

Sistema web de gestión administrativa para PYMES utilizando la tecnología Mean Stack

Administrative management web system for PYMES using Mean Stack



Santiago Rafael Heredia Sayay¹
Elsa Yoconda Narváez Vilema²
Miryan Estela Narváez Vilema¹

✉ <https://orcid.org/0000-0003-4857-7303>
✉ <https://orcid.org/0009-0003-5321-7908>
✉ <https://orcid.org/0000-0002-7734-419X>

¹Universidad Nacional de Chimborazo UNACH | Riobamba, Chimborazo - Ecuador | CP 060108

²Unidad Educativa PCEI Chimborazo | Riobamba, Chimborazo - Ecuador | CP 060101

✉ santiago.heredia@unach.edu.ec

<http://doi.org/10.26423/rctu.v11i2.833>

Páginas: 57- 67

Resumen

En un mercado en constante evolución es indispensable para toda microempresa en crecimiento, mejorar la credibilidad del negocio, generar ahorros y optimizar los procesos de manera sistemática. El objetivo de la investigación fue desarrollar un sistema web de gestión de ventas utilizando Mean Stack, este framework de desarrollo permite manipular el frontend y backend de manera individual. Para la construcción del sistema, se usa la metodología ágil Kanban, permitiendo dividir el trabajo por etapas y clasificarlos adecuadamente siguiendo el formato TEP (Tareas En Progreso). La investigación tuvo un enfoque cuantitativo, se realizaron pruebas de rendimiento con la herramienta de análisis Apache Jmeter. Los resultados más relevantes mostraron una reducción en el tiempo de espera de cada petición, de igual manera, al inducir el sistema a pruebas de carga y estrés se evidenció la robustez con la que el Api Rest (Representational State Transfer Application Programming Interface), resuelve las operaciones que el usuario plantea desde la interfaz. El éxito del sistema web se atribuye a combinar un framework de desarrollo robusto para aplicaciones web modernas, y a una metodología ágil que acompaña cada etapa del trabajo con pruebas unitarias y mantiene una gestión de proyectos más fluida.

Palabras clave: Api Rest, Desarrollo de software, JavaScript, Metodología Kanban.

Abstract

In a constantly evolving market, it is essential for any growing microenterprise to improve the credibility of the business, generate savings and optimize processes systematically. The objective of the research was to develop a web sales management system using Mean Stack, this development framework allows the frontend and backend to be manipulated individually. To build the system, the agile Kanban methodology is used, allowing work to be divided into stages and classified appropriately following the TEP (Tasks in Progress) format. The research had a quantitative approach, performance tests were carried out with the Apache Jmeter analysis tool. The most relevant results showed a reduction in the waiting time for each request, in the same way, by inducing the system to load and stress tests, the robustness with which the Rest API (Representational State Transfer Application Programming Interface) resolves the operations that the user poses from the interface. The success of the web system is attributed to combining a robust development framework for modern web applications, and an agile methodology that accompanies each stage of work with unit tests and maintains more fluid project management.

Keywords: Rest API, Software development, JavaScript, Kanban methodology.

Recepción: 23/08/2024 | Aprobación: 06/12/2024 | Publicación: 26/12/2024

1. Introducción

JavaScript es un lenguaje de programación orientado a objetos, su estructura se basa en prototipos y es interpretativo, es decir, ejecuta las líneas de comando a través de los motores de lectura de scripts incorporados en los navegadores. En los años 90s, la velocidad de transferencia de datos no superaba los 28.8 kbps, las aplicaciones y páginas web eran estáticas con un contenido básicamente en texto plano, posteriormente se fueron añadiendo elementos de interacción como los formularios, los cuales permitían al usuario ingresar valores más estructurados a través del teclado, por medio de botones o mediante listas desplegables. Durante mucho tiempo la validación de datos se realizaba en el servidor, como resultado se obtenía un tiempo de respuesta del sistema demasiado largo, provocando los denominados ‘crash’ o colapso del funcionamiento del software y sistema operativo. A partir del año 1995 Sun Microsystems y Netscape desarrollan un lenguaje de validación de datos desde la capa de aplicación, JavaScript apareció y cumplía la tarea de verificar que los datos cumplan con reglas de ortografía, semántica y coherencia antes de ser enviados al servidor. En el año 1997 el organismo ECMA (European Computer Manufacturers Association) estandariza el lenguaje JavaScript para ser multiplataforma e independiente a cualquier empresa, algunos programadores

prefieren llamarlo también ECMAScript [1].

En el desarrollo web, es crucial evaluar diferentes tecnologías y metodologías para determinar cuál se adapta mejor a las necesidades específicas de las PYMES. JavaScript cumple funciones específicas como crear animaciones y efectos para dinamizar páginas web gracias a su librería JQuery, sin embargo, en la actualidad JavaScript se ha visto inmerso en el desarrollo de servidores web gracias a la incorporación de extensiones que permiten al lenguaje manipular objetos y clases, con estas especificaciones se puede codificar en el cliente y servidor a través de un solo lenguaje, a este tipo de desarrollo se lo conoce como Full Stack, y los framework Mean y Mern (MongoDB, Express.js, React, Node.js) son los más utilizados. Mean Stack es un conjunto de cuatro subsistemas que permiten la construcción de aplicaciones web SPA (Single Page Application) cuya arquitectura es similar a la de Modelo Vista Controlador [2],[3]. La combinación de estas tecnologías permite el desarrollo web de extremo a extremo, su lenguaje natural es JavaScript y sus componentes son: Mongo como base de datos no relacional, Express que provee las funciones y métodos necesarios para el desarrollo de Api Rest, Angular para la interfaz gráfica y desarrollo de páginas SPA, y Node para la creación de servidores multiplataforma [4],[5]. En la Figura 1 se muestra una representación de la arquitectura de Mean Stack y la comunicación bilateral entre cliente y servidor.

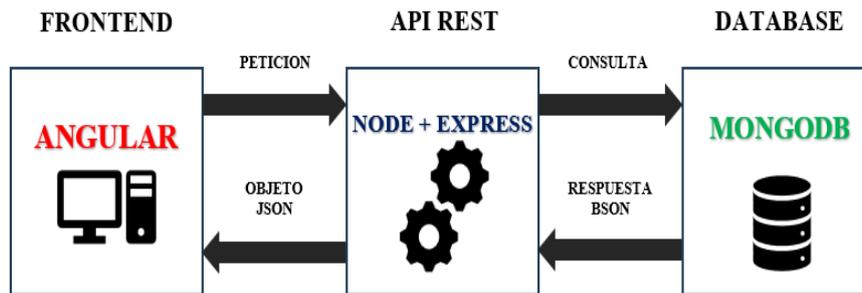


Figura 1. Arquitectura del framework Mean Stack

LAMP stack es popular para el desarrollo web que combina cuatro componentes de software libre fundamentales. El acrónimo LAMP representa las iniciales de sus componentes clave (Linux, Apache, MySQL, PHP). Linux como sistema operativo, Apache como servidor web, MySQL como sistema de base de datos y PHP como lenguaje de programación para el back-end. Cada componente de LAMP se integra de manera fluida, convirtiéndolo en una de las plataformas más potentes para el desarrollo de aplicaciones web. LAMP ofrece control total sobre el servidor y acceso remoto, facilitando la administración a través de servidores Linux desde cualquier lugar [6].

Django es un framework de alto nivel basado en Python, muy popular para el desarrollo de aplicaciones rápida, segura y escalable. Con Django, muchas funcionalidades comunes, como la autenticación de usuarios, la gestión de bases de datos y la administración de contenido, ya están incluidas y listas para usar, lo que reduce la necesidad de escribir código desde cero. Esto optimiza el proceso de desarrollo,

minimiza errores y facilita el mantenimiento de aplicaciones web complejas, haciendo que Django sea una excelente opción para el desarrollo rápido de sistemas de información basados en la web [7].

Ruby on Rails es un lenguaje de programación dinámico y de código abierto que se centra en la simplicidad y la productividad. Su sintaxis elegante lo hace fácil de leer y escribir. Ruby ha ganado reconocimiento, en gran parte, permitiendo un desarrollo ágil de aplicaciones web [8]. Hoy en día, el desarrollo de software exige el cumplimiento de buenos principios y estándares de ingeniería que faciliten el desarrollo, diseño, implementación y mantenimiento del sistema informático. Los proyectos más exitosos han sido aquellos que de principio a fin son administrados y controlados bajo estrictos lineamientos, de no ser así corren el riesgo de fracasar [9].

La ventaja de utilizar el marco de desarrollo MEAN Stack es su arquitectura, en donde cada módulo contiene sus propios

métodos, servicios, funciones y controladores para resolver la lógica de negocio de manera individual, esto beneficia la reutilización y minimiza el código espagueti, utiliza JavaScript en todas las capas de la aplicación (frontend, backend y base de datos), permite a los desarrolladores trabajar con un único lenguaje de programación, reduciendo la complejidad y facilitando la colaboración entre el cliente y el servidor, optimizando el flujo de trabajo. MEAN Stack es ideal para gestionar aplicaciones altamente escalables, su compatibilidad con servicios en la nube y su ecosistema open-source, ofrece una solución ágil y eficiente; ideal para proyectos que requieren rapidez de desarrollo y rendimiento, como aplicaciones en tiempo real o de alto tráfico.

La elección de la tecnología y metodología adecuada para una PYME depende de varios factores, como el presupuesto, la complejidad de la aplicación, la necesidad de escalabilidad y el tiempo de desarrollo. Las metodologías ágiles son excelentes opciones por las pequeñas y medianas empresas debido a su capacidad para organizar, optimizar y mejorar procesos de manera eficiente sin requerir grandes inversiones de tiempo o presupuesto. Metodologías como SCRUM, Extreme Programming (XP), Feature Driven Development (FDD) y Kanban permiten a las PYMES adaptarse rápidamente a cambios, mejorar la calidad de

sus productos y entregar valor de manera continua. Al ser flexibles y escalables, estas metodologías ayudan a las PYMES a ser más competitivas, maximizando el uso de sus recursos disponibles [10].

El objetivo de implementar la metodología Kanban en esta investigación fue restar las denominadas TEP (Tareas En Progreso), separando el desarrollo del sistema web en etapas acordes al ciclo de vida de software, además de tener control de producción más visual y gestionable. Una característica primordial de Kanban es prevenir los cuellos de botella durante el trabajo, por lo que, no existieron tareas inconclusas dentro de la construcción del sistema [11],[12]. De esta manera, cada parte del sistema comprendido entre backend y frontend son independientes y cada uno maneja un control de versiones específico. Para la comunicación entre ambas partes se construyó un Api Rest que integra métodos y funciones HTTP y manipulan la base de datos con funciones CRUD (Create, Read, Update, Delete) [13]. Al no utilizar una base de datos relacional el control de datos se realiza por protocolos de transferencia, similares a las funciones que manipulan una base de datos SQL [14], como se muestra en la Tabla 1, existe una similitud entre el manejo de información en una base de datos SQL y NoSQL.

Tabla 1: Comparación entre métodos CRUD y HTTP

Operación	BD Relacional	BD No Relacional	Descripción
Create/Crear	Insert	Post/Put	Agrega un registro
Read/Leer	Select	Get	Lee o devuelve un registro
Update/Actualizar	Update	Put/Patch	Actualiza un registro
Delete/Eliminar	Delete/Drop	Delete	Elimina un registro

Con este marco de desarrollo acompañado de la metodología ágil, se crea un sistema web administrativo enfocado al sector automotriz, empresas que están adoptando plataformas digitales para gestionar diferentes áreas del negocio, tales como ventas, atención al cliente, mantenimiento de vehículos y pasarelas de pago en línea que facilita el proceso de compra, tanto para productos como para servicios. A través de esta solución tecnológica, la compañía puede mejorar la competitividad en el mercado, brindando una experiencia más ágil, personalizada y orientada a las necesidades de los consumidores.

Finalmente, el sistema web fue sometido a pruebas de carga y estrés con el objetivo de conocer picos de pérdida de información dando como resultado que, los sistemas desarrollados bajo el Stack completo de Mean, son eficientes para el manejo de información masiva, además, pone en práctica la filosofía del diseño centrado en el usuario ya que, al utilizar un framework de frontend como Angular la información siempre está actualizada en la interfaz [15]. De igual manera, el sistema web se evaluó en base a las métricas que la herramienta Apache Jmeter proporciona, estas fueron: mínimo y máximo, media, mediana y desviación estándar; y para el rendimiento se evaluó a través de los listeners o escuchas de Jmeter: tiempo de respuesta, porcentaje de errores, número de bits por segundo, número de páginas por segundo [16],[17].

Este proyecto de investigación, tiene como objetivo diseñar un sistema web de gestión administrativa que cumpla con la definición de arquitectura limpia, modular y escalable, que

beneficie al crecimiento tecnológico en las Pymes y aporte al desarrollo de software enfocado en el sector comercial pequeño y mediano.

2. Materiales y Métodos

La investigación realizada tiene un enfoque cuantitativo con el propósito de obtener datos mensurables, se evaluó el sistema web con Apache Jmeter para obtener los índices de error, porcentaje de carga y tiempo de respuesta del servidor. Para las pruebas, se considera alrededor de 2.000 usuarios con el propósito de conocer el pico máximo que puede soportar sin afectar su funcionamiento. Esto permitirá asegurar que, al añadir nuevos módulos o ejecutar más solicitudes, el sistema mantenga su rendimiento sin sufrir caídas o problemas de estabilidad. En cuanto a la evaluación de la interfaz de usuario se empleó la herramienta BlazeMeter, esta plataforma en línea permitió ejecutar y grabar la interacción usuario-sistema, el resultado final se exportó a Jmeter para evaluar el tiempo de respuesta de la interfaz gráfica, está investigación esclareció que con el framework Mean Stack no solo el desarrollo se puede dividir en partes, también las pruebas se realizaron de manera independiente.

Del mismo modo se usó la metodología de campo y bibliográfica con el fin de tener los datos necesarios para la creación del API Rest que consumirá el sistema web. El proyecto recopiló información publicada en libros, revistas científicas, tesis de grado, páginas web, esto permitió comprender mejor los conceptos relacionados con la temática

abordada y concluir el desarrollo del sistema web.

El desarrollo de software debe ser gestionado por una metodología integrada que aborde de forma homogénea las tareas que conforman el ciclo de vida de un proyecto. Independientemente de la metodología ágil que se seleccione, el desarrollo precisa un enfoque sistemático, cuantitativo y orientado a resultados que garanticen el éxito del trabajo, es importante seleccionar la metodología que mejor se adapte

a las necesidades que se van a satisfacer [18]. Para el cumplimiento de esta investigación se utilizó la metodología Kanban condicionada para el desarrollo de un sistema de gestión administrativa, esto significa que, las tareas avanzan a medida que se cumplen, existe mayor eficiencia y flexibilidad en el cumplimiento, además, se tiene un mejor control del flujo de tiempo. En la Figura 2 se ilustra la disposición de tareas que se realizaron para el desarrollo del sistema web de gestión administrativa.



Figura 2. Tablero Kanban aplicado

La construcción del sistema web se basó en los principios del modelo en cascada, que no permite continuar con el desarrollo de software dejando fases inconclusas. Las etapas comprendidas en el proyecto fueron: • Análisis de requerimientos • Diseño funcional • Implementación y pruebas

Análisis de requerimientos

Se definen las características principales y funcionamiento del sistema web basado en las principales actividades económicas de las Pymes. Por lo general, las startups

o empresas crecientes se apoyan en las Tecnologías de la Información para garantizar un rápido crecimiento económico, el objetivo a corto plazo es solidificar su arquitectura empresarial y afianzar a largo plazo su presencia en el mercado. Las principales actividades tienen que ver con la compra y venta de artículos, control de inventario, fidelización de clientes, reportes de compra y venta, además, control de acceso a la información del sistema. La Figura 3 muestra de qué manera el usuario interactúa con el sistema web desarrollado, esta actividad permitió conocer a detalle la lógica de negocio.

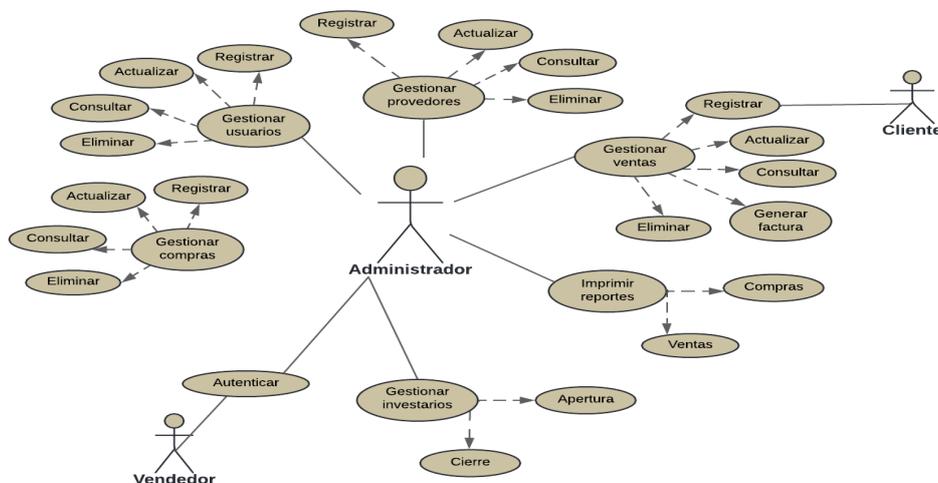


Figura 3. Diagrama de casos de uso del sistema web

Diseño funcional

Se diseñó las funcionalidades de la interfaz gráfica de usuario, tomando en cuenta el borrador inicial representado en el diagrama de casos de uso, existen dos tipos de usuarios con roles diferentes; Administrador ejecuta todas las funcionalidades del sistema, excepto aquellas considerablemente importantes para el historial de la

empresa; mientras que el usuario con rol de Vendedor solo tiene acceso a las funcionalidades que permiten resolver los registros de compra y venta, generación de reportes y facturas, además, de la gestión de clientes. La Figura 4 presenta el bosquejo inicial del sistema siguiendo la filosofía de diseño centrado en el usuario, permitiendo que se cree el sentimiento de pertenencia y permanencia del cliente con las actividades de la empresa.



Figura 4. Plano estructural del sistema web

Se creó el modelado de la base de datos para ser implementada en Mongo, esta es una base de datos no relacional por lo que no existen claves primarias ni foráneas. El objetivo es permitir que el usuario almacene información y la obtenga inmediatamente en formato BSON para el

tratamiento de transacciones con un API Rest. Como ilustra la Figura 5, el modelado de datos es muy similar al de una base de datos relacional, con la diferencia que en Mongo existen un tipo de relación orientado a documentos, lo que garantiza que la información sea más rápida de tratar y consumir.



Figura 5. Modelado de la base de datos de Mongo.

Implementación y pruebas

Para el backend se implementó una arquitectura de eventos, cada evento responde a las peticiones intrínsecas de cada módulo del backend. JavaScript es un lenguaje que permite

el manejo de promesas asíncronas por lo que, las peticiones externas manejan errores y valores eventuales, como se muestra en la Figura 6 la arquitectura de eventos de Node admite gestionar las peticiones según como vayan llegando desde el frontend.

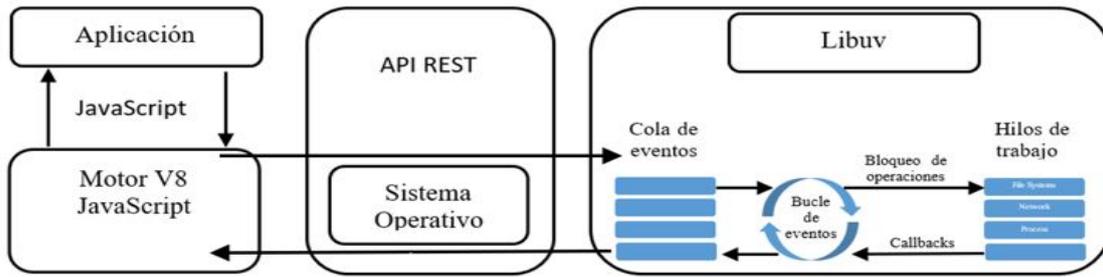


Figura 6. Arquitectura de un servidor web con Node.

Por otra parte, la Figura 7 representa la lógica de negocio que se codificó para automatizar las actividades que comúnmente gestiona una empresa dedicada al comercio de bienes y servicios. Se consideraron como más importantes las siguientes:

- Generar compra y venta de artículos en stock.
- Actualización automática de stock de artículos.
- Generar reportes de compra y venta.
- Generar el listado

- de productos más vendidos.
- Generar alertas cuando un producto tenga poco o excesivo stock.
- Generar reporte de clientes con mayores compras.
- Consulta de compra y venta por diferentes parámetros.
- Apertura y cierre de caja.
- Seguimiento y conversión de clientes.
- Generación de Dashboard de KPIs.

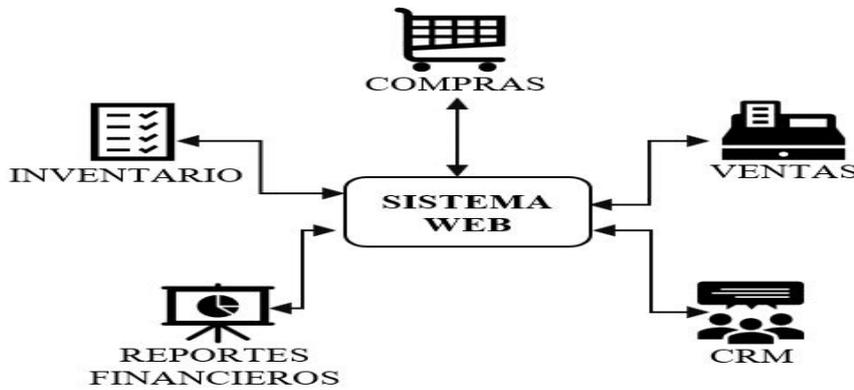


Figura 7. Lógica de negocio del sistema web.

Finalmente, se desarrolló el frontend con Angular, como se mencionó anteriormente cada parte maneja una arquitectura y control de versiones independiente, en el caso de la interfaz gráfica se implementó un Modelo Vista – Vista Modelo, totalmente independiente del backend. Mean Stack permite el desarrollo de aplicaciones web que recargan una sola vez

la página, todos los datos del servidor son capturados por Angular para tenerlos disponibles al usuario, cabe resaltar la importancia de que un sistema de gestión, tenga a disposición los datos para la ejecución de actividades. En la Figura 8 se muestra la arquitectura MVVM con la que Angular permite el manejo de datos en segundo plano.

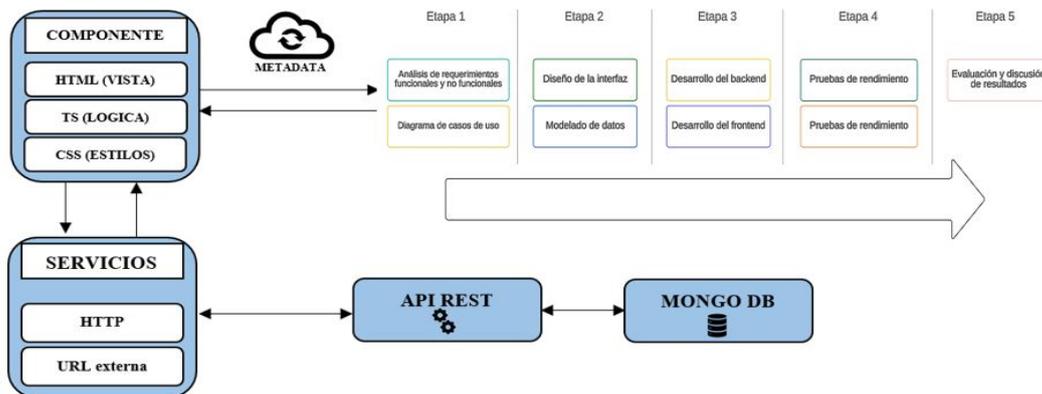


Figura 8. Arquitectura de la Interfaz Gráfica.

3. Resultados y Discusión

Como resultado de la investigación se obtuvo el sistema de gestión administrativa denominado SYSMOTOR V1.0.1, su arquitectura permite la escalabilidad horizontal para

ampliación de actividades, nuevos usuarios y roles, facturación electrónica y consumo de APIs externas. La Figura 9 ilustra en detalle la página principal del sistema web con los KPIs principales, módulos y otras funcionalidades.

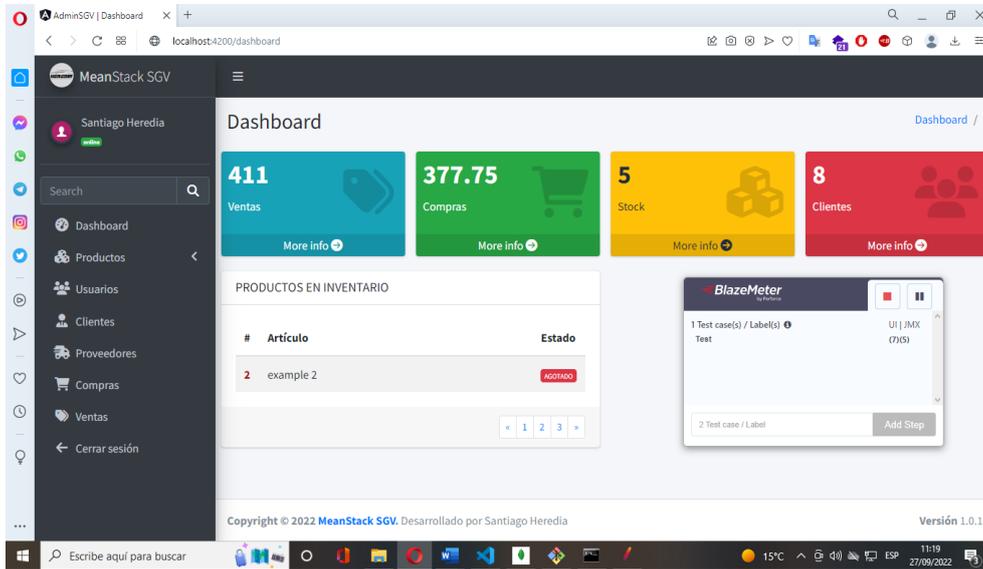


Figura 9. Pantalla principal del sistema web de gestión administrativa.

Los resultados que se obtuvieron de la evaluación del funcionamiento del sistema web fueron tratados con la herramienta Apache Jmeter, en la Figura 10 se muestra el gráfico de respuesta a las peticiones enviadas por el frontend, las líneas en colores representan el rendimiento, la

media, la mediana y la desviación estándar. Cabe recalcar que, para estas pruebas era preciso estresar el sistema con peticiones masivas. A continuación, se realizó 150 peticiones consecutivas.

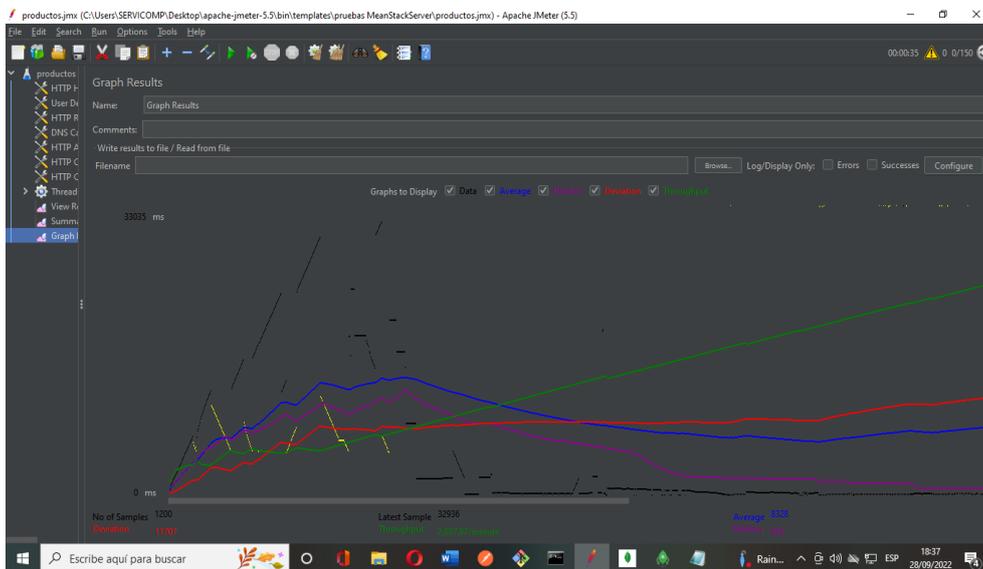


Figura 10. Gráfico de resultados del sistema web.

En color azul se representa el promedio del tiempo de resolución de peticiones con 8,328 milisegundos. El color purpura representa la mediana con un valor de 863 milisegundos la cual no se ve afectada por valores atípicos. El color rojo representa la desviación estándar manteniéndose

en un rango estable, lo que significa que es certera la medición de valores. Y, por último, en color verde se muestra el rendimiento del sistema web, se puede notar a medida que el tiempo transcurre el rendimiento se incrementa.

Tiempo de respuesta

Este indicador muestra el tiempo de respuesta para cada petición, a diferencia del gráfico de resultados este provee

información individual de cada petición realizada al backend. En la Figura 11 se puede apreciar la línea de color rojo con el promedio de tiempo del sistema en general, aproximadamente entre 9,000 y 10,000 milisegundos.



Figura 11. Gráfico de tiempo de respuesta.

Rendimiento

Se elaboró un plan de pruebas con Jmeter con el propósito de valorar los módulos del backend, las consultas de Jmeter se realizaron a través de peticiones HTTP. En la Tabla

2 se ilustran a detalle los resultados del rendimiento de cada módulo con una carga de 50 usuarios concurrentes. La finalidad para esta prueba fue conocer el margen de error para cada petición y reflexionar sobre la implementación de mejoras en la lógica de negocio del sistema web.

Tabla 2: Análisis de varianza para la variable resistencia a compresión (Tabla ANOVA)

Módulo	Samplers	Tiempo de respuesta promedio(segundos)	Error(%)	Rendimiento (petición/s)	KB recibidos (kb/s)	KB enviados (kb/s)	Desviación estándar
PRODUCTOS	50	0.700s	0%	31.0/s	72.29/s	11.81/s	11.81
COMPRAS	50	0.480s	0%	25.4/s	49.75/s	9.62/s	9.62
CLIENTES	50	0.242s	0%	24.8/s	22.0/s	9.40/s	9.40
VENTAS	50	0.463s	0%	20.7/s	51.72/s	7.83/s	7.83
USUARIOS	50	0.239s	0%	28.9/s	41.56/s	10.96/s	10.96
CATEGORIAS	50	0.217s	0%	31.5/s	72.47/s	12.02/s	12.02

Carga y estrés

Para conocer los límites que tiene el backend en la resolución de peticiones es necesario conocer cuál es el pico máximo que soporta antes de llegar al fallo total, de esta manera cuando se

precise añadir más módulos para ejecutar nuevas peticiones estas no provoquen una caída del rendimiento del sistema. En la Tabla 3 se muestran los resultados del sistema web con una carga de 2 000 usuarios.

Tabla 3: Resultados de las pruebas de carga y estrés

Módulo	Samplers	Tiempo de respuesta promedio(segundos)	Error(%)	Rendimiento (petición/s)	KB recibidos (kb/s)	KB enviados (kb/s)	Desviación estándar
PRODUCTOS	2000	0.954s	81.9%	178.9/s	446.81/s	12.30/s	2168.95
COMPRAS	2000	0.327s	81.9%	161.0/s	391.20/s	11.01/s	859.34
CLIENTES	2000	0.168s	81.9%	155.3/s	347.39/s	10.65/s	433.01
VENTAS	2000	0.205s	81.9%	170.1/s	429.85/s	11.60/s	451.26
USUARIOS	2000	0.101s	81.9%	175.4/s	409.78/s	12.02/s	228.74
CATEGORIAS	2000	0.89s	81.9%	173.4/s	432.12/s	11.95/s	203.91

Las métricas evaluadas son indicadores críticos que influyen directamente en el éxito de una empresa en el mercado digital. Al optimizar estos aspectos, las empresas mejoran la satisfacción del cliente, aumentan su rentabilidad, protegen su reputación y preparan su infraestructura para el crecimiento futuro. La habilidad para gestionar estas variables técnicas determina si una empresa puede adaptarse, escalar y competir en mercados globales cada vez más exigentes.

Como se observó en los resultados, Mean Stack es una buena opción para aplicaciones a gran escala con un rendimiento óptimo para la sobrecarga masiva de información. Un resultado similar describe Dunka [19], en la investigación destaca que MEAN stack es una tecnología robusta y de

vanguardia que ha demostrado ser una opción confiable para aplicaciones rápidas, escalables y en tiempo real. Esto significa que la técnica estándar de mezclar JavaScript, CSS y HTML ya no es suficiente [4],[20]. Las aplicaciones web empresariales de gran tamaño desarrolladas mediante MEAN ha demostrado ser una solución completa y potente para satisfacer las demandas de las webs actuales. Los resultados del estudio demuestran el valor añadido del análisis comparativo de MEAN y Java EE, a través de la creación de dos aplicaciones similares, una con cada stack, se analizan aspectos clave como el tiempo de respuesta, el rendimiento y la capacidad de manejar grandes volúmenes de datos [21]. La Tabla 4, presenta los resultados de tiempo de respuesta de la investigación previa (Heredia & Sailema) y del aplicativo web SYSMOTOR.

Tabla 4: Tiempo de respuesta de MEAN y Java EE

Etiqueta	Muestras	Promedio	Error(%)	Rendimiento	(kb/s) Recibidos
Java EE	1000	953 ms	0,00 %	350,30 %	2351,28
MEAN	1000	2 ms	0,00 %	352,50 %	412,12
MEAN	2000	954 ms	81,9 %	178.9 %	446.81 %

La demanda tecnológica para crear sistemas de gestión, abre la oportunidad de introducir un nuevo marco de desarrollo basado únicamente en el lenguaje JavaScript. El sistema web SYSMOTOR V1.0.1 fue desarrollado con MEAN Stack, un framework popular y flexible, combina cuatro de las mejores tecnologías web modernas, de código abierto que, en conjunto, proporcionan un marco de un extremo a otro para desarrollar aplicaciones web dinámicas, rápidas y escalables [2],[22],[23].

En la investigación “Comparative analysis of MEAN stack and MERN stack”, los autores manifiestan que MEAN stack es un marco extremadamente confiable para un desarrollo front-end increíblemente rápido. Da como resultado aplicaciones ligeras basadas en JavaScript. Esto hace que MEAN stack sea una mejor opción para aplicaciones a gran escala [4],[24].

En este contexto, Nirgudkar y Singh (2017) señalan que la evolución tecnológica ha impulsado cambios significativos en el desarrollo de software, especialmente en el ámbito del desarrollo web. Mientras que tecnologías tradicionales como JAVA servlets, PHP, ASP.NET, Ruby, etc. han sido ampliamente utilizadas, presentan limitaciones en comparación con los requisitos actuales en lo que respecta al rendimiento. Para afrontar estos desafíos, han surgido alternativas como MongoDB, Express, Node.js y Angular que ofrecen mejor desempeño, escalabilidad y capacidad para manejar aplicaciones en tiempo real y grandes volúmenes de usuarios de manera eficiente [2].

Una vez analizadas todas las aportaciones de los autores, es evidente que respaldan el uso del MEAN Stack para aplicaciones escalables, de alto rendimiento, fácil de implementar y altamente interactivo, sin embargo, es importante reconocer que MEAN requiere de un análisis más exhaustivo y variables de medición para garantizar su eficacia en el manejo de grandes volúmenes de datos y peticiones en tiempo real. Por lo tanto, aunque el MEAN stack es una opción eficiente en entornos bien desarrollados, su implementación en contextos menos avanzados puede requerir ajustes considerables en cuanto

a infraestructura, capacitación y recursos, aspectos que no fueron profundamente evaluados en la investigación.

4. Conclusiones

Mean Stack, una combinación de tecnologías JavaScript de código abierto, ha demostrado ser una opción sólida y versátil para el desarrollo de aplicaciones web modernas. Su arquitectura basada en MongoDB, Express.js, Angular y Node.js ofrece una serie de ventajas que lo convierten en una elección popular entre los desarrolladores para crear aplicaciones web de alta calidad de manera rápida y eficiente para proyectos tanto pequeños como a gran escala. El desarrollo del API Rest con Node y Express también facilitó a que la comunicación asíncrona con la base de datos de Mongo sea óptima, se concluye que, parte de la comunicación y la resolución de peticiones dependen también de una buena lógica de negocios, siempre enfocada a lo que el sistema debe cumplir.

La adopción de herramientas que soporten comunicaciones asincrónicas robustas entre módulos y bases de datos, junto con prácticas de desarrollo ágiles como Kanban, podrían fortalecer el rendimiento en diversos escenarios, permitiendo la expansión del sistema de forma gradual y adaptable según las necesidades del contexto. Esto facilitaría la creación de sistemas más complejos y eficientes en diferentes sectores.

La investigación realizada concluye con resultados que demuestran un mayor rendimiento del sistema web y que permite la escalabilidad y fácil mantenimiento, la metodología Kanban aplicada al desarrollo de software, permitió la construcción independiente del cliente y servidor, lo que ayudó a realizar las pruebas de funcionamiento separadas.

Este trabajo de investigación deja abierta las puertas a nuevas implementaciones y ampliación de servicios, puede ser el caso de un módulo de facturación independiente, control de la cadena de producción, entre otros. De igual manera, se pretende realizar la continuación del sistema con

una arquitectura de software más enfocada al consumo de microservicios, ya que, Mean Stack facilita la separación de módulos.

Financiamiento:

Los autores declaran que este trabajo no tiene fuentes de financiamiento.

Conflicto de intereses:

Los autores declaran que no existen conflictos de interés de ninguna naturaleza en la presente investigación.

Contribución de autor/es:

Bajo los lineamientos CRediT, los autores dan fe de las contribuciones realizadas al trabajo, que se detallan: Autor principal Santiago Heredia: 35% de autoría; escritura, gestión, recursos y tecnología; investigación Coautor 1 Elsa Narváez: 30% que corresponde a validación de la escritura, supervisión, curación de datos de la investigación Coautor 2 Miryan Narváez: 35% que corresponde a validación de la escritura, supervisión, conducción de la investigación

5. Referencias

- EGUILUZ. Introducción a JavaScript. [Online]. 2024. Disponible en: <https://uniwebsidad.com/libros/javascript?from=librosweb>.
- NIRGUDKAR, NINAAD Y SINGH, POOJA. The MEAN Stack. *International Research Journal of Engineering and Technology* [online]. 2017. ISSN 2395-0072. Disponible en: <https://www.irjet.net/archives/V4/i5/IRJET-V4I5795.pdf>.
- BERA, MARCIO, MINE, ANDERSON Y LOPES, LUIZ. MEAN Stack: Desenvolvendo Aplicações Web Utilizando Tecnologias Baseadas em JavaScript. [Online]. 2015. Disponible en: https://unicv.edu.br/admin/assets/repositorio_arquivo/413a333c0cc0872f3ee8f9d22f7cc166.pdf.
- AGGARWAL, SANCHIT Y VERMA, JYOTI. Comparative analysis of MEAN stack and MERN stack. *International Journal of Recent Research Aspects* [online]. 2018, vol. 5. ISSN 2395-0072. Disponible en: <https://ijrra.net/Vol5issue1/IJRR-05-01-26.pdf>.
- CHOTOMAZA, JHONATAN, ÁVILA-PESANTEZ, DIEGO Y ÁVILA PESANTEZ, LUZ MIRIAM. View of Mobile App development using the MEAN Stack and IONIC framework: A case study in a transport company. [Online]. 2020. Disponible en: <https://journals.gdeon.org/index.php/esj/article/view/74/58>.
- KARANJIT, ARPANA. MEAN vs. LAMP Stack. *Culminating Projects in Computer Science and Information Technology*. [Online]. 2016. Disponible en: <https://journals.gdeon.org/index.php/esj/article/view/74/58>.
- VIDAL-SILVA, CRISTIAN, SÁNCHEZ-ORTIZ, AURORA, SERRANO, JORGE Y RUBIO, JOSÉ. Experiencia académica en desarrollo rápido de sistemas de información web con Python y Django. *Formación universitaria*. [Online]. 2021, vol. 14, n.º 5. ISSN 0718-5006. Disponible en: http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-50062021000500085&lng=es&nrm=iso&tlng=es.
- PLEKHANOVA, JULIA. Evaluating web development frameworks: Django, Ruby on Rails and CakePHP. *Institute for Business and Information Technology*. [Online]. 2009. ISSN 1938-1271. Disponible en: <http://ibit.temple.edu/wp-content/uploads/2011/03/IBITWebframeworks.pdf>.
- DELGADO, LISDANIA DE LA CARIDAD Y DÍAZ, LEXYS MANUEL. Modelos de Desarrollo de Software. *Revista Cubana de Ciencias Informáticas* [online]. 2021, vol. 15, n.º 1. ISSN 2227-1899. Disponible en: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992021000100037&lng=es&nrm=iso&tlng=es.
- MITRE HERNÁNDEZ, HUGO; ORTEGA MARTÍNEZ EDGAR Y LEMUS OLALDE, CUAUHTÉMOC. Estimación y control de costos en métodos ágiles para desarrollo de software: un caso de estudio. *Ingeniería, Investigación y Tecnología. Revista Cubana de Ciencias Informáticas* [online]. 2014, vol. 15, n.º 3. ISSN 1405-7743. Disponible en: <https://www.sciencedirect.com/science/article/pii/S1405774314703506>.
- ARANGO SERNA, MARTIN DARIO; CAMPUZANO ZAPATA, LUIS FELIPE Y ZAPATA CORTES, JULIAN ANDRÉS. Mejoramiento de procesos de manufactura utilizando Kanban. *Revista Ingenierías Universidad de Medellín* [online]. 2015, vol. 14, n.º 27. ISSN 1692-3324. Disponible en: https://repository.udem.edu.co/bitstream/handle/11407/2378/Revista_Ingenierias_UdeM_260.pdf?sequence=2&isAllowed=y.
- FLORES CERNA, FERNANDO; SANHUEZA SALAZAR, VICTOR MANUEL; VALDÉS GONZÁLEZ, HÉCTOR MOISÉS Y REYES BOZO, LORENZO. Metodologías ágiles: un análisis de los desafíos organizacionales

- para su implementación. *Revista científica* [online]. 2022, vol. 43, n.º 43. ISSN 0124-2253. Disponible en: <https://repositorio.udd.cl/server/api/core/bitstreams/c75741f2-5da7-420a-b144-07b2a7d484a1/content>.
13. AMODEO, ENRIQUE. Principios de diseño de APIs REST. [Online]. 2013, vol. 1. Disponible en: http://leanpub.com/introduccion_apis_rest.
 14. LIAO, YING-TI; ZHOU, JIAZHENG; LU, CHIA-HUNG; CHEN, SHIH-CHANG; HSU, CHING-HSIEN; CHEN, WENGUANG; JIANG, MON-FONG Y CHUNG, YEH-CHING. Data adapter for querying and transformation between SQL and NoSQL database. *Future Generation Computer Systems* [online]. 2016, vol. 65. ISSN 0167-739X. Disponible en: <https://doi.org/10.1016/j.future.2016.02.002>.
 15. ASSEMBLER INSTITUTE OF TECHNOLOGY. ¿Qué es Postman? Características y Ventajas - Assembler Institute. [Online]. 2022. Disponible en: <https://assemblerinstitute.com/blog/que-es-postman/>.
 16. MARCO DE DESARROLLO DE LA JUNTA DE ANDALUCÍA. Introducción a JMeter: Conceptos Básicos | Marco de Desarrollo de la Junta de Andalucía. [Online]. 2024. Disponible en: <https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/388>.
 17. MEMON, PIRAH; HAFIZ, TAHSEEN; BHATTI, SANIA Y QURESHI, SAMAN SHAHID. View of COMPARATIVE STUDY OF TESTING TOOLS BLAZEMETER AND APACHE JMETER. [Online]. 2018, vol. 2, n.º 1. Disponible en: <https://journal.iba-suk.edu.pk:8089/sibajournals/index.php/sjcms/article/view/66/91>.
 18. MORALES CARRILLO, JESSICA; CEDEÑO VALAREZO, LUIS; CAJAPE BRAVO, JESÚS Y CALDERÓN ORMAZA, JONATHAN. Metodologías de desarrollo de software y su ámbito de aplicación: Una revisión sistemática. [Online]. 2022, vol. 2, n.º 1. Disponible en: <https://www.risti.xyz/issues/ristie47.pdf>.
 19. DUNKA, BAKWA; EMMANUEL, EDIM Y OYERINDE, DANTALA. Simplifying Web Application Development Using-Mean Stack Technologies. *International Journal of Latest Research in Engineering and Technology* [online]. 2018, vol. 4, n.º 1. ISSN 2454-5031. Disponible en: <http://www.ijlret.com/Papers/Vol-04-issue-01/5.B2018011.pdf>.
 20. BAJPAI, ANUSHKA; MAHESHWARI, ANKUR; SHARMA, ASHWINI Y GUPTA, GAURAV. TRANSFORMING INVENTORY MANAGEMENT SYSTEM USING MEAN STACK. *International Research Journal of Modernization in Engineering Technology and Science* www.irjmets.com @International Research Journal of Modernization in Engineering [online]. 2022, vol. 4, n.º 5. Disponible en: https://www.irjmets.com/uploadedfiles/paper/issue_5_may_2022/22439/final/fin_irjmets1652061574.pdf.
 21. HEREDIA, JAIME SAYAGO Y SAILEMA, GUSTAVO. Comparative Analysis for Web Applications Based on REST Services: MEAN Stack and Java EE Stack. *KnE Engineering* [online]. 2018, vol. 3, n.º 9. Disponible en: <https://doi.org/10.18502/Keg.v3i9.3647>.
 22. HEREDIA, SANTIAGO. Sistema web para la gestión de ventas del taller de reparación y mantenimiento automotriz Heredia utilizando el framework Mean Stack. [Online]. 2018. Disponible en: <https://n9.c1/pbssau>.
 23. SHARMA, RITU; BAJPAI, ANUSHKA; MAHESHWARI, ANKUR; SHARMA, ASHWINI Y GUPTA, GAURAV. Transforming Inventory Management System using MEAN Stack. *International Research Journal of Engineering and Technology* [online]. 2022. ISSN 2395-0056. Disponible en: <https://www.irjet.net/archives/V9/i3/IRJET-V9I3256.pdf>.
 24. NUGRAHA, DIAN Y AHMED, FALAH. MEAN stack to enhance the advancement of parking application: A narrative review. *Journal of Physics: Conference Series* [online]. 2019, vol. 1167, n.º 1. ISSN 1742-6596. Disponible en: <https://iopscience.iop.org/article/10.1088/1742-6596/1167/1/012075/pdf>.



Artículo de **libre acceso** bajo los términos de una **Licencia Creative Commons Reconocimiento – NoComercial – CompartirIgual 4.0 Internacional**. Se permite que otros remezcLEN, adapten y construyan a partir de su obra sin fines comerciales, siempre y cuando se otorgue la oportuna autoría y además licencien sus nuevas creaciones bajo los mismos términos.