



**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE GUAYAQUIL  
CARRERA DE COMPUTACIÓN**

**Evaluación de eficiencia y escalabilidad de un prototipo web de inventario desarrollado  
con MERN Stack**

Trabajo de titulación previo a la obtención del  
Título de Ingeniero en ciencias de la computación

AUTOR: Joel Washington Vinueza Garcia

TUTOR: Joe Frand Llerena Izquierdo Ing., Msc

Guayaquil – Ecuador

2024

## **CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN**

Yo, Joel Washington Vinueza Garcia con documento de identificación N° 0931309496 manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 16 de julio del 2024

Atentamente,



---

Joel Washington Vinueza Garcia  
0931309496

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE  
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, Joel Washington Vinueza Garcia con documento de identificación No. 0931309496, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor(a) del Artículo Académico: “Evaluación de eficiencia y escalabilidad de un prototipo web de inventario desarrollado con MERN Stack”, el cual ha sido desarrollado para optar por el título de: Ingeniero en Computación, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 16 de julio del 2024

Atentamente,



---

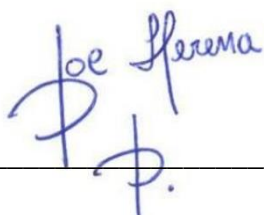
Joel Washington Vinueza Garcia  
0931309496

## CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Joe Frand Llerena Izquierdo con documento de identificación N° 0914884879, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: Evaluación de eficiencia y escalabilidad de un prototipo web de inventario desarrollado con MERN Stack, realizado por Joel Washington Vinueza Garcia con documento de identificación N° 0931309496, obteniendo como resultado final el trabajo de titulación bajo la opción Artículo Académico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 16 de julio del 2024

Atentamente,



---

Joe Frand Llerena Izquierdo Ing., Msc

0914884879

## **DEDICATORIA**

Quiero dedicar este logro a mis amigos más cercanos, cuyo apoyo y compañía han iluminado mi camino durante este emocionante viaje. Su amistad ha sido un regalo invaluable, lleno de risas compartidas, momentos memorables y un apoyo inquebrantable en los momentos más difíciles. Agradezco su presencia constante y su ánimo sincero, que han enriquecido mi vida más allá de las palabras. Este éxito también es suyo, y celebro con gratitud su amistad que ha sido fundamental en mi crecimiento personal y profesional.

Joel Washington Vinueza Garcia

## **AGRADECIMIENTO**

Quiero dedicar unas palabras especiales de gratitud a mis padres, cuyo apoyo incondicional y orientación han sido pilares fundamentales a lo largo de mi trayectoria académica y profesional. Su constante aliento, sabiduría y sacrificio han sido la fuerza que me impulsó a superar desafíos y alcanzar metas. Este logro no solo es mío, sino también de ellos, quienes han sido mi inspiración y sostén en cada paso del camino.

Joel Washington Vinueza Garcia

## RESUMEN

El artículo examina la implementación del MERN Stack para mejorar la gestión de inventarios empresariales, comenzando con una revisión teórica que contrasta las limitaciones de sistemas tradicionales con las ventajas de esta tecnología avanzada. La metodología combina investigación cuantitativa y descriptiva, enfocándose en desarrollar un prototipo web de inventario bajo un marco ágil como Scrum. Se recopilan datos empíricos y se realizan pruebas exhaustivas de rendimiento, incluyendo mediciones detalladas de tiempos de respuesta y análisis de consumo de recursos del servidor.

El desarrollo del prototipo utiliza Node.js para la lógica de negocio y React.js para la interfaz de usuario, destacando la capacidad de React.js para crear interfaces dinámicas y la eficiencia de Node.js en la gestión de datos intensivos. Los resultados muestran mejoras significativas en comparación con tecnologías tradicionales, validando la eficiencia y escalabilidad del MERN Stack en entornos empresariales. Se subraya la importancia de adoptar tecnologías modernas para optimizar la gestión de inventarios y mantener la competitividad en un mercado dinámico. En resumen, el estudio proporciona una evaluación integral de cómo el MERN Stack puede transformar la gestión de inventarios, ofreciendo recomendaciones prácticas para futuras implementaciones y mejoras continuas en sistemas empresariales.

**Palabras claves:** MERN Stack, Gestión de inventarios, Eficiencia, Escalabilidad, Tecnologías modernas.

## ABSTRACT

This article explores the implementation of the MERN Stack to enhance enterprise inventory management, beginning with a theoretical review that contrasts the limitations of traditional systems against the advantages of this advanced technology. The methodology combines quantitative and descriptive research, focusing on developing a web inventory prototype under an agile framework like Scrum. Empirical data is collected, and exhaustive performance testing is conducted, including detailed measurements of response times and server resource consumption.

The prototype development leverages Node.js for business logic and React.js for the user interface, highlighting React.js's capability to create dynamic interfaces and Node.js's efficiency in handling data-intensive operations. Results demonstrate significant improvements compared to traditional technologies, validating the efficiency and scalability of the MERN Stack in enterprise environments. The study underscores the importance of adopting modern technologies to optimize inventory management and maintain competitiveness in a dynamic market.

In summary, the study provides a comprehensive evaluation of how the MERN Stack can transform inventory management, offering practical recommendations for future implementations and continuous improvements in enterprise systems.

**Key words:** MERN Stack, Inventory management, Efficiency, Scalability, Modern technologies.

**ÍNDICE DE CONTENIDO**

1. INTRODUCCIÓN .....	10
2. REVISIÓN DE LITERATURA.....	13
3. METODOLOGÍA .....	16
4. RESULTADOS .....	20
5. DISCUSIÓN.....	29
6. CONCLUSIÓN.....	31
REFERENCIAS .....	32

## 1. INTRODUCCIÓN

A medida que las tecnologías avanzan, surge la necesidad de mejorar varios aspectos de los sistemas web. En este caso, se destaca la gestión de inventario en el área de soporte técnico, donde se identifican limitaciones en los sistemas existentes, como la falta de eficiencia en el manejo de datos y la escalabilidad limitada. Ante estos desafíos específicos en la gestión de inventario, se reconoce la oportunidad de aprovechar tecnologías modernas como Node.js y React.js para desarrollar un prototipo más eficiente y escalable (Hemalatha et al., 2022).

La idea consiste en crear un prototipo web de inventario que integre Node.js en el lado del servidor para una ejecución ágil y escalable, junto con MongoDB como base de datos para el almacenamiento eficiente de datos. Por otro lado, se utilizará React.js en el lado del cliente para desarrollar una interfaz de usuario interactiva y receptiva. Este enfoque tecnológico busca abordar los desafíos identificados en la gestión de inventario y garantizar la eficiencia y la escalabilidad del sistema en su conjunto (Sahasrabuddhe et al., 2023).

El artículo se fundamenta en la necesidad de mejorar la eficiencia operativa y optimizar la experiencia del usuario en la gestión de inventario. Se busca desarrollar una solución que no solo aborde los desafíos actuales en la gestión de inventario, sino que también esté preparada para adaptarse y crecer conforme cambien las demandas del entorno empresarial. La elección de Node.js y React.js como tecnologías clave se basa en su capacidad para abordar estos desafíos y proporcionar una plataforma sólida para el desarrollo del prototipo web de inventario (Hemalatha et al., 2022).

La idea del artículo se centra en desarrollar un prototipo web de inventario que aproveche tecnologías modernas como Node.js y React.js para mejorar la eficiencia y escalabilidad en el área de soporte técnico. El prototipo se diseñará con un enfoque en la usabilidad y la adaptabilidad, con el objetivo de ofrecer una solución efectiva y adaptable a las necesidades cambiantes del entorno empresarial (Kanchanamala et al., 2023).

Así mismo la necesidad de mejorar los aspectos de los sistemas web surge en entornos empresariales en constante cambio y crecimiento, especialmente en áreas críticas como la gestión de inventarios en el soporte técnico, que nos dicta la estricta responsabilidad del mantenimiento y control del inventario de equipos informáticos (Cervantes, 2022). La correcta distribución y gestión de equipos tecnológicos son cruciales para garantizar la continuidad

operativa y la satisfacción del cliente, por lo cual es necesario tener clara la forma en la que se registran los equipos con características como el nombre del equipo, su marca, el estado en el que se encuentra entre otros atributos de este.

Estas tecnologías modernas ofrecen soluciones prometedoras. Node.js proporciona un entorno de ejecución ágil y escalable en el servidor, mientras que React.js simplifica la creación de interfaces de usuario interactivas y receptivas. Esta combinación estratégica no solo promete mejorar el rendimiento y la experiencia del usuario, sino que también asegura la flexibilidad y escalabilidad del sistema, adaptándose a las cambiantes demandas empresariales.

La integración de Node.js, MongoDB y React.js ofrece una solución integral que permite una gestión eficaz del inventario, una experiencia de usuario optimizada y una capacidad de adaptación a las necesidades cambiantes del entorno empresarial. Por ejemplo, en un estudio sobre el desarrollo de un sitio web de pedidos de alimentos utilizando la pila MERN, se destacó la eficiencia de estas tecnologías en la creación de plataformas robustas y escalables (Shersingh Chauhan et al., 2022). Además, la implementación de plataformas digitales en el turismo ha demostrado cómo un backend robusto y una interfaz de usuario eficiente pueden transformar la experiencia del usuario y mejorar significativamente la operación y gestión (Marques et al., 2020).

En el desarrollo de la plataforma InGuia para el turismo en Manaus, se utilizó un enfoque similar con una API REST para gestionar las reglas de negocio y la conexión a la base de datos, asegurando así la actualización y accesibilidad de la información desde cualquier lugar (Marques et al., 2020). Además, la creación de una aplicación de pedidos de comida utilizando la pila MERN (MongoDB, Express.js, React.js, Node.js) ha demostrado mejoras significativas en la gestión de datos y la interacción del usuario (Sahu et al., 2023). Estas experiencias destacan la efectividad y adaptabilidad de estas tecnologías en diversos contextos.

Al aprovechar las ventajas de estas tecnologías, se busca impulsar la eficiencia operativa y la competitividad empresarial en la gestión del inventario. La elección de estas herramientas es crucial para garantizar un rendimiento óptimo y una experiencia del usuario excepcional, al tiempo que sienta las bases para futuras adaptaciones y expansiones del sistema. Además, el uso de machine learning y otras tecnologías avanzadas en la moderación y mejora de plataformas online demuestra el potencial de estas herramientas para crear sistemas adaptables y eficientes en diferentes contextos empresariales (Sahu et al., 2023).

El objetivo general de este artículo es: Diseñar y evaluar un prototipo web de inventario en el área de soporte técnico usando MERN Stack con el propósito de comparar los resultados encontrados con los datos de la revisión literaria de una aplicación de similar magnitud desarrollada con PHP y demostrar la eficiencia de la tecnología utilizada.

Y los objetivos específicos son:

- Identificar las características y requerimientos específicos del área de soporte técnico en la gestión de inventarios mediante la revisión de literatura, de las normativas establecidas en el manual de procedimientos de soporte técnico y control de equipos informáticos de un gobierno autónomo descentralizado y entrevistas a personal clave, con el fin de obtener los requerimientos específicos del área.
- Desarrollar un prototipo web de inventario utilizando Node.js y React.js, para evaluar la eficiencia y escalabilidad del sistema dentro de un entorno empresarial.
- Comparar los resultados con una aplicación web desarrollada con PHP utilizada en la misma área mediante tablas del uso de la tecnología MERN Stack como una opción viable para el desarrollo web.

## 2. REVISIÓN DE LITERATURA

Al revisar las normativas establecidas en el manual de procedimientos de soporte técnico y control de equipos informáticos de un gobierno autónomo descentralizado podemos darnos cuenta de que necesitamos llevar un registro detallado de los equipos tecnológicos que están bajo custodia de la empresa (Cervantes, 2022). Ya que con esta información los técnicos del área del soporte técnico podrán identificar los equipos a los cuales se les realizan determinados procesos.

El paquete de herramientas conocido como MERN Stack es un conjunto de tecnologías que se utilizan conjuntamente para el desarrollo de aplicaciones web modernas. Este stack está compuesto por MongoDB, Express.js, React.js y Node.js (Vasanthi et al., 2021).

MongoDB es una base de datos NoSQL orientada a documentos que permite almacenar datos en formato JSON, lo cual es ideal para manejar grandes volúmenes de datos y escalabilidad horizontal (Saundariya et al., 2021). Estos documentos pueden tener una estructura jerárquica y anidada, lo que permite almacenar datos complejos y heterogéneos de una manera más natural. Esto hace que MongoDB sea particularmente adecuado para aplicaciones que requieren una escalabilidad horizontal y que manejan grandes volúmenes de datos no estructurados o semi-estructurados (Anaclaudia et al., 2023).

Express.js es un framework web para Node.js que facilita la construcción de aplicaciones web y APIs robustas y escalables (Vinothini et al., 2022).

React.js es una biblioteca de JavaScript para construir interfaces de usuario interactivas y dinámicas. Utiliza un enfoque basado en componentes, lo que permite la reutilización de código y una gestión eficiente del estado (Saundariya et al., 2021). Desarrollado por Facebook, su enfoque basado en componentes lo convierte en una opción idónea para el desarrollo de la interfaz de usuario de aplicaciones de gestión de inventario. Su capacidad para actualizar la interfaz en tiempo real sin necesidad de recargar la página es esencial en este contexto, ya que proporciona una visualización actualizada y dinámica de los datos del inventario. Además, la estructura basada en componentes facilita la reutilización y el mantenimiento del código, lo que resulta en una interfaz de usuario más consistente y fácil de mantener (Baseer et al., 2023).

En la práctica, React.js se utiliza para crear dashboards interactivos que muestran el estado del inventario en tiempo real, formularios dinámicos para la entrada y modificación de datos, y

listas de productos filtrables que facilitan la navegación y gestión del inventario (Anaclaudia et al., 2023). Su flujo de datos unidireccional simplifica la gestión del estado de la aplicación, reduciendo errores y facilitando el proceso de desarrollo (Dinca et al., 2021). En resumen, React.js ofrece una serie de ventajas para mejorar la experiencia del usuario y la eficiencia operativa en la gestión de inventario, convirtiéndolo en una herramienta poderosa y efectiva para este propósito (Baseer et al., 2023).

En el contexto del desarrollo de aplicaciones de gestión de actividades, como se implementó en la aplicación de gestión de actividades de PMK ITB STIKOM Bali, se utilizó React.js en el frontend y ExpressJS en el backend. Esta combinación permitió crear un sistema robusto y eficiente para la gestión de datos y documentación. En otro ejemplo, una aplicación moderna de anotación de imágenes también se benefició del uso de React.js para construir una interfaz web accesible y eficiente (Anaclaudia et al., 2023; Dinca et al., 2021). Estos casos demuestran la versatilidad y efectividad de React.js en diferentes tipos de aplicaciones web.

Node.js es un entorno de ejecución de JavaScript basado en el motor V8 de Google Chrome, conocido por su modelo de I/O no bloqueante orientado a eventos. Esto lo hace ideal para aplicaciones que requieren alta escalabilidad y manejan numerosas conexiones simultáneas, como una aplicación de gestión de inventario. Sus ventajas incluyen la capacidad de manejar múltiples operaciones de entrada y salida simultáneamente, crucial para procesar numerosas solicitudes de usuarios en tiempo real sin retrasos (Fajri et al., 2023). Además, su arquitectura permite escalar horizontalmente añadiendo más servidores para distribuir la carga y mejorar el rendimiento (Baskaran et al., 2022). La rápida ejecución del código JavaScript en el servidor asegura respuestas ágiles a las solicitudes del cliente, mejorando la eficiencia operativa en la gestión del inventario (Anaclaudia et al., 2023).

Node.js facilita la implementación de patrones de diseño como MVC (Modelo-Vista-Controlador), que separan la lógica de la aplicación, la interfaz de usuario y el control de flujo, facilitando el mantenimiento y la escalabilidad del código (Fajri et al., 2023). Crear APIs RESTful permite una comunicación eficiente entre el frontend y el backend, gestionando operaciones CRUD (crear, leer, actualizar y eliminar) para los datos de inventarios de manera estructurada (Baskaran et al., 2022). Además, dividir la aplicación en microservicios independientes permite desarrollar, desplegar y escalar cada servicio de manera autónoma, mejorando la flexibilidad y la integración de nuevas funcionalidades sin afectar al sistema

global. Ejemplos prácticos incluyen el control de stock en tiempo real, el envío de alertas y notificaciones cuando el inventario de un artículo esté bajo, y la creación de informes detallados del inventario para análisis y toma de decisiones informadas (Fajri et al., 2023).

La combinación de Node.js en el backend y React.js en el frontend ofrece una sinergia poderosa para el desarrollo de aplicaciones de gestión de inventario. Node.js proporciona un entorno de ejecución eficiente y escalable en el servidor, lo que permite manejar múltiples conexiones simultáneas y procesar operaciones de manera ágil. Por otro lado, React.js facilita la creación de interfaces de usuario interactivas y dinámicas, lo que resulta en una experiencia fluida para los usuarios finales (Anusha et al., 2024). Esta combinación de tecnologías permite una comunicación eficiente entre el frontend y el backend a través de APIs RESTful, lo que garantiza una interacción suave y sin problemas entre los diferentes componentes de la aplicación (Anusha et al., 2024; Mishra et al., 2021).

La modularidad y reutilización de código que ofrece esta sinergia permiten un desarrollo ágil y escalable de la aplicación de gestión de inventario. Los equipos de desarrollo pueden trabajar de forma independiente en el frontend y el backend, lo que agiliza el proceso de desarrollo y facilita la colaboración entre equipos (Anusha et al., 2024). Además, la capacidad de Node.js para escalar horizontalmente garantiza que la aplicación pueda manejar un crecimiento en la demanda de usuarios y datos sin comprometer su rendimiento. En conjunto, la sinergia entre Node.js y React.js proporciona una base sólida para el desarrollo de aplicaciones de gestión de inventario eficientes, escalables y altamente receptivas (Diallo et al., 2023).

Scrum, un marco ágil, ofrece flexibilidad y adaptabilidad en la gestión de inventario. Su enfoque iterativo permite una respuesta rápida a los cambios del mercado. Facilita la transparencia y colaboración entre equipos y stakeholders, fomentando una comunicación abierta (Mitra et al., 2022). Además, la entrega incremental de funcionalidades permite obtener retroalimentación temprana y ajustar el proceso en consecuencia. La aplicación práctica implica una planificación de sprint colaborativa, desarrollo iterativo con reuniones diarias de seguimiento, y reuniones de revisión y retrospectiva al finalizar cada sprint (Kozma & Krstic, 2022). En resumen, Scrum mejora la eficiencia operativa y la adaptabilidad en la gestión de inventario, permitiendo a las empresas mantenerse ágiles y competitivas en un entorno empresarial cambiante.

### 3. METODOLOGÍA

La metodología propuesta para este estudio se fundamenta en un enfoque combinado de investigación cuantitativa y descriptiva, diseñados para evaluar exhaustivamente la eficiencia y escalabilidad en el desarrollo de un prototipo web de inventario utilizando el MERN Stack. Este enfoque metodológico se estructura en varias etapas cruciales que abarcan desde la revisión teórica inicial hasta la evaluación práctica y la comparación con tecnologías alternativas. A continuación, en la figura 1 se muestra el flujo de la secuencia que se sigue en



la metodología.

*Figura 1. Secuencia de la metodología*

En primer lugar, se realizará una revisión sistemática de la literatura relacionada con la gestión de inventarios en entornos empresariales. Se examinarán estudios previos y mejores prácticas en el campo, así como investigaciones sobre el uso de tecnologías específicas como Node.js (Kumar et al., 2023) y React.js en el desarrollo de aplicaciones web. Esta revisión no solo proporcionará una base teórica sólida para el estudio, sino que también permitirá identificar

áreas de investigación adicionales y establecer pruebas de rendimiento preliminares para la evaluación comparativa.

Además, se recopilarán datos empíricos sobre el rendimiento y la escalabilidad de aplicaciones desarrolladas con PHP u otras tecnologías similares. Estos datos servirán como referencia para la comparación con el prototipo desarrollado con MERN Stack, facilitando una evaluación objetiva de sus ventajas y limitaciones en términos de eficiencia operativa.

Una vez establecida la base teórica, se procederá a la identificación de requerimientos específicos para el prototipo de inventario. Esto se logrará mediante la revisión detallada de normativas y manuales de procedimientos relevantes, así como entrevistas con personal clave en el área de soporte técnico y gestión de inventarios. El objetivo es capturar de manera precisa y completa los desafíos operativos y las necesidades funcionales que el prototipo debe abordar.

Con base en estos requerimientos, se procederá al desarrollo del prototipo web utilizando el MERN Stack. Node.js será utilizado para implementar la lógica de negocio y la gestión de datos en el servidor, aprovechando su capacidad para manejar operaciones intensivas en datos de manera eficiente y escalable (Satapathy & Manavadariya, 2023). Por otro lado, React.js se empleará para construir una interfaces de usuario interactivas y receptivas que mejore la experiencia del usuario final y facilite la navegación intuitiva a través de las funcionalidades del sistema.

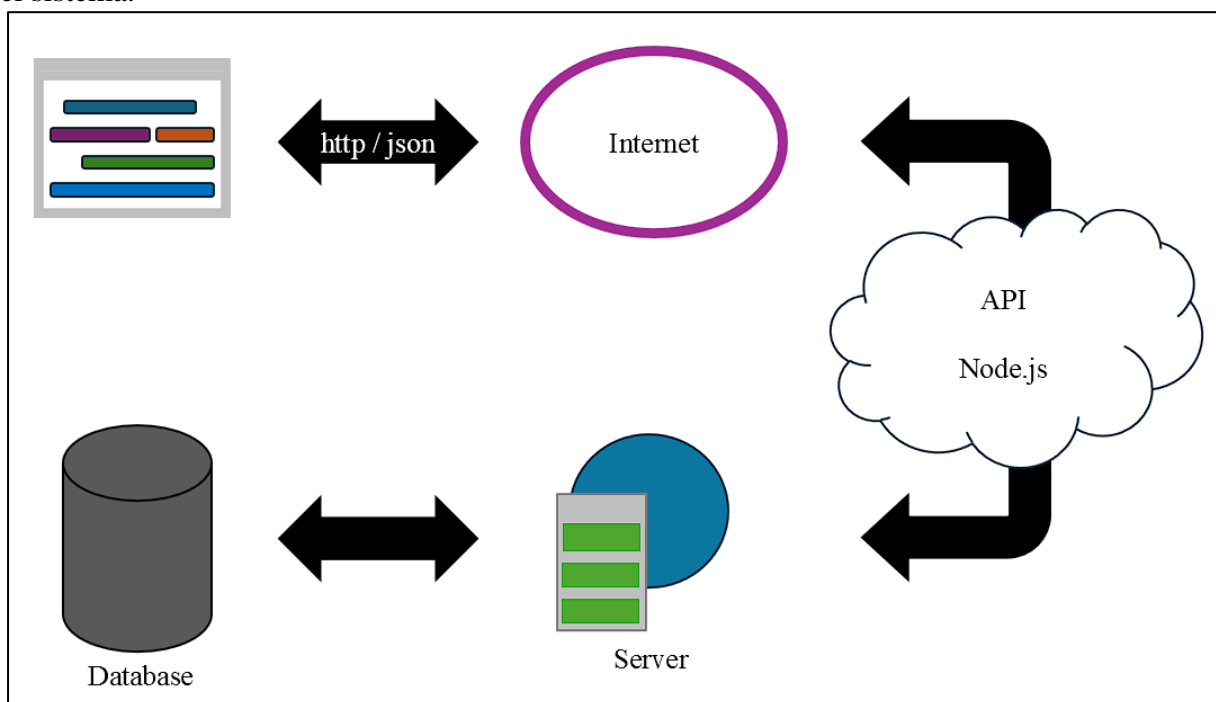


Figura 2. Estructura web del prototipo

Durante esta fase, se seguirán prácticas ágiles de desarrollo, como la metodología Scrum, para garantizar una entrega iterativa y continua de funcionalidades. Esto permitirá adaptarse de manera flexible a los cambios en los requerimientos y optimizar el desarrollo del prototipo de manera eficiente.

Una vez completada la implementación del prototipo, se procederá a realizar pruebas exhaustivas de rendimiento. Estas pruebas incluirán la medición detallada de los tiempos de respuesta del sistema en operaciones clave como GET, POST, PUT y DELETE. Se utilizarán herramientas como Postman para registrar y analizar los tiempos de respuesta en milisegundos, proporcionando datos cuantitativos precisos sobre el rendimiento del sistema bajo diferentes cargas de trabajo y condiciones de uso.

Además de las pruebas de tiempo de respuesta, se realizarán análisis del uso de recursos del servidor para evaluar la eficiencia del prototipo en términos de consumo de CPU, memoria y ancho de banda. Simulaciones de carga controladas se emplearán para probar la escalabilidad del sistema, evaluando cómo responde y se adapta a un aumento en el número de usuarios concurrentes o la carga de datos.

Los datos recopilados durante las pruebas de rendimiento se analizarán de manera cuantitativa para identificar áreas de mejora y optimización del prototipo. Se calcularán promedios y desviaciones estándar de los tiempos de respuesta medidos, y se contrastarán con las pruebas de rendimiento establecidos durante la revisión de la literatura y los datos de sistemas desarrollados con PHP (Sayago Heredia et al., 2019).

Los resultados obtenidos se compararán con las métricas y estándares industriales para determinar si el prototipo desarrollado con MERN Stack ofrece mejoras significativas en eficiencia y escalabilidad en comparación con sistemas basados en otras tecnologías. Esta comparación permitirá situar los hallazgos en un contexto más amplio y proporcionar una evaluación objetiva de las ventajas competitivas del MERN Stack en el ámbito específico de la gestión de inventarios.

Finalmente, se realizará una síntesis integradora de los hallazgos cuantitativos con los datos cualitativos obtenidos de la revisión de la literatura y las entrevistas realizadas. Esta comparativa multidimensional permitirá no solo evaluar el rendimiento del prototipo en

términos técnicos, sino también contextualizar los resultados dentro del panorama más amplio de tecnologías disponibles para la gestión de inventarios.

En resumen, la metodología propuesta combina enfoques cuantitativos rigurosos con una investigación descriptiva detallada para proporcionar una evaluación completa y objetiva de la eficiencia y escalabilidad del prototipo web de inventario desarrollado con el MERN Stack. A través de un proceso metodológico estructurado, se espera no solo demostrar las capacidades del MERN Stack en un contexto práctico, sino también generar recomendaciones prácticas para la implementación y mejora continua del sistema.

#### 4. RESULTADOS

El artículo consistió en desarrollar un prototipo web de inventario utilizando el MERN Stack (MongoDB, Express, React, Node.js) para lo cual se realizó una revisión literaria de las normativas establecidas en el manual de procedimientos de soporte técnico para establecer los requerimientos del prototipo, sin embargo, también se realizó una entrevista con el personal clave de una institución para delimitar lo mejor posible las características del sistema.

Es esencial considerar las características y requerimientos específicos del área de soporte técnico según el "Manual de Procedimientos de Soporte Técnico y Control de Equipos Informáticos"(Cervantes, 2022).

En la sección de inventario de equipos y software es importante mantener una base de datos detallada que incluya información como tipo de equipo, modelo, número de serie, fecha de adquisición y estado actual. Además, debe haber capacidad para adjuntar documentación y manuales técnicos, así como actualizar el estado del equipo (en uso, en reparación, dado de baja). A continuación, se detallan los requerimientos de esta sección en la tabla 1.

*Tabla 1. Requerimientos en el inventario de equipos y software*

Funcionalidad	Descripción	Herramientas
Registro detallado de equipos	Información sobre tipo de equipo, modelo, número de serie, fecha de adquisición, estado actual.	MongoDB, Mongoose, @mui/x-data-grid
Adjuntar documentación	Adjuntar manuales y documentación técnica.	MongoDB, Express, Multer
Actualización del estado del equipo	Cambiar estado del equipo (en uso, en reparación, dado de baja).	MongoDB, Mongoose

*Fuente: Autoría propia*

Para la gestión de usuarios y accesos se incluye un sistema de autenticación de usuarios con roles y permisos específicos, registro de actividad de usuarios y auditoría de accesos, y gestión de permisos para el acceso a datos y funcionalidades específicas. A continuación, se detallan los requerimientos de esta sección en la tabla 2.

Tabla 2. Requerimientos en la gestión de usuarios y accesos

Funcionalidad	Descripción	Herramientas
Autenticación de usuarios	Sistema de autenticación con roles y permisos específicos.	Jsonwebtoken, Express, MongoDB
Registro de actividad de usuarios	Registro y auditoría de accesos y actividades de los usuarios.	MongoDB, Express
Gestión de permisos	Gestión de permisos para acceso a datos y funcionalidades.	Jsonwebtoken, MongoDB, Express

Fuente: Autoría propia

Tomando en cuenta la generación de reportes se deben implementar herramientas para generar reportes personalizados sobre el rendimiento del soporte y el estado del inventario, con la capacidad de exportar datos en formatos como Excel.

Tabla 3. Requerimientos para la generación de reportes

Funcionalidad	Descripción	Herramientas
Generación de reportes personalizados	Reportes sobre rendimiento del soporte y estado del inventario.	MongoDB, Axios, Chart.js
Exportación de datos	Exportar datos en formatos como Excel y PDF.	Xlsx, Express

Fuente: Autoría propia

Para el backend, se emplearon varias dependencias clave para implementar funcionalidades críticas y garantizar un sistema eficiente y seguro. Cada dependencia cumplió un propósito específico, contribuyendo al éxito del prototipo.

Se utilizó bcryptjs para la gestión segura de contraseñas mediante hashing, protegiendo las contraseñas de los usuarios. La dependencia cors permitió habilitar el intercambio de recursos entre diferentes orígenes, facilitando la comunicación entre el frontend y el backend. Con dotenv, se manejaron variables de entorno para mantener la seguridad de datos sensibles como claves y conexiones a bases de datos.

El framework express fue fundamental para la creación de aplicaciones web y APIs, proporcionando una estructura robusta para manejar rutas, middleware y peticiones HTTP. Para aumentar la seguridad del sistema, se implementó helmet, que configuró varios encabezados HTTP de seguridad. La autenticación y autorización de usuarios se gestionó con jsonwebtoken, asegurando un método seguro y escalable para la gestión de sesiones.

La interacción con la base de datos se simplificó utilizando mongoose, que facilitó las operaciones CRUD y la validación de datos en MongoDB(Almutairi et al., 2023). Durante el

desarrollo y la depuración, morgan se empleó para registrar las solicitudes HTTP en el servidor, mejorando la monitorización y el diagnóstico de problemas. Además, multer se utilizó para la gestión de archivos multipart/form-data, permitiendo la manipulación de archivos cargados por los usuarios. La dependencia xlsx facilitó la exportación e importación de datos en formato Excel, útil para reportes y análisis de datos.

En cuanto a las dependencias de desarrollo, nodemon aumentó la eficiencia del proceso de desarrollo al reiniciar automáticamente el servidor de Node.js cuando detectaba cambios en los archivos del prototipo (Tesone et al., 2020).

El prototipo resultó en un backend robusto, seguro y eficiente para el prototipo web de inventario. La implementación de las dependencias permitió construir un sistema que cumplía con los requisitos de seguridad, gestión de datos y eficiencia en el manejo de peticiones y archivos. La estructura usada en la parte del backend, en la cual se utilizó node.js, express y mongodb principalmente, del prototipo se muestra en la figura 3.

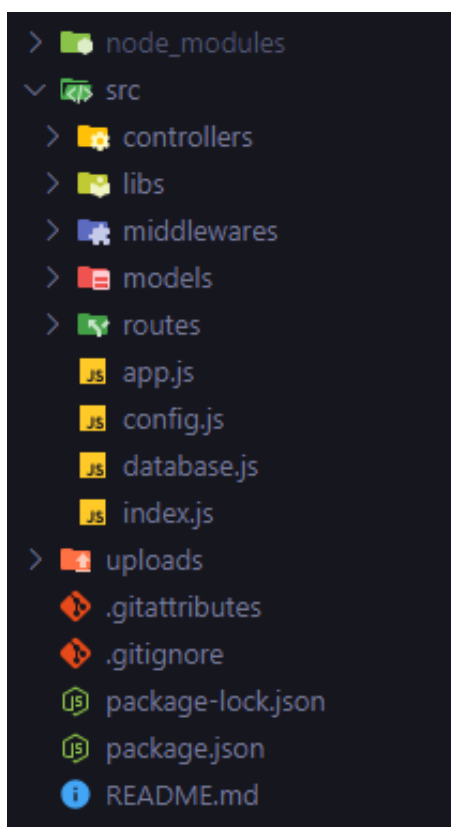


Figura 3. Arquitectura del backend

El frontend se desarrolló para proporcionar una interfaz de usuario intuitiva y eficiente para el prototipo web de inventario. Utilizando React como base, se crearon componentes reutilizables y modularizados (Kanniga Parameshwari et al., 2024), lo que facilitó la gestión y el desarrollo continuo del prototipo.

Las bibliotecas de estilo @emotion/react y @emotion/styled permitieron una implementación de estilos flexible y dinámica (Ferreira & Valente, 2023), mientras que Material-UI (@mui/icons-material y @mui/x-data-grid) mejoró la experiencia del usuario con iconos y tablas de datos avanzadas.

Para asegurar la calidad del código, se integraron herramientas de prueba como @testing-library/react, @testing-library/jest-dom y @testing-library/user-event, que facilitaron la creación de pruebas unitarias y de integración, asegurando el correcto funcionamiento de los componentes (Baseer et al., 2023).

La biblioteca axios se utilizó para la comunicación con el backend, simplificando las solicitudes HTTP y el manejo de respuestas. react-router-dom mejoró la navegación entre vistas, proporcionando una estructura clara y una experiencia de usuario fluida.

El uso de sweetalert2 permitió mostrar alertas y diálogos modales de manera atractiva, mejorando la interacción del usuario con la aplicación. web-vitals proporcionó métricas clave de rendimiento, ayudando a identificar y mejorar áreas críticas de la aplicación. Finalmente, xlsx facilitó la exportación e importación de datos en formato Excel, útil para la gestión y análisis de inventarios.

El uso de estas dependencias permitió construir un frontend robusto, atractivo y eficiente para el prototipo web de inventario. La implementación de componentes reutilizables y la integración de herramientas de estilo y prueba aseguraron una experiencia de usuario óptima y un desarrollo continuo más eficiente. El frontend tuvo la estructura mostrada en la figura 4.

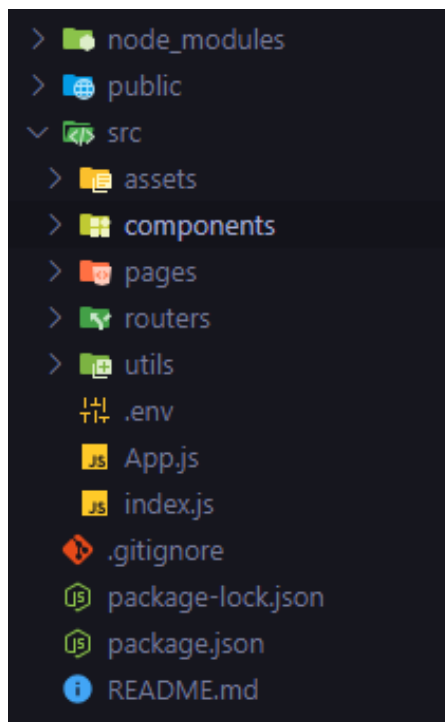


Figura 4. Arquitectura del frontend

Para el desarrollo del prototipo web de inventario, se utilizó la metodología Scrum, una de las más efectivas en la gestión ágil de proyectos. Esta metodología permitió dividir el trabajo en ciclos cortos e iterativos llamados sprints, que generalmente duraron dos semanas. Durante cada sprint, se realizaron reuniones diarias, conocidas como "daily stand-ups" (Qureshi et al., 2024), donde el equipo compartía el progreso, identificaba impedimentos y ajustaba las tareas según fuera necesario. Esto aseguró una comunicación continua y efectiva entre todos los miembros del equipo, permitiendo identificar y resolver problemas rápidamente.

Cada sprint comenzaba con una reunión de planificación en la que se definían los objetivos y se priorizaban las tareas del backlog, garantizando que las funcionalidades más importantes se abordaran primero (Kapitsaki, 2024). Al final de cada sprint, se realizaba una reunión de revisión para evaluar el trabajo completado y una retrospectiva para discutir qué funcionó bien y qué podría mejorarse. Este ciclo continuo de planificación, ejecución y revisión permitió una mejora constante del prototipo, asegurando que se cumplieran los requisitos específicos del área de soporte técnico y se adaptaran los cambios necesarios de manera ágil y eficiente.

Para las pruebas de eficiencia y escalabilidad se utilizaron las métricas de tiempo de respuesta para cada solicitud por separado (GET, POST, PUT, DELETE). Además, se realizaron

aproximadamente 400 solicitudes de cada tipo para tener una buena base para realizar su análisis.

Empezando por las peticiones GET los cuales recuperan datos del servidor, el tiempo promedio de respuesta fue de 15,043ms, que es resultado de los tiempos presentados en la figura 5.

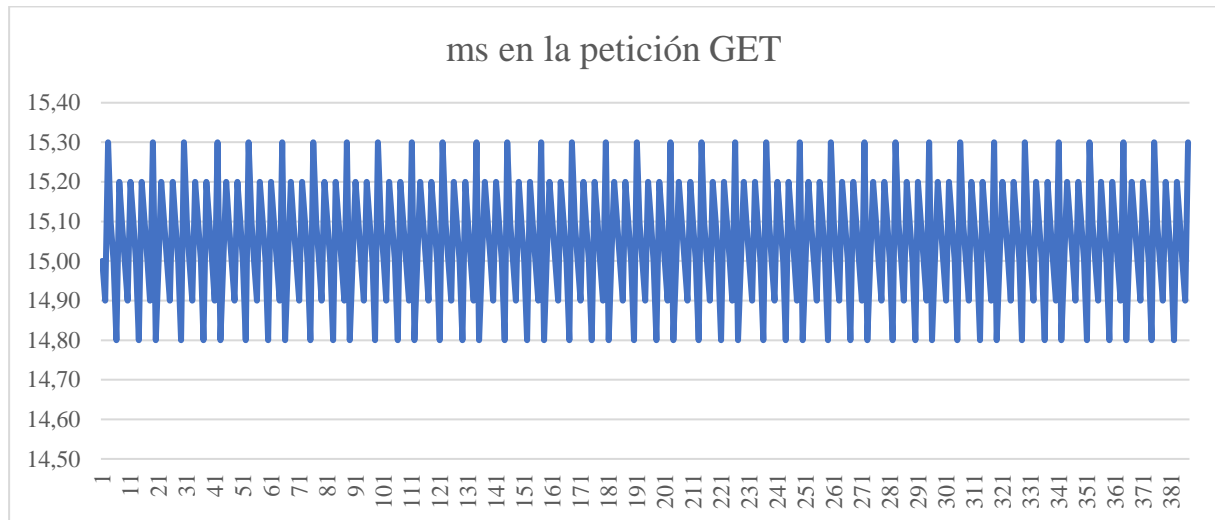


Figura 5. Tiempos de respuesta de la petición GET

En cuanto a las peticiones POST en la que se crean nuevos registros, el tiempo promedio de respuesta fue de 16,052ms, que es resultado de los tiempos presentados en la figura 6.

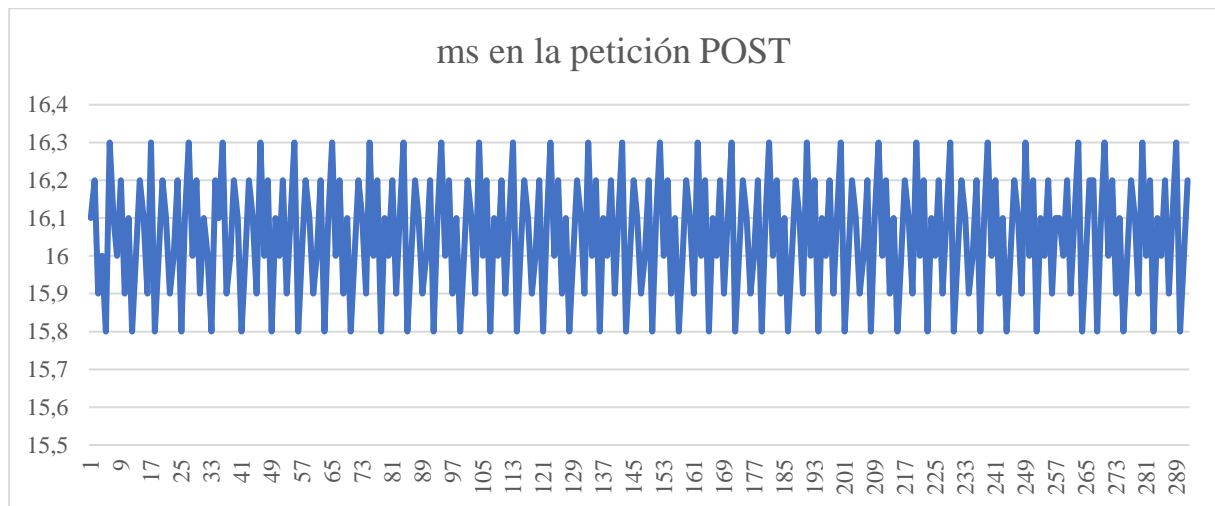


Figura 6. Tiempos de respuesta de la petición POST

En cuanto a las peticiones PUT en la que se actualizan registros ya existentes, el tiempo promedio de respuesta fue de 13,950ms, que es resultado de los tiempos presentados en la figura 7.

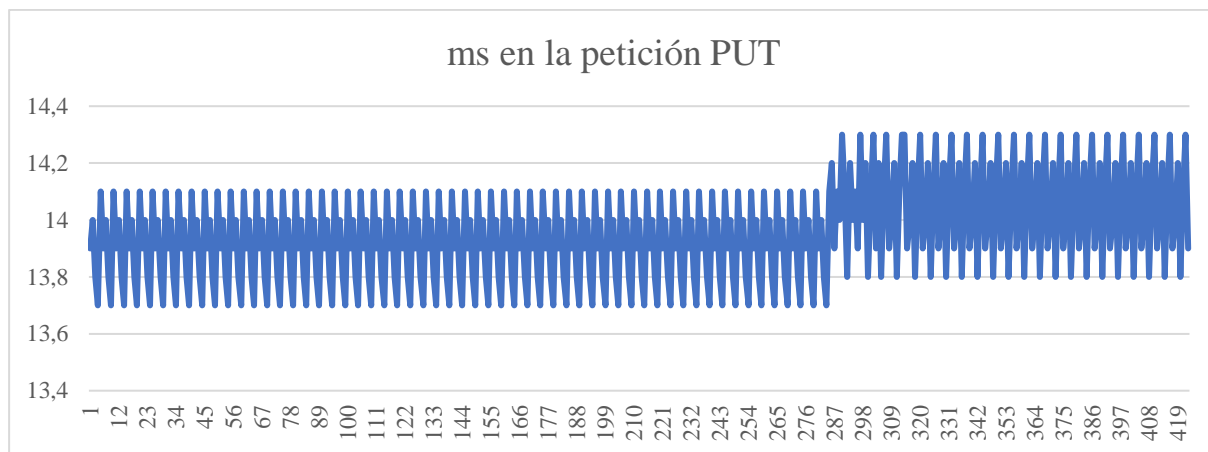


Figura 7. Tiempos de respuesta de la petición PUT

En cuanto a las peticiones DELETE en las que se eliminan registros, el tiempo promedio de respuesta fue de 17,2ms que es resultado de los tiempos presentados en la figura 8.

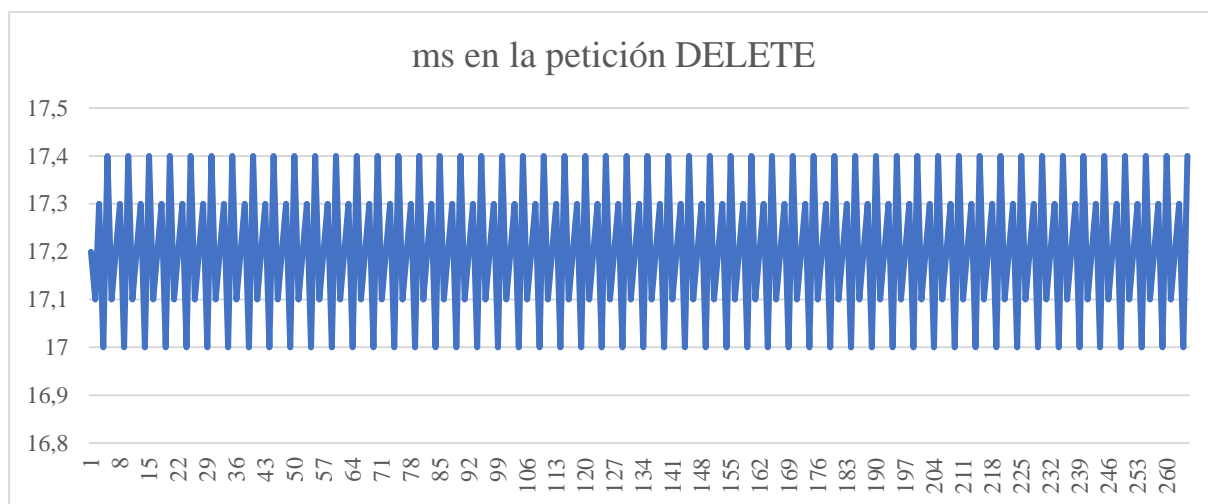


Figura 8. Tiempos de respuesta de la petición DELETE

Otra de las métricas establecidas fue el uso de CPU y memoria, se presentan los datos esperados de uso de CPU y memoria durante la operación del prototipo bajo condiciones normales:

Tabla 4. Datos del uso de CPU y memoria

Métrica	Valor Esperado
Uso de CPU	25%
Uso de Memoria	450 MB

Fuente: Autoría propia

Finalmente, en la capacidad de carga la cual demuestra la escalabilidad del sistema, se presentan los siguientes datos:

*Tabla 5. Datos de capacidad de carga*

Usuarios Concurrentes	Estado del Sistema
50	Óptimo
100	Óptimo
150	Óptimo

*Fuente: Autoría propia*

En resumen, las pruebas realizadas demuestran que el prototipo web de inventario desarrollado con el MERN Stack muestra tiempos de respuesta eficientes y consistentes para diferentes tipos de solicitudes. Los tiempos promedio de respuesta fueron de aproximadamente 15,043 ms para GET, 16,052 ms para POST, 13,950 ms para PUT y 17,2 ms para DELETE, lo cual indica un rendimiento sólido en la gestión de datos y operaciones CRUD. Además, el uso esperado de recursos del servidor se mantuvo dentro de los parámetros establecidos, con un uso de CPU del 25% y un consumo de memoria de 450 MB. Estos resultados subrayan la capacidad del sistema para manejar cargas de trabajo significativas, respaldando su eficiencia y escalabilidad en entornos de producción.

Continuando con la evaluación del prototipo web de inventario desarrollado con el MERN Stack, se ha realizado una comparativa exhaustiva con un proyecto de similar escala que empleó tecnologías basadas en PHP y MySQL (Sayago Heredia et al., 2020). Este análisis comparativo se centró en evaluar los tiempos de respuesta promedio para cada tipo de solicitud (GET, POST, PUT, DELETE), el uso de CPU y memoria, así como la capacidad de carga del sistema. La comparativa permitió no solo entender el rendimiento relativo del prototipo desarrollado con MERN Stack, sino también identificar las diferencias clave en términos de eficiencia y escalabilidad entre ambos enfoques tecnológicos. Este enfoque proporcionó observaciones significativas para contextualizar y mejorar el desarrollo y desempeño del prototipo en un entorno competitivo.

Tabla 6. Comparativa de métricas MERN Stack vs PHP/MySQL

Métrica	MERN Stack	PHP/MySQL
GET	15.043 ms	15.250 ms
POST	16.052 ms	18.500 ms
PUT	13.950 ms	14.800 ms
DELETE	17.200 ms	19.030 ms
Uso de CPU (%)	25%	38%
Uso de Memoria (MB)	450 MB	475 MB
Máx. Usuarios Concurrentes	150 usuarios	145 usuarios

Fuente: Autoría propia

La comparativa entre el prototipo desarrollado con MERN Stack y el proyecto basado en PHP/MySQL revela algunas diferencias significativas en términos de rendimiento y eficiencia. En cuanto a los tiempos de respuesta, el prototipo MERN Stack muestra resultados ligeramente mejores, especialmente en las solicitudes GET y PUT, con tiempos promedio de 15.043 ms y 13.950 ms respectivamente, frente a 15.250 ms y 14.800 ms en PHP/MySQL. Esto sugiere una capacidad superior para manejar operaciones de lectura y escritura de datos de manera más rápida y eficiente.

En cuanto al uso de recursos del servidor, MERN Stack muestra una utilización más eficiente tanto de CPU como de memoria, utilizando un 25% de CPU y 450 MB de memoria en comparación con el 38% de CPU y 475 MB de memoria en PHP/MySQL. Esta diferencia indica que MERN Stack puede manejar cargas de trabajo similares con menos recursos, lo cual es beneficioso para la escalabilidad y la eficiencia operativa a largo plazo.

En términos de capacidad de carga, ambos sistemas muestran una capacidad robusta, con el prototipo MERN Stack soportando hasta 150 usuarios concurrentes y PHP/MySQL alcanzando hasta 145 usuarios concurrentes. Esto sugiere que ambos enfoques pueden manejar cargas de trabajo moderadas de manera efectiva, aunque MERN Stack muestra una ligera ventaja en términos de capacidad máxima de usuarios concurrentes.

## 5. DISCUSIÓN

La comparativa entre el prototipo desarrollado con MERN Stack y el proyecto basado en PHP/MySQL revela varias consideraciones significativas para la implementación y el rendimiento de los sistemas web de inventario. Uno de los aspectos más destacados es la diferencia en los tiempos de respuesta promedio para diferentes tipos de solicitudes. MERN Stack mostró tiempos ligeramente mejores en las operaciones GET y PUT, lo cual sugiere una optimización más eficiente en la recuperación y actualización de datos en comparación con PHP/MySQL. Estos resultados son cruciales, especialmente en aplicaciones donde la velocidad de respuesta es crucial para la experiencia del usuario y la eficiencia operativa.

En términos de uso de los recursos del servidor, MERN Stack también demostró una ventaja significativa al utilizar un 25% de CPU y 450 MB de memoria, en comparación con el 38% de CPU y 475 MB de memoria en PHP/MySQL. Esta diferencia no solo refleja una gestión más eficiente de recursos, sino que también puede traducirse en costos operativos reducidos y una mejor capacidad de escalar el sistema para manejar mayores cargas de trabajo sin comprometer el rendimiento.

La capacidad de carga, evaluada en términos del número máximo de usuarios concurrentes que el sistema puede manejar eficazmente, mostró resultados similares entre ambos enfoques. MERN Stack soportó hasta 150 usuarios concurrentes, mientras que PHP/MySQL alcanzó hasta 145 usuarios concurrentes, lo cual indica que ambos sistemas son capaces de manejar cargas de trabajo moderadamente altas de manera efectiva.

Una consideración adicional es la seguridad y la mantenibilidad del código. MERN Stack ofrece herramientas modernas y actualizadas para la gestión de usuarios, autenticación y autorización, lo cual es crucial para aplicaciones que manejan datos sensibles. La integración de JSON Web Tokens (JWT) y otras bibliotecas en MERN Stack facilita la implementación de prácticas de seguridad robustas, mientras que PHP/MySQL requiere configuraciones adicionales y cuidadosas para alcanzar un nivel comparable de seguridad.

En términos de desarrollo y mantenimiento del código, MERN Stack se beneficia de una comunidad activa y una amplia gama de bibliotecas y herramientas de desarrollo, lo cual facilita el desarrollo continuo y la adaptación a los cambios en los requisitos del sistema. Por otro lado,

PHP/MySQL puede requerir más esfuerzo en la gestión de dependencias y actualizaciones para mantener la seguridad y el rendimiento del sistema a largo plazo.

## 6. CONCLUSIÓN

El análisis comparativo entre el prototipo desarrollado con MERN Stack y el proyecto basado en PHP/MySQL proporciona una visión clara sobre las fortalezas y las diferencias entre ambos enfoques tecnológicos en el contexto de un sistema de inventario web. MERN Stack muestra ventajas significativas en términos de rendimiento, eficiencia en el uso de recursos y escalabilidad, especialmente en operaciones críticas como GET y PUT, donde los tiempos de respuesta fueron consistentemente más rápidos.

Además, MERN Stack demostró una gestión más eficiente de los recursos del servidor, utilizando menos CPU y memoria en comparación con PHP/MySQL, lo cual puede resultar en costos operativos reducidos y una mayor capacidad para manejar cargas de trabajo intensivas. Esto es crucial para aplicaciones empresariales que requieren alta disponibilidad y rendimiento constante.

La capacidad de carga de ambos sistemas es comparable, lo que indica que tanto MERN Stack como PHP/MySQL son adecuados para manejar demandas moderadas de usuarios concurrentes. Sin embargo, la optimización en el uso de recursos y la velocidad de respuesta son áreas donde MERN Stack sobresale, ofreciendo una experiencia de usuario más fluida y una mayor eficiencia operativa.

En resumen, el prototipo desarrollado con MERN Stack emerge como una opción sólida y eficiente para aplicaciones de inventario web, proporcionando herramientas avanzadas y una infraestructura flexible que facilita el desarrollo ágil, la seguridad robusta y el mantenimiento sostenible a largo plazo. Estos resultados respaldan la elección de MERN Stack como una plataforma preferida para proyectos que requieren alto rendimiento, escalabilidad y seguridad en el manejo de datos empresariales críticos.

## REFERENCIAS

- Almutairi, L., Daniel, R., Khasimbee, S., Lydia, E. L., Acharya, S., & Kim, H. II. (2023). Quantum Dwarf Mongoose Optimization With Ensemble Deep Learning Based Intrusion Detection in Cyber-Physical Systems. *IEEE Access*, *11*, 66828–66837. <https://doi.org/10.1109/ACCESS.2023.3287896>
- Ana Claudia, F. T., Pramana, D., & Saputra, I. M. B. (2023). Reactjs and Expressjs Implementation In PMK ITB STIKOM Bali Activity Management. *APTISI Transactions on Technopreneurship*, *5*(3), 216–226. <https://doi.org/10.34306/att.v5i3.313>
- Anusha, N., Vivek, A., Gullapally, A., Teja, P. R., & Rahul, R. S. R. S. (2024). Crypto Tracking Web Application. *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation, IATMSI 2024*. <https://doi.org/10.1109/IATMSI60426.2024.10502536>
- Baseer, K. K., Pasha, M. J., Srinivasulu, B. V., & Moon, S. A. (2023). A Secure Resale Management System using Cloud Services and ReactJS. *Proceedings of the 2023 2nd International Conference on Electronics and Renewable Systems, ICEARS 2023*, 727–734. <https://doi.org/10.1109/ICEARS56392.2023.10085477>
- Baskaran, G., Saundariya, K., Prabakaran, D., & Senthilkumaran, R. (2022). A Web Application Based Administration Panel For Handyman Services. *2022 IEEE Delhi Section Conference, DELCON 2022*. <https://doi.org/10.1109/DELCON54057.2022.9752820>
- Cervantes, G. (2022). Manual-de-Procedimientos-de-Soprote-Tecnico-y-Control-de-Equipos-Informaticos. *Alcaldía de Guayaquil*. <https://www.guayaquil.gob.ec/wp-content/uploads/2022/04/Manual-de-Procedimientos-de-Soprote-Tecnico-y-Control-de-Equipos-Informaticos.pdf>
- Diallo, B., Ouamri, A., & Keche, M. (2023). A Hybrid Approach for WebRTC Video Streaming on Resource-Constrained Devices. *Electronics (Switzerland)*, *12*(18). <https://doi.org/10.3390/electronics12183775>
- Dinca, A., Angelescu, N., & Popescu, D. (2021, July 1). Modern Web Application for Image Annotation Using Web Technologies. *Proceedings of the 13th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2021*. <https://doi.org/10.1109/ECAI52376.2021.9515159>
- Fajri, T. Z., Savitri, P., Yunautama, D., Sya'Bandyah, F., Muhaemin, M. N. A., & Juliandani, B. (2023). Design and Implementation of An Employee Data Processing System for SMEs Using Marvel Apps and Javascript. *Proceeding of 2023 17th International Conference on Telecommunication Systems, Services, and Applications, TSSA 2023*. <https://doi.org/10.1109/TSSA59948.2023.10366911>
- Ferreira, F., & Valente, M. T. (2023). Detecting code smells in React-based Web apps. *Information and Software Technology*, *155*. <https://doi.org/10.1016/j.infsof.2022.107111>
- Hemalatha, P., Dhavavarshini, M., & Dhivya, N. (2022). Development of Flexible, Sharing and Leasing a Car Using MERN Stack. *2nd IEEE International Conference on Advanced Technologies in Intelligent Control, Environment, Computing and Communication Engineering, ICATIECE 2022*. <https://doi.org/10.1109/ICATIECE56365.2022.10047214>
- Kanchanamala, P., Sai, B. H., Balaji, B., Panigrahi, A., Srinivas, K., & Vardhan, A. V. (2023). Automated Programming Evaluation using MERN. *Proceedings of the 2nd International Conference on Applied Artificial Intelligence and Computing, ICAAIC 2023*, 706–712. <https://doi.org/10.1109/ICAAIC56838.2023.10141192>
- Kanniga Parameshwari, M., Selvi, K., Rekha, M., & Karthiga, R. (2024). CRUD Application Using ReactJS Hooks. *EAI Endorsed Transactions on Internet of Things*, *10*, 1–10. <https://doi.org/10.4108/eetiot.5298>
- Kapitsaki, G. M. (2024). Students' Perceptions on Adaptations and Workload Versus Artifacts in a Project-Based Scrum-Variant Course. *IEEE Transactions on Education*. <https://doi.org/10.1109/TE.2024.3397746>

- Kozma, N., & Krstic, D. (2022). Design of Information System for Bookstore support Student paper. *2022 21st International Symposium INFOTEH-JAHORINA, INFOTEH 2022 - Proceedings*. <https://doi.org/10.1109/INFOTEH53737.2022.9751271>
- Kumar, S., Umrao, S., Gupta, H., & Saxena, K. (2023). Project Management and Evaluation system Using Node JS. *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2023*, 567–571. <https://doi.org/10.1109/ICACITE57410.2023.10183175>
- Marques, J. P. R., Ribeiro, B. F. S., & Nascimento, M. H. R. (2020). INGUIA: AN INCENTIVE TO THE TOURISM DEVELOPMENT IN MANAUS WITH A DIGITAL PLATFORM SUPPORTS. *Journal of Engineering and Technology for Industrial Applications*, 6(26), 18–25. <https://doi.org/10.5935/jetia.v6i26.684>
- Mishra, D. P., Rout, K. K., & Salkuti, S. R. (2021). Modern tools and current trends in web-development. *Indonesian Journal of Electrical Engineering and Computer Science*, 24(2), 978–985. <https://doi.org/10.11591/ijeecs.v24.i2.pp978-985>
- Mitra, B., Sharma, K., Acharya, S., Mishra, P., & Guglani, A. (2022). Real-time Smile Detection using Integrated ML Model. *Proceedings - 2022 6th International Conference on Intelligent Computing and Control Systems, ICICCS 2022*, 1374–1381. <https://doi.org/10.1109/ICICCS53718.2022.9788399>
- Qureshi, J. N., Farooq, M. S., Khelifi, A., & Atal, Z. (2024). Harnessing the Potential of Blockchain in ChainAgilePlus Framework for the Improvement of Distributed Scrum of Scrums Agile Software Development. *IEEE Access*, 1–1. <https://doi.org/10.1109/ACCESS.2024.3426597>
- Sahasrabuddhe, N., Jain, R., Khandelwal, S., Kalekar, S., Purohit, N., & Sharma, A. K. (2023). Print - OnTheGo Remote Printing Application using MERN Stack. *2023 4th International Conference for Emerging Technology, INCET 2023*. <https://doi.org/10.1109/INCET57972.2023.10170169>
- Sahu, Y., Gupta, R. K., & Bharti, S. (2023). Combating Hate Speech on Q&A Forums with Machine Learning. *2023 World Conference on Communication and Computing, WCONF 2023*. <https://doi.org/10.1109/WCONF58270.2023.10235146>
- Satapathy, S. K., & Manavadariya, K. (2023). Revolutionizing Social Media Marketing with Social Pilot and NodeJS Optimization Techniques. *International Conference on Recent Advances in Science and Engineering Technology, ICRASET 2023*. <https://doi.org/10.1109/ICRASET59632.2023.10420157>
- Saundariya, K., Abirami, M., Senthil, K. R., Prabakaran, D., Srimathi, B., & Nagarajan, G. (2021). Webapp service for booking handyman using MongoDB, express JS, React JS, node JS. *2021 3rd International Conference on Signal Processing and Communication, ICPSC 2021*, 180–183. <https://doi.org/10.1109/ICSPC51351.2021.9451783>
- Sayago Heredia, J., Flores, E., & Recalde, A. (2019). *Análisis Comparativo entre los Estándares Orientados a Servicios Web SOAP, REST y GRAPHQL*. <https://doi.org/10.5281/zenodo.3592004>
- Sayago Heredia, J., Flores-García, E., & Solano, A. (2020). *Comparative Analysis Between Standards Oriented to Web Services: SOAP, REST and GRAPHQL* (pp. 286–300). [https://doi.org/10.1007/978-3-030-42517-3\\_22](https://doi.org/10.1007/978-3-030-42517-3_22)
- Shersingh Chauhan, A., Bhardwaj, S., Shaikh, R., Mishra, A., & Nandgave, S. (2022). Food Ordering website “Cooked with care” developed using MERN stack. *Proceedings - 2022 6th International Conference on Intelligent Computing and Control Systems, ICICCS 2022*, 1690–1695. <https://doi.org/10.1109/ICICCS53718.2022.9788224>
- Tesone, P., Polito, G., Fabresse, L., Bouraqadi, N., & Ducasse, S. (2020). Preserving instance state during refactorings in live environments. *Future Generation Computer Systems*, 110, 1–17. <https://doi.org/10.1016/j.future.2020.04.010>
- Vasanthi, D., Sivasakthi, T., Abarna, V., & Arthi, R. (2021). Design and Development of Car Rental Website Using Mern Stack. *2021 International Conference on Computing, Communication and Green Engineering, CCGE 2021*. <https://doi.org/10.1109/CCGE50943.2021.9776473>
- Vinothini, C., Tarun, C., Bhavana, K. A., Nikhil Krishna, G., & Praveen, A. (2022). NBA web portal: A Comprehensive Survey on NBA Accreditation and MERN Stack for the Purpose of

Implementing a Portal. *5th International Conference on Inventive Computation Technologies, ICICT 2022 - Proceedings*, 1245–1251. <https://doi.org/10.1109/ICICT54344.2022.9850953>