



ORIGINAL

Evaluation and Analysis of Single Page Applications Using REST Services: a Case Study in Hospital Management Systems

Evaluación y Análisis de Aplicaciones de Página Simple con Servicios de Transferencia de estado representacional: estudio de Caso Sistema de Gestión Hospitalaria

Milton Paúl Lopez-Ramos¹  , Paulina Sofía Valle-Oñate²  , Luis Gonzalo Santillán-Valdiviezo¹  , Miryan Estela Narvaez-Vilema¹  , Lady Marieliza Espinoza-Tinoco¹  

¹Universidad Nacional De Chimborazo, Facultad de Ingeniería. Riobamba, Ecuador.

²Universidad Estatal de Milagro, Facultad de Ciencias e Ingeniería. Milagro, Ecuador.

Citar como: Lopez-Ramos MP, Valle-Oñate PS, Santillán-Valdiviezo LG, Narvaez-Vilema ME, Espinoza-Tinoco LM. Evaluation and Analysis of Single Page Applications Using REST Services: a Case Study in Hospital Management Systems. Salud, Ciencia y Tecnología. 2025; 5:1238. <https://doi.org/10.56294/saludcyt20251238>

Enviado: 21-04-2024

Revisado: 22-08-2024

Aceptado: 03-01-2025

Publicado: 04-01-2025

Editor: Prof. Dr. William Castillo-González 

Autor para la correspondencia: Milton Paúl Lopez-Ramos 

ABSTRACT

Single Page Applications (SPA) complemented by Representational State Transfer (REST) are key tools for data management in critical environments such as hospital systems. This research evaluated the performance of a REST-based SPA designed to manage medical records in a university hospital. Multiple functional modules were developed and integrated into the application using the agile Scrum methodology. Load simulations were performed using Apache JMeter to measure its performance, analyzing response time and server processing capacity metrics.

The results showed an average performance of 99,2 %, with response times that met the ISO/IEC 25010 standard criteria. Statistical analyses showed consistent performance across loads, demonstrating system stability and efficiency. Previous academic studies support the idea that REST-enhanced SPAs are an efficient alternative for hospital environments due to their high responsiveness and functionality.

This study emphasizes the importance of quality metrics in assessing critical applications and proposes that REST-based SPAs represent a practical and efficient solution for healthcare systems. However, testing under real-world operating conditions is recommended to validate these findings. The findings lay a solid foundation for future advancements in similar systems, highlighting the potential of these technologies to improve data management and user experience in hospital management environments.

Keywords: Web SPA; Rest; Scrum; ISO 25010.

RESUMEN

Las Aplicaciones de Página Única (SPA) complementadas por Transferencia de Estado Representacional (REST) son herramientas clave para la gestión de datos en entornos críticos como los sistemas hospitalarios. Esta investigación evaluó el rendimiento de una SPA basada en REST diseñada para gestionar registros médicos en un hospital universitario. Se desarrollaron e integraron múltiples módulos funcionales en la aplicación utilizando la metodología ágil Scrum. Se realizaron simulaciones de carga utilizando Apache JMeter para medir su rendimiento, analizando los tiempos de respuesta y las métricas de capacidad de procesamiento del servidor. Los resultados mostraron un rendimiento promedio del 99,12 %, con tiempos de respuesta que cumplieron con los criterios de la norma ISO/IEC 25010. Los análisis estadísticos mostraron un rendimiento consistente bajo diferentes cargas, demostrando la estabilidad y eficiencia del sistema. Estudios académicos previos

respaldan la idea de que las SPAs mejoradas con REST son una alternativa eficiente para los entornos hospitalarios debido a su alta capacidad de respuesta y funcionalidad.

Este estudio enfatiza la importancia de las métricas de calidad en la evaluación de aplicaciones críticas y propone que las SPAs basadas en REST representan una solución práctica y eficiente para los sistemas de salud. Sin embargo, se recomienda realizar pruebas en condiciones operativas del mundo real para validar estos hallazgos. Los hallazgos sientan una base sólida para futuros avances en sistemas similares, destacando el potencial de estas tecnologías para mejorar la gestión de datos y la experiencia del usuario en entornos de gestión hospitalaria.

Palabras Claves: Web SPA; Rest; Scrum; ISO 25010.

INTRODUCCIÓN

Los avances tecnológicos y la evolución constante de la web están impulsando a las empresas a considerar seriamente la adopción e integración de aplicaciones que requieren el consumo de datos provenientes de una API-REST. Ejemplos de plataformas que operan bajo este modelo incluyen YouTube, Facebook, entre otras destacadas en el mercado actual, un término definido por Roy Fielding. REST proporciona soporte a las arquitecturas orientadas a microservicios, que se están popularizando para la implementación del diseño de servicios basados en web.⁽¹⁾

En la actualidad, los consumidores están cada vez más habituados a la proximidad y conveniencia que brindan las aplicaciones nativas tanto en computadoras personales (PC) como en dispositivos móviles, independientemente del sistema operativo que utilicen. Exigen esa misma cercanía y sencillez en la experiencia que les proporciona una aplicación web, sin embargo, estas últimas siguen estando sujetas a la intermediación de los navegadores web que actúan como puente entre la aplicación y el usuario.⁽²⁾ Las aplicaciones web SPA (Single Page Applications) han ganado popularidad en los últimos años al proporcionar una experiencia de usuario más dinámica y atractiva. Este tipo de aplicaciones ofrecen una ventaja competitiva significativa en el mercado actual.⁽³⁾ Frameworks tan populares como React, Angular y Vue.js siguen dando forma a la manera en que los desarrolladores crean interfaces de usuario y aplicaciones de una sola página. Además, están surgiendo nuevas bibliotecas y marcos de trabajo para abordar necesidades específicas, como la gestión de estados, el enrutamiento y la renderización del lado del servidor.⁽⁴⁾

La tendencia actual en el ámbito del desarrollo de sitios y aplicaciones web consiste en la integración de los beneficios que ofrece la arquitectura API-REST, en conjunción con la versatilidad y eficiencia de las páginas web SPA. Una de las características más sobresalientes y notables de las páginas web de aplicación única es su destacada capacidad para cargar de manera rápida y fluida sus diversas pantallas y secciones, a pesar de que el usuario experimente una navegación que simula la de múltiples páginas individuales. En realidad, lo que sucede es que se trata de una sola página web que se renueva y actualiza dinámicamente en tiempo real.

En comparación con las aplicaciones web basadas en hipertexto, las páginas web SPA no tienen una estructura de navegación explícita. En cambio, su flujo de trabajo es controlado implícitamente por los estados de sus elementos de control (widgets). Por ejemplo, los widgets pueden aparecer o desaparecer para proporcionar información diferente al usuario, la información proporcionada por un widget cambia con el tiempo, y los widgets se pueden deshabilitar para impedir que el usuario realice una determinada acción.⁽⁵⁾

En las SPA, todos los componentes necesarios se descargan en la primera carga de la página. El timeline de twitter y la bandeja de entrada de Gmail son ejemplos de SPA. Sin embargo, algunos contenidos y componentes se cargan dinámicamente a medida que el usuario interactúa con él. Por ejemplo, en la misma página web un usuario puede cambiar de pestaña, abrir un diálogo para publicar nuevos contenidos y ejecutar operaciones de paginación sin cambiar la ruta URL en la del navegador.⁽⁶⁾

Las aplicaciones web SPA utilizan servicios REST para que los clientes (front-end) y servidores (back-end) estén conectados libremente. REST es responsable de proporcionar autenticación, autorización, verificación y procesamiento de datos. Después de una conexión exitosa, se pueden realizar varias acciones como las operaciones CRUD, que son CREATE, READ, UPDATE, y DELETE.⁽⁷⁾

En el trabajo de la bibliografía ⁽⁸⁾ se presenta una arquitectura modular y compleja para una SPA, basada en componentes que abarcan todos los niveles de la aplicación, desde el cliente hasta el almacenamiento. Aunque se enfatiza la funcionalidad del diseño de páginas web SPA, no se abordan aspectos relacionados con su operación conjunta con web service. Por otro lado, Hernández⁽⁹⁾ propone un modelo de operación para páginas web SPA utilizando web services REST, pero no incluye una implementación práctica del modelo.

Para evaluar el rendimiento de aplicaciones SPA basadas en REST sobre entornos críticos de operación, se propone un conjunto de medidas que permitan conocer la operatividad de este tipo de aplicaciones. Las herramientas de software modernas son ampliamente empleadas por las organizaciones con el propósito de

mejorar la eficiencia de sus operaciones, por consiguiente, es fundamental que dichas herramientas cumplan con una variedad de estándares rigurosos para garantizar su fiabilidad y desempeño óptimo. La calidad del software se refiere al nivel de rendimiento que se espera que el software mantenga a lo largo de su ciclo de vida. Estos estándares son fundamentales para asegurar la fiabilidad y eficacia del software en su funcionamiento.⁽¹⁰⁾ La calidad en el desarrollo de software es un concepto fundamental que se emplea para evaluar las múltiples facetas, atributos y capacidades de un sistema informático, así como para analizar cómo estos elementos se comportan y se desempeñan a lo largo del tiempo, considerando tanto su precisión como su eficacia. Este procedimiento es fundamental para evaluar si un programa informático cumple con los requerimientos y demandas de los usuarios y/o clientes que lo utilizan.⁽¹¹⁾

El Institute of Electrical and Electronics Engineers (IEEE), una organización reconocida a nivel internacional en el ámbito de la ingeniería eléctrica y electrónica, establece de manera clara que una métrica de calidad de software se define como “una función matemática cuyas entradas consisten en diversos datos relacionados con el software, y cuya salida es un único valor numérico que puede ser interpretado como el nivel o grado en el que el software en cuestión posee un atributo específico que incide directamente en su calidad general”. Se la puede aplicar a una amplia variedad de productos, desde bienes de consumo hasta servicios de software e infraestructura tecnológica. Si se implementan las métricas correctas y se lleva a cabo un análisis exhaustivo, el objeto de evaluación, que en este caso es el software desarrollado, tendrá la posibilidad de experimentar una reducción significativa en la cantidad de fallos que puedan surgir durante su operatividad.⁽¹²⁾

Las métricas de calidad para productos de software son fundamentales para evaluar la eficacia y fiabilidad de un programa informático. Las métricas consideran una variedad de aspectos, como las dimensiones del software, la complejidad del código, el desempeño del sistema, la accesibilidad y la seguridad del software. Además, se analizan métricas vinculadas al proceso de pruebas, como la cantidad de trabajo dedicada a las pruebas y la efectividad de las pruebas llevadas a cabo, junto con otras variables pertinentes. Para asegurar la calidad y el adecuado desempeño de los productos de software, resulta fundamental realizar un análisis y medición exhaustivos de dichos aspectos.⁽¹²⁾

El rendimiento es un indicador fundamental para evaluar la eficacia de los servicios web, ya que permite medir la cantidad de solicitudes que un servidor es capaz de procesar en un determinado lapso, lo cual resulta crucial para analizar tanto la carga de trabajo como la capacidad de respuesta del servidor. El tiempo que transcurre desde que un servidor recibe una solicitud hasta que envía una respuesta al cliente se denomina tiempo de respuesta, también identificado como tiempo de latencia. Este término hace referencia al lapso que requiere un servidor con operaciones de forma asíncrona, como, por ejemplo, realizar consultas en una base de datos.

En este caso de estudio específico, se analizó la carencia de un sistema de registro de historias clínicas en el Hospital Universitario Andino de Chimborazo. Esta carencia abarcaba desde el momento de ingreso del paciente, pasando por el registro de su estado de salud, la evaluación médica, el seguimiento por parte del personal de enfermería, hasta los procedimientos administrativos asociados a esta labor.

En este contexto específico, se desarrolló de una aplicación web SPA basada en REST, con el propósito de facilitar el registro de pacientes y el seguimiento detallado de las historias clínicas en el entorno de la unidad hospitalaria. Además de cumplir con esta necesidad operativa crucial para el funcionamiento del hospital, se tenía como objetivo principal evaluar y demostrar el rendimiento y la eficacia que ofrecían este tipo de aplicaciones.

Para esta tarea, se optó por emplear la reconocida herramienta Apache JMeter, la cual permitió simular de manera precisa y detallada tanto la operación como la respuesta de la aplicación, como paso previo esencial antes de implementar plenamente este sistema en el entorno hospitalario. La aplicación fue desarrollada considerando los requerimientos de los usuarios del hospital sin embargo para exponer los resultados de esta investigación se consideró únicamente los resultados de la simulación, debido a que en la puesta en marcha el comportamiento del sistema fue completamente normal y no demandó esfuerzo adicional del servidor.

Para evaluar la calidad del aplicativo de manera detallada, se aplicó el estándar internacional ISO/IEC 25010,⁽¹³⁾ el cual según ⁽¹⁴⁾ proporciona un marco para analizar diversas características y subcaracterísticas del software, incluyendo funcionalidad, confiabilidad, usabilidad, eficiencia y mantenibilidad, entre otras. Este estándar, ampliamente reconocido en la industria, establece criterios objetivos y medibles que facilitan la evaluación y mejora sistemática de la calidad del software.

MÉTODO

El objetivo de la investigación es establecer la métrica que permita analizar y evaluar el rendimiento del servidor REST en la gestión de fichas médicas. Para ello, se identificaron recursos académicos que aportan información relevante sobre los beneficios de las aplicaciones web SPA, los servicios REST y las métricas del modelo de nivel de servicio (basadas en el tiempo de respuesta) según la norma ISO/IEC 25010,⁽¹³⁾ con el fin de determinar si el rendimiento del servidor REST es óptimo. Como se puede ver en la figura 1.

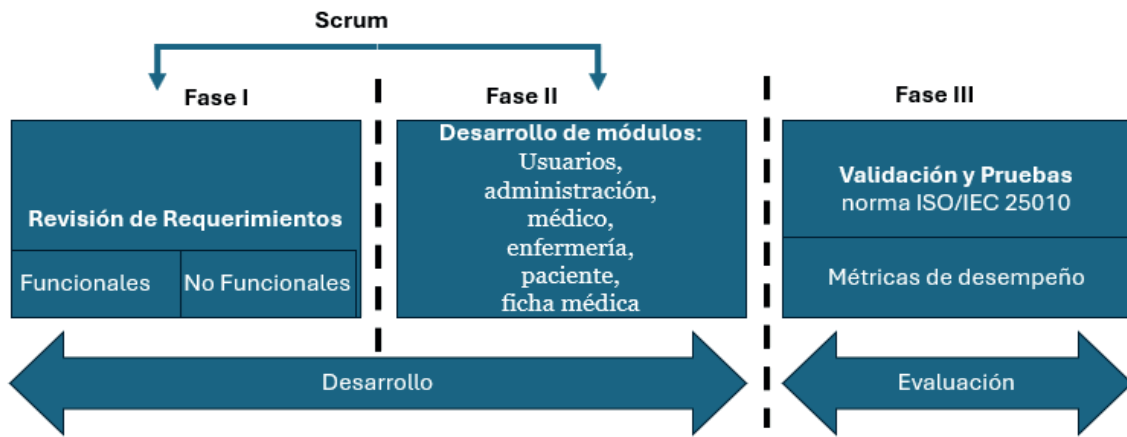


Figura 1. Metodología

Se aplicó la metodología ágil Scrum para planificar y gestionar el desarrollo del software. El proceso se dividió en varios sprints, cada uno con un período breve de tiempo fijo destinado a completar una determinada cantidad de trabajo. Los sprints incluyeron la identificación de las necesidades funcionales y no funcionales, así como la creación e implementación de los módulos de usuario, administración, médico, enfermería, paciente y ficha médica. Posteriormente, se realizaron pruebas de rendimiento de la aplicación utilizando Apache JMeter, que permitió simular las cargas de petición y respuesta del servidor. En cada carga se definieron 496 solicitudes a los endpoints, y el número total de peticiones para el presente trabajo fue de 2 400.

Para medir el rendimiento, se utilizó la ecuación de rendimiento de servicio propuesta en las métricas para la Interoperabilidad de la Información.⁽¹⁵⁾

$$w = \frac{Nro(T^w < T_0; \text{en período } t)}{Nro(\text{Peticiones}^w; \text{en período } t)} \quad (1)$$

Donde:

- w = rendimiento de servicio
- T^w = número de tareas completadas
- T_0 = tiempo requerido

Finalmente, se llevó a cabo un análisis estadístico para comprobar la hipótesis de que una SPA basada en REST es adecuada y eficiente bajo diversas condiciones de carga. Para la parte estadística, se realizaron los siguientes tests:

- Test de Shapiro-Wilk (Menor carga): para verificar la normalidad de los datos bajo condiciones de menor carga.
- Test de Shapiro-Wilk (Mayor carga): para determinar la normalidad bajo condiciones de mayor carga.
- Test de Wilcoxon: utilizado para comparar el rendimiento del servidor REST entre diferentes condiciones de carga, al no cumplir los datos con la normalidad requerida.

Estos análisis permitieron establecer una base sólida para evaluar el comportamiento del servidor REST bajo diferentes escenarios de carga, aportando evidencia sobre su capacidad para gestionar eficientemente las fichas médicas.

DESARROLLO

La figura 2 ilustra de manera detallada el desarrollo de la investigación, iniciando con el análisis documental que incluyó bases de datos científicas como Springer, así como motores de búsqueda indexados como Google Académico y ResearchGate.

La evaluación de la calidad del software se llevó a cabo siguiendo la norma ISO/IEC 25010.⁽¹³⁾ Para la selección de los documentos se consideraron los siguientes parámetros:

- Trabajos relacionados con páginas web SPA.
- Trabajos relacionados sobre páginas SPA con REST.
- Calidad del software.

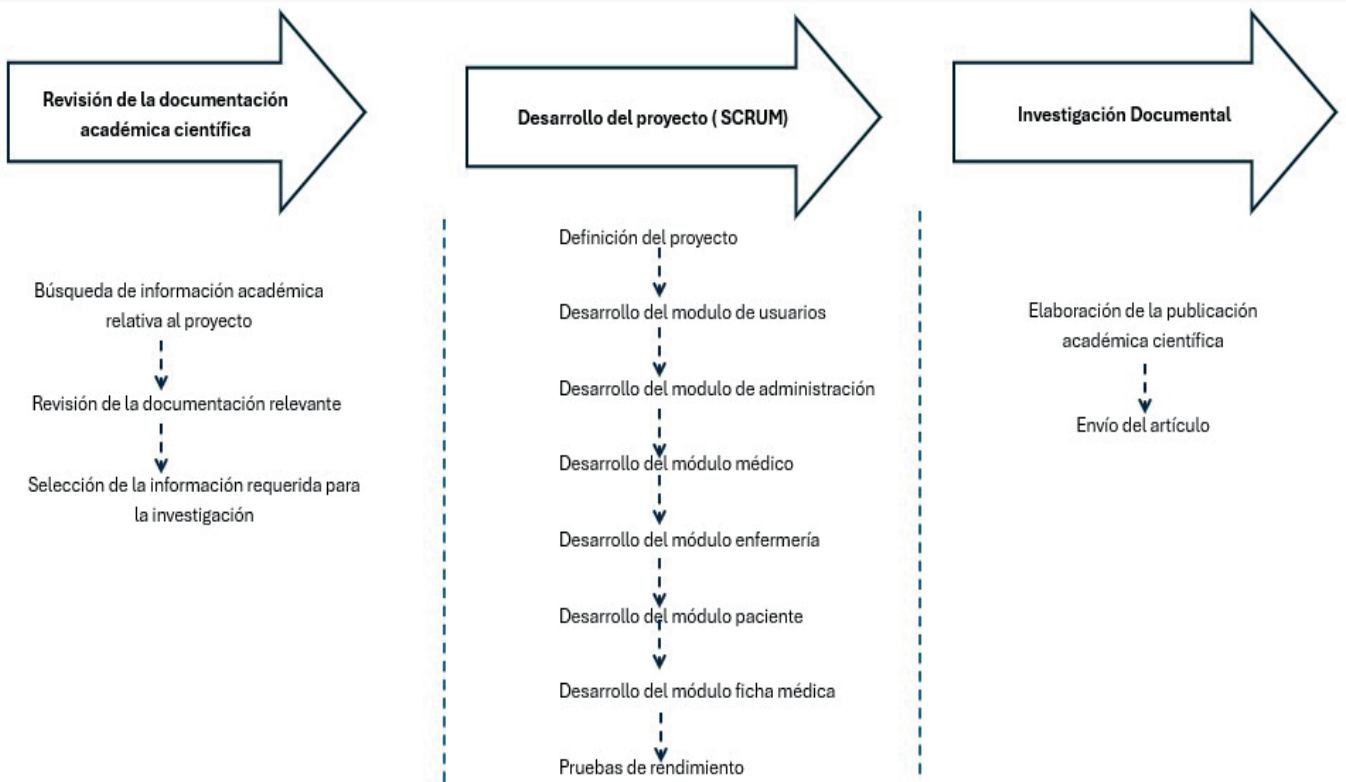


Figura 2. Línea de desarrollo de la investigación

Posteriormente, se procedió a establecer de manera precisa la definición y el alcance detallado del proyecto de desarrollo de software, teniendo en cuenta la amplia gama de requisitos y relatos de experiencia de los diversos usuarios que interactúan con el centro hospitalario. Mediante la adopción de un enfoque iterativo incremental fundamentado en la metodología SCRUM, se llevaron a cabo cuidadosos procesos de planificación de múltiples sprints con el propósito de ejecutar la creación y puesta en marcha de la aplicación web SPA y el registro detallado de las fichas médicas. La estructura y disposición técnica de la aplicación pueden ser visualizadas detalladamente en la representación gráfica proporcionada en la figura 3.

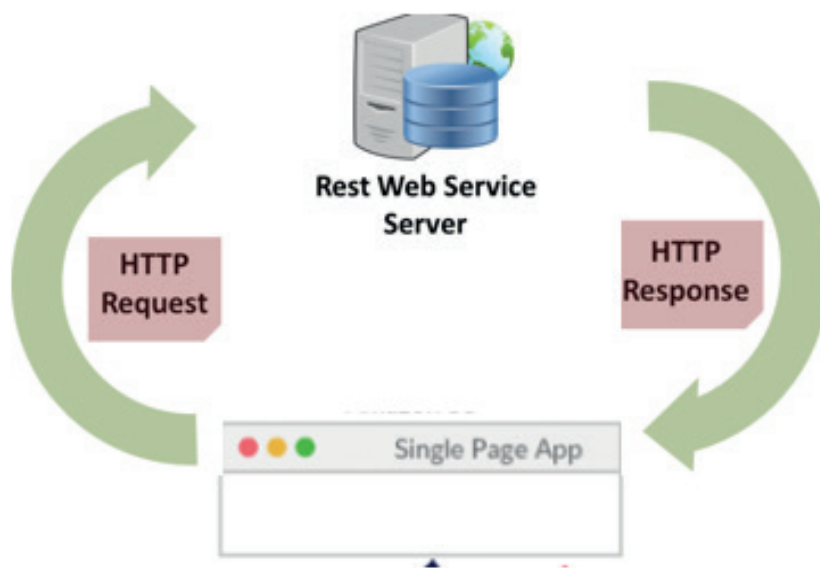


Figura 3. Arquitectura de la aplicación de fichas médicas

La evaluación del rendimiento del software se realizó con cargas simuladas al servidor a través de la aplicación Apache JMeter.

Herramientas utilizadas

Durante el proceso de desarrollo de la aplicación web SPA y servicios REST, se hicieron uso de diversas herramientas tecnológicas que desempeñaron un papel fundamental en la optimización de la creación y la implementación del sistema. A continuación, se presenta en la tabla 1 un resumen de las herramientas más relevantes y sus respectivas aplicaciones en diversos contextos:

Tabla 1. Herramientas de desarrollo	
Herramientas	Definición
ClickUp	Gestión de proyectos basada en metodologías ágiles como Scrum.
Git	Control de versiones para garantizar la integridad y el historial del código.
GitHub	Plataforma para almacenar, colaborar y gestionar proyectos mediante Git.
Node.js	Ambiente de ejecución para JavaScript, usado principalmente en el lado del servidor.
Beekeeper Studio	Herramienta para la gestión y consulta eficiente de bases de datos.
PostgreSQL	Base de datos relacional utilizada en el almacenamiento de información.
Insomnia	Ciente diseñado para la creación y prueba de APIs REST.
Visual Studio Code	Editor de código fuente con extensiones para múltiples lenguajes de programación.

Conforme a la metodología previamente establecida para llevar a cabo la evaluación de desempeño, se procedió a implementar una carga de 496 peticiones en los distintos puntos finales (dispositivo electrónico remoto que establece comunicación a través de una red a la que se encuentra enlazado) que conforman los diversos módulos del sistema. En la figura número 4 se detallan los endpoints a los que se les aplicaron las cargas simuladas durante el experimento. Además, se estableció un período mínimo requerido T_0 de 400 milisegundos, tal como se especifica en la métrica de desempeño del servicio, y se procedió a aplicar la ecuación número 1 correspondiente.

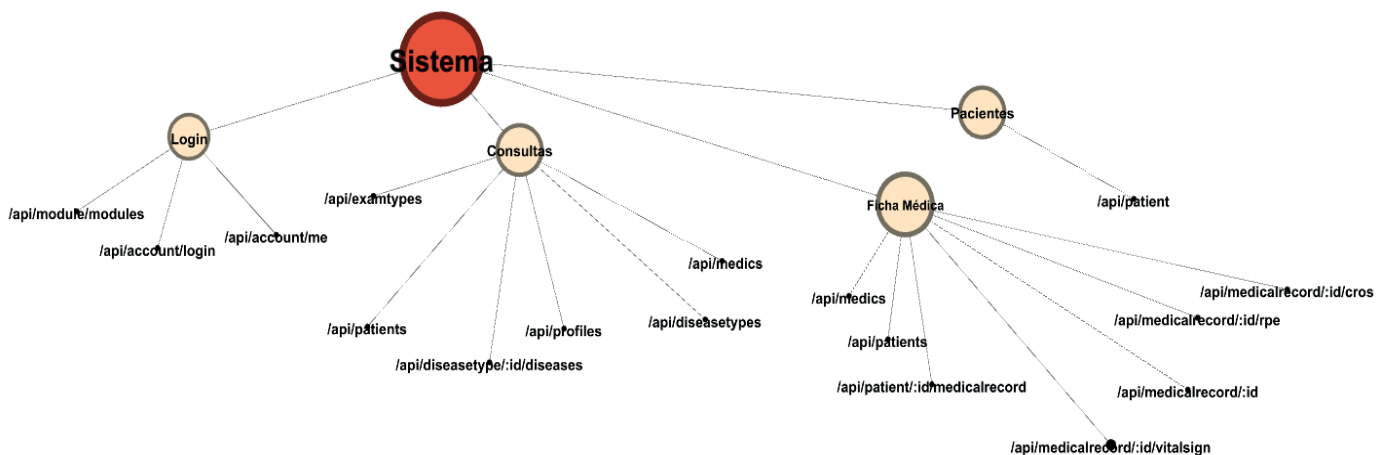


Figura 4. Endpoints

De acuerdo con la información proporcionada, el módulo de Inicio de Sesión dispone de un total de tres endpoints, el de Pacientes cuenta con uno, el de Consultas tiene asignados siete y, por último, el módulo de Ficha Médica presenta un total de nueve endpoints disponibles para su uso.

Cada módulo del sistema se encuentra asignado a una secuencia de operación particular. Por ejemplo, en el módulo de Login, es necesario seguir meticulosamente la serie de instrucciones que se ilustra en la Figura 5 del manual de usuario. De la misma manera, siguiendo el mismo procedimiento con el resto de los módulos, se puede construir una gráfica similar para representar la relación existente entre el tiempo de respuesta por cada endpoint en el sistema.

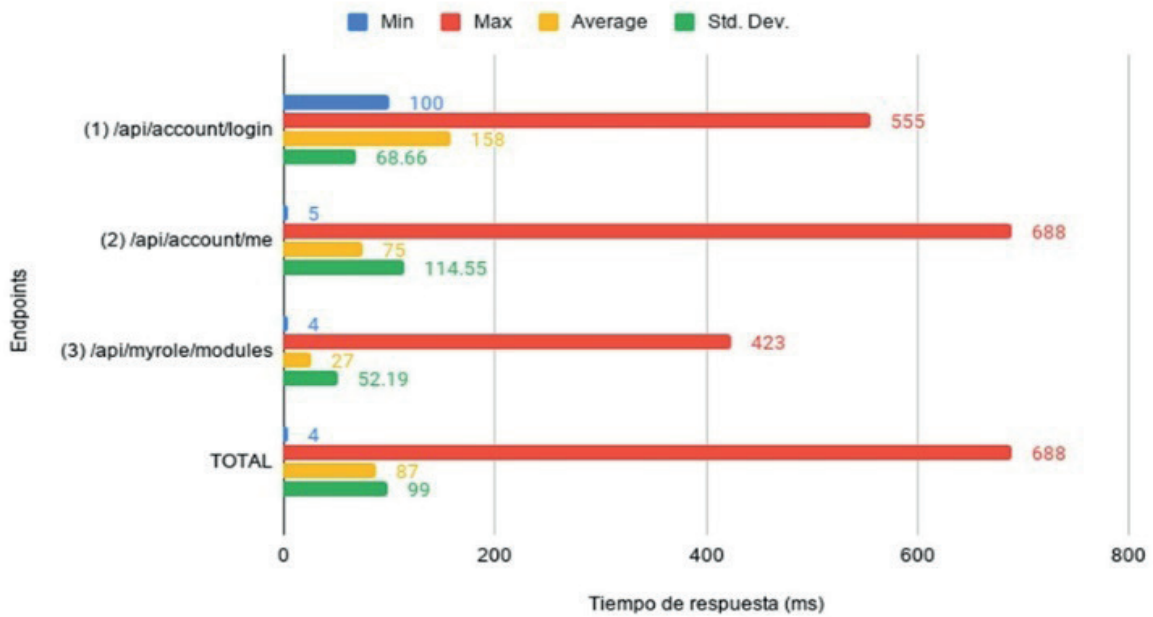


Figura 5. Rendimiento del módulo Login

A continuación, se describe la tabla 2 el resumen tras las pruebas de carga desarrolladas sobre el servidor, así como el análisis de los estadísticos aplicados en la tabla 3.

Módulo	Tiempo de respuesta con menor carga	Tiempo de respuesta con mayor carga	Promedio del tiempo de respuesta	Rendimiento
Login	4 ms	688 ms	99 ms	97,58 %
Pacientes	63 ms	591 ms	79,1 ms	97,58 %
Consultas	3 ms	217 ms	33 ms	100 %
Ficha médica	3 ms	855 ms	119 ms	98,81 %
Total				99,12 %

Prueba	Estadístico	p-valor	Interpretación
Shapiro-Wilk (Menor carga)	0,6418	0,0019	No normal
Shapiro-Wilk (Mayor carga)	0,9440	0,6786	Normal
Wilcoxon	0,0	0,125	No hay diferencia significativa entre menor y mayor carga

Los resultados de la exhaustiva investigación realizada de la aplicación web SPA basada en REST, se puede observar que la métrica del tiempo de respuesta se muestra coherente con la métrica de rendimiento de servicio establecida en la norma ISO/IEC 25010. Esto permite determinar con precisión el rendimiento de la aplicación en relación con la respuesta del servidor REST, demostrando su correcto funcionamiento y eficiencia en una amplia gama de condiciones de carga. Este hecho queda respaldado por el análisis estadístico realizado, el cual indica que las métricas de tiempo de respuesta no presentan correlaciones significativas entre las distintas condiciones de carga evaluadas. Adicionalmente, cabe resaltar que el análisis de la prueba de Wilcoxon indica que no existen diferencias notables en los tiempos de respuesta frente a diversas cargas laborales, tal como se puede observar de manera detallada en la figura número 6 de la presente investigación científica.

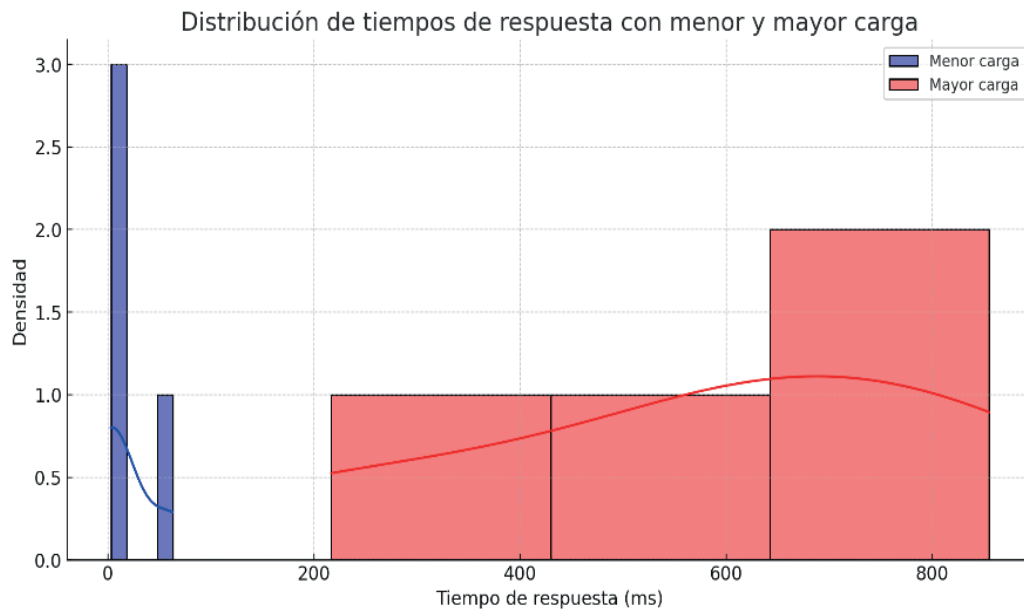


Figura 6. Distribución de tiempo de respuesta con mayor y menor carga

DISCUSIÓN

Los datos recopilados y analizados indican que el servidor REST presenta un nivel de eficiencia satisfactorio para administrar registros médicos en una plataforma web de aplicación de página única (SPA) en diferentes escenarios de carga. La métrica del tiempo de respuesta se mantuvo dentro de los estándares establecidos en la norma ISO/IEC 25010, lo cual indica que el sistema es capaz de manejar múltiples solicitudes sin afectar significativamente su eficiencia. En particular, los resultados de la prueba de Shapiro-Wilk mostraron que los datos bajo menor carga no seguían una distribución normal, mientras que los datos bajo mayor carga sí lo hacían. Esta diferencia sugiere que el comportamiento del sistema varía según el nivel de carga, aunque no de manera que afecte su rendimiento general.

Además, la prueba de Wilcoxon indicó que no hay diferencias significativas entre los tiempos de respuesta bajo diferentes niveles de carga, lo cual respalda la hipótesis de que la aplicación es eficiente independientemente de la carga aplicada. Estos resultados son consistentes con otros estudios que han evaluado el rendimiento de aplicaciones SPA utilizando servicios REST, en los que se ha observado una alta eficiencia y capacidad de respuesta incluso bajo condiciones de carga elevadas.

El uso de Apache JMeter como herramienta para la simulación de cargas permitió evaluar de manera precisa el rendimiento del sistema en diferentes escenarios. Los tiempos de respuesta promedio obtenidos, junto con los análisis estadísticos realizados, demuestran que el servidor REST es capaz de soportar el volumen de solicitudes esperado en un entorno hospitalario, lo cual es fundamental para garantizar un servicio continuo y de calidad.

A pesar de los resultados positivos, es importante señalar que esta investigación se basó en un entorno simulado, por lo que sería recomendable realizar pruebas adicionales en un entorno real para validar la efectividad del sistema bajo condiciones operativas más diversas. Además, futuros trabajos podrían considerar la incorporación de técnicas de optimización adicionales para mejorar aún más el rendimiento, especialmente en escenarios con cargas muy elevadas.

CONCLUSIONES

La aplicación web SPA y el servicio REST operan bajo una arquitectura cliente-servidor, logrando bajos tiempos de respuesta al cargar toda la aplicación una única vez, evitando recargas innecesarias con cada acción del usuario en el sistema de fichas médicas. Las métricas de calidad, como la métrica de rendimiento de servicio definida en la norma ISO/IEC 25010, son fundamentales para garantizar la calidad del producto software. En este contexto, la herramienta Apache JMeter permitió aplicar un conjunto de cargas de trabajo al servidor REST, obteniendo datos que evaluaron el rendimiento de cada endpoint, con un promedio de rendimiento del 99,12 %. Como proyección futura, se propone la integración de aplicaciones web progresivas con tecnologías de internet de las cosas e Industria 4.0, facilitando la captura y gestión de datos desde dispositivos generadores de información digital. Cabe destacar que este proyecto corresponde a un caso de estudio específico, por lo que no es posible generalizar sus resultados; sin embargo, estos pueden contrastarse con proyectos similares para obtener una valoración más precisa del proceso utilizado.

REFERENCIAS BIBLIOGRÁFICAS

1. Kumar K, Jain AK, Tiwari RG, Jain N, Gautam V, Trivedi NK. Analysis of API architecture: A detailed report. In: 2023 IEEE 12th International Conference on Communication Systems and Network Technologies (CSNT). 2023. p. 880-4.
2. Lanza O. Aplicación Web Progresiva (PWA) para la gestión de pagos de estacionamiento en superficie [Internet]. 2019. Available from: <https://repositorio.unican.es/xmlui/handle/10902/17089>
3. Curie DH, Jaison J, Yadav J, Fiona JR. Analysis on web frameworks. In: Journal of Physics: Conference Series. 2019.
4. Abdurakhimovich UA. The Future of JavaScript: Emerging Trends and Technologies. Formation of Psychology and Pedagogy as Interdisciplinary Sciences [Internet]. 2023;2(21):12-4. Available from: <https://interonconf.org/index.php/ity/article/view/6865>
5. Zhang G. Specifying and Model Checking Workflows of Single Page Applications with TLA. In: IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C) [Internet]. 2020. p. 406-10. Available from: <https://arxiv.org/pdf/2005.05627.pdf>
6. Hasan Md, Rahman M, Chowdhury Md, Rahman Md, Abdulle K, Sadia F, et al. Testing React Single Page Web Application using Automated Testing Tools. In 2022. p. 469-76.
7. Chakraborty S, Aithal PS. CRUD Operation on WordPress Database Using C# And REST API. International Journal of Applied Engineering and Management Letters (IJAEML) [Internet]. 2023;7(4):130-8. Available from: <https://supublication.com/index.php/ijaeml/article/view/979>
8. Gavrilă V, Băjenaru L, Dobre C. Modern single page application architecture: A case study. Studies in Informatics and Control. 2019;28(2):231-8.
9. Hernandez-Mendez A, Scholz N, Matthes F. A Model-driven Approach for Generating RESTful Web Services in Single-Page Applications. In: International Conference on Model-Driven Engineering and Software Development. 2018. p. 480-7.
10. Callejas-Cuervo M, Alarcón-Aldana AC, Álvarez-Carreño AM. Modelos de calidad del software, un estado del arte. Entramado [Internet]. 2017;13(1):236-50. Available from: http://www.scielo.org.co/scielo.php?pid=S1900-38032017000100236&script=sci_arttext
11. Goyes FL. Mejora de la calidad del software a través de la integración y entrega continua. Revista Odigos. 2023;4(2):45-55.
12. Castaño JF, Castillo W. Métricas en la evaluación de la calidad del software: una revisión conceptual. CESTA. 2021;2(2):21-6.
13. iso25000. Normas-iso-25000 [Internet]. 2024. Available from: <https://iso25000.com/index.php/normas-iso-25000>
14. Matute SA, Avila-Pesantez D, Avila LM. Desarrollo de sistema Web basado en los frameworks de Laravel y VueJs, para la gestión por procesos: Un estudio de caso. Revista Peruana de Computación y Sistemas. 2020;3(1).
15. Moreno M. Métricas para la Interoperabilidad de la Información en el Gobierno Electrónico [Internet]. 2007. Available from: <https://users.dcc.uchile.cl/~mnmonsal/memoria.pdf>

FINANCIACIÓN

Ninguno de los autores recibió financiación para el desarrollo de la presente investigación.

CONFLICTO DE INTERESES

Los autores declaramos que no existe conflicto de intereses.

CONTRIBUCIÓN DE AUTORÍA

Conceptualización: Milton Paúl Lopez-Ramos, Paulina Sofía Valle-Oñate, Luis Gonzalo Santillán-Valdiviezo,

Miryan Estela Narvaez-Vilema, Lady Marieliza Espinoza-Tinoco.

Curación de datos: Milton Paúl Lopez-Ramos, Paulina Sofía Valle-Oñate, Luis Gonzalo Santillán-Valdiviezo, Miryan Estela Narvaez-Vilema, Lady Marieliza Espinoza-Tinoco.

Análisis formal: Milton Paúl Lopez-Ramos, Paulina Sofía Valle-Oñate, Luis Gonzalo Santillán-Valdiviezo, Miryan Estela Narvaez-Vilema, Lady Marieliza Espinoza-Tinoco.

Investigación: Milton Paúl Lopez-Ramos, Paulina Sofía Valle-Oñate, Luis Gonzalo Santillán-Valdiviezo, Miryan Estela Narvaez-Vilema, Lady Marieliza Espinoza-Tinoco.

Metodología: Paulina Sofía Valle-Oñate, Luis Gonzalo Santillán-Valdiviezo, Miryan Estela Narvaez-Vilema.

Administración del proyecto: Milton Paúl Lopez-Ramos, Miryan Estela Narvaez-Vilema, Lady Marieliza Espinoza-Tinoco.

Recursos: Milton Paúl Lopez-Ramos, Paulina Sofía Valle-Oñate.

Software: Milton Paúl Lopez-Ramos, Miryan Estela Narvaez-Vilema, Lady Marieliza Espinoza-Tinoco.

Supervisión: Milton Paúl Lopez-Ramos.

Validación: Milton Paúl Lopez-Ramos, Paulina Sofía Valle-Oñate, Luis Gonzalo Santillán-Valdiviezo, Miryan Estela Narvaez-Vilema, Lady Marieliza Espinoza-Tinoco.

Visualización: Luis Gonzalo Santillán-Valdiviezo, Miryan Estela Narvaez-Vilema, Lady Marieliza Espinoza-Tinoco.

Redacción-borrador Original: Paulina Sofía Valle-Oñate, Luis Gonzalo Santillán-Valdiviezo, Miryan Estela Narvaez-Vilema.

Redacción - revisión y edición: Paulina Sofía Valle-Oñate, Luis Gonzalo Santillán-Valdiviezo, Miryan Estela Narvaez-Vilema.