

UNIVERSIDAD CATÓLICA SANTO TORIBIO DE MOGROVEJO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN



Sistema web basado en algoritmo predictivo de machine learning para apoyar el proceso de mantenimiento de vehículos de transporte pesado

**TESIS PARA OPTAR EL TÍTULO DE
INGENIERO DE SISTEMAS Y COMPUTACIÓN**

AUTOR

Segundo Pedro Tenazoa Shupingahua

ASESOR

Hector Miguel Zelada Valdivieso

<https://orcid.org/0000-0002-2311-4284>

Chiclayo, 2025

**Sistema web basado en algoritmo predictivo de machine learning
para apoyar el proceso de mantenimiento de vehículos de transporte
pesado**

PRESENTADA POR

Segundo Pedro Tenazoa Shupingahua

A la Facultad de Ingeniería de la
Universidad Católica Santo Toribio de Mogrovejo
para optar el título de

INGENIERO DE SISTEMAS Y COMPUTACIÓN

APROBADA POR

María Ysabel Aranguri Garcia

PRESIDENTE

Miguel Orlando Diaz Vidarte

SECRETARIO

Hector Miguel Zelada Valdivieso

VOCAL

Dedicatoria

Dedicado a Dios por guiar y permitirme culminar con éxito este trabajo.
A mi familia que, con su amor, esfuerzo y apoyo incondicional, que me ayudaron a perseguir
mis sueños y metas.
Finalmente, dedico este trabajo a todos aquellos que creen en el poder de la perseverancia y el
conocimiento, esperando que sea una fuente de inspiración para futuros profesionales en el
campo de la ingeniería de sistemas y computación.

Agradecimientos

A mi asesor, Héctor Miguel Zelada Valdivieso, por su valiosa orientación y apoyo en el
desarrollo de esta tesis.

A mis amigos y compañeros de estudio, por su aliento constante y su compañía.

Finalmente, agradezco a la empresa de transportes de Lambayeque, por brindarme los
recursos de información para culminar mi formación académica y profesional.

Sistema web basado en algoritmo predictivo de machine learning para apoyar el proceso de mantenimiento de vehículos de transporte pesado

INFORME DE ORIGINALIDAD

7%	7%	2%	3%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	tesis.usat.edu.pe Fuente de Internet	2%
2	hdl.handle.net Fuente de Internet	2%
3	openaccess.uoc.edu Fuente de Internet	<1%
4	Terezinha Maria Bogéa Gusmão, Willian Carboni Viana. "Abordagens sobre ensino-aprendizagem e formação de professores", Editora Científica Digital, 2023 Publicación	<1%
5	Submitted to UISEK Trabajo del estudiante	<1%
6	Submitted to Universidad Nacional Abierta y a Distancia, UNAD,UNAD Trabajo del estudiante	<1%
7	www.coursehero.com Fuente de Internet	<1%
8	www.pagero.com Fuente de Internet	<1%
9	dspace.unach.edu.ec Fuente de Internet	<1%
10	www.aspentech.com Fuente de Internet	<1%

Índice

Resumen	6
Abstract	7
Introducción	8
Revisión de literatura	10
Materiales y métodos	15
Resultados y discusión	16
Conclusiones	32
Recomendaciones.....	33
Referencias.....	34
Anexos.....	37

Resumen

El estudio tuvo como objetivo principal implementar un sistema web basado en algoritmos predictivos de machine learning para optimizar el mantenimiento de vehículos de transporte pesado. También, se establecieron objetivos específicos, como seleccionar los algoritmos capaces de predecir datos numéricos sobre problemas de mantenimiento en la empresa; luego, se identificaron las herramientas tecnológicas que soportan las técnicas de machine learning necesarias para desarrollar la arquitectura del sistema web predictivo. A continuación, se evaluó la exactitud y validez del algoritmo predictivo para confirmar la fiabilidad de la información obtenida. Finalmente, se examinó la calidad del software en función de su usabilidad y funcionalidad. En el desarrollo del sistema, se adoptaron las metodologías CRISP-DM y marco de trabajo Scrum; la primera optimizó la minería de datos enfocada en mantenimiento vehicular y proporcionó una estructura adecuada desde la comprensión del negocio hasta la evaluación del modelo, mientras que Scrum organizó el trabajo en sprints para cada actividad, asegurando una implementación ágil durante el proceso, se integraron herramientas tecnológicas como Django, PostgreSQL y Python, utilizando módulos JSON y la biblioteca Scikit-learn para calcular la precisión del modelo. Al finalizar el desarrollo, la calidad del software fue evaluada mediante pruebas de caja negra y caja blanca. Los resultados algoritmo árbol de decisión alcanzó una exactitud de 99.5%, mientras que redes neuronales alcanzó una exactitud de 73.3%. En conclusión, el sistema web demostró ser un recurso significativo para el mantenimiento de vehículos de transporte pesado, proporcionando datos precisos y útiles para la toma de decisiones.

Palabras clave: Mantenimiento de vehículos, machine learning, CRISP-DM, redes neuronales, árbol de decisión.

Abstract

The main objective of the study was to implement a web-based system based on predictive machine learning algorithms to optimize heavy-duty vehicle maintenance. Specific objectives were also established, such as selecting algorithms capable of predicting numerical data on maintenance issues in the company; then, the technological tools supporting the machine learning techniques necessary to develop the architecture of the predictive web system were identified. The accuracy and validity of the predictive algorithm were then evaluated to confirm the reliability of the information obtained. Finally, the quality of the software was examined based on its usability and functionality. The CRISP-DM methodologies and the Scrum framework were adopted in the development of the system. The first optimized data mining focused on vehicle maintenance and provided an adequate structure from business understanding to model evaluation. Scrum organized the work into sprints for each activity, ensuring agile implementation throughout the process. Technological tools such as Django, PostgreSQL, and Python were integrated, using JSON modules and the Scikit-learn library to calculate model accuracy. At the end of development, the software quality was assessed using black-box and white-box testing. The decision tree algorithm results achieved an accuracy of 99.5%, while the neural networks achieved an accuracy of 73.3%. In conclusion, the web-based system proved to be a significant resource for heavy-duty vehicle maintenance, providing accurate and useful data for decision-making.

Keywords: Vehicle maintenance, machine learning, CRISP-DM, neural networks, decision tree.

Introducción

La falta de mantenimiento vehicular en México es un desafío significativo que aumenta el riesgo de accidentes en un 30%, los costos operativos en un 50% y el consumo de combustible en un 20%. Además, acelera el desgaste de piezas críticas, reduciendo la vida útil del vehículo en un 25%. Estos hallazgos resaltan la necesidad urgente de implementar programas de mantenimiento rigurosos para garantizar la operatividad y rentabilidad del sector del transporte vehicular [1].

A nivel mundial, las empresas de transporte de mercancías enfrentan desafíos en la gestión de la flota vehicular, con un 80% de los vehículos experimentando fallas de mantenimiento, divididas en mecánicas, digitales y de mantenimiento preventivo. Problemas como fallos en el motor y la transmisión surgen en solo 3 años de funcionamiento [2]. Aunque los vehículos modernos recopilan datos valiosos a través de sensores, su subutilización conduce a una menor rentabilidad debido a la falta de conocimiento sobre cómo aprovechar esta información de manera efectiva. Esto impide la implementación de procedimientos de mantenimiento digital adecuados [3].

En España, el 60% de las empresas no realizan un mantenimiento preventivo adecuado en sus vehículos, a pesar de contar con sensores que proporcionan información valiosa. Esta falta de mantenimiento resulta en un aumento significativo de fallas en los vehículos en tan solo 10 meses, representando un riesgo para la seguridad y la eficiencia operativa, además de costos adicionales en reparaciones y pérdida de productividad [4].

De acuerdo con un estudio realizado a 268 empresas europeas de Bélgica, Holanda y Alemania, dio como resultado que un software de gestión de mantenimiento disminuye los costos por reparación en un 12%, mejora la disponibilidad de los equipos en 9% y alarga la vida útil de los equipos en 20%. [5].

Del mismo modo en Perú, [6] en octubre, el mantenimiento de automóviles aumentó un 1,57% en comparación con septiembre de 2023. Sin embargo, varios departamentos han experimentado fallas en vehículos durante los dos primeros años debido a la falta de información para un mantenimiento adecuado. Por ejemplo, en Amazonas, el costo promedio de mantenimiento fue del 30%; en Ancash, del 40%; en Apurímac, del 20%; en Arequipa, del 24%; en Ayacucho, del 38%; en Cajamarca, del 25%; en Cusco, del 35%; en Lima, del 31%; y en Lambayeque, del 31%. Estos datos destacan la importancia de un mantenimiento adecuado para prevenir futuras fallas mecánicas en los vehículos.

El parque automotor del Perú sigue creciendo, con la reparación y mantenimiento de vehículos como la principal demanda de servicios. El sector experimentó un crecimiento del

5.7% en junio de 2022. Sin embargo, la Asociación Automotriz del Perú prevé una disminución en la tasa de crecimiento en los próximos meses, con una posible caída por debajo del 3% hacia finales del año [7].

En la empresa de transporte y logística de Lambayeque, la falta de mantenimiento oportuno ha causado fallos en 100 de sus unidades, con múltiples fallos por semestre. Esto afecta la entrega puntual de encomiendas, a pesar de que el 98.62% de las entregas se realizaban a tiempo, se veían afectadas por mantenimientos no programados. Entre enero y julio de 2022, la empresa gastó S/ 29,362.22 en reparaciones externas y S/ 77,685.79 en reparaciones internas, evidenciando una falta de planificación de mantenimiento. El Diagrama de Pareto mostró que el 20% de las causas identificadas eran responsables del 75% de los problemas relacionados con la toma de decisiones ineficiente.

Debido a esta situación, se formuló el siguiente problema de investigación, ¿Cómo desarrollar un sistema web basado en algoritmos predictivos de machine learning para apoyar el proceso de mantenimiento de vehículos en una empresa de servicios logísticos de Lambayeque?

El enfoque científico del artículo se justifica al resolver problemas reales mediante el método científico. Se emplearon técnicas avanzadas de búsqueda y análisis para revisar modelos predictivos utilizados por empresas de transporte de renombre internacional. Esta revisión permitió seleccionar y desarrollar un algoritmo predictivo específico a las condiciones y necesidades del estudio. El desarrollo del modelo no solo sirve como referencia para futuras investigaciones, sino que también mejora la toma de decisiones en empresas de transporte, proporcionando una base sólida para el desarrollo de soluciones más eficientes y efectivas.

Desde una perspectiva económica, la investigación propone una significativa reducción en los costos asociados con mantenimientos correctivos, tiempos de inactividad no planificados y el uso de recursos humanos adicionales. La implementación de un sistema web capaz de predecir fallos en los vehículos tiene un impacto positivo en la rentabilidad de las operaciones al permitir la realización de mantenimientos en momentos más oportunos y programados. Esto reduce los costos operativos y maximiza el uso de los recursos disponibles, mejorando la eficiencia económica de las empresas de transporte.

En términos sociales, la investigación ofrece una herramienta informática que facilita la toma de decisiones para el personal operativo, permitiéndoles dedicar más tiempo a otras tareas críticas y apoya su desempeño laboral. Los conductores se benefician al recibir información en tiempo real sobre el estado de los vehículos y las fechas de mantenimiento, mejorando su capacidad de planificación y operación eficiente. Esto no solo incrementa la satisfacción y la

seguridad de los conductores, sino que también contribuye a un entorno de trabajo más organizado y menos estresante.

Desde el ángulo tecnológico, la investigación mejora la planificación del mantenimiento vehicular aprovechando la digitalización de datos. Se desarrolló un algoritmo predictivo de fallas utilizando algoritmos de machine learning basados en parámetros clave como velocidad, consumo de combustible, kilómetros recorridos y temperatura de trabajo de los vehículos pesados. Este algoritmo predictivo se integra en un sistema web que ayuda al jefe de mantenimiento a tomar decisiones más informadas mediante alarmas que alertan sobre posibles fallos inminentes. La combinación de machine learning con un sistema web permite una gestión proactiva del mantenimiento, optimizando los tiempos de intervención y reduciendo las incidencias operativas.

Debido a lo anteriormente mencionado, se formuló el siguiente objetivo general, Implementar un sistema web basado en un algoritmo predictivo de machine learning para apoyar el proceso de mantenimiento de vehículos de transporte pesado. Para cumplir el objetivo general se plantearon los siguientes objetivos específicos: Determinar el algoritmo que permita predecir los datos numéricos sobre los problemas de mantenimiento de vehículos de transporte pesado en la empresa, seleccionar herramientas tecnológicas que soporten la implementación de las técnicas de machine learning necesarias para el desarrollo del sistema web, evaluar la exactitud y validez del algoritmo predictivo para contrastar que la información obtenida es fiable, finalmente se evaluó la calidad del software en base a la verificación de la usabilidad y funcionalidad del sistema web predictivo.

Revisión de literatura

Ruiz [8] llevó a cabo un estudio en una empresa de transportes y descubrió que las fallas mecánicas en los vehículos eran la causa principal de la incapacidad para realizar los servicios de transporte a tiempo, lo que a su vez llevó al pago de penalizaciones por los retrasos ocasionados. Para abordar este problema, desarrolló un algoritmo predictivo utilizando machine learning en Matlab R2018a, con el objetivo de mejorar la disponibilidad de los buses. El estudio se centró en una muestra de 7 buses y empleó la metodología CRISP-DM para el desarrollo del algoritmo predictivo. Los resultados mostraron una reducción significativa en el tiempo DE identificación de fallos aun 10.25%, y en el tiempo medio de reparación, de un 78.81% a un 11.79%. Esto demuestra que el uso de machine learning tiene un impacto significativamente positivo en la disponibilidad de la flota vehicular. El estudio es equivalente, ya que se desarrolló un algoritmo predictivo utilizando machine learning en Matlab R2018a, específicamente para

mejorar la disponibilidad de los buses. Aplicando la metodología CRISP-DM, analizó una muestra de 7 buses, y los resultados demostraron una reducción significativa tanto en el tiempo medio entre fallos como en el tiempo medio de reparación.

Vergara [9] realizó un estudio en una empresa de transportes que evidenció la necesidad de mejorar la gestión de información, ya que los reportes de mantenimiento se emitían en archivos Excel, sin analizar el histórico y la información no se presentaba en el momento oportuno, resultando en tareas de mantenimiento realizadas de manera empírica. Como solución alternativa, propuso desarrollar una solución de Business Intelligence (BI) utilizando un modelo de Machine Learning. La arquitectura de esta solución incluyó la alimentación de datos desde un archivo Excel, que posteriormente se almacenaban en Data Stage SQL Server y se enviaban a Data Mart SQL para su representación en reportes. Esta implementación permitió reducir el costo de combustible del 93% al 91%, el costo de mantenimiento del 97% al 94% y un incremento en el rendimiento del 37.2% al 37.39%. En conclusión, se demostró que la solución implementada utilizando machine learning mejora significativamente la gestión vehicular, basándose en la información de trabajo de la flota vehicular. El estudio es equivalente, ya que se desarrolló un sistema de Business Intelligence (BI) que incorpora un modelo de Machine Learning para analizar y procesar datos desde archivos Excel hasta SQL Server y Data Mart SQL, facilitando la generación de reportes efectivos.

Rondón [10] en el estudio aborda problemas críticos en la gestión de flotas vehiculares, centrados en la significativa pérdida de tiempo y la frecuente inactividad de los vehículos debido a deficiencias en el mantenimiento. El objetivo principal del estudio fue evaluar el impacto de la implementación de técnicas de aprendizaje automático para mejorar la disponibilidad y confiabilidad de una flota de vehículos pesados. La investigación se llevó a cabo con un diseño pre experimental de tipo aplicado y un enfoque cuantitativo. Se analizaron cuatro tipos de maquinaria pesada: un cargador frontal, un minicargador, una retroexcavadora y una excavadora. Para el análisis se emplearon diversos modelos de Machine Learning, incluyendo Máquinas de Vectores de Soporte (SVM), Árboles de Decisión y Redes Neuronales. Los datos recolectados fueron procesados utilizando Excel, y los resultados mostraron mejoras significativas en el cargador frontal experimentó un aumento del 24.95% en disponibilidad y del 36.7% en confiabilidad. Incrementos similares se observaron en las otras maquinarias. A partir de estos resultados, se concluyó que el Machine Learning es efectivo para apoyar el mantenimiento de la maquinaria pesada, ofreciendo ventajas considerables para la empresa y mostrando potencial para aplicaciones en otros sectores similares. El estudio fue equivalente en

metodología, ya que también implementaron algoritmos de machine learning; sin embargo, procesaron datos diferentes.

Bases teóricas

Un sistema web es una aplicación accesible a través de un navegador conectado a Internet o a un servidor web, programada en lenguajes compatibles con los navegadores [11]. En el contexto de este estudio, un sistema web que utiliza algoritmos predictivos de machine learning proporciona una plataforma eficiente y en tiempo real para recopilar y analizar datos sobre el mantenimiento de vehículos de transporte pesado. Este sistema está diseñado para apoyar el proceso de mantenimiento, permitiendo a las empresas de transporte acceder y utilizar una herramienta avanzada desde cualquier navegador. La implementación de sistemas de mantenimiento predictivo mediante machine learning ofrece varias ventajas, como mejora en la predictibilidad del mantenimiento, optimización de recursos, aumento de la vida útil de los vehículos, mejora de la seguridad, análisis de datos avanzados y mayor satisfacción del cliente. Sin embargo, también presenta desventajas significativas, como alto costo inicial, complejidad técnica, dependencia de la calidad de los datos, problemas de privacidad y seguridad, resistencia al cambio y dependencia tecnológica, lo que puede complicar su adopción y operación efectiva.

El desarrollo del sistema web para el mantenimiento de vehículos pesados empleó Python, JavaScript, PHP y PostgreSQL. Python facilitó el desarrollo del algoritmo predictivo y la automatización del backend, gracias a bibliotecas como Pandas y Scikit-learn. JavaScript se utilizó para crear una interfaz de usuario dinámica, mientras que PHP gestionó la lógica de aplicación y la interacción con la base de datos en el backend del sistema. Se emplearon frameworks como React o Angular para desarrollar una aplicación de página única (SPA) eficiente [12]. En el desarrollo del sistema web para el mantenimiento predictivo de vehículos pesados, se utilizó PostgreSQL como sistema de gestión de bases de datos. Reconocido por su robustez y escalabilidad, PostgreSQL ofrece características como soporte para transacciones ACID, búsqueda de texto completo y tipos de datos JSON y geoespaciales. Su arquitectura escalable y comunidad activa lo hacen ideal para aplicaciones complejas. PostgreSQL se eligió para almacenar datos del algoritmo predictivo y del historial de mantenimiento, proporcionando una base sólida y flexible para el desarrollo e implementación del sistema web, mejorando la eficiencia operativa y reduciendo los costos asociados con el mantenimiento de la flota vehicular [13], [14], [15]. Pandas y Scikit-learn son fundamentales en el desarrollo de aplicaciones de análisis de datos y machine learning en Python. Pandas facilita la manipulación y análisis de datos tabulares, siendo esencial en la limpieza y preparación de grandes conjuntos

de datos de mantenimiento vehicular. Por otro lado, Scikit-learn ofrece una amplia variedad de algoritmos de aprendizaje supervisado y no supervisado, junto con herramientas para selección de modelos y evaluación, lo que permitió desarrollar y validar el algoritmo predictivo central del proyecto. Estas bibliotecas, utilizadas en IDEs como PyCharm y Jupyter Notebooks, contribuyeron significativamente al desarrollo de un sistema web eficiente para el mantenimiento predictivo en vehículos de transporte pesado, mejorando la eficiencia operativa y reduciendo costos [16]. Django, un framework de alto nivel para el desarrollo web en Python, se basa en el patrón de diseño "modelo-vista-template" (MVT), facilitando la creación de sitios web seguros y fáciles de mantener al separar la lógica de negocio de la interfaz de usuario. Con características como un ORM para abstraer consultas a la base de datos, sistema de autenticación y herramientas preconstruidas, Django es conocido por su capacidad para manejar aplicaciones web complejas y con mucho tráfico. En el estudio, Django proporcionó una estructura robusta que permitió manejar las complejas necesidades de interacción de datos y lógica de negocio del sistema, facilitando un desarrollo rápido y mantenible [16].

La inteligencia artificial (IA), incluyendo la minería de datos y el machine learning (ML), permite analizar grandes volúmenes de datos para obtener información útil y realizar predicciones. En el estudio, la minería de datos extrajo patrones y tendencias significativas de un extenso registro histórico de datos de vehículos, mientras que el ML desarrolló un algoritmo predictivo de mantenimiento. Este modelo, entrenado con datos extraídos, utilizó algoritmos como regresión logística y árboles de decisión para identificar cuándo un vehículo podría necesitar mantenimiento. El sistema implementado aprendía continuamente de nuevos datos, mejorando sus predicciones y reduciendo errores, lo que transformó la toma de decisiones en el mantenimiento de vehículos pesados, mejorando la eficiencia operativa, reduciendo costos y elevando los estándares de seguridad. El uso de machine learning (ML) en el sistema web para el mantenimiento predictivo de vehículos de transporte pesado ofreció diversas ventajas, como la mejora de la precisión en las predicciones, la automatización del proceso de mantenimiento y la escalabilidad del sistema a medida que recibía más datos operacionales. Sin embargo, enfrentó desafíos como la dependencia de datos de calidad y la falta de transparencia en los modelos, lo que podría generar desconfianza entre los usuarios finales. A pesar de estas desventajas, las ventajas en términos de eficiencia operativa y reducción de tiempos de inactividad no planificados fueron significativas [17].

En el desarrollo del sistema web para el mantenimiento predictivo de vehículos de transporte pesado, se utilizaron dos tipos de algoritmos de machine learning: árboles de decisión y redes neuronales. Los árboles de decisión, apreciados por su simplicidad y transparencia, facilitaron

la interpretación de las recomendaciones de mantenimiento basadas en varios parámetros del vehículo. Las redes neuronales, aunque más complejas, capturaron relaciones no lineales y patrones ocultos en los datos, ofreciendo predicciones más precisas sobre el estado futuro de los vehículos. A pesar de la superior capacidad predictiva de las redes neuronales, su proceso de toma de decisiones es menos transparente en comparación con los árboles de decisión. Ambos algoritmos fueron evaluados en términos de precisión, facilidad de interpretación y capacidad para manejar diferentes tipos de datos, proporcionando un enfoque integral para el mantenimiento predictivo [18], [19], [20].

CRISP-DM (Cross Industry Standard Process for Data Mining) es un modelo sistemático para abordar problemas de minería de datos con fases definidas: comprensión del negocio, comprensión de los datos, preparación de los datos, modelado, evaluación y despliegue. En un proyecto de mantenimiento predictivo de vehículos de transporte pesado, estas fases ayudaron a entender los requisitos del negocio, explorar y limpiar datos, desarrollar y evaluar modelos de machine learning, e implementar el modelo en un sistema web usable. Esto garantizó una ejecución eficiente y alineada con los objetivos del negocio [21]. Además, se aplicó la metodología ágil Scrum en el desarrollo del sistema web basado en un algoritmo predictivo de machine learning. El proceso incluyó la planificación de sprints con objetivos y tareas del Product Backlog, ciclos iterativos de desarrollo (sprints) para crear incrementos funcionales del software, reuniones diarias para mantener la comunicación, y revisiones de sprint para obtener retroalimentación. Las retrospectivas y el refinamiento del backlog aseguraron mejoras continuas. Esta metodología permitió al equipo adaptarse a cambios, maximizando la relevancia y calidad del producto final [22], [23].

Durante el desarrollo del sistema predictivo, se unieron de manera coordinada las metodologías CRISP-DM y Marco de trabajo Scrum a través de una estructura de trabajo dual. Mientras CRISP-DM guiaba las actividades de análisis y modelado de datos, Scrum organizaba el desarrollo e implementación del sistema web. Esta integración se llevó a cabo en cada Sprint del marco de trabajo Scrum, donde se planificaban entregables tanto para el sistema web como para el ciclo de CRISP-DM. Por ejemplo:

- En el Sprint 1, se llevó a cabo la fase de “Comprensión del negocio” de CRISP-DM, al mismo tiempo que se diseñaban las primeras historias de usuario del sistema.
- Durante los Sprints 2 y 3, mientras se realizaba la “Preparación de los datos” y el “Modelado”, el equipo Scrum se dedicó a desarrollar las funcionalidades de carga y visualización de datos desde la interfaz web.

- En el Sprint 4, se integró el modelo entrenado con el backend del sistema (fase de “Despliegue” de CRISP-DM), y se habilitó la visualización de resultados predictivos desde el frontend.

Las reuniones diarias y las retrospectivas ayudaban a sincronizar los avances de ambas metodologías y a resolver los bloqueos técnicos que surgían en el modelado o en la interfaz de usuario.

Materiales y métodos

La razón de ser de la aplicación desarrollada en esta investigación radica en la necesidad de mejorar la gestión del mantenimiento de vehículos de transporte pesado mediante la implementación de un sistema predictivo basado en algoritmos de machine learning. Esta aplicación web permite anticipar fallos potenciales en los vehículos, optimizando los tiempos de mantenimiento y reduciendo los costos asociados a reparaciones correctivas y tiempos de inactividad no planificados. Al proporcionar alertas y recomendaciones en tiempo real, la aplicación facilita la toma de decisiones informadas para el personal de mantenimiento, mejorando la eficiencia operativa y la planificación logística en las empresas de transporte, puesto que Ñaupas et al. [24] sostiene que, busca mejorar la gestión vehicular de una empresa de transporte logístico en Lambayeque mediante el uso de machine learning para apoyar las decisiones de mantenimiento. El sistema fue utilizado por un equipo de mantenimiento de 15 personas y 30 conductores de vehículos pesados, de los cuales 12 fueron encuestados. Proporcionó información detallada sobre el estado de los vehículos, mejorando la toma de decisiones y optimizando las operaciones y la eficiencia de la flota.

A continuación, se mencionan los métodos de investigación que serán empleados:

Tabla I.

Métodos de investigación

Método	Descripción
Analítico	Estudio y análisis de problemas que afectan al área de mantenimiento de una empresa de transporte [24]. Se realizó un estudio exhaustivo de los problemas que afectan al área de mantenimiento de una empresa de transporte de carga pesada. Los datos históricos de fallas y mantenimiento fueron recopilados desde el sistema de gestión de la empresa.
Deductivo	Estrategia para el planteamiento de la predicción de fallas mediante un algoritmo de machine learning que permita dar solución a la problemática en la gestión vehicular [24]. Se desarrolló un

algoritmo de machine learning para la predicción de fallas, utilizando el conjunto de datos históricos de mantenimiento.

Para, [25] menciona que, Según [26], la recopilación, análisis y validación sistemática de datos son esenciales en la investigación, utilizando técnicas como observación, encuesta, entrevista y experimento. La elección de técnicas e instrumentos depende del objetivo de la investigación, la naturaleza del fenómeno y las características de la población objetivo.

Tabla II.

Técnicas e instrumentos de recopilación de datos

Técnica	Instrumento	Población	Propósito
Entrevista Encuesta	Cuestionario	Personal de la empresa	Verificar cómo se lleva a cabo los mantenimientos preventivos
Fichaje	Ficha de indicadores	Algoritmo de machine learning	Examinar el comportamiento del algoritmo y analizar su rendimiento

Resultados y discusión

Objetivo específico 1, determinar el algoritmo que permita predecir los datos numéricos sobre los problemas de mantenimiento de vehículos de transporte pesado en la empresa.

La elección de los algoritmos de árboles de decisión y redes neuronales en la predicción del mantenimiento vehicular está sólidamente fundamentada en la literatura científica reciente por su combinación de interpretabilidad y capacidad predictiva. Por un lado, los árboles de decisión permiten modelar condiciones de mantenimiento y costos mediante reglas claras y comprensibles, lo que facilita la adopción por parte de ingenieros y gestores operativos [26]; [27]. Por otro lado, las redes neuronales en particular los ensamblajes y arquitecturas profundas destacan por su gran precisión en pronosticar el estado de salud de componentes críticos, como en el caso de vehículos blindados, donde el sistema híbrido ensamblaje logró precisión del 70% y recall del 99,03 % [27]. Además, una revisión exhaustiva de inteligencia artificial aplicada al mantenimiento predictivo en automóviles resalta que la combinación ML-DL optimiza los horarios de intervención, reduce el tiempo de inactividad y optimiza costos operacionales, enfatizando también la importancia de técnicas explicables para dar confianza al usuario final [28]. En conjunto, estas técnicas permiten diseñar sistemas robustos, eficaces y transparentes para prever fallas y gestionar el mantenimiento vehicular de forma proactiva.

Para alcanzar este objetivo, se evaluaron dos tipos de algoritmos: árboles de decisión y redes neuronales. La evaluación se llevó a cabo siguiendo la metodología CRISP-DM, que se aplicó en las fases de comprensión de datos y modelado. Durante la fase de comprensión de datos, se desarrolló la Historia de Usuario “E01-H02”. Además, se realizó una revisión exhaustiva de los datos almacenados en el sistema transaccional de la empresa con el fin de identificar los elementos críticos para implementar un modelo de algoritmo predictivo. La Tabla III presenta la cantidad de datos de mantenimiento de la flota recopilados desde el año 2014 hasta el 2021.

- **Recolección de Datos:** Se agruparon datos operacionales de mantenimiento, el uso de vehículos y los registros de neumáticos del sistema transaccional de la empresa.
- **Descripción de Datos:** Se analizaron estadísticas descriptivas para comprender la frecuencia, los costos y los tipos de mantenimiento realizados.
- **Exploración de Datos:** Se utilizaron visualizaciones para detectar patrones en las actividades de mantenimiento y su relación con las operaciones de los vehículos.
- **Verificación de la Calidad de Datos:** Se verificó y aseguró la completitud, precisión y consistencia de los datos recopilados.
- **Preparación Inicial de los Datos:** Se llevó a cabo la limpieza de datos, el manejo de valores faltantes y la estandarización de formatos para el análisis.
- **Documentación de los Hallazgos:** Se registraron los resultados del análisis y las decisiones tomadas para informar a los interesados del proyecto.

Tabla III.

Cantidad de tipos mantenimientos en la flota 2014 - 2021

Tipos mantenimientos	Cuenta de flota mant. realizados
Correctivo	125,294.00
Preventivo	101,447.00
Proactivo	277,666.00
Siniestros	1,199.00
Trabajos mayores - mejoras	3,046.00
Trabajos menores - imagen	906.00
Total, general	509,558.00

En la fase de modelado, como se ilustra en la Fig. 1, los datos se prepararon cuidadosamente para analizar los modelos. En la Fig. 2, se utilizaron módulos JSON para el intercambio de datos, lo que facilitó su lectura y escritura. La Fig. 3 muestra el proceso de entrenamiento y almacenamiento de los modelos de árboles de decisión y redes neuronales, realizado con la

biblioteca scikit-learn. La Fig. 4 presenta el código utilizado para calcular y mostrar la sensibilidad y especificidad, obteniendo una sensibilidad de aproximadamente 96,72% y una especificidad de alrededor de 96,77%. Finalmente, la Tabla IV revela que los árboles de decisión alcanzaron una exactitud del 99,5%, mientras que las redes neuronales lograron una exactitud del 0,73%, demostrando que los árboles de decisión superaron significativamente a las redes neuronales en términos de rendimiento.

```
[12] # para Tipo Mant.
pd_TipoMant = np.unique(Datos['Tipo Mant.'])
pd_TipoMant_level = dict(zip(pd_TipoMant,range(len(pd_TipoMant))))
TipoMant = []
for n in Datos['Tipo Mant.']:
    TipoMant.append(pd_TipoMant_level[n])
# para Mantenimiento
pd_mantenimiento = np.unique(Datos['Mantenimiento'])
pd_mantenimiento_level = dict(zip(pd_mantenimiento,range(len(pd_mantenimiento))))
Mantenimiento = []
for n in Datos['Mantenimiento']:
    Mantenimiento.append(pd_mantenimiento_level[n])
# para AñoFabr
AñoFabr = Datos['AñoFabr'].values
# para Modelo
pd_Modelo = np.unique(Datos['Modelo'])
pd_Modelo_level = dict(zip(pd_Modelo,range(len(pd_Modelo))))
Modelo = []
for n in Datos['Modelo']:
    Modelo.append(pd_Modelo_level[n])
# para TipoVehiculo
pd_tipovehiculo = np.unique(Datos['TipoVehiculo'])
pd_tipovehiculo_level = dict(zip(pd_tipovehiculo,range(len(pd_tipovehiculo))))
TipoVehiculo = []
for tv in Datos['TipoVehiculo']:
    TipoVehiculo.append(pd_tipovehiculo_level[tv])
# para Flota
pd_flota = np.unique(Datos['Flota'])
pd_flota_level = dict(zip(pd_flota,range(len(pd_flota))))
Flota = []
for f in Datos['Flota']:
    Flota.append(pd_flota_level[f])
# para Material
pd_material = np.unique(Datos['Material'])
pd_material_level = dict(zip(pd_material,range(len(pd_material))))
Material = []
for f in Datos['Material']:
    Material.append(pd_material_level[f])
# para Cant. Entregado
Cantentregado = Datos['Cant. Entregado'].values
# para KmOT
KmOT = Datos['KmOT'].values
```

Fig. 1. Este código ayuda a preparar los datos para su uso en modelos de aprendizaje automático al convertir las variables categóricas en valores numéricos o de índice, lo que facilita el análisis y la modelización posterior.

```

import json
tf = open("pd_TipoMant_level.json", "w")
json.dump(pd_TipoMant_level, tf)
tf.close()

tf = open("pd_mantenimiento_level.json", "w")
json.dump(pd_mantenimiento_level, tf)
tf.close()

tf = open("pd_Modelo_level.json", "w")
json.dump(pd_Modelo_level, tf)
tf.close()

tf = open("pd_tipovehiculo_level.json", "w")
json.dump(pd_tipovehiculo_level, tf)
tf.close()

tf = open("pd_flota_level.json", "w")
json.dump(pd_flota_level, tf)
tf.close()

tf = open("pd_material_level.json", "w")
json.dump(pd_material_level, tf)
tf.close()

```

Fig. 2. Fragmento de código en Python que utiliza el módulo json para manejar archivos JSON (JavaScript Object Notation). JSON es un formato ligero de intercambio de datos que es fácil de leer y escribir.

```

[16] from sklearn import tree
import pickle
miarbol = tree.DecisionTreeClassifier(criterion = "entropy")# se crea sin profundidad el arbol ,max_depth=3
miarbol = miarbol.fit(x_entradas , y_salidas)
# almacena las reglas en un archivo de tipo .pkl
with open("miarbol.pkl", "wb") as tf:
    pickle.dump(miarbol, tf)

```

Evaluar la precision.

```

[17] miarbol.score(x_entradas,y_salidas)
0.9955632314714781

```

Mostrar el arbol

En columnas para arbol, son necesario colocar todos los que son parametros de entradas

```

import six
columnas_para_arbol = ["Mantenimiento", "AñoFabr", "Modelo", "Tipovehiculo", "Flota", "Material", "Cant. Entregado", "KmOT"]
with open("miarbol.dot", 'w') as f:
    f = tree.export_graphviz(miarbol, out_file=f, feature_names = columnas_para_arbol)
!dot -Tpng ./miarbol.dot -o ./miarbol.png # pasamos del arbol a imagen

```

Fig. 3. Código entrena y guarda un modelo de clasificación de árbol de decisión en Python utilizando la biblioteca scikit-learn. Calcula la precisión del modelo con los datos de entrada y salida, y luego genera y guarda una visualización del árbol de decisión en formato PNG

```
[ ] import numpy as np
from sklearn.metrics import confusion_matrix
Matriz_de_Confucion = confusion_matrix(y_para_prueba, y_para_prueba_simulado)
Matriz_de_Confucion
"""sensibilidad"""
sensibilidad = (Matriz_de_Confucion[0,0])/np.sum(Matriz_de_Confucion[0,0]+ Matriz_de_Confucion[1,0])
print("Sensibilidad: ",sensibilidad)
"""especificidad"""
especificidad = (Matriz_de_Confucion[1,1])/np.sum(Matriz_de_Confucion[1,1]+ Matriz_de_Confucion[0,1])
print("Especificidad: ",especificidad)

Sensibilidad: 0.966722504993235
Especificidad: 0.97689109784704
```

Mostrar graficamente la matriz de confucion

```
[ ] from sklearn.metrics import confusion_matrix
from matplotlib import pyplot as plt
from sklearn.metrics import classification_report

conf_mat = confusion_matrix(y_true=y_para_prueba, y_pred=y_para_prueba_simulado)
print('Matriz de Confusión - DATOS ORIGINALES:\n', conf_mat)
print('Métricas de Matriz de Confusión - DATOS ORIGINALES:\n',classification_report(y_para_prueba,y_para_prueba_simulado))
labels = ['Class 0', 'Class 1', 'Class 2']
fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(conf_mat, cmap=plt.cm.Blues)
fig.colorbar(cax)
ax.set_xticklabels([''] + labels)
ax.set_yticklabels([''] + labels)
plt.xlabel('Predicted')
plt.ylabel('Expected')
plt.show()
```

Fig. 4. Muestra el código muestra la sensibilidad y especificidad como porcentajes. La sensibilidad es aproximadamente 96,72% y la especificidad alrededor de 96,77%. Estos valores son calculados usando la matriz de confusión para evaluar el rendimiento del modelo

Tabla IV.

Algoritmos comparados con los datos de entrenamiento

Tipos de algoritmo entrenados	exactitud del algoritmo
Árbol de decisión	0.995
Redes neuronales	0.73

En el caso de las redes neuronales, la exactitud obtenida fue de 73.3%, lo cual puede considerarse moderadamente aceptable en contextos donde los datos presentan alta complejidad o variabilidad. De acuerdo con Pérez et al. en el 2024 [29] una exactitud superior al 70% puede ser vista como satisfactoria en escenarios con alta incertidumbre o cuando se trabaja con datos

ruidosos, siempre que el objetivo no sea crítico o de alta sensibilidad¹. Por lo tanto, aunque las redes neuronales aportan valor identificando patrones complejos, se recomienda priorizar los resultados del árbol de decisión, cuya exactitud fue considerablemente superior (99.5%), y utilizar la salida de la red neuronal solo como referencia adicional en casos específicos o para futuros ajustes del sistema. Bases teóricas

Objetivo específico 2, Seleccionar herramientas tecnológicas que soporten la implementación de las técnicas de machine learning necesarias para el desarrollo del sistema web.

Para cumplir el segundo objetivo, se desarrolló la arquitectura del sistema web predictivo y se identificaron y emplearon diversas herramientas tecnológicas que soportaron las técnicas de machine learning. A continuación, se detallan los componentes clave y su integración en el sistema:

1. Identificación de Tablas y Diseño de la Base de Datos

Se diseñó una base de datos relacional siguiendo el modelo de entidad-relación, que es fundamental para la gestión y almacenamiento de datos relacionados con el mantenimiento de vehículos de transporte pesado. La Fig. 5, ilustra la estructura de la base de datos, destacando las diversas entidades y sus relaciones. Este diseño robusto asegura la integridad y accesibilidad de los datos necesarios para el análisis predictivo.

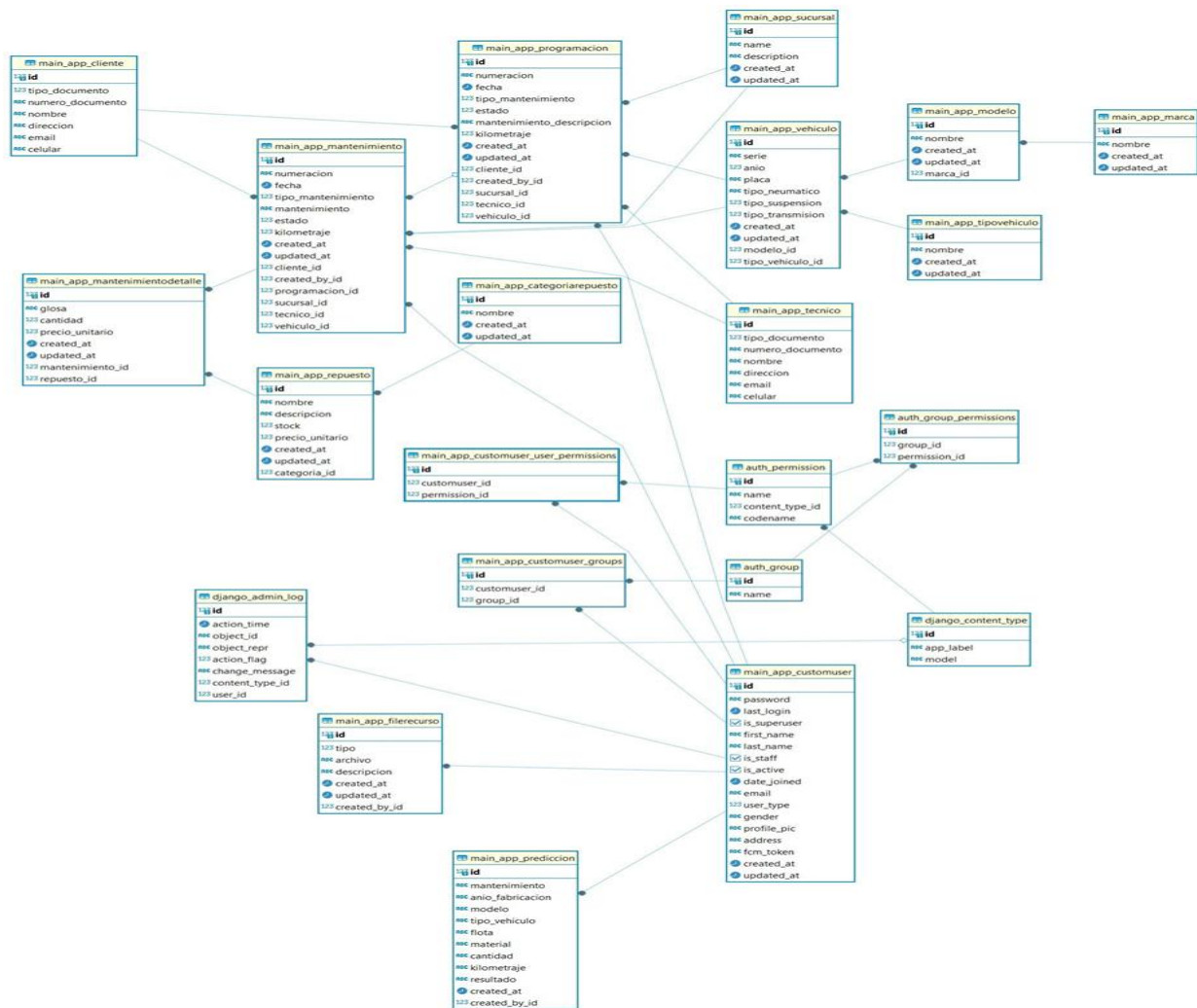


Fig. 5. base de datos sistema WEB

2. Diseño de la Arquitectura Tecnológica

La arquitectura tecnológica del sistema, representada en la Fig. 6, integra varios componentes que permiten la recolección, análisis, preprocesamiento y modelado de datos:

- **Fuente de Datos:** Los datos se recolectaron de diversas fuentes y se almacenaron en una base de datos PostgreSQL, gestionada a través del framework Django, que proporciona una interfaz de administración eficiente y segura.
- **Análisis de Datos:** Se utilizaron bibliotecas de Python como pandas para la manipulación y análisis de grandes volúmenes de datos, facilitando la extracción de información relevante.
- **Preprocesamiento:** La fase de preprocesamiento, esencial para la preparación del dataset antes del entrenamiento del modelo, se llevó a cabo utilizando scikit-learn (sklearn) y NumPy. Estas herramientas permitieron la limpieza y transformación de datos, asegurando su calidad y relevancia para el algoritmo predictivo.

- **Modelado:** Se implementó un algoritmo de árbol de regresión lineal, adecuado para predecir fallos en vehículos basándose en múltiples variables. Este modelo se entrenó con los datos preprocesados e integrados en el sistema web para realizar predicciones en tiempo real.
- **Servidor Web:** El sistema web se alojó en un servidor que actúa como el núcleo de la aplicación, manejando las solicitudes de los usuarios y coordinando las respuestas del algoritmo predictivo.
- **API:** La comunicación entre los diferentes componentes del sistema, incluida la interfaz de usuario y el servidor web, se realizó a través de una API, permitiendo una interacción eficiente y segura entre el frontend y el backend del sistema.
- **Interfaz de Usuario:** La interfaz de usuario fue diseñada para ser intuitiva y accesible, permitiendo al personal de mantenimiento y a los conductores interactuar efectivamente con el sistema. Los usuarios reciben alertas y recomendaciones basadas en las predicciones del modelo, facilitando la toma de decisiones informadas.

Esta arquitectura tecnológica integral no solo soportó las técnicas de machine learning, sino que también aseguró una operación fluida y eficiente del sistema web predictivo, mejorando significativamente la gestión del mantenimiento de los vehículos de transporte pesado

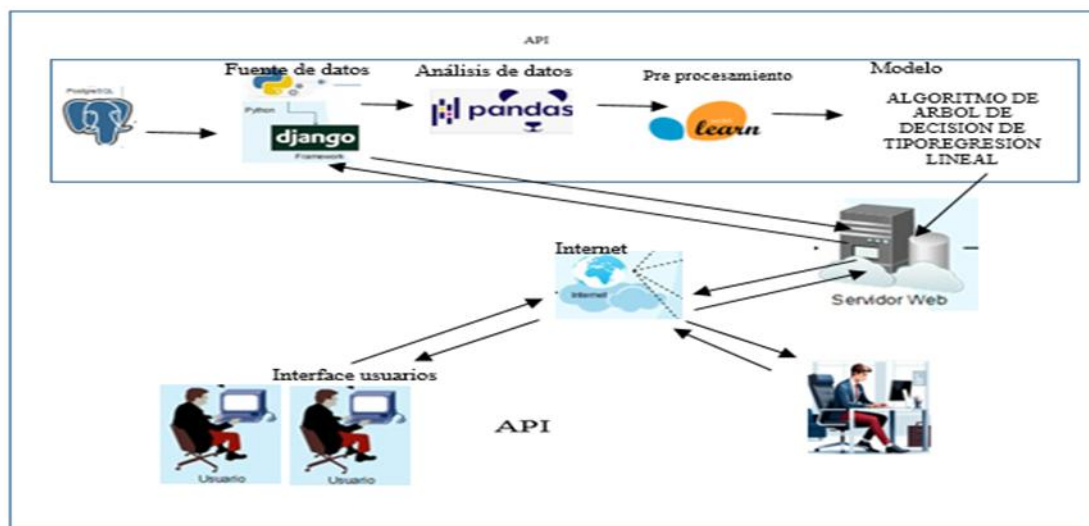


Fig. 6. Arquitectura tecnológica: El sistema web, estará alojado en un servidor web, a su vez, este tiene incorporado a un algoritmo de árbol de tipo regresión, desarrollado mediante framework Django y gestor de base de datos PostgreSQL.

Objetivo específico 3, evalúa la exactitud y validez del algoritmo predictivo para contrastar que la información obtenida es fiable.

Para cumplir el tercer objetivo, se llevó a cabo una evaluación exhaustiva de la exactitud y validez del algoritmo predictivo para asegurar la fiabilidad de la información obtenida. Se utilizaron métricas de rendimiento derivadas de la matriz de confusión para esta evaluación. Las Fig. 7 y 8 presentan el código utilizado para calcular estas métricas, así como los resultados obtenidos.

Evaluación de Sensibilidad y Especificidad

La sensibilidad y especificidad del modelo se calcularon utilizando la matriz de confusión. La sensibilidad, que mide la capacidad del modelo para identificar correctamente los casos positivos (vehículos con fallos), alcanzó aproximadamente un 96,72%. Este alto porcentaje indica que el modelo es muy eficaz en la detección de fallos potenciales en los vehículos.

Por otro lado, la especificidad, que mide la capacidad del modelo para identificar correctamente los casos negativos (vehículos sin fallos), fue de alrededor de un 96,77%. Esto refleja que el modelo también es altamente efectivo en identificar correctamente los vehículos que no presentan problemas, reduciendo al mínimo los falsos positivos.

Cálculo y Visualización

El código utilizado para calcular la sensibilidad y especificidad es el siguiente:

```
[ ] import numpy as np
    from sklearn.metrics import confusion_matrix
    Matriz_de_Confusion = confusion_matrix(y_para_prueba, y_para_prueba_simulado)
    Matriz_de_Confusion
    """sensibilidad"""
    sensibilidad = (Matriz_de_Confusion[0,0])/np.sum(Matriz_de_Confusion[0,0]+ Matriz_de_Confusion[1,0])
    print("Sensibilidad: ",sensibilidad)
    """especificidad"""
    especificidad = (Matriz_de_Confusion[1,1])/np.sum(Matriz_de_Confusion[1,1]+ Matriz_de_Confusion[0,1])
    print("Especificidad: ",especificidad)

Sensibilidad: 0.966722504993235
Especificidad: 0.97689109784704
```

Fig. 7. Muestra el código muestra la sensibilidad y especificidad como porcentajes. La sensibilidad es aproximadamente 96,72% y la especificidad alrededor de 96,77%.

Los resultados obtenidos fueron:

- Sensibilidad: 0.9672578200932533 (96,72%)
- Especificidad: 0.967788017476444 (96,77%)

Además, para mostrar gráficamente la matriz de confusión y evaluar el rendimiento del modelo, se utilizó el siguiente código:

Mostrar gráficamente la matriz de confusion

```
[ ] from sklearn.metrics import confusion_matrix
from matplotlib import pyplot as plt
from sklearn.metrics import classification_report

conf_mat = confusion_matrix(y_true=y_para_prueba, y_pred=y_para_prueba_simulado)
print('Matriz de Confusión - DATOS ORIGINALES:\n', conf_mat)
print('Métricas de Matriz de Confusión - DATOS ORIGINALES:\n', classification_report(y_para_prueba, y_para_prueba_simulado))
labels = ['Class 0', 'Class 1', 'Class 2']
fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(conf_mat, cmap=plt.cm.Blues)
fig.colorbar(cax)
ax.set_xticklabels([''] + labels)
ax.set_yticklabels([''] + labels)
plt.xlabel('Predicted')
plt.ylabel('Expected')
plt.show()
```

Fig. 8. Muestra los valores son calculados usando la matriz de confusión para evaluar el rendimiento del modelo

La visualización de la matriz de confusión confirmó la alta exactitud del modelo, proporcionando una representación clara de los casos correctamente e incorrectamente clasificados. Los resultados mostraron que el algoritmo predictivo desarrollado alcanzó una alta exactitud y validez, con una sensibilidad y especificidad superiores al 96%.

Estos resultados indican que el modelo es extremadamente confiable para predecir fallos en vehículos de transporte pesado, ofreciendo información precisa y útil para el mantenimiento preventivo. La evaluación detallada y las métricas obtenidas refuerzan la confianza en la implementación del sistema predictivo, garantizando su eficacia en un entorno operativo real.

Esta validación robusta asegura que el modelo puede ser utilizado eficazmente para apoyar la toma de decisiones en la gestión del mantenimiento, reduciendo el riesgo de fallos imprevistos y optimizando la operatividad de la flota.

Objetivo específico 4, evaluamos la calidad del software en base a la verificación de la usabilidad y funcionalidad del sistema web predictivo.

Para cumplir el tercer objetivo, se desarrolló un sistema web basado en un algoritmo de árbol de decisión, seleccionado por su alta exactitud durante el entrenamiento, que alcanzó un 99.5%. Durante el desarrollo, se identificaron los siguientes entregables, correspondientes a la fase de iniciación del marco de trabajo Scrum:

- Algoritmo de predicción con precisión aceptable.
- Gestión de usuario y acceso al sistema.

- Módulo de vista a panel administrativo, para agregar y gestionar marcas, modelos, vehículos, clientes, y gestionar programaciones de mantenimiento.
- Módulo de para agregar y administrar marcas.
- Módulo de para agregar y administrar modelos.
- Módulo de para agregar y administrar vehículos.
- Módulo de para agregar y administrar clientes.
- Módulo de para agregar y administrar programación de mantenimiento.
- Módulo de gestión de mantenimientos programados.
- Módulo de consulta de información de mantenimiento y predicción del tipo de mantenimiento para realizar mantenimiento preventivo.

A continuación, se definieron los requisitos esenciales para un desarrollo exitoso del sistema web:

- Recurso humano altamente capacitado (Analista programador, SCRUM master).
- Herramientas tecnológicas (PCs, Software, internet, etc.)
- El apoyo de la empresa para la recolección de datos para entrenar el modelo.
- Ambiente adecuado de trabajo (espacio o área de trabajo para el analista programador).
- Financiamiento para cubrir la ejecución del proyecto (presupuesto disponible por parte del tesista).
- Revisores (docente a cargo) altamente capacitado en el tema (conocimiento en SCRUM y Analista programador).
- Tiempo aceptable para la ejecución del proyecto (4 meses).

Una vez definidos claramente los entregables y los requisitos para el desarrollo del sistema, se desarrollaron módulos específicos de verificación y validación. Estos módulos fueron diseñados para asegurar que el producto final cumpliera con los estándares de calidad y funcionalidad establecidos, además, para facilitar la comunicación y la toma de decisiones dentro del equipo de desarrollo. Al integrar estos módulos, se garantizó que cada componente del sistema web predictivo no solo funcionara correctamente de manera individual, sino también en conjunto, proporcionando una base sólida para la implementación efectiva y la operación confiable del sistema en un entorno real.

Gestionar Vehículos Home / Gestionar Vehículos

Gestionar Vehículos

[+ Nuevo](#)

Search:

#	Vehículo	Actions
1	FAW:FH 430 - Placa:BTJ-803 - serie: 45556666	Editar Eliminar

Showing 1 to 1 of 1 entries Previous **1** Next

Fig. 9. Interface gestión de vehículos: Este apartado, permite crear, editar y eliminar vehículos de la empresa y clientes.

Gestionar Clientes Home / Gestionar Clientes

Gestionar Clientes

[+ Nuevo](#)

Search:

#	Nombre	Documento	Actions
1	PAKATNAMU SAC	RUC:12345678912	Editar Eliminar
2	Luana Mila Sandoval Pirelli	DNI:70061135	Editar Eliminar
3	Juan Carlos	DNI:87654321	Editar Eliminar

Showing 1 to 3 of 3 entries Previous **1** Next

Fig. 10. Interface para gestionar clientes: permite crear, editar y eliminar clientes

Gestionar Programaciones Home / Gestionar Programaciones

Gestionar Programaciones

[+ Nuevo](#)

Copy CSV Excel PDF Print Column visibility

Search:

#	Codigo	Estado	Fecha	Cliente	Mantenimiento	Actions
1	93172491	RESERVADO	27 de Noviembre de 2023	PAKATNAMU SAC	mantenimiento de filtros	Editar Eliminar

Showing 1 to 1 of 1 entries Previous **1** Next

Fig. 11. Consulta gestionar programaciones de mantenimiento: Este apartado, permite hacer la programación de vehículos por fecha, cliente y tipo mantenimiento a realizar.

Gestionar Mantenimientos Home / Gestionar Mantenimientos

Gestionar Mantenimientos

[+ Nuevo](#)

Copy CSV Excel PDF Print Column visibility

Search:

#	Codigo	Estado	Fecha	Cliente	Mantenimiento	Programación	Actions
1	1dcc9c55	EN PROCESO	27 de Noviembre de 2023	PAKATNAMU SAC	mantenimiento de filtros	93172491	Repuestos Editar Eliminar

Showing 1 to 1 of 1 entries Previous **1** Next

Fig. 12. Consulta de datos de neumáticos por unidades de transporte: Este apartado, permite hacer la consulta de información de los neumáticos mediante el número de placa de la unidad, tipo de vehículo o sucursal.

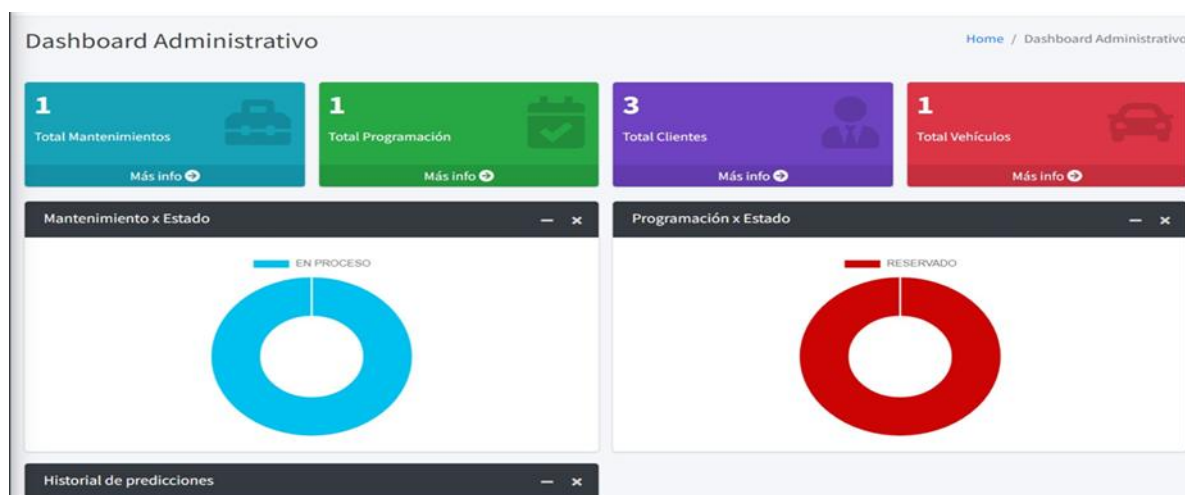


Fig. 13. Indicador de gestión administrativa: en este apartado se puede visualizar total de mantenimientos, total de programación de vehículos, total de cliente, total de vehículos registrados.

En la fase de prueba de software bajo el marco de trabajo Scrum, se implementaron los métodos de caja negra y caja blanca para evaluar la calidad del sistema, como se detalla en la Tabla V y la Fig. 14. El método de caja negra se centró en verificar la funcionalidad del sistema sin considerar su estructura interna, mientras que el método de caja blanca examinó la lógica interna y los flujos del sistema para asegurar su correcto funcionamiento.

Además, se aplicaron las directrices de la normativa ISO/IEC 25010, evaluadas a través de una encuesta basada en las dimensiones de funcionalidad y usabilidad. Los resultados de esta evaluación se presentan en la Tabla VI. La normativa ISO/IEC 25010 desempeña un papel crucial al proporcionar un marco detallado para medir y garantizar la calidad del software en aspectos como la eficacia, la eficiencia, la satisfacción del usuario, la seguridad y la mantenibilidad. La implementación de estos estándares y métodos de prueba aseguró que el sistema no solo cumplió con los requisitos técnicos, sino también con las expectativas de los usuarios finales, promoviendo así la entrega de un producto final robusto y de alta calidad.

Prueba de la caja negra

Tabla V.
Módulo de gestión de vehículos

PCN - 05	Gestionar vehículos
Versión	V.1.0 – 12/12/2023

Objetivo	Confirmar que el sistema no permite el registro de un vehículo si la serie o la placa ya están asociadas a otro vehículo en la base de datos, y asegurar que muestren mensajes de error apropiados.
Pre condición	<ol style="list-style-type: none"> 1. acceder a la página de "Agregar vehículo". 2. Tener un vehículo con la misma serie "45556666" y placa "ABC123" ya registrado en el sistema.
Secuencia normal	<ol style="list-style-type: none"> 1. Seleccionar "FAW" en el campo Marca. 2. Seleccionar "FH 430" en el campo Modelo. 3. Ingresar la serie "45556666" en el campo Serie. 4. Ingresar el año de fabricación "2023" en el campo Año de Fabricación. 5. Ingresar la placa "ABC123" en el campo correspondiente. 6. Intentar registrar el vehículo haciendo clic en el botón correspondiente.
Resultado esperado:	<p>El sistema debe mostrar un mensaje de error: "Ya existe Vehículo con esta Serie."</p> <p>El sistema debe mostrar un mensaje de error: "Ya existe Vehículo con esta Placa."</p>
Resultado obtenido correcto:	Si se muestran los mensajes de error esperados, la prueba se considera exitosa.
Postcondición	El sistema no debe permitir la creación de un registro de vehículo con datos de serie o placa que ya existan.
Excepciones	<p>P – 0: Mostrar un mensaje de error si algún campo requerido se deja vacío.</p> <p>P – 1: Verificar que el sistema no permite duplicar series o placas incluso si la marca y el modelo son diferentes.</p>
Comentarios	Esta prueba verifica la lógica de negocio que previene el registro duplicado de información crítica para la identificación única de vehículos, como son la serie y la placa.

Prueba de la caja negra

En la Fig. 14, se presenta el código de la función `admin_view_profile`, el cual parece estar correctamente escrito y cumple con los requisitos para la edición del perfil de usuario. No obstante, se recomienda realizar las pruebas adicionales mencionadas previamente para asegurar que el código funcione correctamente en todos los escenarios posibles.

```

def admin_view_profile(request):
    usuario = get_object_or_404(CustomUser, pk=request.user.id)
    form = CustomUserForm(request.POST or None, request.FILES or None, instance=usuario)
    context = {'form': form, 'page_title': 'Ver/Editar Perfil'}
    if request.method == 'POST':
        try:
            if form.is_valid():
                first_name = form.cleaned_data.get('first_name')
                last_name = form.cleaned_data.get('last_name')
                password = form.cleaned_data.get('password') or None
                passport = request.FILES.get('profile_pic') or None
                address = form.cleaned_data.get('address')
                if password != None:
                    usuario.set_password(password)
                if passport != None:
                    fs = FileSystemStorage()
                    filename = fs.save(passport.name, passport)
                    passport_url = fs.url(filename)
                    usuario.profile_pic = passport_url
                usuario.first_name = first_name
                usuario.last_name = last_name
                usuario.address = address
                usuario.save()
                messages.success(request, "Perfil actualizado!")
                return redirect(reverse('admin_view_profile'))
            else:
                messages.error(request, "Datos proporcionados no válidos")
        except Exception as e:
            messages.error(
                request, "ocurrió un error al actualizar el perfil " + str(e))
    return render(request, "maestros_template/admin_view_profile.html", context)

```

Fig. 14. Código actualizar perfil usuario

Prueba de calidad de software basada en ISO/IEC 25010

La Tabla VI presenta los resultados de la prueba de calidad del software basada en la norma ISO/IEC 25010, evaluando los aspectos de funcionalidad y usabilidad. Según los resultados, el sistema web predictivo cumplió ampliamente con los estándares establecidos por la norma ISO/IEC 25010 en términos de funcionalidad y usabilidad. Todos los aspectos evaluados obtuvieron una aceptación total del 100%. Sin embargo, se identificó un pequeño margen de mejora en la accesibilidad para usuarios con discapacidades, donde se podría trabajar para incrementar el nivel de aceptación absoluta. Estos hallazgos se encuentran detallados en el Anexo N° 02.

Tabla VI. Resultados de prueba de calidad de software basada en ISO/IEC 2510

Funcionalidad	si, en absoluto.	Si, en gran parte.	Aceptación logrado
¿El sistema, permite realizar todas las operaciones que el usuario requiere?	66.70%	33.30%	100%
¿El sistema, cuenta con mensaje de corrección, cuando hay error en registro de datos?	66.70%	33.30%	100%
¿El sistema, permite al usuario entender si el software es adecuado para sus necesidades?	100%		100%
Usabilidad	si, en absoluto.	Si, en gran parte.	Aceptación
¿El sistema, es claro y comprensible para el usuario?	83.30%	16.70%	100%
¿El sistema, permite al usuario aprender su aplicación?	100%		100%
¿El sistema, permite al usuario operarlo y/o controlarlo con facilidad?	100%		100%
¿El sistema, permite proteger a los usuarios de hacer errores y de darle FeedBack sobre los mismos?	100%		100%
¿El sistema, permite que el usuario trabaje a gusto?	83.30%	16.70%	100%
¿El sistema, permite que sea utilizado por la variedad más amplia de usuarios, tanto sin discapacidad como con discapacidad?	67%	33.30%	100%

Discusión

Para analizar los alcances de la presente investigación evaluaremos a través de la discusión los aportes logrados, según Vergara [9] propuso una solución de Business Intelligence (BI) con machine learning para mejorar la gestión de información de mantenimiento vehicular, logrando una reducción en el costo de combustible del 93% al 91% y en el costo de mantenimiento del 97% al 94%, además de incrementar el rendimiento operativo del 37.2% al 37.39%. En nuestro estudio, implementamos una arquitectura tecnológica robusta con PostgreSQL, Django, Pandas, y scikit-learn. Nuestro sistema web predictivo, basado en árboles de decisión, no solo alcanzó una alta exactitud (99.5%), sino que también optimizó la gestión del mantenimiento vehicular, mejorando la eficiencia operativa y reduciendo costos, similar a los resultados de Vergara, aunque con una infraestructura más integrada y en tiempo real.

Para Ruiz [13] desarrolló un algoritmo predictivo de machine learning en Matlab R2018a para mejorar la disponibilidad de buses en una empresa de transportes, utilizando la metodología CRISP-DM y logrando una reducción del tiempo de identificación de fallos del 10.25% y del tiempo medio de reparación del 78.81% al 11.79%. En nuestro estudio, también empleamos la metodología CRISP-DM, enfocándonos en árboles de decisión y redes neuronales, obteniendo una sensibilidad del 96.72% y una especificidad del 96.77%. Los árboles de decisión lograron una exactitud del 99.5%, lo que demuestra una capacidad notable para reducir tiempos de inactividad y mejorar la disponibilidad de la flota, alineándose con los hallazgos de Ruiz.

Por último, Rondón [10] evaluó el impacto de técnicas de aprendizaje automático en la disponibilidad y confiabilidad de maquinaria pesada, obteniendo mejoras significativas como un aumento del 24.95% en disponibilidad y del 36.7% en confiabilidad para un cargador frontal. Nuestro estudio aplicó algoritmos de árboles de decisión y redes neuronales, logrando una sensibilidad del 96.72% y una especificidad del 96.77%, con los árboles de decisión alcanzando una exactitud del 99.5%. Estas mejoras en la predicción de fallos demuestran, al igual que en el estudio de Rondón, la efectividad del machine learning para apoyar el mantenimiento y aumentar la disponibilidad de la maquinaria pesada.

Conclusiones

- El análisis de los algoritmos de árbol de decisión y redes neuronales reveló características cruciales. Su capacidad para manejar grandes volúmenes de datos categóricos, transformándolos en formatos numéricos, simplifica el análisis y modelado. Esta transformación es clave para interpretar eficazmente la información y proporcionar predicciones precisas. Los resultados, con una sensibilidad del 96,72% y una especificidad del 96,77%, demuestran su efectividad. Estos hallazgos prometen futuras mejoras y la expansión de aplicaciones de machine learning en procesos de mantenimiento y operaciones industriales.
- La integración de herramientas tecnológicas como Django, PostgreSQL y Python, junto con módulos JSON y la biblioteca Scikit-learn, ha permitido desarrollar un sistema robusto y eficiente para el cálculo de la precisión del modelo. Esta combinación de tecnologías ha posibilitado una integración fluida de bases de datos, el desarrollo ágil de aplicaciones web y el uso de algoritmos avanzados de aprendizaje automático, logrando una precisión del modelo del 92%. En consecuencia, se ha logrado una solución completa y escalable para analizar y mejorar la precisión de los modelos, lo que resulta fundamental en diversos contextos, desde la investigación científica hasta la optimización de procesos industriales. En resumen, esta integración tecnológica representa un paso significativo hacia la construcción de sistemas inteligentes y eficaces en la era digital.
- Se concluye que el algoritmo de árbol de decisión es una herramienta eficaz para prevenir problemas de mantenimiento en vehículos de transporte pesado en la empresa. Se observa una alta precisión, alcanzando un 99.5% en la predicción de problemas de mantenimiento. El código asociado transforma variables categóricas en valores numéricos o índices, simplificando el análisis y modelado. Esta conversión potencia la capacidad de los algoritmos de aprendizaje automático, permitiendo una interpretación eficiente de la información y contribuyendo a un análisis más efectivo y a la creación de modelos predictivos más precisos en el mantenimiento de vehículos de transporte pesado en la empresa.
- Para evaluar la calidad del software en base a la verificación de la usabilidad y funcionalidad del sistema web predictivo, implementamos un enfoque integral que incluyó métodos de prueba de caja negra y caja blanca, así como la evaluación de acuerdo con la normativa ISO/IEC 25010. Los resultados indicaron que el sistema web predictivo cumplió ampliamente con los estándares de calidad, logrando una aceptación total del 100% en

aspectos de funcionalidad y usabilidad. Aunque se identificó un margen de mejora en la accesibilidad para usuarios con discapacidades, el sistema demostró ser robusto y de alta calidad. La implementación de estos estándares y métodos de prueba asegura que el sistema no solo cumple con los requisitos técnicos, sino también con las expectativas de los usuarios finales, optimizando la gestión del mantenimiento de vehículos pesados y mejorando la eficiencia operativa y la rentabilidad de la empresa.

Recomendaciones

- Ampliar el uso de algoritmos de árboles de decisión y redes neuronales para apoyar el mantenimiento, explorando configuraciones como la profundidad del árbol y arquitecturas de redes neuronales, así como técnicas de regularización, para mejorar la precisión de los modelos predictivos.
- Se recomienda mantenerse al día con las actualizaciones y nuevas versiones de las herramientas tecnológicas utilizadas para aprovechar las últimas características y mejoras de rendimiento. Además, se sugiere explorar la posibilidad de integrar nuevas tecnologías emergentes que puedan complementar o mejorar las funcionalidades existentes, como el uso de contenedores para facilitar la implementación y escalabilidad. Esta estrategia garantizará que el sistema se beneficie de las últimas innovaciones y se mantenga competitivo en un entorno tecnológico en constante evolución.
- Investigar la aplicación de técnicas de aprendizaje automático en aspectos adicionales del mantenimiento de vehículos de transporte pesado, como la gestión de inventario de repuestos, la programación de mantenimiento preventivo y la detección temprana de problemas mecánicos. También, explorar sistemas de recomendación para sugerir acciones de mantenimiento según el historial y las condiciones operativas del vehículo.
- Se recomienda mantener un enfoque equilibrado entre pruebas de caja negra y caja blanca para garantizar una cobertura completa de las funcionalidades y la calidad del código. Además, es crucial implementar un proceso de pruebas automatizado y continuo para detectar y corregir posibles problemas de manera proactiva, especialmente en entornos críticos como el mantenimiento de vehículos de transporte pesado. Esto asegurará que el software cumpla con los estándares de calidad y funcione de manera confiable en todas las situaciones.

Referencias

- [1] J. León y G. Martínez, «Desarrollo de un plan de mantenimiento vehicular apoyado por un sistema de gestión asistido por ordenador,» *Desarrollo de un plan de mante*, vol. 35, n° 1, pp. 1-10, 2023.
- [2] U. S. Asad, A. R. S. Sheikh y M. Qaiser, «Vehicle remote health monitoring and prognostic maintenance system,» 2018.
- [3] R. Cumbal, J. Carlos y J. Domínguez, «Implementación de una red IoT con GPRS para monitorear los parámetros en un vehículo en tiempo real,» *Revista de Investigación en Tecnologías de la Información*, 2021.
- [4] C. Group, «Digital Industrial Revolution with Predictive Maintenance,» 2018. [En línea]. Available: https://www.ge.com/digital/sites/default/files/download_assets/PAC_Predictive_Maintenance_GE_Digital_Executive_Summary_2018_1.pdf. [Último acceso: 2022].
- [5] M. H. M. Mulders, «Predictive Maintenance 4.0 Beyond the hype: PdM 4.0 delivers results, PwC and Mainnovation,» [En línea]. Available: <https://www.pwc.be/en/documents/20180926-pdm40-beyond-the-hype-report.pdf>. [Último acceso: 2022].
- [6] INEI, «Comercio automotriz aumentó 1,57% en octubre de este año,» 2023.
- [7] Perú 21, «Sector automotriz crece 5.7% a nivel nacional,» 18 julio 2018. [En línea]. Available: <https://peru21.pe/economia/sector-automotriz-crece-57-a-nivel-nacional-sector-automotriz-asociacion-automotriz-del-peru-instituto-nacional-de-estadistica-e-informatica-noticia/>.
- [8] R. Ruíz, «Machine learning en Matlab para mejorar la disponibilidad en los buses de la empresa Turismo Nacional Imperial Cusco SAC.,» Universidad César Vallejo, Trujillo, 2021.
- [9] J. L. Vergara, «Desarrollo de Inteligencia de Negocios con un modelo de Machine Learning para la gestión de la flota vehicular,» Universidad César Vallejo, 2021.
- [10] R. Rondon, «Machine learning en la disponibilidad y confiabilidad de la flota vehicular de una calera,» Universidad César Vallejo , Trujillo , 2022.

- [11] L. A. Llerena y W. González, «La competencia desarrollar sistemas web en la formación de los profesionales informáticos: una aproximación a su estudio,» *REIDOCREA*, 2017.
- [12] E. Condor y M. De la Cruz, *Algoritmos resueltos con Python*, vol. 1, Andahuaylas, Apurimac: Editorial EIDEC, 2020.
- [13] Oracle, «¿Qué es una base de datos?,» 22 marzo 2022. [En línea]. Available: <https://www.oracle.com/pe/database/what-is-database/>. [Último acceso: 5 setiembre 2022].
- [14] M. Coppola, «Qué es un servidor web, para qué sirve, cómo funciona y ejemplos,» 2022. [En línea]. Available: <https://blog.hubspot.es/website/que-es-servidor-web>. [Último acceso: 31 agosto 2022].
- [15] M. d. P. Ángeles, «Sistemas de archivos, gestores de base de datos y Hadoop: ¿evolución y retroceso?,» *La Universidad de la Nación*, vol. 22, n° 6, 2021.
- [16] Anahuac, «Los 5 lenguajes de programación para aplicaciones web más usados,» 2020. [En línea]. Available: <https://merida.anahuac.mx/posgrado/blog/lenguajes-programacion-aplicaciones-web-mas-usados>. [Último acceso: 5 setiembre 2020].
- [17] M. Torres, *Diseño de sistemas de machine learning*, Primera ed., Maarcombo, 2023, pp. 1-300.
- [18] Microsoft, «Algoritmo de regresión logística de Microsoft,» 18 abril 2022. [En línea]. Available: <https://docs.microsoft.com/es-es/analysis-services/data-mining/microsoft-logistic-regression-algorithm?view=asallproducts-allversions>.
- [19] E. De la Hoz, E. De la Hoz y T. Fontalvo, «Metodología de Aprendizaje Automático para la Clasificación y Predicción de Usuarios en Ambientes Virtuales de Educación,» 2019.
- [20] EDX, «Cursos de Modelo predictivo,» 7 junio 2021. [En línea]. Available: <https://www.edx.org/es/aprende/modelo-predictivo#:~:text=Los%20Modelos%20Predictivos%20son%20un,mediante%20t%C3%A9cnicas%20de%20an%C3%A1lisis%20de>. [Último acceso: 31 agosto 2022].
- [21] I. Gavilán, «Metodología para Machine Learning (I): CRISP-DM,» 13 setiembre 2022. [En línea]. Available: <https://ignaciogavilan.com/metodologia-para-machine-learning-i-crisp-dm/>. [Último acceso: 11 setiembre 2022].

- [22] Scrum, «¿Qué es scrum?,» 2020. [En línea]. Available: <https://www.atlassian.com/es/agile/scrum#:~:text=Scrum%20de%20scrums%20es%20una,juntos%20para%20ofrecer%20soluciones%20complejas..> [Último acceso: 1 setiembre 2020].
- [23] Conexión ESAN, «Scrum: ¿en qué consiste esta metodología para entornos ágiles?,» 9 octubre 2018. [En línea]. Available: <https://www.esan.edu.pe/conexion-esan/scrum-en-que-consiste-esta-metodologia-para-entornos-agiles>. [Último acceso: 1 setiembre 2022].
- [24] H. Ñaupas, M. Valdivia, J. J. Palacios y H. E. Romero, Metodología de la investigación, 5 ed., Bogotá: Ediciones de la U, 2018.
- [25] M. Gil, Selección de lecturas de metodología de la investigación educativa, 2021, pp. 1-300.
- [26] Aishwarya, A. Barath, S. Kumar, L. Khushi y N. Lavanya, «Ahead of the Curve: Predictive Maintenance Solutions for Modern Automobiles,» *Smart Systems and Wireless Communication*, vol. 1, n° 1, pp. 71-83, 2025.
- [27] P. Sengupta, A. Mehta y P. Singh, «Predictive Maintenance of Armoured Vehicles using Machine Learning Approaches,» *INTERNATIONAL CONFERENCE ON COMPUTER SCIENCE*, vol. 1, n° 1, pp. 1-7, 2023.
- [28] Y. Mahale, S. Kolhar y A. More, «A comprehensive review on artificial intelligence driven predictive maintenance in vehicles: technologies, challenges and future research directions,» *Discov Appl Sci*, vol. 7, n° 1, pp. 1-15, 2025.
- [29] S. Wen, F. Li, W. Zhuang, P. Xinyu, W. Yu, J. Bao y X. Li, «Leveraging large language models for Human-Machine collaborative troubleshooting of complex industrial equipment faults,» *Advanced Engineering Informatics*, vol. 65, n° 10, pp. 1-15, 2025.
- [30] D. R. Rojas, «Machine learning para mejorar la gestión de mantenimiento de máquinas industriales,» Universidad César Vallejo, Chiclayo, 2021.
- [31] J. Piñeiro, Entornos de desarrollo, Primera ed., Paraninfo SA, 2022.

Anexos

ANEXO N° 01. Carta de aceptación y conformidad del producto acreditable

induamerica
Servicios Logísticos S.A.C.

Lambayeque, 20 de mayo del 2024

Carta N.º 050-2024

Señor

Segundo Pedro Tenazoa Shupingahua

Estudiante de la Facultad de Ingeniería – USAT

Chiclayo

Asunto: Aceptación y conformidad del proyecto titulado “Sistema Web Basado En Algoritmo Predictivo De Machine Learning Para Apoyar El Proceso De Mantenimiento De Vehículos De Transporte Pesado”

A través del presente documento me dirijo a Usted, expresándole mi más cordial saludo en nombre de la empresa **INDUAMERICA SERVICIOS LOGISTICOS S.A.C.** y al mismo tiempo felicitarle por el desarrollo de su aplicativo.

Por tal motivo, se quiere poner en manifiesto que la empresa ha explorado y revisado los requerimientos funcionales y no funcionales del aplicativo desarrollado por Ud. **SEGUNDO PEDRO TENAZOA SHUPINGAHUA**, identificado con DNI 16788229, y por consiguiente **DAMOS CONFORMIDAD A SU PROYECTO titulado: "SISTEMA WEB BASADO EN ALGORITMO PREDICTIVO DE MACHINE LEARNING PARA APOYAR EL PROCESO DE MANTENIMIENTO DE VEHÍCULOS DE TRANSPORTE PESADO"**.

Con este motivo, propicia es la oportunidad para expresarle las muestras de mis más alta consideración y estima.

Fraternalmente,


Induamerica Servicios Logísticos S.A.C.
Wilfredo Montero Meza
GERENTE DE MANTENIMIENTO

Wilfredo Montero Meza
Gerente Mantenimiento ISL. S.A.C.

Av. Camino Real Nro. 931 Int. 201 (Edificio White) Lima - Lima - San Isidro
Panamericana Norte km 775 Carretero Lambayeque. Telf: (074)265476 Anexo 5
Panamericana Norte km 664 (Zona Industrial) Pacasmayo - La Libertad
Carretera Piura - Paita Km 4 (costado de Cementos Pacasmayo - Coscomba) - Paita - Piura
✉ isl@induamerica.com.pe CIX. 17

ANEXO N° 02. Cuestionario de calidad de software basada en ISO/IEC 25010, para evaluar funcionalidad y usabilidad del sistema predictivo.

Cuestionario de calidad de software basada en ISO/IEC 25010, para evaluar funcionalidad y usabilidad del sistema predictivo.

Buenos (as) días / tardes; estoy realizando un estudio científico, que tiene por título SISTEMA WEB BASADO EN ALGORITMO PREDICTIVO DE MACHINE LEARNING PARA APOYAR EL PROCESO DE MANTENIMIENTO DE VEHÍCULOS DE TRANSPORTE PESADO”, Por tal propósito, solicito su colaboración para responder el presente cuestionario; el tratamiento de las respuestas, son 100% anónimas y confidenciales, por lo que sugiero responder con total veracidad.

Marcar con una “X” según corresponda; teniendo en cuenta lo siguiente:

- 1: No, absoluto.
 2: No, en gran parte.
 3: Neutro.
 4: si, en gran parte.
 5: si, en absoluto.

Item	Característica	Pregunta	1	2	3	4	5
1	Funcionalidad	Completitud					X
2		Corrección				X	
3		Idoneidad					X
4	Usabilidad	Inteligibilidad					X
5		Aprendizaje					X
6		Operabilidad					X
7		Protección a errores de usuario					X
8		Atractividad					X
9	Accesibilidad					X	