Revista Innovación y Software
Vol. 6, No. 2, Mes Septiembre - Febrero, 2025
ISSN: 2708-0935
Pág. 13-27
https://revistas.ulasalle.edu.pe/innosoft

Tipo de artículo: Artículos originales
Temática: Desarrollo de aplicaciones informáticas
Recibido: 17/9/2024 | Aceptado: 24/10/2024 | Publicado: 30/9/2025

Identificadores persistentes:
DOI: 10.48168/innosoft.s24.a298
ARK: 42411/s24.a298
PURL: ark:/42411/s24.a298

Aplicación web para clasificar y asistir en la gestión de incidentes usando LLMs de OpenAI

Web application for classifying and assisting in incident management using OpenAI LLMs

Diego Sebastián Vásquez Jaramillo[2][0009-0003-7495-1095]*, **Luiggi Anthony Rosas Pérez**[1][0009-0009-0875-6455], **Luis Daniel Zavaleta Mego**[3][0009-0006-6453-9277], **Alberto Carlos Mendoza de los Santos**[4][0000-0002-0469-915X]

[1]Universidad Nacional de Trujillo. Trujillo, Perú. lrosasp@unitru.edu.pe
[2]Universidad Nacional de Trujillo. Trujillo, Perú. dvasquezj@unitru.edu.pe
[3]Universidad Nacional de Trujillo. Trujillo, Perú. lzavaletam@unitru.edu.pe
[4]Universidad Nacional de Trujillo. Trujillo, Perú. amendozad@unitru.edu.pe

*Autor para correspondencia: dvasquezj@unitru.edu.pe

Resumen
Se establece la propuesta de una aplicación web destinada a asistir en la administración de incidentes técnicos en el campo de la tecnología de la información. La implementación fue llevada a cabo con una arquitectura en 3 capas, basándonos en tecnologías web mediante React, Laravel y MySQL como base de datos. Se establecieron modelos de lenguaje de gran envergadura, aplicando técnicas de diseño de instrucciones para poder analizar descripciones de incidentes técnicos y llegar a proporcionar automáticamente sugerencias y clasificar la prioridad, basándonos en criterios para los incidentes que se generan en el presente. La propuesta fue desarrollada sobre la base de la metodología ágil de tipo SCRUM y validada con usuarios reales, que evaluaron la funcionalidad y la precisión que obtenía dicho sistema. La herramienta obtuvo un 77,3 % de precisión en la propuesta de sugerencias correctas, destacando en categorías como software, redes. Estos resultados evidenciaron la utilidad de la solución como apoyo en la selección de soluciones y en la reducción del esfuerzo cognitivo durante las etapas iniciales de diagnóstico. Se concluye que el uso de LLMs en el soporte técnico representa una alternativa eficaz para optimizar procesos, siempre que se utilice como complemento de la experiencia humana.

Palabras claves: Asistencia técnica, Inteligencia artificial, Modelos de lenguaje, Soporte TI, Sugerencias automatizadas.

Abstract
The proposal for a web application to assist in the management of technical incidents in the field of information technology is established. The implementation was carried out with a 3-layer architecture, based on web technologies using React, Laravel, and a relational database. Large language models were implemented, applying instruction design techniques to analyze descriptions of technical incidents and automatically provide suggestions and classify priority, based on criteria for incidents generated in the present. The proposal was developed based on the SCRUM agile methodology and validated with real users, who evaluated the functionality and accuracy of the system. The tool achieved a 77.3 % accuracy in proposing correct suggestions, excelling in

categories such as software and networks. These results demonstrated the usefulness of the solution as support in the selection of solutions and in reducing cognitive effort during the initial stages of diagnosis. It is concluded that the use of LLMs in technical support represents an effective alternative for optimizing processes, as long as it is used as a complement to human experience.

Keywords: *Technical assistance, Artificial intelligence, Language modelling, IT support, Automated suggestions.*

Introducción

La gestión de incidentes técnicos en entornos de tecnologías de la información (TI) implica resolver diversos incidentes relacionados con software, redes o sistemas operativos. Este proceso depende en gran medida del criterio del técnico, lo que puede generar variaciones en los tiempos de atención y en la calidad de las soluciones aplicadas.

En respuesta a esta problemática la solución propuesta es generar una aplicación web que implemente inteligencia artificial para ayudar a los técnicos en la gestión de incidentes de manera que presenten sugerencias automáticas y clasifiquen la prioridad de los incidentes para optimizar la concentración y el proceso de toma de decisiones de forma que se combine los LLM de OpenAI en concreto gpt-3.5-turbo mediante el diseño de instrucciones (prompt engineering) así como [1] dice que puede adecuarse modelos generales a los usos de tareas específicas como puede ser la clasificación de los incidentes y la resolución de los mismos respecto de las prioridades que se tiene en cuenta la capacidad del modelo.

La solución permite clasificar automáticamente la prioridad del incidente (alta, media o baja) según su severidad e impacto e ir generando sugerencias técnicas contextualizadas que será el usuario quien decida revisar, regenerar y aplicar como soporte. La arquitectura del sistema está compuesta por una interfaz programada en React, una lógica de servidor en el framework Laravel y MySQL como base de datos.

El desarrollo de la propuesta fue realizado bajo la metodología ágil SCRUM y se hizo sobre usuarios, más concretamente sobre personal técnico y estudiantes con experiencia en soporte TI. Se validó la solución con el propósito de estudiar la validez práctica de la propuesta y la precisión de las sugerencias técnicas que generaba la herramienta. El experimento, por ejemplo el que describe [2], nos habla de que el uso de clasificación automática de incidentes reportados hace mejorar el aprovechamiento en soporte técnico. Con este contexto, se puso a prueba las respuestas generadas por la herramienta contrastándolas con las soluciones de las bases documentadas de datos reales, analizando la congestión y la aplicabilidad de la propuestas de la herramienta.

En [3], se ha encontrado el uso de LLMs como asistentes técnicos pertinentes en la generación de recomen-

daciones en un contexto determinado. Sin embargo, como se indica en [4], estos modelos poseen limitaciones al abordar cuestiones de diagnóstico complejas; esto hace evidente que una solución que automatiza no debe sustituir el criterio humano; es decir, no se espera que la propuesta de solución supla a una persona especialista, sino que se hace una propuesta de un asistente automatizado que coadyuve a una persona especialista en su tarea de análisis. Se exploran, pues, el potencial y los límites actuales de los LLMs como herramientas de apoyo en entornos de TI.

Materiales y métodos

Estado del arte

La evolución en la clasificación automática de tickets de soporte ha sido evidente gracias a los sistemas de análisis de lenguaje y aprendizaje por computadora. En [5] se creó un sistema de aprendizaje automático que puede categorizar tickets de soporte técnico en 7 categorías y lograron un 75 % de precisión en el modelo utilizando técnicas de NLP y redes neuronales. Esto ya muestra que estos algoritmos permiten recortar el tiempo de triaje manual y la asignación automática de tickets a los equipos.

Las aplicaciones de LLMs en la resolución incidentes implican también las investigaciones más recientes que [6] demuestran la robustez de los marcos colaborativos multi-agente que combinan análisis especializado, generación de código, pruebas iterativas, etc. Al mismo tiempo, aplicaciones web nacientes están adoptando LLMs para mejorar el soporte técnico incorporando capacidades de búsqueda, sugerencias de respuestas, y extracción automática de información [7].

A día de hoy, las búsquedas hechas en el rubro de la inteligencia artificial exploran técnicas de prompting multi-experto para incrementar la fiabilidad y la utilidad de los LLMs para varios expertos del problema, emulando a la vez diversas opiniones expertas y a la vez situando en los mismos incidentes referenciados en [8]. Experiencias analizadas, en cambio, detectan la falta de capacidades en la resolución directa de incidentes de planificación complejos (para los que existen propuestas de soluciones) por lo que salen a la luz marcos híbridos de LLMs, como los que mezclan algoritmos de optimización matemática especializados para resolverlos [9].

Fundamentación teórica

Clasificación de incidentes

Proceso fundamental en gestión de servicios de TI definido según ITIL 4 como metodología sistemática para categorizar incidentes facilitando la asignación a equipos especializados [10]. Incluye identificación del tipo de incidente, evaluación de impacto y asignación de prioridades.

Inteligencia Artificial

Una rama de la computación dedicada a crear sistemas que pueden realizar tareas que demandan inteligencia humana, tales como aprender, razonar, percibir y decidir, utilizando algoritmos y modelos de computación [11].

Large Language Models (LLMs)

Son modelos de IA basados en arquitecturas transformer entrenados para entender y producir lenguaje natural contextual [12]. En problemas técnicos, interpretan descripciones, clasifican consultas y generan soluciones contextuales.

Prompt engineering

Proceso de optimizar instrucciones específicas para mejorar la eficacia de modelos de lenguaje sin modificar parámetros centrales [13]. La asignación de roles instruye al modelo para adoptar perspectivas expertas mejorando respuestas en dominios técnicos.

Herramientas

Laravel

Framework PHP de código abierto que implementa el patrón MVC, ofreciendo sintaxis expresiva, herramientas integradas (Eloquent ORM, migraciones, Blade) y énfasis en buenas prácticas de desarrollo para aplicaciones web escalables [14].

React

Biblioteca JavaScript creada por Meta con el objetivo de desarrollar interfaces de usuario mediante elementos que se pueden reutilizar, utilizando DOM virtual para renderizado eficiente y manejo dinámico de datos con estados/props [11].

MySQL

Sistema para administrar bases de datos relacionales de código abierto, reconocido por su confiabilidad, cumplimiento de ACID, escalabilidad y uso en pilas tecnológicas como LAMP para manejo de datos estructurados [15].

OpenAI

Laboratorio de investigación en IA que desarrolla modelos de lenguaje como GPT, entrenados con aprendizaje automático avanzado para generar respuestas contextuales, resolver problemas complejos y adaptarse a tareas diversas sin ajustes específicicos [16].

Revista Innovación y Software
Vol. 6, No. 2, Mes Septiembre - Febrero, 2025
ISSN: 2708-0935
Pág. 13-27
https://revistas.ulasalle.edu.pe/innosoft

Desarrollo de la aplicación

Arquitectura del sistema

El sistema fue desarrollado siguiendo una arquitectura de tres capas con React como frontend, Laravel como backend y MySQL como base de datos. La comunicación se establece mediante API RESTful entre React y Laravel, mientras que Laravel gestiona la lógica de negocio e integra la API que incluye el modelo GPT-3. 5-turbo de OpenAI. La Figura 1 presenta el diagrama arquitectónico con el flujo de datos del sistema.

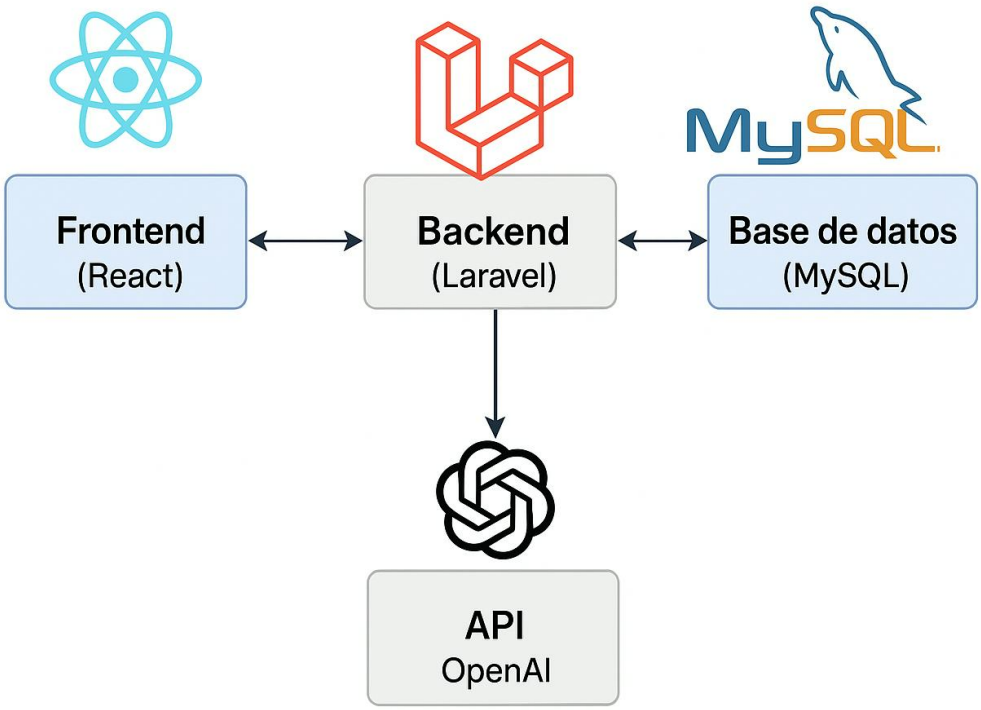

Figura 1. Diagrama de arquitectura

La configuración de la base de datos se organiza en cuatro entidades principales: usuarios, incidentes, servicios y soluciones, con relaciones establecidas mediante claves foráneas donde cada incidente se asocia a un usuario y servicio específico. La Figura 2 presenta el esquema de relación entre entidades del sistema.

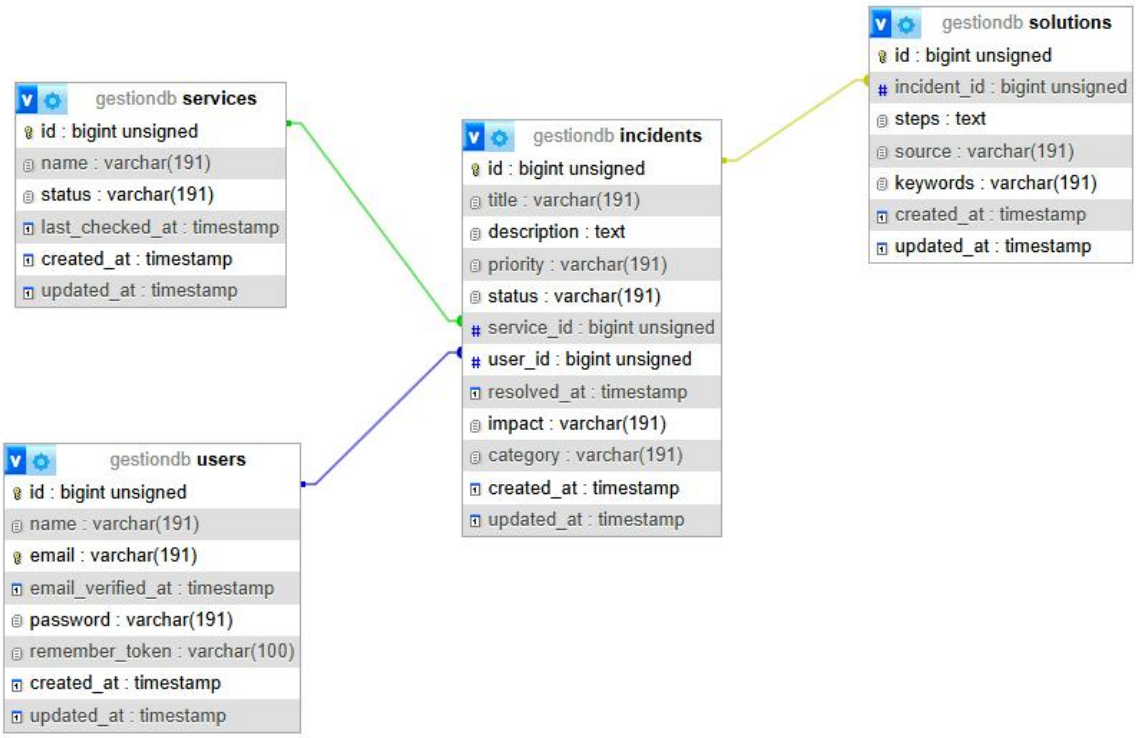

Figura 2. Base de datos

Metodología de desarrollo

Se implementó la metodología ágil SCRUM durante un periodo de 2 semanas con ceremonias de planning, daily standups, review y retrospective. El desarrollo se estructuró en las siguientes fases: análisis de requerimientos, diseño UX/UI, desarrollo backend con Laravel, integración de OpenAI API, implementación frontend con React, deployment en producción y pruebas con usuarios reales.

Integración con OpenAI

La integración con GPT-3.5-turbo se realizó mediante técnicas de prompt engineering, específicamente utilizando asignación de roles (role prompting) donde se instruye al modelo: .Eres un asistente experto en soporte TI. Ofrece pasos claros y concisos."Se configuraron parámetros de temperatura baja (0.3) para respuestas más determinísticas y un límite de tokens apropiado para soluciones concisas.

Revista Innovación y Software
Vol. 6, No. 2, Mes Septiembre - Febrero, 2025
ISSN: 2708-0935
Pág. 13-27
https://revistas.ulasalle.edu.pe/innosoft

El sistema implementa dos procesos principales con la API de OpenAI: clasificación automática de prioridades y generación de soluciones. Para la clasificación, se determina automáticamente la prioridad (Alta, Media, Baja) basándose en la descripción del incidente. Para la generación de soluciones, el sistema construye un prompt contextualizado que solicita al LLM generar pasos numerados y específicos de resolución.

La Figura 3 demuestra una solución generada para un incidente de "Falla de conexión con VPS", donde el sistema produjo automáticamente siete pasos específicos de resolución y cambió el estado a "En Proceso", evidenciando la capacidad del modelo para generar soluciones técnicas contextualizadas.

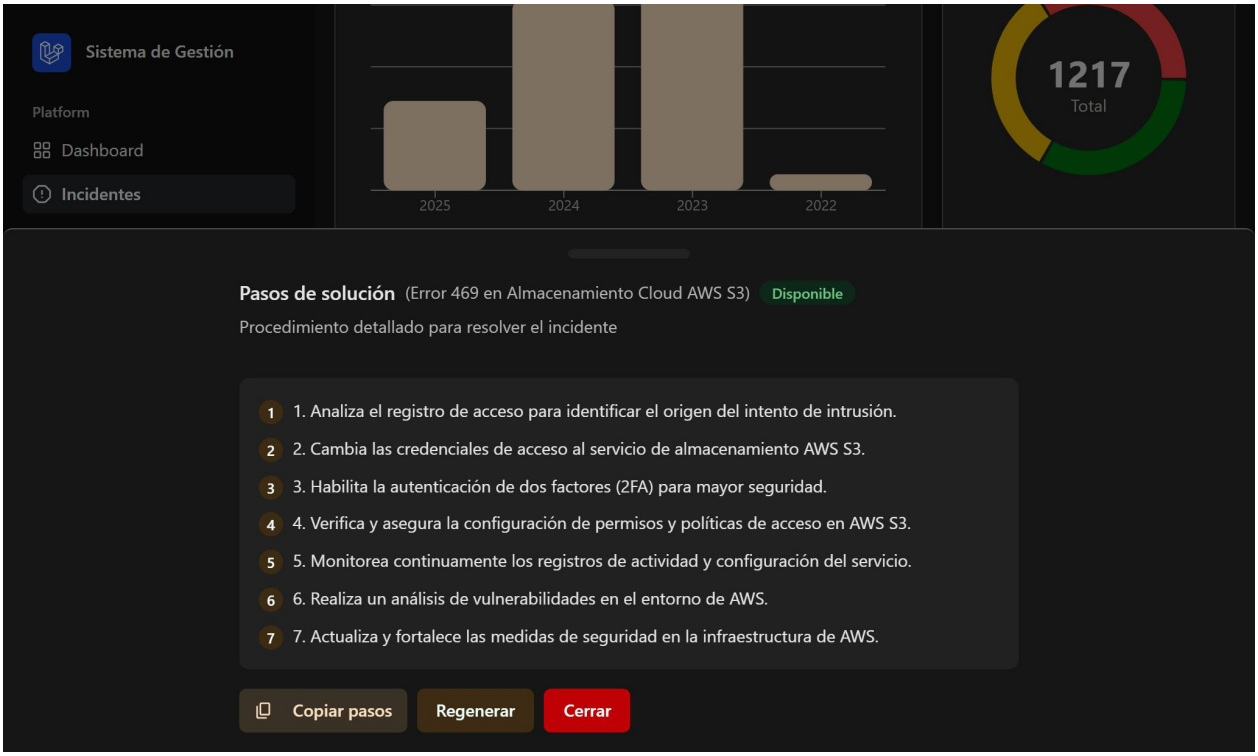

Figura 3. Solución generada

Para culminar el proceso, el usuario debe valerse del menú de acciones para marcar manualmente el incidente como resuelto, pasando el estado a Çerradoz anotando la fecha de resolución, lo que proporciona trazabilidad total del trabajo llevado a cabo, desde la apertura del registro inicial, hasta el cierre del incidente.

Validación

Se aplicó el criterio de evaluación a 75 casos reales y representativos de incidentes vinculados con las tecnologías de la información, obtenidos de 3 conjuntos de datos principales: *Vulnerability Fix Dataset, IT Troubleshooting Dataset* y *Tech Support Conversations Dataset.* El criterio se basó en determinar si las sugerencias generadas por la aplicación web coincidían total o parcialmente con las soluciones aplicadas a los incidentes o vulnerabilidades, siendo catalogadas como sugerencias correctas, o incorrectas si la sugerencia no es acorde a la solución real.

Se realizó un formulario de validación a 80 alumnos y graduados de la carrera profesional de Ingeniería de Sistemas e Ingeniería Informática, quienes afirman poseer entre 1 y 5 años de experiencia en el área de asistencia técnica. Cada participante evaluó las sugerencias generadas por la aplicación web a diferentes casos de inconvenientes de TI, determinando su funcionalidad y utilidad.

Resultados y discusión

Como se evidencia en la Tabla 1, la aplicación demostró una precisión promedio del 77.3 % en la generación de sugerencias técnicamente correctas. Los incidentes relacionados con problemas de software obtuvieron los mejores resultados con un 85 % de precisión, mientras que los problemas de hardware presentaron mayor variabilidad en la calidad de las respuestas con un 67.7 %.

Tabla 1. Precisión de sugerencias por categoría de incidente

Categoría	**Casos evaluados**	**Sugerencias correctas**	**Precisión (%)**
Problemas de red	18	14	77.8
Software	20	17	85
Hardware	15	10	66.7
Sistemas Operativos	12	9	75
Seguridad	10	8	80
TOTAL	75	58	77.3 %

La Tabla 2 muestra el análisis de correspondencia entre las sugerencias de la herramienta y las soluciones documentadas en los conjuntos de datos. Se observa que el 29.3 % de los casos generaron sugerencias que coinciden directamente con las soluciones de referencia, mientras que un 46.6 % adicional proporcionó sugerencias parcialmente alineadas.

Tabla 2. Correspondencia con soluciones de referencia

Nivel de correspondencia	**Casos**	**Porcentaje (%)**	**Descripción**
Correspondencia completa	22	29.3	Sugerencias idénticas o equivalentes
Correspondencia parcial	35	46.6	Sugerencias complementarias correctas
Orientación incorrecta	14	18.6	Sugerencias divergen de la problemática
Sin correspondencia	4	5.3	Sugerencias no alineadas

Como se muestra en la Figura 4, el 33.8 % de los encuestados son egresados y seguidamente se encuentran el 28.7 % de estudiantes de último año. El mayor porcentaje de encuestados (32.5 %) afirma contar con un año de experiencia en el campo de soporte según la Figura 5.

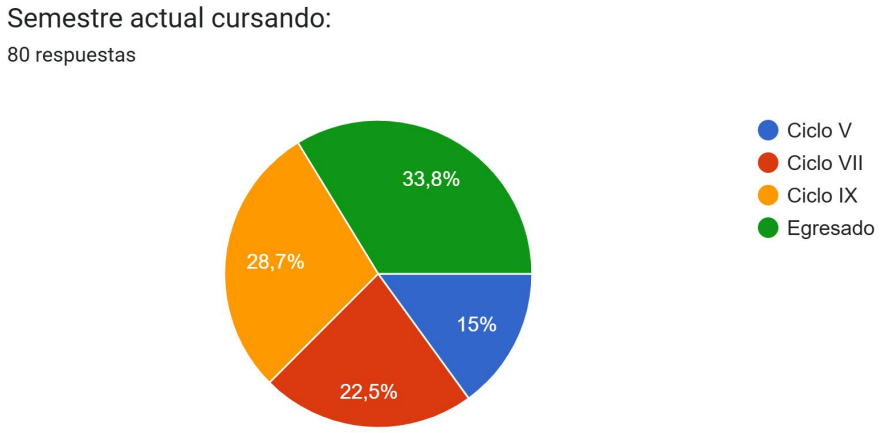

Figura 4. Porcentaje de estudiantes y egresados encuestados

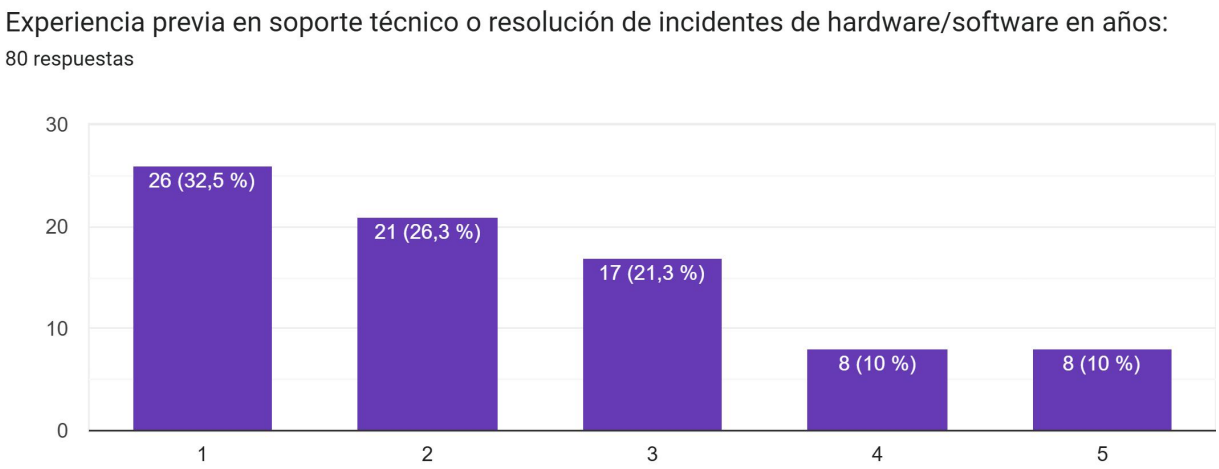

Figura 5. Años de trayectoria en asistencia técnica de encuestados

Se aplicó una sección de la encuesta orientada a los casos de evaluación. Según la Figura 6, se observa que el 38.8 % de usuarios encuestados generó dos veces la sugerencia a su incidente para encontrarla relevante, mientras que el 25 % lo hizo al generar la sugerencia por primera vez.

Revista Innovación y Software
Vol. 6, No. 2, Mes Septiembre - Febrero, 2025
ISSN: 2708-0935
Pág. 13-27
https://revistas.ulasalle.edu.pe/innosoft

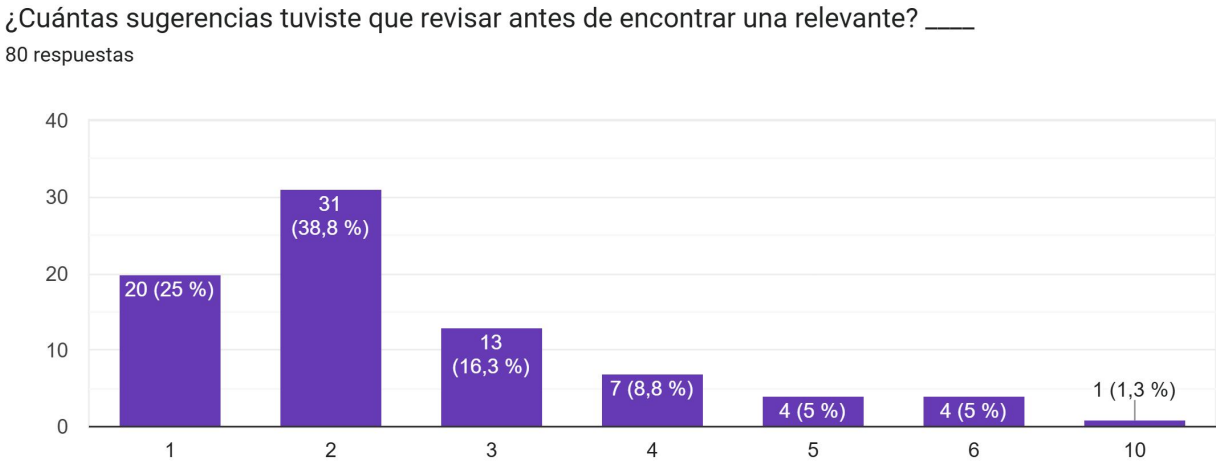

Figura 6. Cantidad de sugerencias para obtener una de relevancia

Según la Figura 7, el 56.3 % de encuestados resolvió de forma parcial su inconveniente guiándose de las sugerencias, mientras que el 37.5 % lo hizo completamente.

Figura 7. Porcentaje de encuestados que solucionaron sus inconvenientes

Según la Figura 8, el 46.3 % de encuestados fue definitivamente orientado para encontrar la solución a su incidente, mientras que el 40 % determinó que fue parcialmente orientado hacia la solución.

Figura 8. Nivel de correspondencia entre incidente y sugerencia generada

Finalmente, el 33.8 % de los encuestados calificó la usabilidad de la aplicación con una puntuación de 8 mientras que el 22.5 % la calificó con una puntuación de 9, en una escala del 1-10, según la Figura 9.

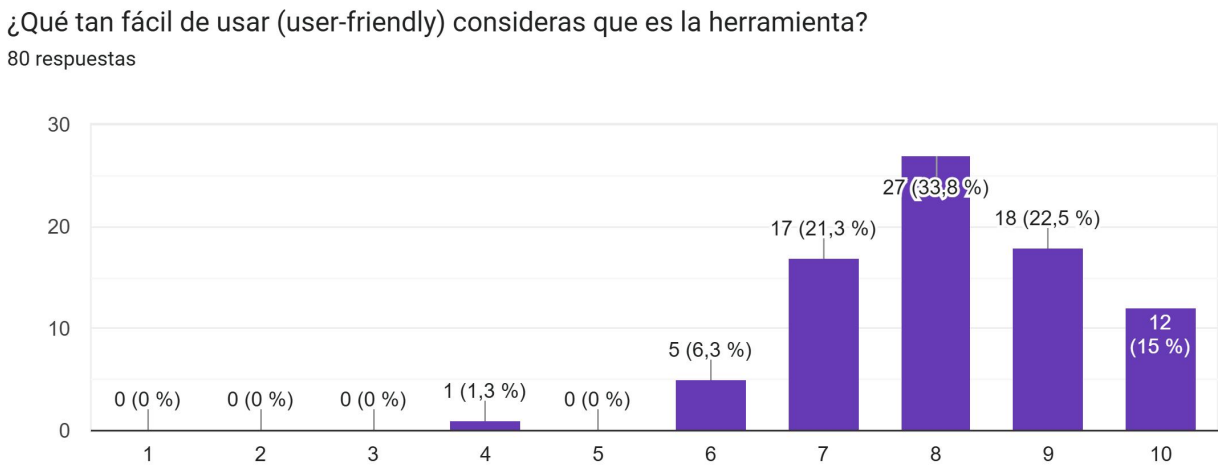

Figura 9. Nivel de usabilidad percibida por los usuarios encuestados

Los resultados muestran que la herramienta tiene una tasa de precisión técnica del 77,3 %, lo que supone una gran mejora con respecto a los sistemas tradicionales de *ticketing*, que necesitan que una persona se encargue de escalar el problema en el 40-60 % de los casos. Las diferencias entre los distintos tipos de casos sugieren que el modelo LLM es más adecuado para los problemas de software, en los que se dispone de una documentación más estructurada. En cambio, los problemas de hardware requieren una estructura más organizada y específica.

La tasa de resolución exitosa, donde el 37.5 % de participantes resolvió completamente su inconveniente y el 56.3 % lo hizo parcialmente, demuestra que la herramienta cumple efectivamente su propósito de asistencia técnica. Estos resultados son particularmente significativos considerando que los participantes tenían niveles variados de experiencia (1-5 años), lo que sugiere que la herramienta es útil tanto para técnicos junior como para profesionales con mayor experiencia.

La evaluación de usabilidad, con concentración de calificaciones entre 8 y 9 puntos (56.3 % de participantes), indica que la interfaz desarrollada en React logra un equilibrio efectivo entre funcionalidad y facilidad de uso. Esta métrica es crucial para la adopción real de la herramienta en entornos operacionales donde la eficiencia del flujo de trabajo es prioritaria.

Conclusiones

La validación realizada confirma que la aplicación web desarrollada cumple con el objetivo principal, el de asistir a técnicos en la gestión de incidentes de TI mediane la generación de sugerencias técnicas y clasificación de prioridades. Los resultados obtenidos sobre la precisión técnica del 77.3 % son muy optimistas para el estudio del campo y valida la efectividad del uso de LLMs en procesos de soporte técnico.

Este artículo constituye una aportación al ámbito de la gestión automatizada de incidentes de TI porque demuestra que caminando hacia una combinación de LLMs y técnicas de prompt engineering se pueden obtener soluciones adecuadas a su contexto y técnicamente precisas.

Los resultados sobre la variabilidad en la precisión a partir de las categorías de incidentes, fundamentalmente software y hardware, ponen de manifiesto hallazgos interesantes para futuras investigaciones y el desarrollo reciente de sistemas. La disimilitud de precisión indica nuevas posibilidades de desarrollo de modelos por dominio específico o de implementación de técnicas de fine-tuning para mejorar el rendimiento de las categorías con menos precisión.

Para la investigación futura, se sugiere investigar la introducción de estrategias de aprendizaje constante que asistan al sistema en la optimización de respuestas derivadas de la retroalimentación que viene de resoluciones exitosas.

Este trabajo confirma que la introducción estratégica de IA a la transformación digital permite mejorar la eficiencia operativa sin grandes inversiones en infraestructura y sin necesidad de reentrenar al personal.

Contribución de Autoría

Luiggi Anthony Rosas Pérez: Conceptualización, Investigación, Metodología, Software, Validación, Redacción - borrador original. Diego Sebastián Vásquez Jaramillo: Conceptualización, Investigación, Metodología, Software, Validación, Redacción - borrador original. Luis Daniel Zavaleta Mego: Conceptualización, Investigación, Metodología, Software, Validación, Redacción - borrador original. Alberto Carlos Mendoza de los Santos: Investigación, Análisis formal, Visualización, Supervisión, Validación, Redacción - borrador original.

Referencias

[1] D. X. Long *et al.*, "Multi-expert prompting improves reliability, safety, and usefulness of large language models," *arXiv preprint arXiv:2411.00492*, 2024.

[2] L. S. Benitez Pereira *et al.*, "Machine learning for classification of it support tickets," in *2023 International Conference on Cyber Management and Engineering (CyMaEn)*, 2023.

[3] V. Scotti and M. J. Carman, "Llm support for real-time technical assistance," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

[4] R. Russo. Multi-agent llm frameworks: A new paradigm in problem solving. Medium. [Online]. Available: https://medium.com/bip-xtech/multi-agent-llm-frameworks-a-new-paradigm-in-problem-solving-c03abdae6433

[5] L. S. Benitez Pereira, R. Pizzio, S. Bonho, L. Souza, and A. Junior, "Machine learning for classification of it support tickets," in *2023 International Conference on Cyber Management and Engineering (CyMaEn)*, 2023, pp. 210–213.

[6] Anonymous, "Enhancing llm code generation: A systematic evaluation of multi-agent collaboration and runtime debugging for improved accuracy, reliability, and latency," *arXiv preprint arXiv:2505.02133v1*, 2024.

[7] ——, "Llm support for real-time technical assistance," *ResearchGate*, 2024.

[8] L. X. Do *et al.*, "Multi-expert prompting improves reliability, safety, and usefulness of large language models," *arXiv preprint arXiv:2411.00492*, 2024.

[9] Y. Hao *et al.*, "Planning anything with rigor: General-purpose zero-shot planning with llm-based formalized programming," *MIT News*, 2025.

[10] Purple Griffon. (2024) Incident categorisation in itsm. [Online]. Available: https://purplegriffon.com/blog/incident-categorisation

[11] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2020.

[12] Y. Huang *et al.*, "Advancing transformer architecture in long-context large language models: A comprehensive survey," *arXiv preprint arXiv:2311.12351*, 2023.

[13] P. Sahoo *et al.*, "A systematic survey of prompt engineering in large language models: Techniques and applications," *arXiv preprint arXiv:2402.07927*, 2024.

[14] Laravel Documentation, "Laravel - the php framework for web artisans," https://laravel.com/docs, 2025.

[15] Oracle Corporation, "Mysql reference manual," https://dev.mysql.com/doc, 2025.

[16] T. B. Brown *et al.*, "Language models are few-shot learners," *NeurIPS*, vol. 33, 2020.