPO DE RESPUESTA		
	MINIMA (ms)	MEDIA (ms)	MAXIMA (ms)
CONCURRENCIA BAJA (10)	242	256	265
CONCURRENCIA MEDIA (20)	238	261	344
CONCURRENCIA ALTA (30)	236	290	490

Fuente: Elaboración propia

Sistema con tecnología de Aplicaciones web progresivas

En las funcionalidades del sistema Tradicional con un grado concurrencia de 10,20,30 usuarios se obtuvieron como resultados

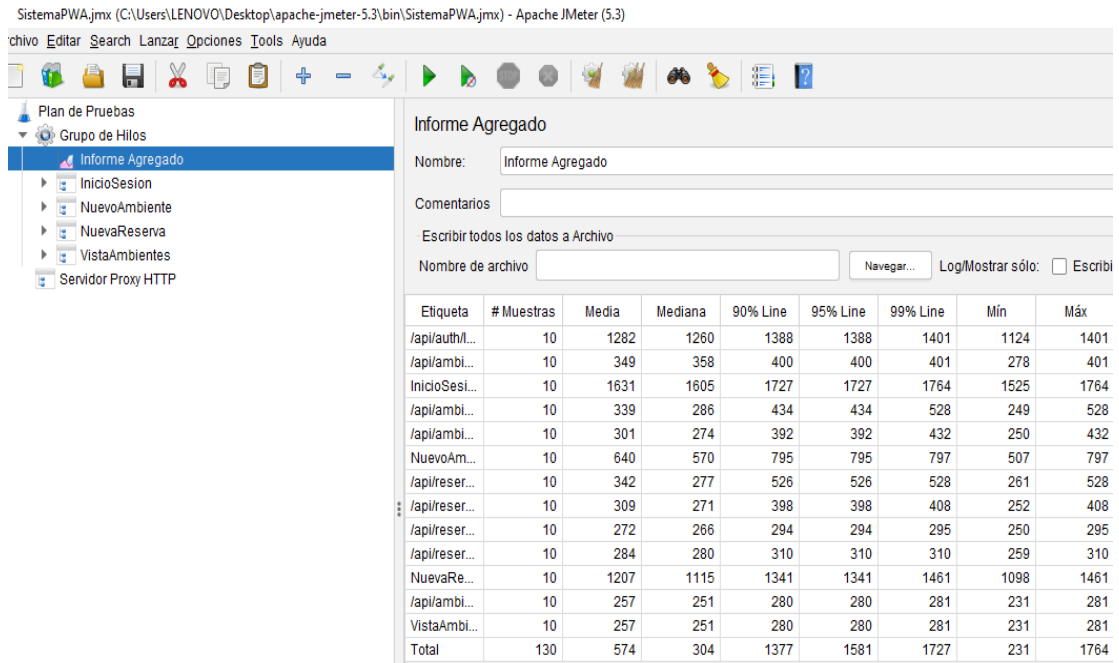


Figura 54 Concurrencia baja - funcionalidades del Sistema con PWA. Fuente: Elaboración propia.

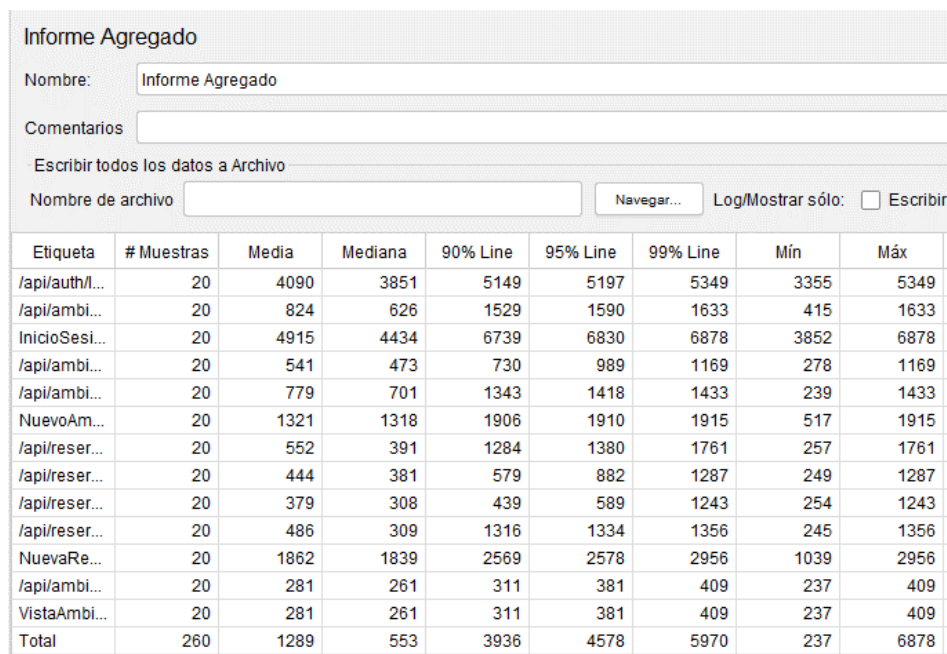


Figura 55 Figura N° 48 Concurrencia media - funcionalidades del Sistema PWA. Fuente: Elaboración propia.

Informe Agregado

Nombre: Informe Agregado

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo: Escribi

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx
/api/auth/...	30	4864	4853	6069	6077	6910	3201	6910
/api/ambi...	30	1104	1068	1285	1386	3227	246	3227
InicioSesi...	30	5969	6640	7261	7290	8634	3745	8634
/api/ambi...	30	679	310	935	3229	3336	247	3336
/api/ambi...	30	1442	389	3299	3307	4214	231	4214
NuevoAm...	30	2121	1312	3858	3877	4506	486	4506
/api/reser...	30	663	393	1315	2368	4228	259	4228
/api/reser...	30	610	484	710	1881	2381	250	2381
/api/reser...	30	526	398	596	1193	2196	242	2196
/api/reser...	30	505	383	508	1198	2305	241	2305
NuevaRe...	30	2305	1831	3370	3453	5325	1041	5325
/api/ambi...	30	384	262	495	1203	1296	234	1296
VistaAmbi...	30	384	262	495	1203	1296	234	1296
Total	390	1658	500	4609	6064	7210	231	8634

Figura 56Figura N° 49 Concurrencia alta - funcionalidades del Sistema PWA.
Fuente: Elaboración propia.

Tabla 26 Funcionalidad Inicio de sesión

SISTEMA CON PWA	TIEMPO DE RESPUESTA		
	MINIMA (ms)	MEDIA (ms)	MAXIMA (ms)
CONCURRENCIA BAJA (10)	1124	1631	1041
CONCURRENCIA MEDIA (20)	3852	4915	6878
CONCURRENCIA ALTA (30)	3745	5969	8634

Fuente: Elaboración propia

Tabla 2 Funcionalidad Nuevo ambiente

SISTEMA CON PWA	TIEMPO DE RESPUESTA		
	MINIMA (ms)	MEDIA (ms)	MAXIMA (ms)
CONCURRENCIA BAJA (10)	507	640	797
CONCURRENCIA MEDIA (20)	517	1321	1915
CONCURRENCIA ALTA (30)	486	2121	4506

Fuente: Elaboración propia

Tabla 3 Funcionalidad Nueva reserva

SISTEMA CON PWA	TIEMPO DE RESPUESTA		
	MINIMA (ms)	MEDIA (ms)	MAXIMA (ms)
CONCURRENCIA BAJA (10)	1098	1027	1461
CONCURRENCIA MEDIA (20)	1039	1862	2956
CONCURRENCIA ALTA (30)	1041	2305	5325

Fuente: Elaboración propia

Tabla 4 Funcionalidad Vista ambientes

SISTEMA CON PWA	TIEMPO DE RESPUESTA		
	MINIMA (ms)	MEDIA (ms)	MAXIMA (ms)
CONCURRENCIA BAJA (10)	231	257	281

CONCURRENCIA MEDIA (20)	237	281	409
CONCURRENCIA ALTA (30)	234	384	1296

Fuente: Elaboración propia

CASO N° 02

Se realizó la sumatoria del tamaño del contenido (script, Font, css, etc.) del Sistema tradicional y el Sistema web aplicando tecnología de Aplicaciones web progresivas. Usando las 3 tecnologías GTmetrix, Pingdom y Webpagetest.

Tabla 27 Resultado del tamaño de la web del sistema tradicional

Tamaño de la web	
Herramientas	Medida (Kb)
gtmetrix	686
pingdom	730,4
webpagetest	772

Fuente: Elaboración propia

Tabla 28 Resultado del tamaño de la web con tecnología de aplicaciones web progresivas

Tamaño de la web	
Herramientas	Medida (Kb)
gtmetrix	683
pingdom	706,9
webpagetest	683

Fuente: Elaboración propia

CASO DE PRUEBA N°03

Se hizo la medición de la utilización de memoria RAM, mientras carga el navegador con todos los elementos del sistema web. Para las funcionalidades de cada sistema respectivamente.

Sistema Tradicional

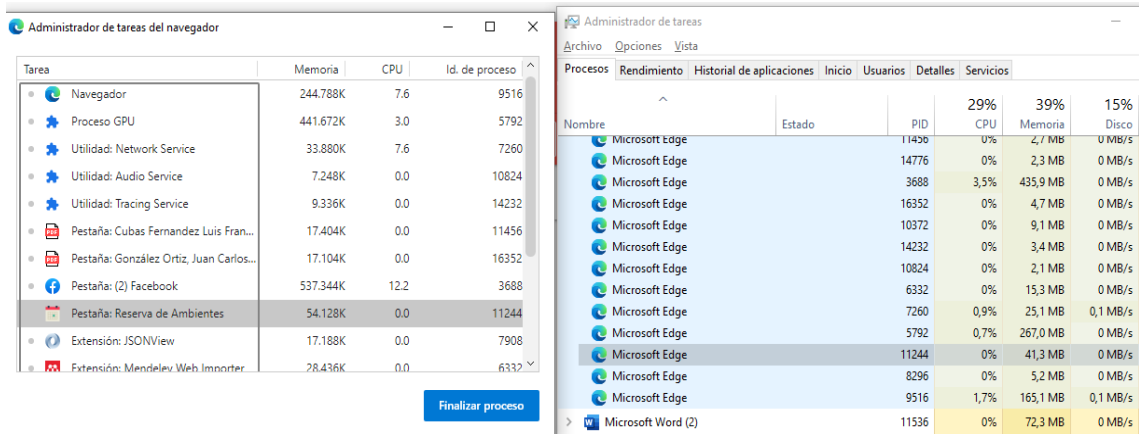


Figura 57 Uso de memoria RAM de Inicio de Sesión. Fuente: Elaboración propia

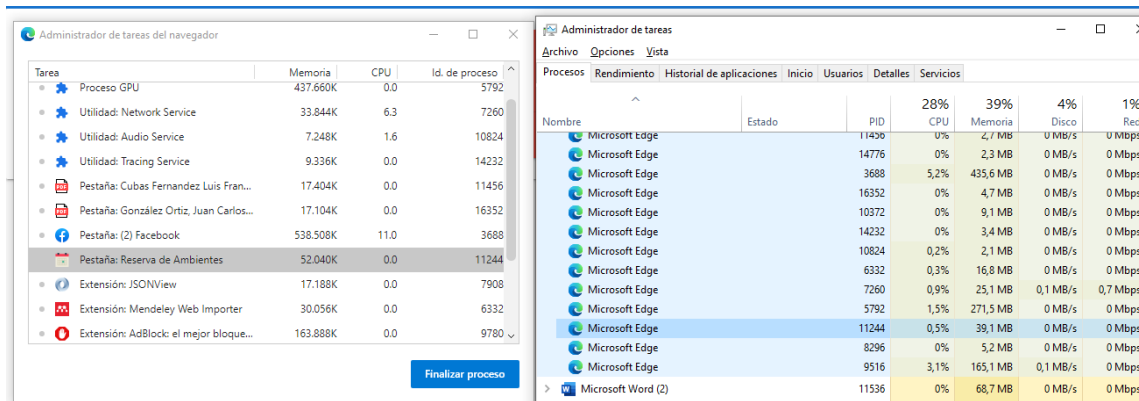


Figura 58 Uso de memoria RAM de vistas ambientes. Fuente: Elaboración propia

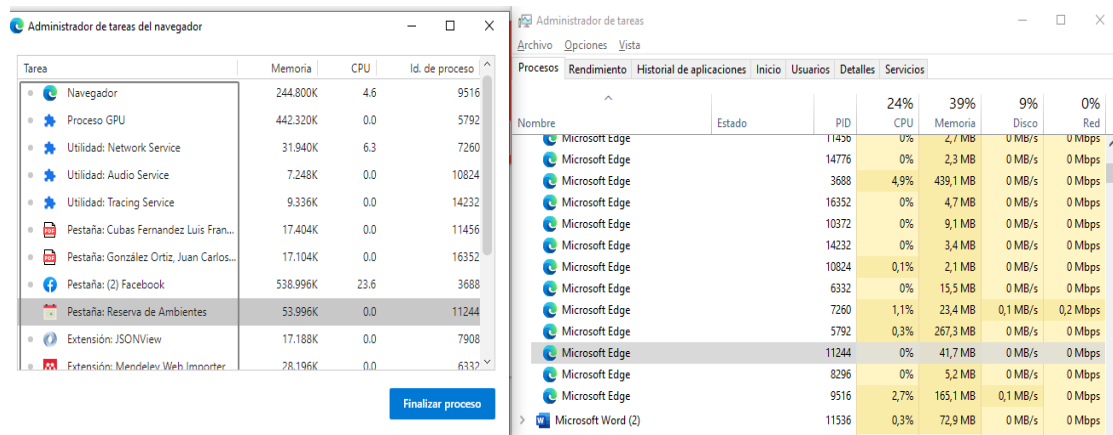


Figura 59 Uso de memoria RAM de Nuevo Ambiente. Fuente: Elaboración propia

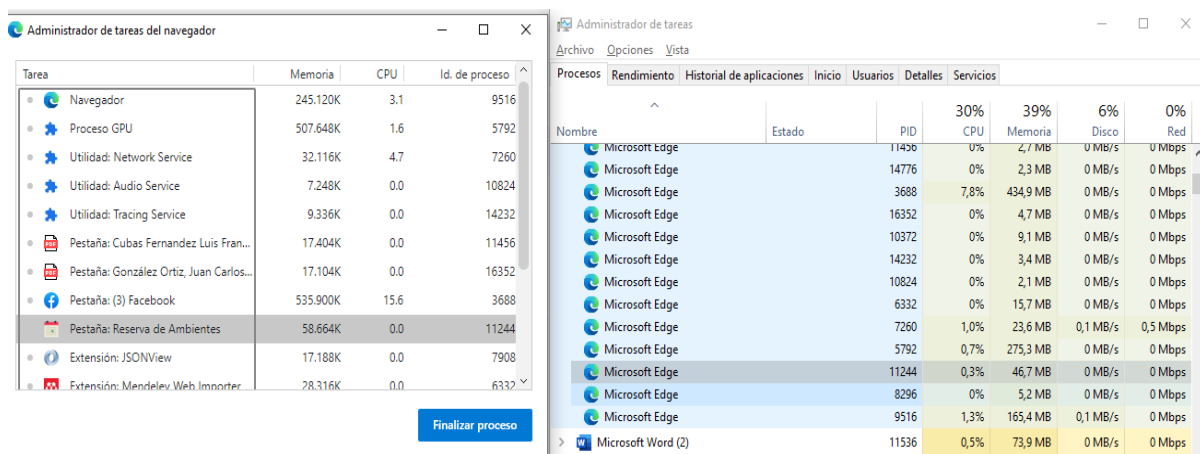


Figura 60 Uso de memoria RAM de Nueva Reserva. Fuente: Elaboración propia

Tabla 29 Consumo de memoria RAM por funcionalidad

ID / PID	CONSUMO DE MEMORIA Funcionalidad	administrador de tareas de Windows	MEDIDA
11244	Inicio de sesión	41.3	MB
11244	Nuevo ambiente	41.7	MB
11244	Nueva reserva	46.7	MB
11244	Vista de ambientes	39.1	MB

Fuente: Elaboración propia

Sistema con tecnología de Aplicaciones web progresivas

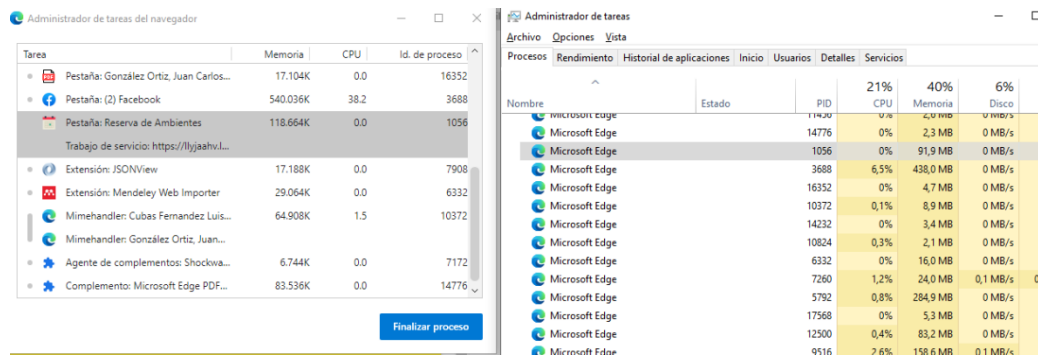


Figura 61 Uso de memoria RAM de Inicio de sesión. Fuente: Elaboración propia

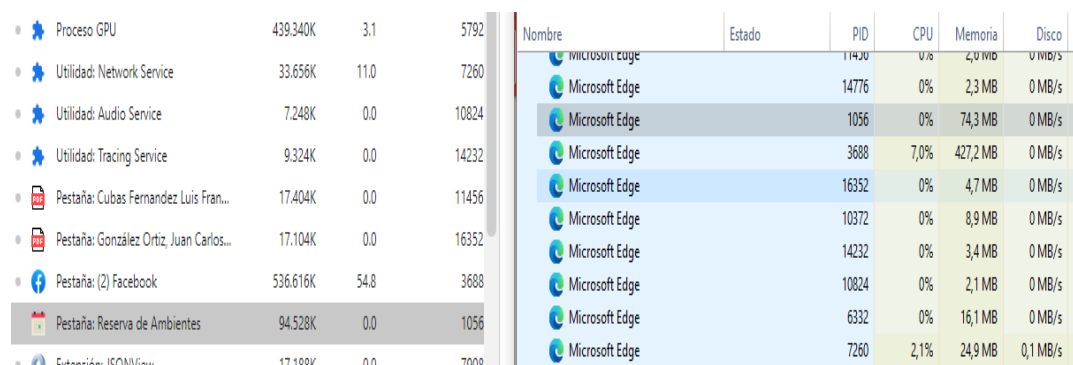


Figura 62 Uso de memoria RAM de Nuevo ambiente. Fuente: Elaboración propia

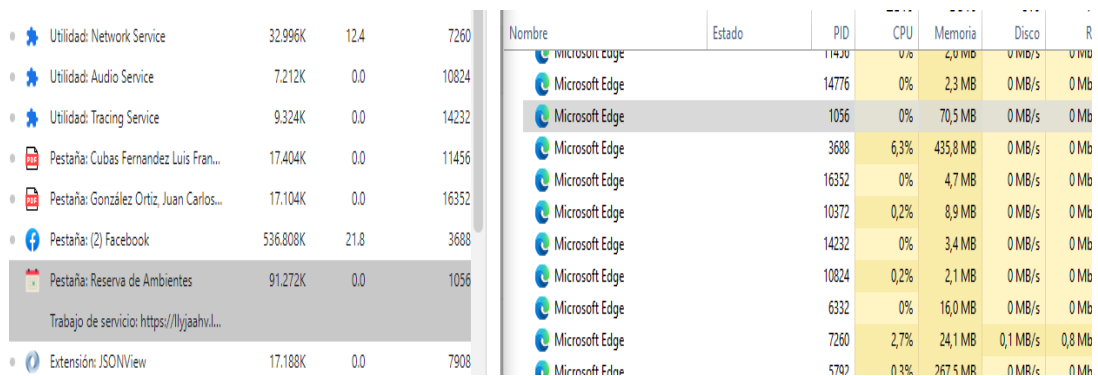


Figura 63 Uso de memoria RAM de Nueva reserva. Fuente: Elaboración propia

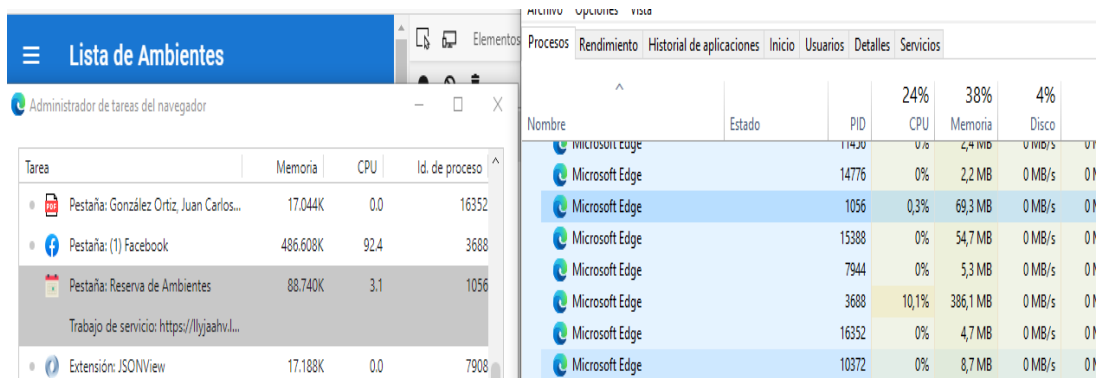


Figura 64 Uso de memoria RAM de Vista de Ambientes. Fuente: Elaboración propia

Tabla 30 Consumo de memoria RAM por funcionalidad

ID / PID	CONSUMO DE MEMORIA Funcionalidad	administrador de tareas de Windows	MEDIDA
1056	Inicio de sesión	91.9	MB
1056	Nuevo ambiente	74.3	MB
1056	Nueva reserva	70.5	MB
1056	Vista de ambientes	69.3	MB

Fuente: Elaboración propia

CASO DE PRUEBA N°04

Se realizó 10 iteraciones para obtener el índice de velocidad con el uso 3 diferentes herramientas de evaluación, esto se aplicó en ambos sistemas.

Sistema web Tradicional

Datos de campo — El informe "Experiencia de Usuario de Chrome" **no tiene suficientes datos a tiempo real sobre la velocidad** de esta página.

Origin Summary — El informe Experiencia de Usuario de Chrome **no tiene suficientes datos a tiempo real sobre la velocidad** de este origen.



Datos de experimentos

■ Primer renderizado con contenido	1,4 s	● Tiempo hasta que está interactiva	1,4 s
■ Índice de velocidad	2,1 s	● Tiempo total de bloqueo	0 ms

Figura 65 Índice de velocidad con la herramienta PageSpeed Insights. Fuente: Elaboración propia

Performance Results (Median Run - SpeedIndex)

First Byte	Start Render	First Contentful Paint	Speed Index	Web Vitals		Document Complete			Fully Loaded			
				Largest Contentful Paint	Cumulative Layout Shift	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
0.808s	3.600s	3.568s	3.885s	3.944s	0	3.641s	8	682 KB	3.910s	9	683 KB	\$\$\$---

[Plot Full Results](#)

Figura 66 Índice de velocidad con la herramienta webpagetest. Fuente: Elaboración propia

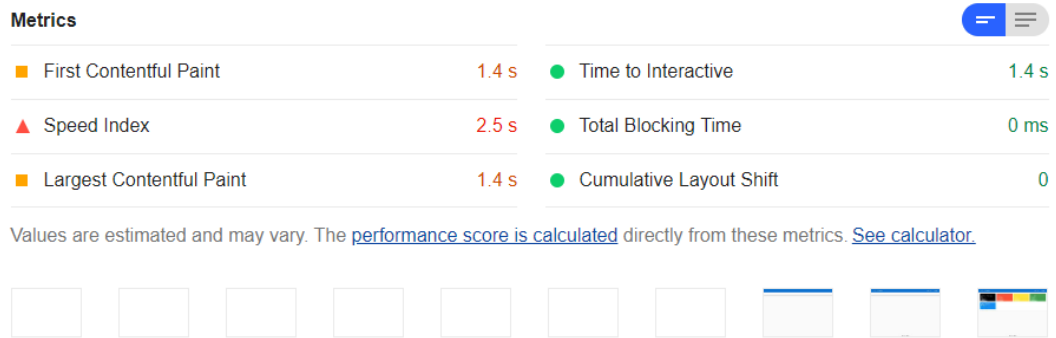


Figura 67 Índice de velocidad con la herramienta Lighthouse. Fuente: Elaboración propia

Tabla 31 Índice de velocidad del Sistemas con 3 herramientas diferentes

INDICE DE VELOCIDAD AD	N° de iteraciones										promedio	
	1	2	3	4	5	6	7	8	9	10		
Herramientas												
PageSpeed Insights	2,30	2,10	1,90	2,00	2,00	2,00	2,20	2,10	2,00	2,00	2,00	2,07
Lighthouse	2,30	2,30	2,00	1,90	1,90	2,30	2,30	2,20	1,90	2,00	2,00	2,15
webpagetest	3,90	3,80	3,80	3,80	3,80	3,90	3,80	4,30	3,80	3,80	3,80	3,87

Fuente: Elaboración propia

Sistema web con Tecnología de aplicaciones web progresivas

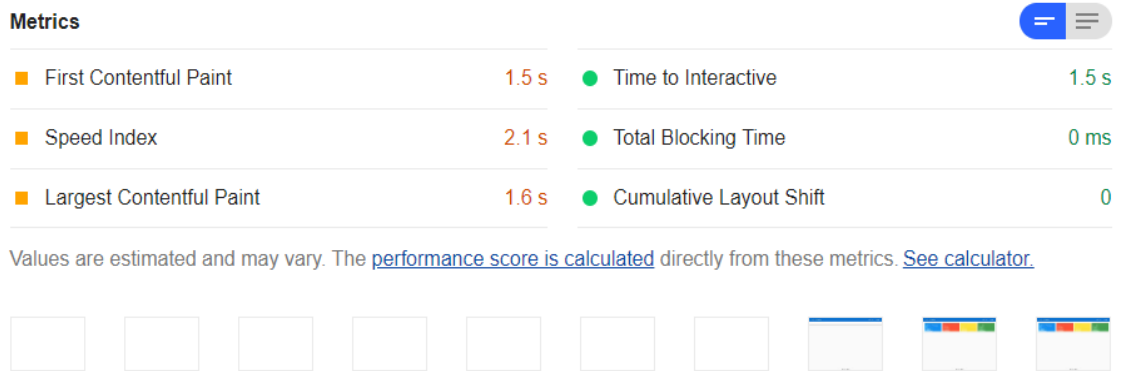


Figura 68 Índice de velocidad con la herramienta PageSpeed Insights. Fuente: Elaboración propia

	First Byte	Start Render	First Contentful Paint	Speed Index	Web Vitals		Document Complete			Fully Loaded			
					Largest Contentful Paint	Cumulative Layout Shift	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View (Run 1)	0.420s	3.200s	2.969s	3.335s	3.135s	0	3.054s	18	769 KB	3.476s	20	772 KB	\$---

[Plot Full Results](#)

Test Results

Run 1:

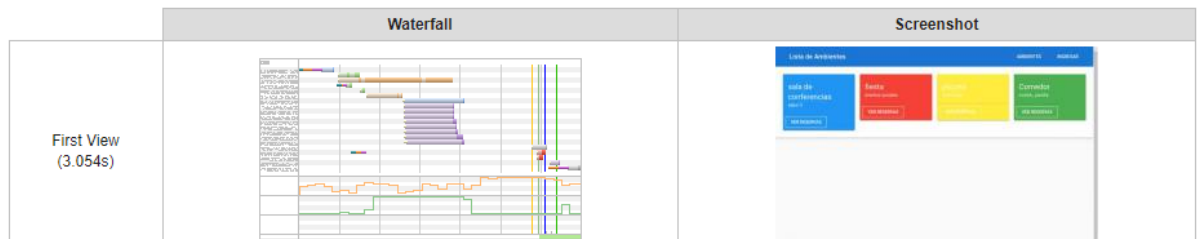


Figura 69 Índice de velocidad con la herramienta Pagewebtest. Fuente: Elaboración propia

Datos de experimentos



■ Primer renderizado con contenido	1,3 s	● Tiempo hasta que está interactiva	1,3 s
■ Índice de velocidad	1,4 s	● Tiempo total de bloqueo	0 ms
■ Largest Contentful Paint	1,4 s	● Cambios de diseño acumulados	0



Figura 70 Índice de velocidad con la herramienta Lighthouse. Fuente: Elaboración propia

Tabla 32 Índice de velocidad del Sistema con 3 herramientas diferentes

INDICE DE VELOCIDAD	N° de iteraciones										promedio
	1	2	3	4	5	6	7	8	9	10	
Herramientas	1	2	3	4	5	6	7	8	9	10	
PageSpeed Insights	2,	1,	2,	2,	2,	2,	1,	1,	2,	1,	1,97
Lighthouse	10	40	10	10	10	30	80	80	10	90	
webpagetest	2,	2,	1,	1,	2,	1,	2,	1,	1,	2,	1,90
	30	10	60	80	00	60	00	80	80	00	
	3,	4,	3,	4,	4,	4,	4,	4,	4,	4,	3,98
	30	20	30	40	10	00	20	20	00	10	

Fuente: Elaboración propia

Tabla 33 Índice de velocidad promedio de ambos sistemas

INDICE DE VELOCIDAD	
Sistema web con tecnología de aplicaciones web progresivas	2,62 seg.
Sistema web tradicional	2,70 seg.

Fuente: Elaboración propia

IV. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones.

- A. Al realizar una revisión y un estudio teórico de los frameworks más utilizados apoyadas en investigaciones, encuestas hechas por las páginas oficiales de los frameworks y además de estadísticas presentadas por Google Trends y npmtrends, de esta manera se determinó a evaluar 3 frameworks, tanto para el desarrollo como para el diseño.
- B. Se optó a tomar como caso de estudio un Sistema de Reservas de Ambientes, para el cual se determinó por conveniencia utilizar para el desarrollo del Sistema el Framework Laravel y para el diseño el Framework Vue, ya que ambas son muy compatibles especialmente en su arquitectura MVC de esta manera se cumple el trabajar con el enfoque ligero.
- C. Se realizó los casos de prueba correspondientes para evaluar los indicadores tiempo mínimo, medio y máximo de respuesta, consumo de memoria, tamaño de la web y por último el índice de velocidad con diferentes herramientas de evaluación para obtener un mejor promedio en nuestros indicadores.
- D. Al evaluar las pruebas de rendimiento de los sistemas web se llegó a la conclusión que no todos indicadores de rendimiento no tuvieron una mejora como se esperaba; ya que al agregar al sistema web tradicional la tecnología de aplicaciones web progresivas tuvo un crecimiento en el tamaño de la web como en el consumo de memoria RAM, pero en los indicadores como índice de velocidad y tiempo de respuesta sí tuvo una mejora.

4.2. Recomendaciones.

- A. Realizar un estudio que permita seleccionar los frameworks más utilizados por los desarrolladores.
- B. Desarrollar un Sistema utilizando el Framework Laravel y Vue, por la compatibilidad en su arquitectura MVC y trabajar de manera ligera.
- C. Se recomienda crear los casos de prueba que permitan evaluar ambos sistemas y nos den como resultados los indicadores propuestos.
- D. Evaluar las pruebas de rendimiento de ambos sistemas para ver la mejora mediante el uso de la tecnología de aplicaciones web progresivas.

REFERENCIAS.

- Acosta, V. M. (2019). Frontend con Angular: Todo lo que debes saber sobre esta herramienta. *Revistadigital INESEM*.
- An, D. (Mayo de 2018). Obtenido de thinkwithgoogle: <https://www.thinkwithgoogle.com/intl/es-es/canales-de-publicidad/movil/por-que%C3%A9-esperar-consigue-m%C3%A1s-ingresos-con-las-aplicaciones-web-progresivas-ahora/>
- Baca, J. (2018).
- Baker, D. L. (2003). *HTML Complete Course*.
- Cardieri, G. d., & Martinez, L. (2018). Analyzing User Experience in MobileWeb, Native and ProgressiveWeb Applications: A User and HCI Specialist Perspectives. *Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems*. doi:10.1145/3274192.3274201
- Coe, M. E. (2019). Cada segundo cuenta: la velocidad de tu sitio móvil puede definir el crecimiento de tu marca. Retrieved from <https://www.thinkwithgoogle.com/intl/es-419/recursos-y-herramientas/móvil/cada-segundo-cuenta-la-velocidad-de-tu-sitio-movil-puede-definir-el-crecimiento-de-tu-marca/>
- Conallen, J. (2002). *Building Web Applications with UML Second Edition*.
- Dayley, B., Dayley, B., & Dayley, C. (2018). *Node.js, MongoDB and Angular Web Development, Second Edition*.
- De Andrade Cardieri, G., & Zaina, L. M. (2018). Analyzing user experience in mobile web, native and progressive web applications: A user and HCI specialist perspectives. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/3274192.3274201>
- Gaunt, M. (sf). *Introducción a los service workers*. Obtenido de <https://developers.google.com/web/fundamentals/primers/service-workers/?hl=es>
- Delia, L., Galdamez, N., Corbalan, L., Pesado, P., & Thomas, P. (2017). Approaches to mobile application development: Comparative performance

- analysis. 2017 Computing Conference, 2018-Janua(July), 652–659.
<https://doi.org/10.1109/SAI.2017.8252165>
- Fortunato, D., & Bernardino, J. (2018). Progressive web apps: An alternative to the native mobile Apps. Iberian Conference on Information Systems and Technologies, CISTI, 2018-June, 1–6.
<https://doi.org/10.23919/CISTI.2018.8399228>
- Frankston, B. (2018). Progressive Web Apps [Bits Versus Electrons]. IEEE Consumer Electronics Magazine, 7(march), 106–117.
<https://doi.org/10.1109/MCE.2017.2776463>
- Gambhir, A., & Raj, G. (2018). Analysis of Cache in Service Worker and Performance Scoring of Progressive Web Application. 2018 International Conference on Advances in Computing and Communication Engineering (ICACCE), 294–299. <https://doi.org/10.1109/ICACCE.2018.8441715>
- Gilmore, W. J. (2018). Easy Laravel 5 A Hands On Introduction Using a Real-World Project. Lean Publishing, (Febuary), 1–289.
- González Ortiz, J. C. (2016). Diseño y Aplicación de una Metodología para la Ejecución de Pruebas de Carga y Stress Basada en Estándares. 217.
- Google and Nodus. (2018). Sitios Móviles en Hispanoamérica Panorama y Oportunidades.
- Gore, A. (2015). Full-Stack Vue.js 2 and Laravel 5. In Dk (Vol. 53).
<https://doi.org/10.1017/CBO9781107415324.004>
- Gourley, D., & Totty, B. (2002). HTTP the definitive guide.
- Griffith, C. (2017). *Mobile App Development with Ionic*. O'Reilly. doi:ISBN: 978-1-491-99812-0
- Hajian, M. (2019). *Progressive Web Apps with Angular*. Apress.
[doi:https://doi.org/10.1007/978-1-4842-4448-7](https://doi.org/10.1007/978-1-4842-4448-7)
- Jimenez, H., Rodriguez, R., & Tiparra, J. (1978). *Diagnóstico de TEA*. Madrid: Latinoamérica SA.
- Kappel G., P. B. (2006). *Web Engineering: The Discipline Of Systematic Development Of Web Applications*.
- Kappel, G., Pröll, B., Reich, S., & Retschitzegger, W. (2006). *Web Engineering*. Heidelberg: John Wiley & Sons, Ltd.

- Khan, A. I., Al-Badi, A., & Al-Kindi, M. (2019). Progressive Web Application Assessment Using AHP. *Procedia Computer Science*, 155, 289–294. <https://doi.org/10.1016/j.procs.2019.08.041>
- Lee, J., Kim, H., & Park, J. (2018). Pride and Prejudice in ProgressiveWeb Apps: Abusing Native App-like Features inWeb Applications. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. doi:10.1145 / 3243734.3243867
- Lepage, P. (sf). *Progressive Web Apps on Desktop*. Obtenido de <https://developers.google.com/web/progressive-web-apps/desktop>
- Lerma, R., Murcia, J. A., & Mifsud, E. (2013). *Aplicaciones web*. España.
- Lerma-Blasco, R. V, Alfredo, J., Andrés, M., Mifsud, E., Lerma-Blasco, T. R. V, Murcia, J. A., & Mifsud, E. (2013). *Aplicaciones web*. Retrieved from www.mhe.es/cf/informatica
- López, J. M. (2017). Los principales problemas de las aplicaciones móviles. Retrieved September 1, 2019, from <https://hipertextual.com/2017/07/principales-problemas-aplicaciones-moviles>
- Malavolta, I. (2016). Beyond Native Apps: Web Technologies to the Rescue! (Keynote). *Mobile! 2016 - Proceedings of the 1st International Workshop on Mobile Development, Co-Located with SPLASH 2016*, 1–2. <https://doi.org/10.1145/3001854.3001863>
- Malavolta, I., Procaccianti, G., Noorland, P., & Vukmirovi, P. (2017). Assessing the Impact of Service Workers on the Energy Efficiency of Progressive Web Apps. <https://doi.org/10.1109/MOBILESoft.2017.7>
- Martín, Á. J. (18 de Junio de 2019). *openwebinars*. Obtenido de openwebinars: <https://openwebinars.net/blog/react-native-que-es-para-que-sirve/>
- Mejia, I., Ramirez, R., Jimenez, H., & Rosas, J. (2019). A new method a architecture entreprise. *Conference IEEE bussines*, 200-215.
- Mejia, I., Tuesta, M., & Forero, M. (2020). A new method of enterprise archicture small organizations. *Computer Science Techology*, 150-170.
- Mika, P. (2007). *Social Networks and the Semantic Web*. Barcelona.
- Mozilla. (18 de marzo de 2019). *Cache*. Obtenido de <https://developer.mozilla.org/en-US/docs/Web/API/Cache>

- Nash, M. (2003). *Java Frameworks and Components: Accelerate Your Web Application Development*.
- npm trends. (2020). *npm trends*. Obtenido de <https://www.npmtrends.com/>
- Nugroho, L. E., Pratama, A. G. H., Mustika, I. W., & Ferdiana, R. (2017). Development of monitoring system for smart farming using Progressive Web App. 2017 9th International Conference on Information Technology and Electrical Engineering, ICITEE 2017, 2018-Janua, 1–5. <https://doi.org/10.1109/ICITEED.2017.8250513>
- O'Reilly, T. (2007). *What Is Web 2.0: Design Patterns and Business for the next generation of software*.
- Palatnik, S. (2017). PWA & APPs, No VS. - El Riesgo Es No Cambiar - Medium. Retrieved September 1, 2019, from <https://medium.com/el-riesgo-es-no-cambiar/pwa-y-app-experiencia-mobile-31b8c533f07b>
- Palma, J., & Marín, R. (2008). *Inteligencia Artificial*. Madrid: McGrawHill. doi:978-84-481-5618-3
- Pande, N., Somani, A., Samal, S. P., & Kakkirala, V. (2018). Enhanced Web Application and Browsing performance through Service-worker Infusion Framework. *IEEE International Conference on Web Services*, 195-202.
- Rieger, C., & Majchrzak, T. A. (2019). Towards the definitive evaluation framework for cross-platform app development approaches. *Journal of Systems and Software*, 153, 175–199. <https://doi.org/10.1016/j.jss.2019.04.001>
- Rojas, K. (2018). Identificación de efectos negativos de la TEA en el aprendizaje. *IEEE conference Techology children especial*, 200-215.
- Sansano M., H. (2017). *Pruebas sobre sitios web*.
- Seshadri, S. (2018). *Angular: Up and Running: Learning Angular, Step by Step*.
- Spero, J. (2017). Los comportamientos que están definiendo la nueva generación de experiencias móviles. Retrieved October 16, 2019, from Google website: https://www.thinkwithgoogle.com/intl/es-419/recursos-y-herramientas/móvil/consumer-behavior-mobile-digital-experiences/?utm_medium=social&utm_campaign=thinkwithgoogle&utm_source=email&utm_content=consumer-behavior-mobile-digital-experiences
- Sukoco, A., Marzuki, & Cucus, A. (2012). Concept of Quality Measurement System Software Based on Standard ISO 9126 and ISO 19011.

- SZNAJDLEDER, P. (2012). *Java a fondo - estudio del lenguaje y desarrollo de aplicaciones - 2a ed.* México: Alfaomega.
- Telefonica. (2014). Framework para el desarrollo ágil de. *AcensTechonologies*.
- WebPagetest. (2012). *webpagetest*. Obtenido de <https://sites.google.com/a/webpagetest.org/docs/using-webpagetest/metrics/speed-index>
- whatwg. (2018). *Offline Web Applications*. Obtenido de <https://html.spec.whatwg.org/multipage/offline.html#supporting-offline-caching-for-legacy-applications>
- Zaninotto, F., & Potencier, F. (2009). The Definitive Guide to symfony. In Access Zaninotto, & Potencier. (2007).

ANEXOS.

Anexo N°1 Resolución del Proyecto



**UNIVERSIDAD
SEÑOR DE SIPÁN**

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

RESOLUCIÓN N° 1813-2019/FIAU-USS

Chiclayo, 09 de diciembre de 2019

VISTO:

El Acta de Reunión N° de fecha 09 de diciembre de 2019., para la ejecución de la Tesis titulada: **"EVALUACIÓN DE RENDIMIENTO DE UN SISTEMA WEB DESARROLLADO MEDIANTE LA TECNOLOGÍA DE APLICACIONES WEB PROGRESIVAS"**, presentada por el(los) estudiante(s) **TELLO PEÑA CRISTHIAN** de la Escuela Académico Profesional de **INGENIERÍA DE SISTEMAS** y;

CONSIDERANDO:

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48° que a letra dice: *"La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas."*;

Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

SE RESUELVE:

ARTÍCULO 1°: APROBAR, el Proyecto de Tesis denominado **"EVALUACIÓN DE RENDIMIENTO DE UN SISTEMA WEB DESARROLLADO MEDIANTE LA TECNOLOGÍA DE APLICACIONES WEB PROGRESIVAS"**, perteneciente a la Línea de Investigación **INFRAESTRUCTURA, TECNOLOGÍA Y MEDIO AMBIENTE - INFRAESTRUCTURA, TECNOLOGÍA Y MEDIO AMBIENTE**, a cargo del(los) estudiante(s) **TELLO PEÑA CRISTHIAN**, de la Escuela Académico Profesional de **INGENIERÍA DE SISTEMAS**.

ARTÍCULO 2°: ESTABLECER, que la inscripción de la Tesis se realice a partir de emitida la presente resolución, y tendrá una vigencia máxima de 02 años.

REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE


UNIVERSIDAD SEÑOR DE SIPÁN S.A.
Dr. Andrés Alberto Ruiz Gómez
DECANO DE LA FACULTAD DE INGENIERÍA
ARQUITECTURA Y URBANISMO


UNIVERSIDAD SEÑOR DE SIPÁN S.A.C.

Mg. Luis Roberto Larrea Colchado
SEC. ACADEMICO FACULTAD DE INGENIERIA
ARQUITECTURA Y URBANISMO

Cc: Dirección de Investigación, CPGYT, Interesados, Archivo

ADMISIÓN E INFORMES

074 481610 - 074 481632

CAMPUS USS

Km. 5, carretera a Pimentel
Chiclayo, Perú

www.uss.edu.pe

Anexo N° 2 Ranking de los Frameworks más utilizados

Framework	Github Score	Stack Overflow Score	Overall Score
<u>React</u>	99	94	96
<u>Ruby on Rails</u>	89	99	94
<u>ASP.NET MVC</u>		94	94
<u>AngularJS</u>	91	97	94
<u>Angular</u>	90	95	92
<u>Django</u>	89	96	92
<u>Vue.js</u>	100	83	91
<u>Laravel</u>	91	92	91
<u>ASP.NET</u>	78	100	89
<u>Spring</u>	86	93	89
<u>Express</u>	89	85	87
<u>Meteor</u>	88	80	84
<u>Flask</u>	89	80	84
<u>Symfony</u>	82	86	84

Anexo N° 3 Vistas del Sistema Tradicional

Vista Login

The screenshot shows a web browser window with the URL <https://bxxwqjec.lucusvirtual.es/#/login>. The page has a blue header with the text "Login" on the left and "AMBIENTES" and "INGRESAR" on the right. The main content area features a white box titled "Inicio de Sesion" with two input fields: "Ingrese su correo electronico" and "Ingrese su password". Below the password field is a small eye icon and a "0" character. At the bottom of the box are two buttons: "INGRESAR" (blue) and "LIMPIAR" (grey). The browser's address bar and tabs are visible at the top.

Reserva de Ambientes

Vista ambientes

The screenshot shows a web browser window with the URL <https://bxxwqjec.lucusvirtual.es/#/lista-ambientes>. The page has a blue header with the text "Lista de Ambientes" on the left and "USUARIOS", "AMBIENTES", and "SALIR" on the right. The main content area displays a grid of environment cards. Each card has a title, a subtitle, and two buttons: "VER RESERVAS" and "EDITAR". The cards are: "sala de conferencias" (salon 3, black card), "fiesta" (eventos sociales, red card), "piscina" (recreacion, yellow card), "Comedor" (cocina, parrilla, green card), and "cancha de futbol" (area deportiva, blue card). A red circular button with a white plus sign is located in the bottom right corner. The browser's address bar and tabs are visible at the top.

Reserva de Ambientes

Vista reserva de ambiente

The screenshot shows a web browser window with the URL <https://bxxwqjec.lucusvirtual.es/#/reserva>. The page title is "sala de conferencias". A modal window titled "Reservar Ambiente" is open, containing the following fields:

- Fecha: 09/07/2020
- Hora (24 Hr):
 - Hora Inicio: --:--
 - Hora Final: --:--
- Motivo Reserva: Motivo Reserva
- Usuario: Tello Peña Cristhian

A "Reservar" button is located at the bottom right of the modal. The background shows a calendar grid for the month of July 2020.

Vista nuevo usuario

The screenshot shows a web browser window with the URL <https://bxxwqjec.lucusvirtual.es/#/lista-usuarios>. The page title is "Lista de Usuarios". A "NUEVO USUARIO" button is located in the top right corner. Below the button is a table of registered users:

ID ↑	Nombre	Email	Rol	Estado	Ultima Actualizacion	Opciones
2	Tello Peña Cristhian	tellocris@gmail.com	ADMINISTRADOR	HABILITADO	Invalid date	
3	oliver tello	oliver123@gmail.com	USUARIO	HABILITADO	28-06-2020	

At the bottom right, there is a pagination control showing "Rows per page: 5" and "1-2 of 2".

Vista registro de usuario

The screenshot shows a web browser window with the URL <https://bxxwjec.lucusvirtual.es/#/registrar-usuario>. The browser's address bar and tabs are visible at the top. The page has a blue header with the text "Lista de Usuarios" on the left and navigation links "USUARIOS", "AMBIENTES", and "SALIR" on the right. The main content area features a white registration form titled "Registro de Usuario". The form contains four input fields: "Nombre", "Email", "Password", and "Rol" (a dropdown menu). Below the "Rol" field is a blue button labeled "INGRESAR". At the bottom of the page, the text "Reserva de Ambientes" is visible.

Lista de Usuarios USUARIOS AMBIENTES SALIR

Registro de Usuario

Nombre

Email

Password

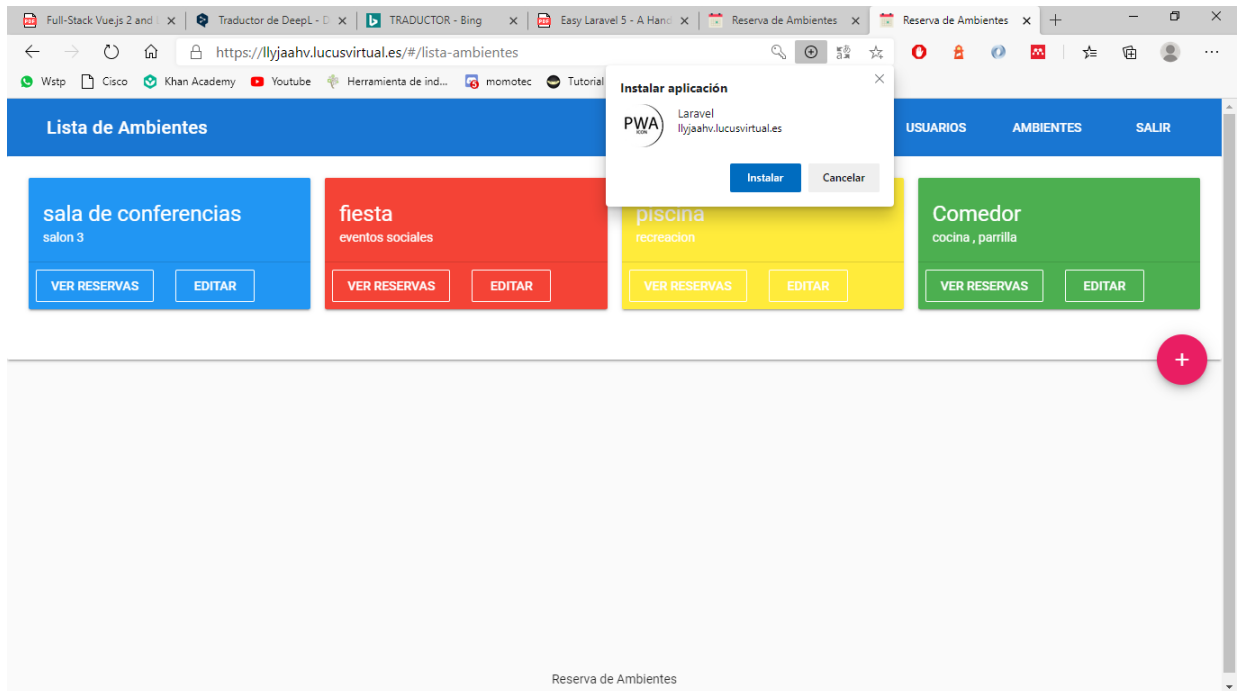
Rol

[INGRESAR](#)

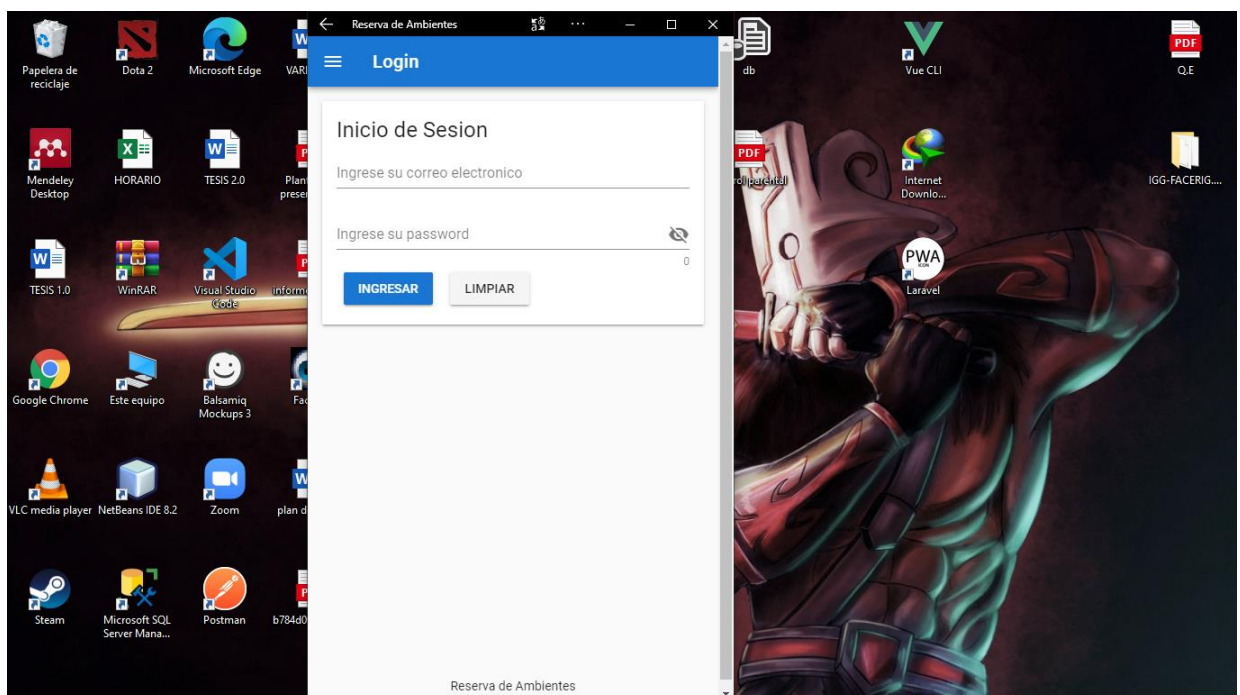
Reserva de Ambientes

Anexo N° 4 Vistas del Sistema con tecnología de aplicaciones web progresivas

Vista de instalacion de sistema



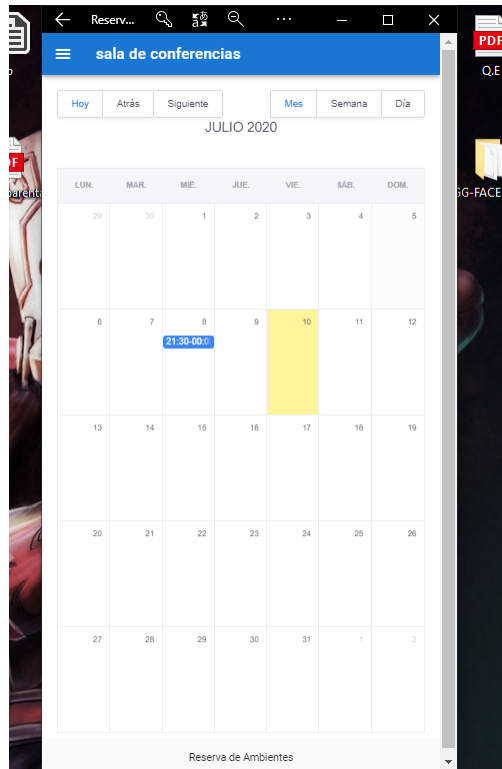
Vista login



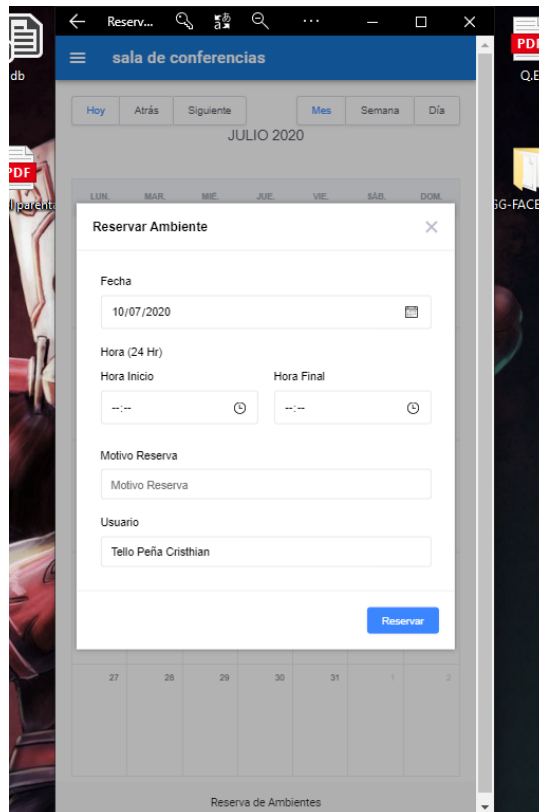
Vista Ambientes



Vista reserva hecha



Vista nueva reserva



Vista de usuarios

The screenshot shows a web application interface for managing users. The page title is "Lista de Usuarios" and it displays a list of registered users. The interface includes a search bar, a "NUEVO USUARIO" button, and a table with columns for "Estado", "Ultima Actualizacion", and "Opciones".

Estado	Ultima Actualizacion	Opciones
HABILITADO	Invalid date	
HABILITADO	28-06-2020	

1-2 of 2

Rows per page: 5

Reserva de Ambientes