

UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Tema:

SISTEMA DE PREPROCESAMIENTO DE DATOS SENSORIALES UTILIZANDO PYTHON EN LA GESTIÓN DE MANTENIMIENTO PREDICTIVO DE BOMBAS HIDRAÚLICAS EN LA PRODUCCIÓN DE CRUDO

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la obtención del título de Ingeniera en Tecnologías de la Información

ÁREA: Gestión de Tecnologías de la Información

LÍNEA DE INVESTIGACIÓN: Software, Tecnologías de la Información y Ciencia de Datos

AUTOR: Carmen Alexandra Rea Ramírez

TUTOR: Ing. Edison Homero Álvarez Mayorga, Mg.

Ambato - Ecuador

enero – 2026

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: SISTEMA DE PREPROCESAMIENTO DE DATOS SENSORIALES UTILIZANDO PYTHON EN LA GESTIÓN DE MANTENIMIENTO PREDICTIVO DE BOMBAS HIDRAÚLICAS EN LA PRODUCCIÓN DE CRUDO, desarrollado bajo la modalidad Proyecto de Investigación por la señorita Rea Ramírez Carmen Alexandra, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que la estudiante ha sido tutorada durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, enero 2026.

Firmado electrónicamente por:
EDISON HOMERO ALVAREZ MAYORGA
Validar únicamente con FirmaEC

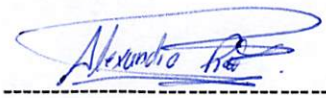
Ing. Edison Homero Álvarez Mayorga, Mg.

TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: SISTEMA DE PREPROCESAMIENTO DE DATOS SENSORIALES UTILIZANDO PYTHON EN LA GESTIÓN DE MANTENIMIENTO PREDICTIVO DE BOMBAS HIDRAÚLICAS EN LA PRODUCCIÓN DE CRUDO es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, enero 2026.



Carmen Alexandra Rea Ramírez

C.C. 1850651983

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, enero 2026.



Carmen Alexandra Rea Ramírez
C.C. 1850651983
AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por la señorita Carmen Alexandra Rea Ramírez, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA DE PREPROCESAMIENTO DE DATOS SENSORIALES UTILIZANDO PYTHON EN LA GESTIÓN DE MANTENIMIENTO PREDICTIVO DE BOMBAS HIDRAÚLICAS EN LA PRODUCCIÓN DE CRUDO, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con el señor presidente del Tribunal.

Ambato, enero 2026.

Firmado electrónicamente por:
FRANKLIN OSWALDO MAYORGA MAYORGA
Validar únicamente con FirmaEC

Ing. Franklin Oswaldo Mayorga Mayorga, Mg.
PRESIDENTE DEL TRIBUNAL

Firmado electrónicamente por:
KLEVER RENATO URVINA BARRIONUEVO
Validar únicamente con FirmaEC

Ing. Klever Urvina Barrionuevo, Mg.
PROFESOR CALIFICADOR

Firmado electrónicamente por:
MARCELO VLADIMIR GARCIA SANCHEZ
Validar únicamente con FirmaEC

Ing. Marcelo García Sánchez, PhD.
PROFESOR CALIFICADOR

DEDICATORIA

Dedico este trabajo a las personas más importantes de mi vida, a quienes con su amor, apoyo y compañía han marcado cada paso de este camino y han hecho posible este sueño.

En primer lugar, a Dios, mi roca, mi fuerza y mi esperanza. A Él que me sostuvo en cada paso, me levanto en los momentos difíciles y me mostró que nada es imposible cuando camino de su mano. Este logro es fruto de su gracia.

A mis padres, Ángel y Carmen, a quienes les debo todo lo que soy. Su amor, esfuerzo y sacrificio han guiado mi vida. Han sido mi refugio, mi impulso y el ejemplo que me enseña cada día que los sueños se alcanzan con perseverancia y fe.

A mis hermanos, Daniel, Mirian y Sarita, por su amor y apoyo incondicional. Por creer en mí incluso cuando yo dudaba. Han sido una parte esencial en mi vida y nada habrá sido igual sin ustedes.

A mi sobrino, Esteban, cuya alegría ilumina incluso los días más cansados y me inspira a seguir adelante con ilusión y valentía.

Y a todos quienes han creído en mí. Gracias por su confianza, por sus palabras de ánimo y por acompañarme en cada paso.

Este logro es tan mío como suyo.

AGRADECIMIENTO

Agradezco, en primer lugar, a Dios, por sostenerme, por fortalecerme y por nunca soltar mi mano.

A mis padres, hermanos y a toda mi familia, por su amor incondicional, su paciencia infinita y por acompañarme en cada paso de este camino. Gracias por sostenerme cuando las fuerzas faltaban y por celebrar conmigo cada pequeño avance. Gracias por creer en mí, por ser mi refugio, mi impulso y mi hogar. En ustedes encontré la motivación, la esperanza y la fortaleza que me permitieron llegar hasta aquí.

Extiendo mi agradecimiento a cada miembro del proyecto de investigación "Desarrollo e Implementación de un Sistema de Mantenimiento Predictivo Basado en Inteligencia Artificial para Optimizar Equipos de Bombeo Hidráulico en la Producción de Crudo", por confiar en mis capacidades. Agradezco al Dr. Marcelo García por su guía constante, al Ing. Renato Urvina, Mg., por sus valiosos aportes que permitieron mejorar este trabajo, y en especial a mi tutor, el Ing. Edison Álvarez, Mg., por la confianza depositada en mí y por orientarme con dedicación durante el desarrollo de esta investigación.

También quiero agradecer a mis amigos, quienes han estado a mi lado brindándome compañía, apoyo emocional y momentos de respiro cuando más los necesité.

Finalmente, agradezco a la Universidad Técnica de Ambato por abrirme las puertas y por ser el espacio donde pude crecer, aprender y demostrar lo mejor de mí.

ÍNDICE GENERAL DE CONTENIDOS

PORTADA ..... **i**

APROBACIÓN DEL TUTOR ..... **ii**

AUTORÍA ..... **iii**

DERECHOS DE AUTOR ..... **iv**

APROBACIÓN DEL TRIBUNAL DE GRADO ..... **v**

DEDICATORIA ..... **vi**

AGRADECIMIENTO ..... **vii**

ÍNDICE GENERAL DE CONTENIDOS ..... **viii**

ÍNDICE DE TABLAS ..... **xii**

ÍNDICE DE FIGURAS ..... **xiv**

RESUMEN EJECUTIVO ..... **xvi**

ABSTRACT ..... **xvii**

CAPÍTULO I. MARCO TEÓRICO **1**

1.1 Tema de investigación 1

1.1.1 Planteamiento del problema 1

1.2 Antecedentes investigativos 2

1.3 Fundamentación teórica 7

1.3.1 Tecnologías de la Información (TI) 7

1.3.2 Ciencia de datos 7

1.3.3 Python para preprocesamiento de datos 8

1.3.4 Mantenimiento predictivo 8

1.3.5 Equipos de bombeo en la industria petrolera 8

1.3.6 Optimización de la gestión del mantenimiento 9

1.3.7 Análisis exploratorio de datos (EDA) 9

1.4 Objetivos 10

1.4.1 Objetivo general 10

1.4.2 Objetivos específicos 10

CAPÍTULO II. METODOLOGÍA 11

2.1 Materiales 11

2.2 Métodos 11

2.2.1 Modalidad de la investigación 11

2.2.2 Población y muestra 12

2.2.3 Recolección de información 12

2.2.4 Procesamiento y análisis de datos 20

CAPÍTULO III. RESULTADOS Y DISCUSIÓN 22

3.1 Análisis y discusión de resultados 22

3.2 Metodología aplicada al proyecto 22

3.2.1 Selección de metodología 22

3.2.2 Aplicación de metodología CRISP-DM 24

3.3 Fase 1: Comprensión del Negocio (Business Understanding) 26

3.4 Fase 2: Comprensión de los Datos (Data Understanding) 29

3.4.1 Fuente y origen de los datos sensoriales 29

3.4.2 Estructura y características de la tabla de datos crudos 30

3.4.3 Tipos de variables sensoriales 32

3.4.4 Análisis exploratorio de datos (EDA) 34

3.5 Fase 3: Preparación de Datos (Data Preparation) 51

3.5.1 Diseño del sistema de preprocesamiento de datos 52

3.5.2 Herramientas de software y librerías utilizadas 53

3.5.3 Fase 1: Detección y clasificación de anomalías 54

3.5.4 Fase 2: Corrección, imputación y reconstrucción temporal 64

3.5.5 Fase 3: Construcción de Características (Feature Engineering) 73

3.6 Fase 4: Modelado (Modeling) 78

3.6.1 Selección de Variables 78

3.6.2 Construcción del Dataset 79

3.6.3 Modelos de clasificación entrenados 79

3.6.4 Entrenamiento y validación 82

3.7 Fase 5: Evaluación (Evaluation) 85

3.8 Fase 6: Despliegue (Deployment) 87

3.8.1 Interoperabilidad con los módulos existentes 87

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES 89

4.1 Conclusiones 89

4.2 Recomendaciones 91

REFERENCIAS BIBLIOGRÁFICAS 92

ÍNDICE DE TABLAS

Tabla 1. Cadenas de búsqueda en base de datos científicas 3

Tabla 2. Cadenas de búsqueda aplicado en repositorio de universidades de Ecuador. 4

Tabla 3. Criterios de inclusión y exclusión. 4

Tabla 4. Ficha de contenido 1 12

Tabla 5. Ficha de contenido 2 13

Tabla 6. Ficha de contenido 3 14

Tabla 7. Ficha de contenido 4 14

Tabla 8. Ficha de contenido 5 15

Tabla 9. Ficha de contenido 6 16

Tabla 10. Ficha de Contenido 7 17

Tabla 11. Ficha de contenido 8 17

Tabla 12. Ficha de contenido 9 18

Tabla 13. Ficha de contenido 10 19

Tabla 14. Ficha de contenido 12 19

Tabla 15. Comparativa de metodologías analíticas 23

Tabla 16. Variables capturadas por el sensor ABB Ability™ Smart Sensor 32

Tabla 17. Clasificación de variables según el tipo de dato 34

Tabla 18. Resumen estadístico de variables continuas 40

Tabla 19. Resumen de valores atípicos en variables continuas 43

Tabla 20. Resumen de valores categóricos u ordinales 47

Tabla 21. Banderas generadas en diagnostico temporal 57

Tabla 22. Resumen de indicadores de calidad 61

Tabla 23. Métodos de imputación por tipo de variable 70

Tabla 24. Resumen de indicadores de preprocesamiento 70

Tabla 25. Matriz de confusión de Regresión Logística 84

Tabla 26. Matriz de confusión de Random Forest 85

ÍNDICE DE FIGURAS

Figura 1. Fases de la metodología CRISP-DM 25

Figura 2. ABB Ability Smart Sensor ubicado en la bomba hidráulica 27

Figura 3. Diagrama conceptual de adquisición de datos 29

Figura 4. Registros de la tabla de datos crudos INGESTAS 32

Figura 5. Matriz de correlaciones de variables continuas 35

Figura 6. Resumen global de dataset 37

Figura 7. Resumen del despliegue seleccionado 38

Figura 8. Resultado estadístico de Acceleration RMS (Radial) 39

Figura 9. Boxplot de Acceleration RMS (Radial) 44

Figura 10. Histograma de Acceleration RMS (Radial) 45

Figura 11. Gráfico lineal de Acceleration RMS (Radial) 46

Figura 12. Resumen estadístico de estructura temporal 49

Figura 13. Resumen de gaps detectados 50

Figura 14. Arquitectura del sistema de preprocesamiento de datos 53

Figura 15. Endpoints de API Resfull 55

Figura 16. Consulta por despliegue 55

Figura 17. Resumen de diagnóstico temporal del despliegue 57

Figura 18. Método de detección de valores faltantes 58

Figura 19. Método de detección de valores físicamente imposibles 58

Figura 20. Estadísticos de variables de vibración e IQR 59

Figura 21. Método de detección de categorías fuera de dominio 60

Figura 22. Método de detección de integridad de variables acumulativas 60

Figura 23. Registros de la tabla indicador_calidad 61

Figura 24. Resumen de calidad de datos 62

Figura 25. Almacenamiento de datos con indicador de calidad 64

Figura 26. Método de conversión de anomalías a NaN 65

Figura 27. Estadístico descriptivo temporal antes y después de corrección de jitter . 67

Figura 28. Dataset en formato ancho y reconstrucción de rejilla temporal 68

Figura 29. Métodos de imputación 69

Figura 30. Resumen de mediciones preprocesadas 71

Figura 31. Almacenamiento de mediciones limpias 73

Figura 32. Métodos de creación de características 76

Figura 33. Dataset con características nuevas 77

Figura 34. Almacenamiento de características en la base de datos 77

Figura 35. Parámetros principales de los modelos 82

Figura 36. Entrenamiento de modelos 82

Figura 37. Evaluación de métricas claves de modelos entrenados 83

Figura 38. Resumen de métricas de Regresión logística 83

Figura 39. Resumen de métricas claves de Random Forest 84

RESUMEN EJECUTIVO

La gestión del mantenimiento predictivo en la industria petrolera depende de la calidad, consistencia y confiabilidad de los datos sensoriales, los cuales suelen presentarse como flujos crudos con irregularidades temporales, pérdidas de información y variaciones no controladas que dificultan su análisis. Frente a este desafío, el propósito de este trabajo fue implementar un sistema de preprocesamiento de datos sensoriales utilizando Python en la gestión del mantenimiento predictivo de bombas hidráulicas en la producción de crudo. Para ello se adoptó la metodología CRISP-DM, que permitió estructurar el proceso desde la comprensión del problema hasta la preparación final de los datos para su integración en modelos predictivos.

El sistema desarrollado integra tres fases principales: diagnóstico y etiquetado de calidad, reconstrucción temporal con imputación controlada y generación de características estadísticas. Cada fase produce y almacena sus resultados en tablas independientes, garantizando la trazabilidad y el uso organizado de la información.

Como resultado, se obtuvo un sistema capaz de mejorar notablemente la calidad y utilidad de los datos sensoriales. El impacto de la calidad del dato fue demostrado al entrenar dos modelos sobre los datos de la Fase 1 donde Regresión Logística alcanzó una exactitud de 34.92%, mostrando su limitada capacidad frente al ruido, los outliers y la variabilidad temporal; mientras que Random Forest obtuvo 99.57%, confirmando que una mejor calidad de datos influye directamente en el aprendizaje y desempeño de los modelos. El sistema final quedó disponible para su uso por el equipo de investigación y para su ampliación en futuros trabajos.

Palabras claves: Calidad de datos, sensores industriales, mantenimiento predictivo, preprocesamiento de datos, CRISP-DM.

ABSTRACT

Predictive maintenance management in the oil industry heavily depends on the quality, consistency, and reliability of sensor data, which often appear as raw streams with temporal irregularities, information loss, and uncontrolled variations that hinder analysis. To address this challenge, the purpose of this work was to implement a sensor data preprocessing system using Python to support predictive maintenance of hydraulic pumps in crude oil production. The CRISP-DM methodology was adopted, structuring the process from problem understanding to the final preparation of data for integration into predictive models.

The developed system comprises three main phases: quality diagnosis and labeling, temporal reconstruction with controlled imputation, and statistical feature generation. Each phase produces and stores its results in independent tables, ensuring traceability and organized use of information.

As a result, the system significantly improved the quality and usefulness of sensor data. The impact of data quality was demonstrated by training two models on the Phase 1 dataset, where Logistic Regression achieved an accuracy of 34.92%, showing its limited ability to handle noise, outliers, and temporal variability, while Random Forest reached 99.57%, confirming that higher data quality directly influences model learning and performance. The final system is available for use by the research team and for future extension in subsequent work.

Keywords: Data quality, industrial sensors, predictive maintenance, data preprocessing, CRISP-DM.

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

SISTEMA DE PREPROCESAMIENTO DE DATOS SENSORIALES UTILIZANDO PYTHON EN LA GESTIÓN DE MANTENIMIENTO PREDICTIVO DE BOMBAS HIDRÁULICAS EN LA PRODUCCIÓN DE CRUDO

1.1.1 Planteamiento del problema

La industria petrolera a nivel mundial es un sector estratégico para la economía global, siendo el petróleo una de las principales fuentes de energía utilizadas en actividades industriales, comerciales y de transporte según la Agencia Internacional de la Energía (AIE). Sin embargo, enfrenta desafíos como la necesidad de optimizar la eficiencia operativa [1]. En este entorno, el mantenimiento predictivo ha emergido como una estrategia clave para anticipar fallas en equipos críticos, reducir tiempos de inactividad no planificados y minimizar riesgos operacionales y ambientales [2].

En Ecuador, el sector petrolero representa uno de los pilares fundamentales de la economía, contribuyendo con aproximadamente el 7% al Producto Interno Bruto emitido por el Banco Central del Ecuador en 2022 [3]. La optimización de los procesos de producción, especialmente en el mantenimiento de equipos como bombas hidráulicas utilizadas en la extracción de crudo, es esencial para garantizar la continuidad operativa y minimizar pérdidas económicas. Sin embargo, muchas operaciones aún dependen de métodos de mantenimiento correctivo y preventivo, lo que puede generar altos costos y tiempos de inactividad. La implementación de estrategias basadas en mantenimiento predictivo en empresas petroleras ecuatorianas se presenta como una necesidad urgente para modernizar los procesos, mejorar la eficiencia y garantizar la sostenibilidad de la producción [4].

Empresas como Sertecpet S.A. han desarrollado iniciativas para implementar soluciones innovadoras en la gestión de mantenimiento de equipos críticos. En colaboración con instituciones académicas como la Universidad Técnica de Ambato (UTA) y la Universidad Técnica de Cotopaxi (UTC), se desarrolla el proyecto

"Desarrollo e Implementación de un Sistema de Mantenimiento Predictivo Basado en Inteligencia Artificial para Optimizar Equipos de Bombeo Hidráulico en la Producción de Crudo", financiado por la SENESCYT. Este proyecto busca integrar herramientas de IA y análisis de datos para mejorar la eficiencia operativa en la industria petrolera nacional.

La implementación de tecnologías avanzadas, como el Internet de las Cosas (IoT) y la inteligencia artificial (IA), ha permitido a las empresas petroleras monitorear en tiempo real el estado de sus activos y predecir posibles fallas antes de que ocurran. Esto no solo mejora la confiabilidad de los equipos, sino que también contribuye a una operación más segura y eficiente [5]. Dentro del marco del proyecto mencionado, se ha identificado que los datos recopilados por sensores en bombas hidráulicas presentan inconsistencias, dispersión y valores atípicos, lo que dificulta la implementación efectiva de estrategias de mantenimiento predictivo. Estos problemas en la calidad de los datos pueden llevar a decisiones incorrectas y aumentar el riesgo de fallas no anticipadas en los equipos [6].

1.2 Antecedentes investigativos

Se llevó a cabo una revisión sistemática de la literatura con el objetivo de recolectar información académica y científica relacionada con el preprocesamiento de datos sensoriales, el uso de Python como herramienta tecnológica y su aplicación en la gestión de mantenimiento predictivo de bombas hidráulicas dentro del sector petrolero. Las cadenas de búsqueda se configuraron meticulosamente utilizando palabras clave derivadas de las variables dependiente e independiente, como son “data preprocessing”, “sensor data”, “predictive maintenance”, “python”, “oil production”, “hydraulic pumps”, en las bases de datos académicas más prominentes, Scopus y IEEE Xplore, entre otras. Estas búsquedas se realizaron en el período comprendido entre los años 2020 y 2025.

La Tabla 1 muestra los resultados obtenidos de las cadenas de búsqueda aplicados a diferentes repositorios académicos y científicos.

Tabla 1. Cadenas de búsqueda en base de datos científicas

Base de datos	Cadena de búsqueda	Núm. de artículos
SCOPUS	TITLE-ABS-KEY (("data preprocessing" OR "data cleaning" OR "data preparation") AND ("sensor data" OR "sensor-based monitoring" OR "sensory data") AND ("predictive maintenance" OR "condition-based maintenance" OR "maintenance forecasting")) AND PUBYEAR > 2020 AND PUBYEAR < 2025 AND LANGUAGE (english) AND (SUBJAREA (comp) OR SUBJAREA (engi))	23
	(TITLE-ABS-KEY (data AND preprocessing) AND ALL (sensor AND data) AND ALL (predictive AND maintenance)) AND PUBYEAR > 2020 AND PUBYEAR < 2025 AND (LIMIT-TO (LANGUAGE , "English")) AND (LIMIT-TO (SUBJAREA , "COMP") OR LIMIT-TO (SUBJAREA , "ENGI")) AND (LIMIT-TO (EXACTKEYWORD , "Data Preprocessing") OR LIMIT-TO (EXACTKEYWORD , "Machine Learning"))	161
IEEE XPLORE	(("All Metadata":data preprocessing) AND ("All Metadata":predictive maintenance)) AND (("Publication Year":2020) OR ("Publication Year":2021) OR ("Publication Year":2022) OR ("Publication Year":2023) OR ("Publication Year":2024) OR ("Publication Year":2025))	186
GOOGLE SCHOLAR	"data preprocessing" AND "sensor data" AND "predictive maintenance"	334
RESEARCHGATE	"data preprocessing" AND " oil production " AND "predictive maintenance"	1767
	Total	2471

Además, se amplió la búsqueda explorando repositorios académicos de instituciones en Ecuador con el fin de identificar investigaciones previas desarrolladas en contextos similares o que aporten perspectivas locales sobre el mantenimiento predictivo y el uso de tecnologías como Python en aplicaciones industriales, incluyendo la Universidad Técnica de Ambato (UTA), Universidad San Francisco de Quito (USFQ), Universidad Central del Ecuador (UCE) y la Universidad de Cuenca (UCUENCA) en el periodo comprendido desde el año 2020 al 2025.

La Tabla 2 muestra los resultados obtenidos de las cadenas de búsqueda aplicado en repositorios de universidades del Ecuador

Tabla 2. Cadenas de búsqueda aplicado en repositorio de universidades de Ecuador.

Universidad	Cadena de búsqueda	Núm. artículos
Universidad Técnica de Ambato	data preprocessing AND python	1
Universidad San Francisco de Quito	data preprocessing AND predictive maintenance AND python	2
Universidad Central del Ecuador	data preprocessing AND sensor data	3
Universidad de Cuenca	data preprocessing AND predictive maintenance	6
	Total	12

El objetivo de esta revisión bibliográfica fue proporcionar un análisis de los antecedentes relevantes relacionados con la tecnología y prácticas en el desarrollo de sistemas de preprocesamiento de datos sensoriales en el contexto de la industria de la petrolera. Se aplicaron criterios de inclusión y exclusión para garantizar la calidad y pertinencia de las fuentes recopiladas.

En la Tabla 3 se muestra los criterios de inclusión y exclusión.

Tabla 3. Criterios de inclusión y exclusión.

Criterios de inclusión	Criterios de exclusión
Investigaciones de acceso libre (open access)	Investigaciones anteriores al año 2020 (excepto si son base teórica fundamental).
Investigaciones que incluyan aplicación de mantenimiento predictivo en industria petrolera o equipos hidráulicos	Investigaciones centradas en otros sectores industriales no relacionados (agricultura, medicina, etc.).
Investigaciones que apliquen preprocesamiento de datos o Python en procesos industriales	Investigaciones que no aporten herramientas o conceptos aplicables al problema.
Estudios que aborden problemas de calidad de datos sensoriales.	Investigaciones de revisión que no incluyan datos o resultados prácticos

Como resultado de este proceso de selección, se identificaron y detallan cinco artículos que cumplen con los criterios establecidos y que se consideran fundamentales para enriquecer el marco teórico de esta investigación.

Cofre-Martel et al. desarrollaron una metodología integral de preprocesamiento de datos para sistemas de mantenimiento predictivo, orientada a maquinaria pesada en entornos industriales. El objetivo principal fue garantizar que los datos recopilados por sensores sean confiables, útiles y estén adecuadamente preparados para alimentar modelos de machine learning. El estudio evidencia que procesos como la limpieza de ruido, la detección de valores atípicos y la segmentación temporal, sumados al conocimiento experto del dominio, son cruciales para mejorar el desempeño de los

algoritmos. La metodología propuesta fue centrada en Modelos basados en Datos (DDM) aplicados a marcos de Pronóstico y Gestión de la Salud (PHM) en grandes volúmenes de datos provenientes de equipos industriales que permitió construir modelos predictivos más robustos. También se propuso una metodología para la generación de etiquetas de estado de salud para modelos de diagnóstico. Se analizaron dos casos de estudio de sistemas complejos reales: una unidad de recuperación de vapor (VRU) en una plataforma de producción de petróleo en alta mar y un sistema de trituración de minería de cobre (CMC). Los autores concluyen diciendo que un buen preprocesamiento tiene un impacto significativo en la precisión del diagnóstico y la predicción de fallos, reduciendo errores y costos operativos [7].

Ji-Hyun et al. desarrollaron una técnica de Mantenimiento Predictivo (PdM), un nuevo método de mantenimiento, para desarrollar un algoritmo que prediga fallas en robots de transferencia de obleas con anticipación. Los datos del sensor de aceleración utilizados en el experimento se obtuvieron instalando un sensor en el robot de transferencia de obleas. Para analizar estos datos, se realizó el preprocesamiento de datos y el proceso FFT. Estos datos se dividieron en datos normales, datos de primer error, datos de segundo error y datos de tercer error (datos de falla) en etapas. Mediante la agrupación de los datos con el algoritmo K-means, se analizó la distribución del punto central de los clústeres y se extrajeron las características de los datos de error y los datos normales. Con estas características, se diseñó un modelo de red neuronal artificial para predecir el punto de falla del robot. Investigaciones previas sobre sistemas de mantenimiento del robot de transferencia utilizaron menos de 50 datos de error, pero en este experimento se utilizaron 1686. La fiabilidad del modelo se mejora seleccionando aleatoriamente datos de un total de 2248 conjuntos. Además, se confirmó la posibilidad de clasificar datos normales y de error con una precisión del 97 % y predecir fallos del equipo mediante el modelado de redes neuronales [8].

Kutubuddin propuso una solución de mantenimiento predictivo basada en Python para el monitoreo de lubricantes industriales, centrándose en sensores que capturan propiedades fisicoquímicas como viscosidad y temperatura. El objetivo fue demostrar cómo herramientas libres permiten desarrollar sistemas de alerta temprana sobre degradación de componentes. La metodología incluyó recolección automatizada de datos sensoriales, limpieza con Pandas, visualización con Matplotlib y clasificación

con modelos como Random Forest y SVM. El autor destaca que Python facilita todo el flujo de trabajo, desde la adquisición de datos hasta la toma de decisiones, siendo especialmente útil en industrias que requieren soluciones económicas. Los resultados mostraron un aumento en la precisión de mantenimiento programado, reduciendo paradas inesperadas [9].

Wang et al. exploraron la aplicación de técnicas de ensemble learning como bagging y boosting en el mantenimiento predictivo de equipos de extracción de petróleo en China. El estudio tuvo como objetivo optimizar la identificación temprana de fallas en componentes críticos, utilizando datos históricos de presión, vibración, temperatura y flujo, capturados por sensores. Para asegurar la calidad de los modelos, se aplicó un preprocesamiento riguroso que incluyó interpolación, balanceo de clases y selección de características. Los algoritmos ensamblados mostraron mejoras significativas en la precisión del pronóstico frente a métodos tradicionales. Se destaca la relevancia del análisis multivariable y la integración del conocimiento experto para entrenar los modelos. Este antecedente apoya la viabilidad del uso de machine learning avanzado en sectores como el petrolero, donde las condiciones operativas son complejas y los errores pueden resultar costosos. Además, demuestra que el preprocesamiento personalizado, según las variables sensoriales, es determinante en la eficacia de los sistemas predictivos [10].

En un estudio centrado en la detección de anomalías en maquinaria indutrial, Chevetchenko et al. presentan un estudio sistemático sobre Detección de Anomalías (AD) para maquinaria industrial utilizando dispositivos IoT y algoritmos de Machine Learning (ML) para abordar esta brecha. El estudio evalúa exhaustivamente 84 estudios relevantes del período comprendido entre 2016 y 2023, ofreciendo una revisión exhaustiva de la investigación en AD. Los hallazgos identifican los algoritmos, las técnicas de preprocesamiento y los tipos de sensores más utilizados. El análisis sugiere que estos tipos de maquinaria suelen estar sujetos a condiciones que pueden provocar anomalías, como desgaste en las herramientas de fresado y corte, fluctuaciones de temperatura y presión en los sistemas hidráulicos, y fallas térmicas en los rodamientos. Se empleó una amplia gama de sensores para detectar anomalías. Si bien los trabajos principales a menudo no detallaban las características específicas de los sensores, destacaron los más utilizados, como los de vibración y temperatura.

La combinación de estos sensores se considera una estrategia común para detectar y monitorear eficientemente anomalías en máquinas. Además, se utilizó diversas técnicas de aprendizaje automático para el análisis de datos automatizado, incluyendo métodos supervisados, no supervisados y heurísticos. La combinación de estas técnicas, como la FFT y los autocodificadores, resulta eficaz para mejorar la precisión del AD [11].

1.3 Fundamentación teórica

1.3.1 Tecnologías de la Información (TI)

Las TI han transformado los procesos industriales mediante la integración de herramientas digitales capaces de adquirir, almacenar y procesar datos en tiempo real. Esta infraestructura digital, que incluye sensores IoT y sistemas de adquisición de datos, es fundamental para habilitar el mantenimiento predictivo, ya que facilita la captura y la posterior limpieza y análisis de grandes volúmenes de información. La implementación de sistemas de mantenimiento basados en inteligencia artificial requiere, como paso previo, un tratamiento riguroso de los datos, en especial en ambientes industriales donde el ruido y los valores atípicos pueden distorsionar los modelos de predicción [7].

1.3.2 Ciencia de datos

La Ciencia de Datos proporciona las técnicas que permiten convertir los datos en conocimiento útil. A través de métodos estadísticos, minería de datos y aprendizaje automático, los datos de sensores industriales como vibración, presión y temperatura se transforman en señales diagnósticas y patrones de fallo. Modelos como árboles de decisión, Random Forest, redes neuronales y técnicas de boosting (XGBoost, LGBM) han demostrado su efectividad en el pronóstico de fallas en bombas hidráulicas y otros sistemas mecánicos críticos [12]. Esta disciplina permite diseñar modelos entrenados con datos históricos que anticipan eventos de falla con precisión, optimizando la planificación del mantenimiento y reduciendo los tiempos de inactividad no planificados.

1.3.3 Python para preprocesamiento de datos

El lenguaje de programación Python se posiciona como una de las herramientas más versátiles y potentes en el campo del análisis de datos, gracias a su sintaxis clara y la amplia disponibilidad de bibliotecas especializadas como Pandas, NumPy, SciPy y Scikit-learn. Estas bibliotecas permiten realizar tareas clave como imputación de datos faltantes, normalización, detección de valores atípicos, escalado, y transformación de estructuras temporales. Su integración con entornos industriales y arquitecturas de IoT ha facilitado el desarrollo de sistemas predictivos en tiempo real para maquinaria pesada [13].

1.3.4 Mantenimiento predictivo

El mantenimiento predictivo se define como una estrategia basada en la supervisión continua de variables operativas, con el fin de anticipar fallas y ejecutar acciones de mantenimiento antes de que se presenten averías. Esta metodología se apoya en el análisis histórico y en tiempo real de datos sensoriales para generar modelos que identifican patrones de degradación [14]. En el caso particular de bombas hidráulicas, esta estrategia ha permitido una reducción considerable en paradas no planificadas, además de extender la vida útil de los equipos, mejorar la seguridad operativa y reducir costos [15].

1.3.5 Equipos de bombeo en la industria petrolera

Las bombas hidráulicas son elementos vitales dentro de la cadena de extracción y producción de petróleo, especialmente en aplicaciones como el bombeo de agua de inyección, levantamiento artificial y sistemas de distribución de crudo. Fallas en estos equipos pueden provocar la interrupción del proceso productivo y pérdidas económicas significativas. Por esta razón, la aplicación de modelos predictivos a partir de datos sensoriales (presión, caudal, temperatura, vibración, etc.) se ha convertido en una prioridad dentro de los sistemas de gestión del mantenimiento industrial [16].

1.3.6 Optimización de la gestión del mantenimiento

La optimización de la gestión del mantenimiento se logra mediante el uso de plataformas inteligentes que incorporan análisis avanzado de datos, algoritmos de predicción y herramientas de visualización. Estas soluciones permiten jerarquizar fallas, planificar intervenciones según criticidad, automatizar reportes técnicos y mejorar la toma de decisiones en áreas operativas. Estudios recientes evidencian que los sistemas de mantenimiento predictivo integrados con algoritmos de inteligencia artificial no solo mejoran el desempeño técnico, sino que también generan beneficios económicos y sostenibilidad operativa en sectores altamente demandantes como el petrolero [17].

1.3.7 Análisis exploratorio de datos (EDA)

EDA constituye una fase crítica en la cadena de procesamiento de datos sensoriales industriales porque permite identificar patrones, valores atípicos, distribución de variables y relaciones subyacentes antes de aplicar modelos predictivos. Una metodología asistida que registra las interacciones del analista durante EDA sobre datos de sensores industriales para reproducir y guiar futuras sesiones exploratorias demuestra que el EDA manual aporta en promedio un 86,5 % de los conocimientos generados por expertos, lo que evidencia su importancia para interpretar datos en entornos reales [18].

Un enfoque visual interactivo para que expertos de proceso puedan explorar, agrupar, etiquetar y clasificar datos sensoriales sin necesidad de codificación avanzada corresponde a EDA. Este enfoque facilita que los usuarios identifiquen anomalías, distribuciones y relaciones entre variables críticas en procesos industriales. El análisis visual y estadístico previo al modelado es esencial en ambientes industriales: ayuda a detectar inconsistencias, guiar decisiones de limpieza y transformación, y asegurar que los modelos predictivos trabajen con datos confiables [19].

El Análisis Exploratorio de Datos (EDA) se constituye como una metodología crucial para resumir las características principales de un conjunto de datos, identificar patrones, detectar valores atípicos y guiar el posterior modelado estadístico o de

Machine Learning. El lenguaje Python, junto con librerías como Pandas para la manipulación y Matplotlib, Seaborn para la visualización permite realizar de manera eficiente la inspección, limpieza y el análisis de los datos, transformando datos crudos en recursos comprensibles [20].

1.4 Objetivos

1.4.1 Objetivo general

Implementar un sistema de preprocesamiento de datos sensoriales utilizando Python en la gestión de mantenimiento predictivo de bombas hidráulicas en la producción de crudo.

1.4.2 Objetivos específicos

- Analizar el impacto de la calidad de los datos sensoriales en la gestión del mantenimiento predictivo de bombas hidráulicas en procesos de producción de crudo.

- Diseñar un sistema de preprocesamiento de datos sensoriales utilizando Python.

- Integrar los resultados de la presente investigación al proyecto institucional colaborativo “Desarrollo e Implementación de un Sistema de Mantenimiento Predictivo Basado en Inteligencia Artificial para Optimizar Equipos de Bombeo Hidráulico en la Producción de Crudo” de la Universidad Técnica de Ambato.

CAPÍTULO II. METODOLOGÍA

2.1 Materiales

En el presente proyecto de investigación se llevó a cabo la técnica de revisión bibliográfica/documental, utilizando como instrumento fichas de contenido que recolectan información relevante relacionada al tema de estudio, la cual proporcionó una base sólida para el análisis y desarrollo del proyecto.

2.2 Métodos

2.2.1 Modalidad de la investigación

La modalidad básica de investigación para este proyecto abarca tanto la investigación de campo como la experimental.

a. Investigación de Campo

La investigación adopta una modalidad de campo, ya que se basa en el análisis de datos reales obtenidos del entorno operativo de una empresa del sector petrolero (Sertepect S.A.). Aunque los datos no serán recolectados directamente por el investigador, su origen proviene de sistemas sensoriales que monitorean bombas hidráulicas utilizadas en la producción de crudo, lo que permite establecer una conexión directa con el fenómeno estudiado.

b. Investigación experimental

Se emplea la modalidad experimental, ya que se propone diseñar, desarrollar y validar un sistema de preprocesamiento de datos sensoriales utilizando Python. A través de esta modalidad, se comprobará el impacto de distintas técnicas de limpieza y transformación de datos en la efectividad del mantenimiento predictivo, utilizando pruebas controladas sobre conjuntos de datos reales. La experimentación permitirá verificar si el preprocesamiento mejora los resultados de los modelos de predicción aplicados a la gestión del mantenimiento.

2.2.2 Población y muestra

Debido a que la investigación se desarrolló en el marco de un proyecto institucional, no se estableció una población ni muestra convencional. En lugar de recolectar información de personas o grupos sociales, el estudio se basó en datos generados por sensores instalados en sistemas de bombeo hidráulico proporcionados directamente por la empresa Sertepect S.A.

Por tanto, la unidad de análisis de esta investigación estuvo compuesta por el conjunto de datos sensoriales entregados por los colaboradores del proyecto, sobre los cuales se aplicaron técnicas de limpieza, transformación y estructuración. Dado a que se trabajó con un volumen de datos ya definido y completo, no se requirió establecer un muestreo estadístico ni seleccionar una población específica para la recolección de información.

2.2.3 Recolección de información

Para la recolección de información se empleó la técnica de revisión bibliográfica-documental, en coherencia con la modalidad de la investigación, cuyo propósito fue analizar de manera crítica y sistemática la producción científica relacionada con el preprocesamiento de datos sensoriales y el mantenimiento predictivo en la industria petrolera. El instrumento utilizado para organizar los hallazgos fue fichas de contenido, en la cual se registraron elementos clave de cada referencia, este proceso permitió recopilar de manera estructurada información esencial acerca de técnicas de preprocesamiento, librerías de Python aplicadas al tratamiento de datos, estándares internacionales de mantenimiento, así como experiencias de implementación de sistemas predictivos en bombas hidráulicas utilizadas en la producción de crudo.

Tabla 4. Ficha de contenido 1

Ficha de contenido N°- 1	
Tema	Big machinery data preprocessing methodology for data-driven models in prognostics and health management [7].
Año de publicación	2021
Tesis	Se plantea un pipeline de preprocesamiento de datos sensoriales, fundamentado en la combinación de técnicas estadísticas y conocimiento experto, con el objetivo de mejorar la calidad de los datos y optimizar el

Ficha de contenido N°- 1	
	rendimiento de modelos de diagnóstico y mantenimiento predictivo en entornos industriales.
Ideas Centrales	Los datos sensoriales de maquinaria suelen presentar ruido, valores faltantes y anomalías que afectan los sistemas de mantenimiento predictivo. El estudio propone un proceso estructurado que incluye: selección de datos, limpieza, generación de etiquetas de estado de salud, reducción de dimensionalidad y validación. La metodología se valida en casos reales de sistemas industriales, destacando la necesidad de aplicar un preprocesamiento riguroso antes de entrenar modelos de pronóstico y diagnóstico.
Conceptos Clave	Data preprocessing, PHM (Prognostics and Health Management), Modelos basados en Datos (DDM), maquinaria industrial.
Conclusiones	La implementación de un pipeline estructurado de preprocesamiento mejora la confiabilidad y precisión de los modelos de mantenimiento predictivo. El conocimiento experto resulta fundamental para guiar el tratamiento de los datos sensoriales y garantizar la utilidad de los resultados en entornos reales de operación.
Aporte al tema	Este trabajo constituye un respaldo teórico y metodológico para la presente investigación, ya que evidencia que el preprocesamiento de datos sensoriales es un componente indispensable en la construcción de sistemas predictivos. Sus aportes permiten fundamentar la variable independiente del estudio y ofrecen un marco de referencia para el diseño del sistema propuesto en Python.
Observaciones	Aunque la investigación se orienta a maquinaria industrial en general, sus principios metodológicos son aplicables al contexto específico de bombas hidráulicas en la producción de crudo. La relevancia del artículo radica en que aborda directamente el problema central del proyecto, relacionado con la presencia de datos inconsistentes y dispersos en entornos industriales reales

Tabla 5. Ficha de contenido 2

Ficha de contenido N°- 2	
Tema	Predictive Maintenance System for Wafer Transport Robot Using K-Means Algorithm and Neural Network Model [8].
Año de publicación	2022
Tesis	El artículo propone un sistema de mantenimiento predictivo (PdM) para robots de transporte de obleas, basado en clustering K-Means para agrupar datos sensoriales y luego un modelo de red neuronal que predice fallas con alta precisión.
Ideas Centrales	Se utilizan datos de aceleración captados por sensores instalados en el robot, los cuales pasan por un proceso de preprocesamiento y transformación (FFT) para extraer características. Mediante clustering con K-Means, los datos se segmentan en categorías como normales y errores de distintos órdenes, permitiendo caracterizar los estados del equipo. Las características derivadas del clustering se emplean como entradas en una red neuronal (con varias capas ocultas), entrenada para clasificar estados normales vs. estados de error y anticipar fallas. El modelo alcanza una precisión cercana al 97 % al distinguir entre datos normales y de falla, lo que demuestra su viabilidad como sistema PdM.
Conceptos Clave	K-Means, red neuronal, preprocesamiento, FFT (Transformada Rápida de Fourier), clustering, datos sensoriales, mantenimiento predictivo, robot de transporte de obleas.

Ficha de contenido N°- 2	
Conclusiones	El sistema propuesto puede clasificar con alta precisión datos normales y de falla, permitiendo anticipar necesidades de mantenimiento antes de que ocurra una avería. La combinación de clustering con modelos supervisados es efectiva para el monitoreo predictivo en entornos automatizados.
Aporte al tema	Sus métodos: preprocesamiento, extracción de características, clustering y predicción pueden adaptarse al contexto de bombas hidráulicas en la producción de crudo usando Python, fortaleciendo la propuesta metodológica de esta investigación.
Observaciones	El artículo destaca el uso de datos reales y un volumen significativo de instancias (2248 datos en total) para entrenar el modelo, lo que refuerza la aplicabilidad del enfoque en contextos industriales más exigentes.

Tabla 6. Ficha de contenido 3

Ficha de contenido N°- 3	
Tema	A Python-Based Approach for Predictive Maintenance Condition Monitoring of Lubricants [9].
Año de publicación	2025
Tesis	Se propone una solución escalable y práctica basada en Python para realizar la monitorización de la condición de lubricantes, con el fin de identificar condiciones seguras o inseguras que anticipen fallas en los equipos.
Ideas Centrales	El enfoque utiliza Python y la biblioteca Pandas para manipular y analizar datos experimentales relacionados con la viscosidad de lubricantes. Los datos experimentales se procesan para distinguir entre condiciones normales y degradadas, apoyando decisiones de mantenimiento oportunas. El sistema permite visualización e interpretación eficiente de los datos, facilitando la identificación de tendencias de deterioro en lubricantes. Se destaca la escalabilidad del enfoque, lo que lo hace aplicable en distintos entornos industriales con capacidades de adaptación.
Conceptos Clave	Condition monitoring, lubricantes, Python, Pandas, viscosidad, mantenimiento predictivo, confiabilidad, escalabilidad
Conclusiones	El estudio concluye que el enfoque basado en Python optimiza el tratamiento de datos sensoriales para lubricantes, permitiendo predicciones útiles que ayudan a reducir fallas inesperadas y mejorar eficiencia operativa.
Aporte al tema	Este artículo aporta un caso concreto de cómo puede utilizarse Python para el análisis de datos sensoriales (en este caso de lubricantes) para mantenimiento predictivo. Aunque no trata directamente bombas hidráulicas, refuerza la viabilidad de aplicar un módulo de preprocesamiento automático con Python en entornos industriales.
Observaciones	Subraya la importancia de contar con datos fiables de viscosidad como indicador del estado de salud, lo que implica que futuros desarrollos deben contemplar sensores adecuados y calibrados para capturar parámetros relevantes en tu contexto de producción de crudo.

Tabla 7. Ficha de contenido 4

Ficha de contenido N°- 4	
Tema	Enhancing Predictive Maintenance Strategies for Oil and Gas Equipment through Ensemble Learning Modeling [10].

Ficha de contenido N°- 4	
Año de publicación	2025
Tesis	Se propone un método híbrido basado en aprendizaje por conjunto que integra redes neuronales convolucionales con meta-aprendizaje, reglas de expertos y SVM para evaluar el estado de salud de los equipos y predecir su vida útil restante, superando limitaciones de modelos únicos.
Ideas Centrales	La dependencia de un solo modelo de IA limita la precisión y robustez, especialmente cuando los datos presentan calidad variable. Se realiza un análisis de "big data" del historial operativo de equipos petroleros para extraer instancias de falla y agruparlas. Se construye un modelo híbrido que combina ML-CSNN (meta-learning convolutional shrinkage neural networks), reglas de dominio y SVM. El método fue validado con datos operativos del campo petrolero Shengli, alcanzando métricas como precisión = 0,98, precisión (precision) = 0,93, recall = 0,94 y F1 = 0,93.
Conceptos Clave	Ensemble learning, meta-learning, redes neuronales convolucionales (CNN), SVM, vida útil restante (Remaining Useful Life), diagnóstico híbrido.
Conclusiones	El enfoque híbrido propuesto mejora la precisión y robustez del mantenimiento predictivo en comparación con modelos individuales como GBDT o XGBoost, logrando mejoras entre 9 % y 25 % en las métricas de rendimiento. El modelo es capaz de anticipar fallas, apoyando estrategias predictivas más eficaces.
Aporte al tema	Sus resultados respaldan la idea de que un sistema de preprocesamiento es esencial para mejorar el desempeño de modelos complejos aplicados a la gestión del mantenimiento predictivo.
Observaciones	Aunque el estudio se aplica a equipos petroleros con condiciones específicas del campo Shengli (China), la metodología es transferible al entorno de bombas hidráulicas en producción de crudo. El alto desempeño obtenido refuerza la necesidad de aplicar un preprocesamiento riguroso, tal como plantea el problema central del proyecto (datos inconsistentes, dispersos y atípicos). Además, señal de la exigencia de buenas prácticas en la selección de modelos y en la integración de conocimientos expertos para maximizar el resultado.

Tabla 8. Ficha de contenido 5

Ficha de contenido N°- 5	
Tema	Anomaly Detection in Industrial Machinery Using IoT Devices and Machine Learning: A Systematic Mapping [11].
Año de publicación	2023
Tesis	El estudio presenta un mapeo sistemático de literatura que analiza cómo se emplean técnicas de Machine Learning e IoT para la detección de anomalías en maquinaria industrial, identificando los algoritmos más usados, técnicas de preprocesamiento y tipos de sensores, así como los desafíos y oportunidades futuras.
Ideas Centrales	Se evaluaron 84 estudios publicados entre 2016 y 2023 para mapear el uso de técnicas de detección de anomalías en entornos industriales. Se identificaron los algoritmos más comunes, las técnicas de preprocesamiento aplicadas y los tipos de sensores empleados en los casos estudiados. Se destaca que muchos estudios previos en detección de anomalías se enfocan en redes y seguridad informática, con menor atención al contexto industrial real.

Ficha de contenido N°- 5	
	El mapeo también resume las brechas existentes, como la falta de trabajos que integren preprocesamiento robusto en entornos IoT industriales complejos.
Conceptos Clave	Detección de anomalías, IoT industrial, aprendizaje automático, preprocesamiento de datos, sensores industriales, mapeo sistemático
Conclusiones	El estudio concluye que, aunque hay un creciente interés en la detección de anomalías en maquinaria industrial, muchos trabajos no abordan los desafíos del preprocesamiento de datos en entornos reales. Se recomienda más investigación en la integración de preprocesamiento, modelos escalables y validación en campo.
Aporte al tema	Subraya la necesidad de un módulo sólido de preprocesamiento antes de aplicar modelos predictivos en contextos industriales reales.
Observaciones	Al tratarse de un mapeo sistemático, el artículo no propone un modelo específico, pero es valioso para identificar tendencias, vacíos investigativos y técnicas emergentes que pueden orientarte en el diseño del sistema.

Tabla 9. Ficha de contenido 6

Ficha de contenido N°- 6	
Tema	Deep Learning Models for Predictive Maintenance: A Survey, Comparison, Challenges and Prospect [12].
Año de publicación	2022
Tesis	Se presenta un estudio exhaustivo de modelos de deep learning empleados en mantenimiento predictivo, categorizando arquitecturas, comparando su desempeño en diferentes etapas del ciclo de PdM y discutiendo los desafíos y direcciones futuras del campo
Ideas Centrales	El estudio revisa técnicas como CNN, LSTM, autoencoders y modelos generativos aplicados a la predicción de fallas e "Remaining Useful Life" (RUL). Identifica cómo cada arquitectura aborda distintas etapas del proceso de mantenimiento predictivo (detección de anomalías, diagnóstico, pronóstico, mitigación). Se discuten los retos de la generalización de modelos en entornos reales, la interpretabilidad, la escasez de datos etiquetados y la variabilidad del dominio industrial. Los autores proponen que el uso combinado de arquitecturas y técnicas de transferencia de conocimiento podría mitigar las limitaciones de modelos únicos.
Conceptos Clave	Deep learning, LSTM, CNN, autoencoders, RUL, diagnóstico de fallas, generalización, interpretación de modelos
Conclusiones	El análisis concluye que no existe una arquitectura universalmente ideal; cada modelo debe adaptarse al tipo de datos y condición operativa. Se recomienda investigar enfoques híbridos, técnicas de adaptación de dominio, y métodos que combinen conocimiento experto con aprendizaje automatizado para superar barreras en implementación industrial.
Aporte al tema	Proporciona directrices para seleccionar modelos apropiados para bombas hidráulicas y subraya la importancia del preprocesamiento de datos (problema central del proyecto) para que los modelos de deep learning funcionen adecuadamente.
Observaciones	Aunque el artículo tiene carácter de encuesta y no implementa un modelo específico, su valor radica en su visión general del estado del arte y en la identificación de brechas, especialmente en entornos industriales reales.

Ficha de contenido N°- 6	
	Reafirma la necesidad de que el sistema de preprocesamiento asegure alta calidad de datos antes de aplicar modelos complejos.

Tabla 10. Ficha de Contenido 7

Ficha de contenido N°- 7	
Tema	Prognostic Health Management of Pumps Using Artificial Intelligence in the Oil and Gas Sector: A Review [14].
Año de publicación	2022
Tesis	Este artículo revisa críticamente los avances en inteligencia artificial aplicada a la salud predictiva de bombas, identificando algoritmos, tipos de datos, retos y oportunidades en el pronóstico de fallas en el sector petrolero.
Ideas Centrales	Se analiza cómo los modelos de aprendizaje automático se aplican a diagnóstico y pronóstico en bombas, destacando que las redes neuronales (NN) son las más utilizadas, seguidas por Bayesian Networks, SVM y modelos híbridos. Se identifican los defectos de bombas más estudiados: fallas de rodamiento (≈ 46 %), cavitación, daño de sellos, entre otros. Señala que los datos más usados son vibración y flujo, pero estos pueden no ser suficientes para caracterizar el estado integral de una bomba. Se discuten los desafíos de la adopción amplia: falta de métodos end-to-end, necesidad de calibración de algoritmos según el equipo y los datos, y ausencia de estándares consolidados.
Conceptos Clave	PHM (Prognostic Health Management), diagnóstico de fallas, pronóstico, redes neuronales, SVM, datos sensoriales, vibración, flujo, bombas industriales, algoritmos híbridos.
Conclusiones	Los autores concluyen que la inteligencia artificial tiene un gran potencial para mejorar el mantenimiento predictivo de bombas en la industria petrolera, pero su implementación real se ve obstaculizada por retos como la calidad de los datos sensoriales, la falta de metodologías estándar y la adaptación de algoritmos al dominio específico de operación.
Aporte al tema	Este trabajo sustenta directamente la importancia del preprocesamiento de datos sensoriales, al evidenciar que uno de los limitantes principales en los estudios de PHM de bombas es precisamente la calidad y cantidad de datos disponibles. Además, proporciona un panorama de los algoritmos más comunes y los defectos más estudiados.
Observaciones	Aunque es un estudio de revisión, no propone un sistema concreto sino un análisis de estudios previos; sin embargo, su valor radica en identificar las brechas y retos más recurrentes en el campo, especialmente en la aplicación de IA a bombas en petróleo.

Tabla 11. Ficha de contenido 8

Ficha de contenido N°- 8	
Tema	An Ensemble Machine Learning Model for Predictive Maintenance on Water Injection Pumps in the Oil and Gas Industry [15].
Año de publicación	2024

Ficha de contenido N°- 8	
Tesis	Se propone e implementa un modelo de ensemble que combina los resultados de algoritmos XGBoost, LGBM y Random Forest para mejorar la predicción de fallas en bombas de inyección de agua, superando el desempeño de modelos individuales en precisión y confiabilidad
Ideas Centrales	Se monitorean cinco parámetros operativos críticos: presión de sello (Seal Pressure 1 y 2), vibración en extremo de accionamiento (VIB DE), vibración en el extremo no accionado (VIB NDE) y corriente (Ampere). Cada algoritmo (XGBoost, LGBM y Random Forest) se entrena por separado con esos parámetros para predecir el comportamiento de cada bomba. Se construye un modelo de ensemble que combina las predicciones individuales para mejorar métricas como R^2, MAE y RMSE. El modelo ensemble demostró un mejor desempeño general que los modelos independientes, ofreciendo resultados más estables y precisos.
Conceptos Clave	Ensemble learning, XGBoost, LGBM, Random Forest, mantenimiento predictivo, bombas de inyección de agua, parámetros operativos, predicción de fallas
Conclusiones	El estudio concluye que la combinación de múltiples modelos mediante el enfoque ensemble mejora significativamente la precisión de las predicciones frente a modelos individuales, logrando menores errores y mayor coeficiente de determinación. Esto valida la eficacia del sistema propuesto para mantenimiento predictivo en bombas de inyección de agua.
Aporte al tema	La investigación aporta un caso concreto cercano al ámbito de las bombas hidráulicas en el sector petrolero, demostrando que el uso de un sistema de preprocesamiento de datos sensoriales soportado por modelos ensemble puede elevar la confiabilidad y precisión de predicciones.
Observaciones	El artículo subraya la importancia de seleccionar parámetros críticos y elaborar un preprocesamiento riguroso para alimentar los modelos, lo cual coincide con la problemática central planteada (datos inconsistentes y atípicos).

Tabla 12. Ficha de contenido 9

Ficha de contenido N°- 9	
Tema	A Machine Learning Implementation to Predictive Maintenance and Monitoring of Industrial Compressors [16].
Año de publicación	2025
Tesis	Se presenta una integración de dispositivos IoT, adquisición de datos multi-sensor, minería estructurada de datos y algoritmos de machine learning (regresión lineal) para anticipar fallas en compresores industriales con alta precisión.
Ideas Centrales	Se instalan sensores para capturar parámetros como temperatura, presión y caudal en un compresor industrial. Los datos se almacenan en una base de datos SQL para su análisis. Se aplica un modelo de regresión lineal entrenado con los datos históricos para predecir potenciales fallas y emitir alertas cuando los valores exceden umbrales predeterminados. El modelo integrado con IoT permite una vigilancia continua con mínima intervención humana, transicionando de un mantenimiento reactivo a uno proactivo. En pruebas, el modelo logró una precisión del 98 % en error cuadrático medio (MSE), indicando su viabilidad para escenarios industriales reales.
Conceptos Clave	IoT, sensores múltiples, regresión lineal, almacenamiento SQL, alerta automática, mantenimiento predictivo, monitoreo continuo

Ficha de contenido N°- 9	
Conclusiones	El estudio demuestra que una integración bien estructurada de adquisición de datos, preprocesamiento y modelos de regresión puede predecir fallas con alta precisión, disminuir tiempos de inactividad y optimizar recursos de mantenimiento.
Aporte al tema	El uso de IoT y la arquitectura propuesta sirven como referencia para diseñar tu sistema en Python con un flujo de datos similar.
Observaciones	Aunque el modelo es relativamente simple al usar regresión lineal y no técnicas más complejas, el alto rendimiento obtenido (98 %) sugiere que un buen preprocesamiento puede compensar en muchos casos la simplicidad del modelo.

Tabla 13. Ficha de contenido 10

Ficha de contenido N°- 10	
Tema	Guided Exploration of Industrial Sensor Data [18].
Año de publicación	2024
Tesis	Se propone un sistema que registra las interacciones del experto durante el proceso de EDA sobre datos sensoriales industriales, y genera sesiones guiadas para analistas novatos, con el fin de replicar el conocimiento tácito extraído durante la exploración de datos.
Ideas Centrales	Los insights generados durante EDA como características clave de los datos industriales son esenciales para construir procesos guiados de análisis de datos. Dado lo complejo de los procesos productivos, los científicos de datos requieren colaboración continua con expertos del dominio para interpretar los datos correctamente. El sistema propuesto captura estados del sistema e interacciones del usuario para producir exploraciones guiadas reproducibles para otros casos
Conceptos Clave	Exploratory Data Analysis (EDA), interacción guiada, transferencia de conocimiento, datos sensoriales industriales, sesiones guiadas
Conclusiones	El enfoque permite documentar el conocimiento experto generado durante la exploración de datos y reutilizarlo en nuevos análisis, reduciendo la dependencia directa del dominio humano en cada instancia de EDA.
Aporte al tema	Los procesos exploratorios pueden sistematizarse y documentarse, lo que aporta mayor rigor al preprocesamiento de datos sensoriales en la investigación.
Observaciones	Aunque no se aplica directamente a bombas hidráulicas, el método es transferible a cualquier sistema con datos sensoriales industriales. Es útil para guiar a nuevos analistas al replicar rutas efectivas de exploración.

Tabla 14. Ficha de contenido 12

Ficha de contenido N°- 11	
Tema	Visual Analytics for Industrial Sensor Data Analysis [19].
Año de publicación	2021
Tesis	Se presenta un enfoque de visual analytics para que expertos del proceso puedan explorar, agrupar, etiquetar y analizar datos sensoriales industriales,

Ficha de contenido N°- 11	
	sin necesidad de conocimientos profundos en programación o gestión de datos.
Ideas Centrales	Se desarrolló una herramienta interactiva con visualizaciones específicas para la exploración de datos sensoriales, clustering y etiquetado. La herramienta facilita que expertos en procesos (no científicos de datos) participen en el análisis de datos industriales mediante visualización asistida. A través de casos de uso reales, se evaluó la utilidad de la interfaz con usuarios, quienes valoraron positivamente su capacidad para interpretar patrones sin escribir código.
Conceptos Clave	Visual analytics, EDA interactivo, expertos de proceso, etiquetado visual, datos sensoriales industrialesVisual analytics, EDA interactivo, expertos de proceso, etiquetado visual, datos sensoriales industriales
Conclusiones	El enfoque visual permite disminuir la brecha técnica entre los expertos del dominio y los científicos de datos, haciendo más accesible el análisis de datos sensoriales industriales. Su herramienta demostró utilidad en escenarios reales para facilitar el descubrimiento de patrones sin programación.
Aporte al tema	Esta ficha sostiene la necesidad de incorporar métodos visuales y exploratorios en el procesamiento de datos sensoriales.
Observaciones	El enfoque es aplicable a cualquier entorno industrial con sensores y es útil para reducir la dependencia técnica en el análisis preliminar de datos.

2.2.4 Procesamiento y análisis de datos

El procesamiento y análisis de datos en la presente investigación se fundamentó en la información obtenida a través de doce fichas de contenido que permitieron sistematizar el conocimiento existente respecto a las etapas de preprocesamiento de datos sensoriales, la gestión de mantenimiento predictivo y el impacto de la calidad de los datos en la precisión de los modelos de predicción de fallas.

Según las fichas de contenido 1 y 2, que sintetizan los estudios de Cofre-Martel et al. [7] y Yoo et al. [8], el preprocesamiento de datos sensoriales es una fase crítica para garantizar la confiabilidad de los sistemas predictivos, ya que la presencia de datos inconsistentes, dispersos o atípicos puede afectar significativamente la precisión de los modelos. De igual manera, los trabajos de Wang et al. [21] y Kutubuddin et al. [9], referenciados en las fichas 3 y 4, demuestran que el uso de bibliotecas en Python como Pandas, NumPy y SciPy permite desarrollar rutinas eficientes para la limpieza, normalización y transformación de los datos antes de su integración a los modelos predictivos.

Por otro lado, los resultados presentados en las fichas 5 y 6, basadas en los aportes de Chevtchenko et al. [11] y Serradilla et al. [12], destacan la importancia del Exploratory

Data Analysis (EDA) como fase intermedia entre la recolección y el modelado, al permitir identificar patrones, tendencias y correlaciones ocultas que orientan la depuración y la reducción de dimensionalidad de los conjuntos de datos. A su vez, los estudios de Langer y Meisen [18] [19], incorporados en las fichas de contenido 10 y 11, refuerzan que las metodologías de análisis exploratorio visual y guiado contribuyen a la comprensión contextual de los datos industriales y facilitan la toma de decisiones fundamentadas.

El análisis comparativo entre datos crudos y preprocesados evidenciado en las fichas 7 y 8 confirma que la depuración de registros sensoriales incrementa la estabilidad de los indicadores de salud de las bombas hidráulicas y reduce la varianza en los modelos de predicción [14].

Finalmente, la información analizada valida que el preprocesamiento adecuado de los datos sensoriales, mediante técnicas estadísticas y computacionales, constituye una etapa esencial para la implementación efectiva de modelos de mantenimiento predictivo en la producción de crudo.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de resultados

El presente capítulo presenta los resultados obtenidos durante el desarrollo del sistema de preprocesamiento de datos sensoriales aplicado a bombas hidráulicas en la producción de crudo. A través de la metodología CRISP-DM, se estructura el análisis desde la comprensión del problema de negocio y la naturaleza de los datos, hasta la preparación, depuración y organización de la información, necesaria para modelos futuros de diagnóstico y mantenimiento predictivo. El desarrollo del sistema no fue un proceso aislado, sino que se ejecutó en paralelo con la implementación de la Base de Datos (Time-Series) y la API RESTful (FastAPI), módulos desarrollados por otros miembros del equipo de investigación. Esta concurrencia en el desarrollo implicó una alta adaptabilidad metodológica, comunicación constante y la integración continua de ajustes, asegurando que todos los módulos fueran compatibles y funcionales entre sí.

3.2 Metodología aplicada al proyecto

3.2.1 Selección de metodología

El desarrollo del sistema de preprocesamiento de datos sensoriales requerido en esta investigación demanda una metodología capaz de gestionar de manera ordenada y sistemática las etapas de exploración, análisis, limpieza y transformación de datos provenientes de sensores industriales. Debido a la naturaleza del proyecto centrado en la preparación de series temporales, tratamiento de calidad de datos y estructuración de un pipeline ETL (Extract, Transform, Load) para uso posterior en modelos de mantenimiento predictivo se consideró necesario evaluar metodologías ampliamente empleadas en el ámbito del análisis de datos y la minería de información.

Para ello, se realizó una comparación técnica entre metodologías reconocidas en el campo, con la finalidad de determinar cuál se ajusta mejor a las exigencias del estudio. Las metodologías seleccionadas para la comparación fueron:

- CRISP-DM (Cross Industry Standard Process for Data Mining

- KDD (Knowledge Discovery in Databases)
- SEMMA (Sample, Explore, Modify, Model, Assess)
- TDSP (Team Data Science Process).

Estas representan los enfoques más utilizados en proyectos de análisis, descubrimiento de patrones y ciencia de datos, por lo que constituyen el marco adecuado para evaluar alternativas metodológicas.

La Tabla 15 muestra la comparativa de metodologías analíticas.

Tabla 15. Comparativa de metodologías analíticas

Característica	CRISP-DM	KDD	SEMMA	TDSP
Fases o componentes principales	Comprensión del negocio, comprensión de los datos, preparación, modelado, evaluación, despliegue	Selección, preprocesamiento, transformación, minería de datos, interpretación	Sample, Explore, Modify, Model, Assess	Ingesta, exploración, modelado, evaluación, implementación
Orientación técnica	Estándar industrial para minería de datos; flexible y adaptable a distintos tipos de datos	Enfoque profundo en descubrimiento de patrones y extracción de conocimiento	Marco operacional enfocado en modelamiento estadístico dentro del ecosistema SAS	Proceso estructurado para proyectos de ciencia de datos colaborativos
Ventajas principales	Fuerte énfasis en la comprensión y preparación de los datos; adaptable; independiente de herramientas	Rigurosidad analítica en etapas de minería de datos	Ideal para exploración estadística y modelamiento rápido	Alto nivel de formalidad y soporte para gestión de proyectos grandes
Limitaciones	No especifica técnicas concretas; requiere interpretación contextual	Menor orientación al negocio y a la calidad del dato crudo; menos práctico para entornos industriales	Dependencia del entorno SAS; limitado para series temporales industriales	Metodología compleja y pesada para proyectos individuales
Uso ideal o aplicación típica	Proyectos donde la preparación del dato es crítica; construcción de pipelines ETL; análisis industrial	Descubrimiento de patrones y análisis predictivo en bases de datos	Modelos predictivos en SAS; análisis exploratorio	Proyectos de ciencia de datos con equipos multidisciplinarios

La comparación realizada evidencia que CRISP-DM es la metodología que mejor responde a las características del presente estudio, tanto por su estructura como por su enfoque analítico.

3.2.2 Aplicación de metodología CRISP-DM

CRISP-DM (Cross-Industry Standard Process for Data Mining) en español significa Proceso Estándar Intersectorial para la Minería de Datos es un estándar ampliamente adoptado en proyectos de análisis y minería de datos debido a su estructura flexible, iterativa y orientada a la comprensión y preparación exhaustiva de la información. El modelo se compone de seis fases que se retroalimentan entre sí: comprensión del negocio, comprensión de los datos, preparación de datos, modelado, evaluación y despliegue [22].

Fases de CRISP-DM:

1. *Comprensión del negocio*: En esta primera fase se identifican los objetivos estratégicos del proyecto y los problemas que se desean resolver. Se define qué decisión se quiere mejorar y que beneficio se busca obtener y cómo el análisis de datos puede aportar valor al negocio o proceso [22].

2. *Comprensión de los datos*: Esta fase se encarga de recolectar los datos iniciales, analizarlos de manera exploratoria y evaluar su calidad. Se identifican patrones preliminares, inconsistencias, valores faltantes y posibles problemas que podrían influir en el modelado [22].

3. *Preparación de los datos*: Esta fase realiza la limpieza, transformación, integración y selección de datos relevantes. Aquí se construye el conjunto de datos final que se utilizará para entrenar el modelo, aplicando técnicas como imputación, normalización o creación de nuevas variables [22].

4. *Modelado*: En esta fase se seleccionan técnicas de modelado adecuadas, se entrenan distintos modelos y se ajustan sus parámetros. El objetivo es construir una solución capaz de predecir, clasificar o describir patrones relevantes [22].

5. *Evaluación*: Los modelos desarrollados se analizan para comprobar su desempeño, consistencia y utilidad. Se valida si los resultados cumplen los

objetivos definidos en la fase inicial y si los modelos son confiables para su implementación [22].

6. *Despliegue*: Consiste en llevar el modelo a la práctica, integrarlo en los procesos operativos, generar reportes, automatizarlo o ponerlo en producción. También incluye monitoreo y mantenimiento para asegurar su utilidad continua [22].

Estas seis fases permiten planificar, ejecutar y validar un proyecto de manera sistemática. CRISP-DM destaca por su carácter iterativo y flexible, permitiendo retroalimentaciones continuas entre fases para mejorar la calidad del análisis. Además, su independencia de herramientas, sectores o tipos de datos la ha convertido en un estándar ampliamente adoptado por organizaciones y plataformas analíticas, este modelo actúa como un nuevo plano de referencia para estructurar proyectos de data mining [23].

La Figura 1 muestra de manera esquemática la estructura del modelo CRISP-DM.

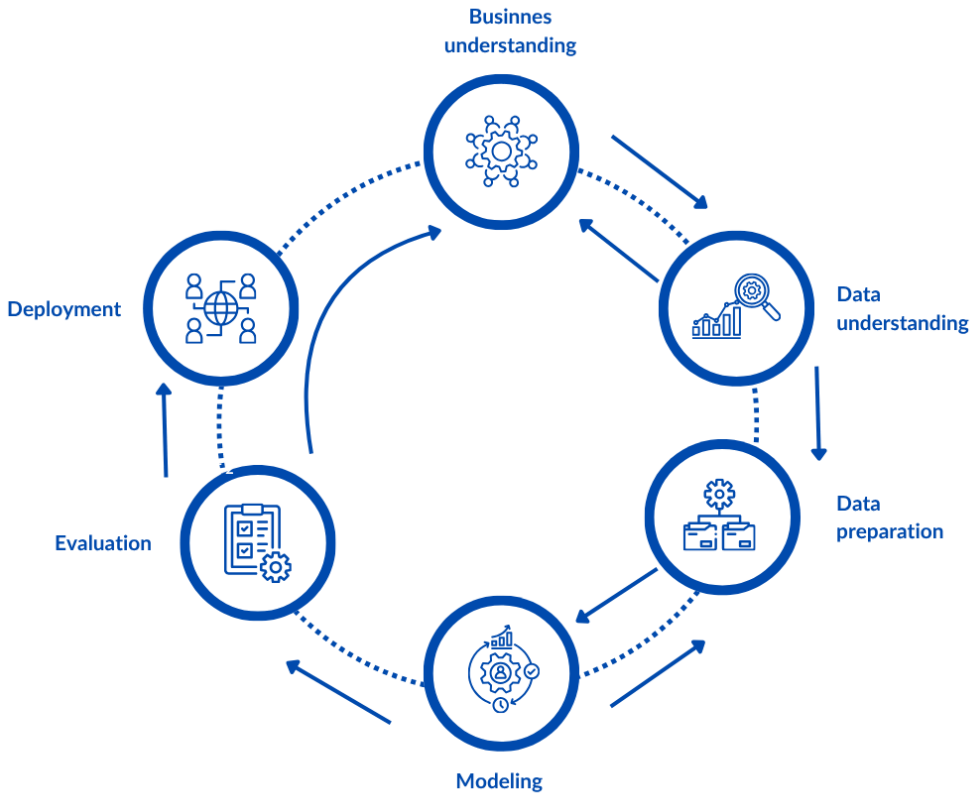


Figura 1. Fases de la metodología CRISP-DM

3.3 Fase 1: Comprensión del Negocio (Business Understanding)

El presente proyecto forma parte del estudio institucional denominado "*Desarrollo e Implementación de un Sistema de Mantenimiento Predictivo Basado en Inteligencia Artificial para Optimizar Equipos de Bombeo Hidráulico en la Producción de Crudo*", desarrollado en colaboración con la Universidad Técnica de Ambato, la Universidad Técnica de Cotopaxi y la empresa Sertecpet S.A, entidad ecuatoriana especializada en soluciones tecnológicas aplicadas al sector petrolero. El objetivo general de este proyecto es diseñar e implementar un sistema inteligente capaz de monitorear de manera continua el estado operativo de bombas hidráulicas, identificar patrones de deterioro y anticipar fallas mediante algoritmos de aprendizaje automático, con el fin de reducir costos operativos, minimizar tiempos de inactividad y mejorar la sostenibilidad de los procesos extractivos.

Las prácticas de mantenimiento actualmente utilizadas en la industria tradicionalmente se basan en mantenimiento correctivo, preventivo y rutinas de inspección manual, como mediciones puntuales de vibración con vibrómetros portátiles o reportes termográficos mensuales, aunque algunas de estas actividades suelen clasificarse como predictivas resultan insuficientes para detectar de forma temprana fallas incipientes en equipos críticos. Estos métodos se basan en intervalos fijos y no en la condición real del equipo, lo que genera limitaciones importantes como intervenciones tardías, mayores riesgos de fallos inesperados y elevados costos por paradas no programadas. Frente a este contexto, la transición hacia un enfoque de mantenimiento predictivo basado en inteligencia artificial constituye una necesidad urgente para optimizar la operación de las bombas hidráulicas y mejorar la toma de decisiones técnicas en continuo.

Dentro de este proyecto institucional, el presente trabajo se ubica en la fase inicial del flujo de procesamiento, desempeñando un rol fundamental que corresponde a la construcción de un sistema de preprocesamiento de datos sensoriales que garantice que los modelos de inteligencia artificial dispongan de datos limpios, coherentes, confiables y representativos del comportamiento real de los equipos monitoreados. Específicamente, este módulo toma como insumo los datos crudos generados por el sensor industrial ABB Ability™ Smart Sensor (Firmware 6.4.7).

La Figura 2 muestra el sensor ubicado en la bomba hidráulica.



Figura 2. ABB Ability Smart Sensor ubicado en la bomba hidráulica

El flujo de adquisición de datos sigue el mecanismo propio de la plataforma ABB Ability Digital Powertrain, el sensor captura las señales del motor y transmite los datos a un módulo intermedio o gateway, que posteriormente los sincroniza y sube a la nube de la plataforma, desde donde son descargados manualmente en archivos Excel debido a que no se proporcionaron las credenciales necesarias para acceder a la API oficial. Estos datos, previamente ingeridos en la base de datos times series, representan la fuente primaria utilizada en este estudio.

Durante el análisis inicial se identificaron múltiples problemas de calidad propios del entorno industrial: ruido, valores faltantes, registros físicamente imposibles, picos abruptos producidos por el desmontaje y montaje del sensor, dado que el mismo dispositivo fue rotado entre motores distintos, desincronización entre variables, gaps temporales y diferencias irregulares en los intervalos de muestreo. Asimismo, se detectó que los timestamps de las 22 variables no siempre coincidían en un mismo instante, generando inconsistencias que imposibilitan utilizar los datos directamente en modelos de aprendizaje automático. Entrenar un modelo predictivo con datos crudos que contienen series no continuas, ruido sin tratar o mediciones no representativas del estado real del motor puede derivar en predicciones incorrectas,

falsas alarmas, o la incapacidad del modelo para aprender patrones robustos de deterioro.

En este contexto, el sistema propuesto en esta investigación constituye un aporte esencial, pues establece un mecanismo estructurado de preprocesamiento compuesto por dos fases:

1. Detección y etiquetado de anomalías, donde los datos crudos son analizados sin modificaciones previas y se les asigna un indicador de calidad según el tipo de problema detectado.

2. Corrección e imputación, donde se reconstruye la serie temporal, se resuelven las inconsistencias, se sincronizan los registros de todas las variables, se imputan valores faltantes y se corrigen valores atípicos o físicamente inválidos.

Una contribución significativa de este trabajo es la preservación explícita de la trazabilidad mediante dos tablas en la base de datos, una para almacenar los datos originales con sus indicadores de calidad y otra para registrar los datos ya limpios y preprocesados. Este diseño permite comprender qué problemas presentaba cada medición y cómo fue tratada durante el proceso de limpieza, facilitando el análisis de auditoría del dato y el trabajo colaborativo con otros módulos del proyecto.

Finalmente, los datos limpios sirven como base para la generación de características, y la posterior aplicación de un modelo de clasificación sencillo orientado exclusivamente a analizar el impacto de la calidad del dato, cumpliendo así los objetivos planteados de este tema de estudio. Cabe recalcar que este proyecto no pretende predecir fallas reales, sino establecer la infraestructura técnica necesaria para que futuros modelos predictivos operen sobre datos de alta calidad.

En síntesis, esta fase de comprensión del negocio evidencia que la calidad del dato es un factor crítico en sistemas de mantenimiento predictivo con inteligencia artificial. El módulo desarrollado en esta investigación responde a esta necesidad, garantizando consistencia, completitud y fiabilidad en las series temporales que serán utilizadas por los modelos del proyecto institucional y aportando un mecanismo sólido de preparación de datos sensoriales para aplicaciones avanzadas en el sector petrolero.

3.4 Fase 2: Comprensión de los Datos (Data Understanding)

3.4.1 Fuente y origen de los datos sensoriales

La adquisición de datos en esta investigación se enmarca en la arquitectura de datos del Proyecto Institucional, originalmente los datos provienen del ABB Ability™ Smart Sensor, un dispositivo industrial diseñado para monitorear continuamente el estado operativo de motores eléctricos mediante la medición de parámetros mecánicos, energéticos y ambientales. Debido a que no fue posible acceder a la API oficial de la plataforma ABB por falta de credenciales corporativas los datos fueron descargados manualmente en archivos Excel y posteriormente ingeridos en una base de datos de series de tiempo (TimescaleDB) desarrollada como parte del proyecto institucional.

La Figura 3 muestra el diagrama conceptual de adquisición de datos

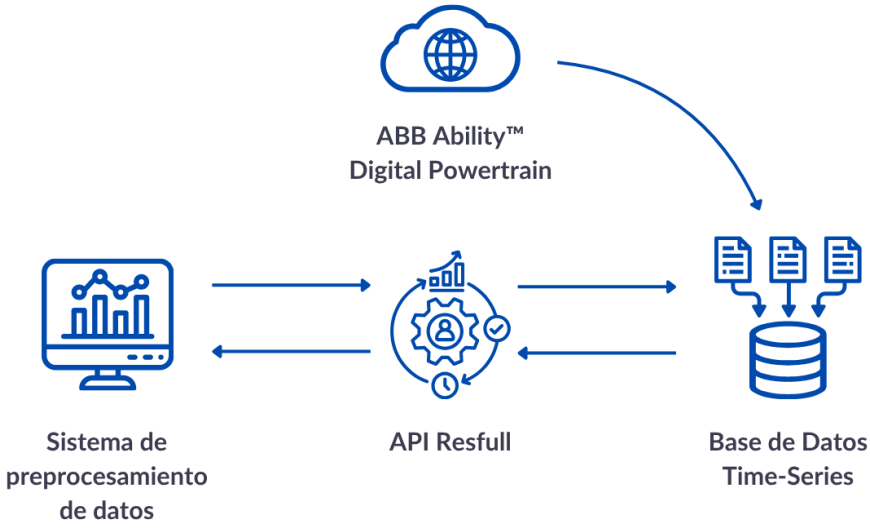


Figura 3. Diagrama conceptual de adquisición de datos

Esta arquitectura se compone de tres módulos principales, de los cuales el presente estudio se integra al inicio de la cadena de procesamiento:

Base de Datos de Series de Tiempo (Time-Series Database): Este módulo fue implementado utilizando PostgreSQL con la extensión TimescaleDB. Es el repositorio

central que almacena la información sensorial de las bombas hidráulicas, optimizado para el manejo eficiente de datos temporales (time-series).

Interfaz de Acceso (API RESTful): Desarrollada utilizando FastAPI, esta API sirve como un intermediario seguro y estandarizado para la consulta de datos. Su propósito es encapsular la lógica de acceso a la base de datos y estandarizar el formato de salida. Esta API consume datos en tiempo real mediante protocolos industriales como MQTT y OPC UA, garantizando que los datos disponibles para el preprocesamiento sean los más actuales y relevantes.

Sistema de Preprocesamiento de Datos: Constituye el alcance central de este estudio, siendo el módulo que consume los datos crudos de la BASE DE DATOS TIMES SERIES a través de la API RESTFULL para aplicar métodos de limpieza, corrección y feature engineering antes de ser utilizados en el modelado predictivo.

Para este estudio se recibieron aproximadamente 140.000 registros, correspondientes al monitoreo de dos despliegues del sensor en dos motores distintos. Esto incluye un historial completo de mediciones tomadas a intervalos de 15 minutos, con un total de 22 variables sensoriales por timestamp. El conjunto de datos resultante constituye la base primaria sobre la cual se desarrolló el sistema de preprocesamiento.

Los datos fueron almacenados en TimescaleDB en una estructura diseñada específicamente para soportar volúmenes altos de series temporales y permitir consultas eficientes. Cada registro representa una única medición para una variable particular en un timestamp determinado. A continuación, se describe la estructura general de la tabla utilizada.

3.4.2 Estructura y características de la tabla de datos crudos

La tabla de almacenamiento en la base de datos de series de tiempo utiliza un formato largo (long format) con el nombre INGESTAS, donde cada fila corresponde a un timestamp, una variable y un valor. Este formato es estándar en sistemas TSDB y facilita la escalabilidad y el ingestamiento de datos provenientes de sensores industriales. La tabla contiene las siguientes columnas principales:

- **ingesta_id(BIGSERIAL)**: Identificador interno autoincremental utilizado como clave técnica para auditoría y control del orden de ingesta. No representa un identificador lógico de medición.

- **asset_codigo (VARCHAR 50)**: Código del sensor ABB que generó la medición. Permite diferenciar registros cuando existen múltiples dispositivos en la planta o cuando el sensor se desplaza entre motores.

- **motor_codigo (VARCHAR 50)**: Identificador del motor monitoreado. En este estudio, este campo es especialmente relevante debido a que el sensor fue rotado entre dos equipos en distintos períodos, lo que introduce variabilidad en el comportamiento registrado.

- **ts_utc (TIMESTAMPTZ)**: Marca temporal original en UTC proveniente del archivo exportado. Es el eje principal de indexación y particionamiento en la hypertable de TimescaleDB.

- **ts_local_tz (TIMESTAMP)**: Columna generada automáticamente que convierte ts_utc a la zona horaria America/Guayaquil. Esto facilita análisis contextualizados según la operación real de la maquinaria.

- **variable (VARCHAR 70)**: Nombre de la variable sensorial medida. Corresponde a una de las 22 señales del ABB Smart Sensor.

- **valor (NUMERIC)**: Medición asociada a la variable. Según la variable, este valor puede representar vibración, aceleración, temperatura, frecuencia, condición del rodamiento, entre otras magnitudes físicas o indicadores de estado.

- **archivo_origen (VARCHAR 100)**: Nombre del archivo excel desde el cual se importó la medición. Este campo permite identificar el origen del lote y facilita reprocesos o auditorías.

- **recibido_en (TIMESTAMPTZ)**: Fecha y hora en que el dato fue ingresado en TimescaleDB. Permite calcular latencias entre captura y almacenamiento.

La tabla raw.ingestas no representa datos puramente crudos, sino datos enriquecidos durante la ingesta manual, donde se asigna cada tramo al motor y sensor correspondiente

La Figura 4 muestra registros de la tabla INGESTAS

ingesta_id bigint	asset_codigo character var	motor_codigo character varying	ts_utc timestamp with time zone	ts_local_tz timestamp without tim	variable character varying (70)	valor numeric	archivo_origen character varying (100)
120121	37156	Motor_002	2025-10-08 19:04:01-05	2025-10-08 19:04:...	Acceleration RMS (Axial)	1.584	acc_rms_axial.csv
120122	37156	Motor_002	2025-10-08 19:04:01-05	2025-10-08 19:04:...	Acceleration RMS (Radial)	0.7437	acc_rms_radial.csv
120123	37156	Motor_002	2025-10-08 19:04:01-05	2025-10-08 19:04:...	Acceleration RMS (Tange...	1.3213	acc_rms_tangential.csv
120124	37156	Motor_002	2025-10-08 19:04:01-05	2025-10-08 19:04:...	Bearing Condition	3.0	bearing_condition.csv
120125	37156	Motor_002	2025-10-08 19:04:01-05	2025-10-08 19:04:...	Looseness (BETA)	0.0	looseness_beta.csv
120126	37156	Motor_002	2025-10-08 19:04:01-05	2025-10-08 19:04:...	Misalignment (BETA)	0.0	misalignment_beta.csv
120127	37156	Motor_002	2025-10-08 19:04:01-05	2025-10-08 19:04:...	Motor Supply Frequency	50.5	motor_supply_freq.csv
120128	37156	Motor_002	2025-10-08 19:04:01-05	2025-10-08 19:04:...	Number of Starts Betwee...	0.0	starts_between_meas.csv
120129	37156	Motor_002	2025-10-08 19:04:01-05	2025-10-08 19:04:...	Output Power	118.18	output_power.csv
120130	37156	Motor_002	2025-10-08 19:04:01-05	2025-10-08 19:04:...	Overall Vibration	3.695	overall_vibration.csv
120131	37156	Motor_002	2025-10-08 19:04:01-05	2025-10-08 19:04:...	Peak to Peak	3.9199	peak_to_peak.csv

Figura 4. Registros de la tabla de datos crudos INGESTAS

3.4.3 Tipos de variables sensoriales

El sensor registra un conjunto de variables relacionadas con el comportamiento mecánico, térmico y vibracional de los motores. Estas variables pueden clasificarse en numéricas continuas, acumulativas y categóricas o de diagnóstico, según su naturaleza y forma de representación.

La Tabla 16 detalla las métricas capturadas por el sensor y su relevancia para la detección temprana de fallas.

Tabla 16. Variables capturadas por el sensor ABB Ability™ Smart Sensor

Variable	Unidad	Descripción	Relevancia para mantenimiento predictivo
Acceleration RMS (Axial/Radial/Tangential)	g RMS	Aceleración promedio por componente en los ejes axial, radial y tangencial.	Detecta desalineación del eje, problemas de montaje, y resonancias mecánicas en etapas tempranas.
Bearing Condition	Indicador numérico	Evaluación del estado del rodamiento calculado por el sensor.	Valores descendentes advierten desgaste o posible fallo del rodamiento.
Looseness (BETA)	Indicador numérico	Detecta holguras mecánicas (tornillos sueltos, piezas flojas).	Permite identificar estructuras sin fijación adecuada que generan vibraciones irregulares.
Misalignment (BETA)	Indicador numérico	Indicador de desalineación angular	Valores altos sugieren desalineación del acoplamiento, lo que genera desgaste acelerado.

Variable	Unidad	Descripción	Relevancia para mantenimiento predictivo
		entre el motor y la bomba.	
Motor Supply Frequency	Hz	Frecuencia eléctrica de alimentación del motor.	Permite diagnosticar fallas eléctricas (variadores, armónicos, fluctuaciones de carga).
Number of Starts Between Measurements	Indicador numérico	Arranques ocurridos entre dos mediciones consecutivas.	Ayuda a correlacionar incrementos de vibración o temperatura con ciclos de arranque.
Output Power	kW	Potencia real entregada por el motor a la bomba.	Cambios abruptos indican sobrecarga hidráulica, cavitación o fallas en la bomba.
Overall Vibration	mm/s RMS	Representa el nivel global de vibraciones del motor en todas las direcciones.	Un aumento progresivo indica desgaste de rodamientos, desbalanceo o desalineación.
Peak-to-Peak (Axial/Radial/ Tangential)	g	Diferencia entre valores máximos y mínimos de vibración en cada eje.	Indica impactos mecánicos, holgura o golpes internos característicos de rodamientos dañados.
Skin Temperature	°C	Temperatura externa del motor.	Incrementos indican fricción excesiva, sobrecarga eléctrica, lubricación deficiente o bloqueos mecánicos.
Speed	RPM	Mide la velocidad de rotación del eje del motor.	Cambios bruscos indican problemas de carga, fallas en el variador o condiciones anormales de operación.
Total Number of Starts	Conteo	Número de arranques acumulados del motor.	Un número elevado afecta los rodamientos y el aislamiento del motor.
Total Running Time	h	Total acumulado de horas operativas.	Se usa como variable de salud, útil para calcular ciclos de vida y fallas por desgaste.
Unbalance (BETA)	Indicador númerico	Representa el desbalance rotacional del motor.	Un desbalance provoca aumento de vibraciones y acorta vida útil de rodamientos.
Vibration (Axial/Radial /Tangential)	mm/s RMS	Medición directa de vibración en cada eje del motor.	Permite identificar desbalanceo (radial), desalineación (axial) o problemas estructurales.

Cada variable tiene un comportamiento particular y niveles típicos asociados al estado del motor. Por ello, las variables fueron agrupan en cuatro categorías funcionales, cada una con un esquema de limpieza específico y fundamentado en criterios de ingeniería, comportamiento físico y lineamientos de la literatura en mantenimiento predictivo.

Tabla 17. Clasificación de variables según el tipo de dato

Grupo de variables	Variables que contiene
Continuas físicas de vibración	• Vibration (Axial, Radial, Tangential y general) • Acceleration RMS (Axial, Radial, Tangential) • Peak-to-Peak (Axial, Radial, Tangential y general)
Continuas físicas no relacionadas con vibración	• Skin Temperature • Motor Supply Frequency • Speed • Output Power
Categóricas o diagnósticas	• Bearing Condition (0–4) • Unbalance (BETA) (0-1) • Misalignment (BETA) (0-1) • Looseness (BETA) (0-1) • Number of Starts Between Measurements (0–1)
Acumulativas (contadores)	• Total Running Time • Total Number Of Starts

Para esta investigación se tuvieron:

- N° total de registros: ~140.000
- N° de variables: 22
- N° de despliegues: 2
- N° de motores monitoreados: 2
- Intervalo de muestreo: 15 minutos
- Periodo total monitoreado: aproximadamente tres meses (agosto, septiembre, octubre)

El volumen, complejidad y variedad del dataset justifican el uso de técnicas de preprocesamiento y la aplicación de CRISP-DM como metodología de trabajo.

3.4.4 Análisis exploratorio de datos (EDA)

El análisis exploratorio de datos constituye el núcleo de esta fase, ya que permite comprender cómo se distribuyen las mediciones en el tiempo, cómo se comportan las variables bajo condiciones normales y qué anomalías aparecen recurrentemente. Para ello se desarrolló un cuaderno Jupyter utilizando Python y la biblioteca Pandas donde se generaron visualizaciones y estadísticas descriptivas utilizadas en este apartado.

En este proyecto, el EDA se orientó exclusivamente a evaluar la calidad del dato, identificar patrones de anomalías y comprender la estructura temporal de las mediciones registradas por el sensor ABB Ability™ Smart Sensor.

Aunque el dataset incluye dos despliegues correspondientes a dos motores monitorizados en diferentes periodos, se determinó que las anomalías presentes son inherentes al sensor, al proceso de transmisión y al mecanismo de ingesta. Por esta razón, el análisis exploratorio detallado se realizó únicamente sobre el despliegue con mayor presencia de irregularidades. Esto permite documentar con mayor precisión el comportamiento anómalo de los datos sin redundar en análisis repetitivos.

Se realizó un análisis de interdependencia para identificar qué descriptores contienen la mayor carga de información sobre la condición del activo.

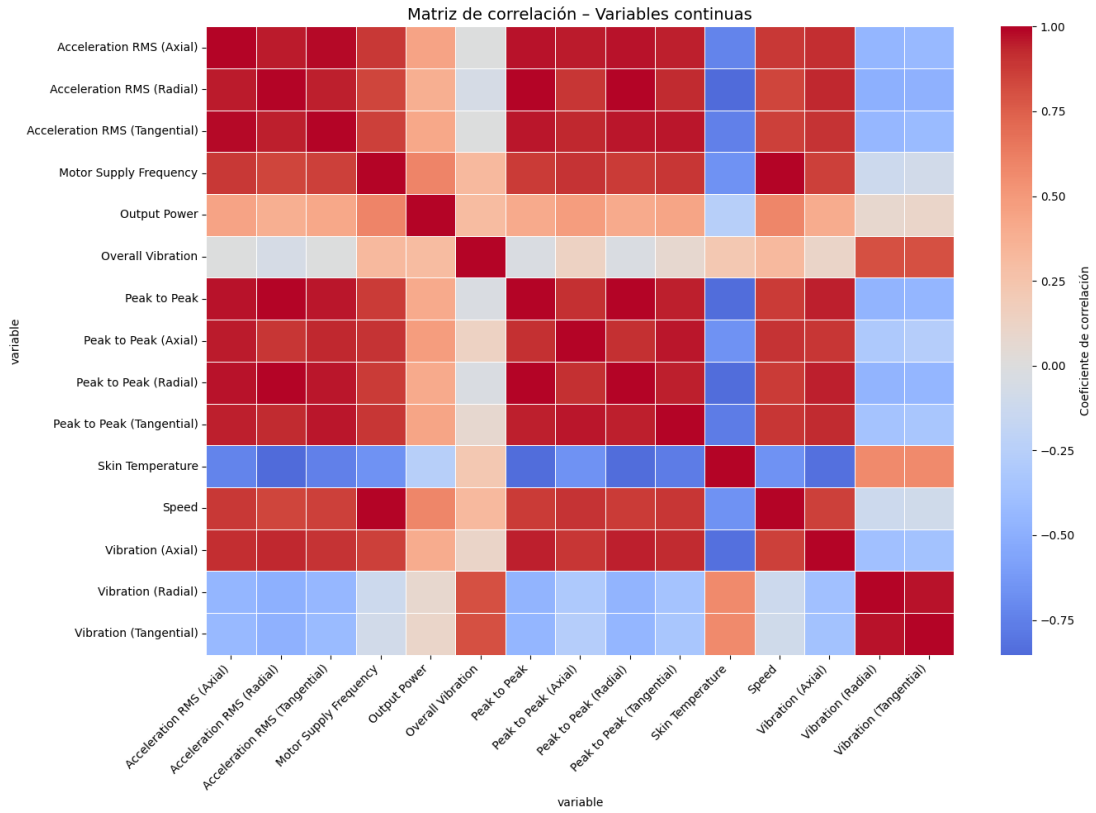


Figura 5. Matriz de correlaciones de variables continuas

Tras el análisis de la matriz de correlación de las variables continuas, se determinó que la Aceleración RMS en el eje Radial es la variable más representativa para el diagnóstico de mantenimiento por las siguientes razones:

- En bombas hidráulicas, los modos de fallo más comunes (desalineación, cavitación y desequilibrio del impulsor) se manifiestan con mayor energía en el plano radial.
- La matriz muestra una correlación superior al 0.90 con los indicadores Peak to Peak, lo que significa que la Aceleración RMS actúa como un resumen estadístico eficiente de la severidad vibratoria global.
- A diferencia de la temperatura, la aceleración radial muestra una respuesta inmediata a cambios mecánicos, permitiendo una detección temprana antes de que ocurra una degradación térmica o una falla catastrófica.

En la matriz se observan tres fenómenos clave que definen el comportamiento de la bomba hidráulica entre ellas:

Redundancia Crítica (Colinealidad): Existe una correlación casi perfecta entre las métricas de Acceleration RMS y Peak to Peak en sus respectivos ejes. Esto indica que ambas variables capturan la misma firma mecánica. Desde el punto de vista del mantenimiento predictivo, esto permite una validación cruzada de sensores: si una variable sube y su "par" no, estamos ante un error de medición y no ante una falla real.

Acoplamiento Operativo: Las variables de vibración muestran una correlación positiva moderada-alta con Speed y Motor Supply Frequency. Esto confirma que la energía vibratoria es dependiente del régimen de giro. Sin embargo, la falta de una correlación perfecta sugiere que existen componentes de vibración independientes de la velocidad, los cuales son los candidatos principales para identificar fallos (como cavitación o desgaste) que no dependen solo de qué tan rápido gire la bomba.

Comportamiento Térmico Inverso: La Skin Temperature presenta una correlación negativa con los niveles de aceleración y picos de vibración. Esto indica que, en este sistema, el aumento de temperatura no es un síntoma inmediato de la vibración mecánica, sino que responde a otras variables (probablemente la carga o el tiempo de operación continua), lo que posiciona a la temperatura como un indicador secundario frente a la vibración para el diagnóstico temprano.

- ***Resumen global del dataset***

A nivel general, el dataset contiene aproximadamente 142.000 registros distribuidos en 22 variables sensoriales. El rango temporal total corresponde al periodo comprendido entre 11-08-2025 y 19-10-2025, con un intervalo de muestreo nominal de 15 minutos por medición.

La estructura está en formato largo, lo que significa que cada fila representa el valor de una variable en un único timestamp. Este formato es adecuado para bases de series temporales como TimescaleDB, pero dificulta el análisis global sin realizar transformaciones previas.

- El análisis global que se evidencia en la Figura 5, permitió identificar los siguientes patrones preliminares:
- No existen valores faltantes ni duplicados; todas las variables presentan registros completos para cada evento de adquisición.
- Existen diferencias en el número de registros por despliegue y por variable, atribuibles a periodos de desconexión o reasignación del sensor.
- Algunas variables presentan una dispersión significativamente mayor que otras, especialmente aquellas relacionadas con vibración y aceleración.

```
Total de registros: 142443
Número de variables: 22
Variables: ['Total Number of Starts' 'Acceleration RMS (Axial)'
 'Acceleration RMS (Radial)' 'Acceleration RMS (Tangential)'
 'Bearing Condition' 'Looseness (BETA)' 'Misalignment (BETA)'
 'Motor Supply Frequency' 'Output Power' 'Overall Vibration'
 'Peak to Peak' 'Peak to Peak (Axial)' 'Peak to Peak (Radial)'
 'Peak to Peak (Tangential)' 'Skin Temperature' 'Speed'
 'Number of Starts Between Measurements' 'Total Running Time'
 'Unbalance (BETA)' 'Vibration (Axial)' 'Vibration (Radial)'
 'Vibration (Tangential)']
Motores / despliegues: ['Motor_001' 'Motor_002']
Rango temporal global: 2025-08-11 07:40:01+00:00 → 2025-10-19 13:29:48+00:00
```

Figura 6. Resumen global de dataset

- ***Resumen de despliegue seleccionado***

El despliegue seleccionado para el análisis detallado contiene 82702 registros y abarca el periodo 09-08-2025 al 19-10-202. Este despliegue fue escogido debido a que

concentra la mayor cantidad de irregularidades, incluyendo picos atribuibles al montaje y desmontaje del sensor, desincronización temporal, outliers y variabilidad elevada en varias señales.

El número de variables registradas es coincidente con el conjunto original (22 variables), permitiendo realizar un análisis completo de las mediciones.

```
Despliegue representativo: Motor_002
Registros en el despliegue representativo: 82702
Rango temporal: 2025-09-08 19:33:01+00:00 → 2025-10-19 13:29:48+00:00
Variables en este despliegue: 22
```

Figura 7. Resumen del despliegue seleccionado

- ***Análisis estadístico en variables continuas***

Para cada variable se calcularon estadísticas descriptivas del despliegue anteriormente seleccionado con el fin de identificar patrones de dispersión y posibles valores atípicos en las variables continuas.

Las estadísticas descriptivas son un conjunto de medidas que permiten resumir, describir y entender rápidamente cómo se comportan los datos sin necesidad de analizarlos uno por uno. Entre las más importantes están:

- Media (mean): Es el valor promedio de un conjunto de datos. Se calcula sumando todos los valores y dividiéndolos para la cantidad de datos.
- Mediana (median): Es el valor central cuando los datos están ordenados de menor a mayor dividiendo el conjunto de datos en dos partes igual.
- Desviación estándar (std): Mide qué tanto se dispersan los datos alrededor de la media. Si es pequeña indica que los valores están cerca del promedio, pero si es grande significa que los datos están muy dispersos o hay variaciones fuertes.
- Mínimo (min): Es el valor más pequeño del conjunto de datos.
- Máximo (max): Es el valor más grande del conjunto.

El objetivo de la estadística descriptiva es describir los datos observados de forma significativa y comprensible, naturalmente los datos brutos serían difíciles de analizar,

y la determinación de tendencias y patrones podrían ser un reto significativo en la minería de datos [24].

La Figura 8 muestra el resultado estadístico de la variable principal seleccionada.

variable	count	mean	std	min	median	max
Acceleration RMS (Radial)	3764	0.828722	0.160344	0.0052	0.8389	2.2207

Figura 8. Resultado estadístico de Acceleration RMS (Radial)

La estadística descriptiva de la variable Acceleration RMS (Radial) presenta una distribución concentrada alrededor de **0.83** g, lo que indica que la mayoría de las mediciones se encuentran en un rango estable y consistente. Sin embargo, los valores mínimo y máximo están muy alejados del rango central: el mínimo está prácticamente en cero y el máximo supera ampliamente **2.2** g. Esto implica que esta variable contiene valores atípicos importantes, tanto inferiores como superiores.

En términos de calidad, esta variable es adecuada para análisis, pero requiere un tratamiento de outliers para mejorar la robustez del modelo predictivo.

El conjunto de variables continuas presenta distribuciones mayoritariamente estables, con medias y medianas cercanas que sugieren comportamientos centrales consistentes. Sin embargo, todas las variables muestran la presencia de valores extremos, evidenciados por mínimos muy bajos y máximos considerablemente altos en comparación con sus rangos típicos de operación. Esto indica la coexistencia de datos válidos con picos atípicos o ruido sensorial, lo cual afecta la calidad general del dataset. Por lo tanto, antes del modelado es necesario aplicar técnicas de depuración, verificación de rangos físicos y tratamiento de outliers para asegurar que las señales representen adecuadamente el comportamiento del equipo

La Tabla 18 resume estadísticos descriptivos del despliegue seleccionado.

Tabla 18. Resumen estadístico de variables continuas

Variable continua	Conteo	Media	Desviación estándar	Mediana	Mínimo	Máximo
Acceleration RMS (Axial)	3764	1.6054	0.3123	1.5996	0.0049	3.3516
Acceleration RMS (Radial)	3764	0.8287	0.1603	0.8389	0.0052	2.2207
Acceleration RMS (Tangential)	3764	1.3339	0.2897	1.3164	0.0052	3.3047
Motor Supply Frequency	3764	48.7203	7.7958	50.03	0	53.06
Output Power	3706	106.8110	29.7320	109.08	0	300.5
Overall Vibration	3764	3.4630	0.7097	3.514	0.047	6.797
Peak to Peak	3764	3.8973	0.6936	3.9414	0.0372	9.1562
Peak to Peak (Axial)	3764	7.9084	1.5010	8.0391	0.0347	13.5078
Peak to Peak (Radial)	3764	3.8973	0.6936	3.9414	0.0372	9.1562
Peak to Peak (Tangential)	3764	7.0271	1.3863	7.0742	0.0366	13.7812
Skin Temperature	3764	32.7480	3.5074	31.6	24.6	49.9
Speed	3764	1454.1986	232.6818	1493.45	0	1588.8
Total Number of Starts	3778	1054.9455	542.2248	1051	116	1988
Total Running Time	3778	704.0759	18.4901	704.3	672	735.5
Vibration (Axial)	3764	3.4329	0.7268	3.506	0.037	6.797
Vibration (Radial)	3764	2.6427	0.6676	2.807	0.043	5.215
Vibration (Tangential)	3764	2.0379	0.4641	2.1	0.04	4.789

- ***Identificación de Outliers***

Un outlier (valor atípico) es un dato que se aleja significativamente del comportamiento general del conjunto de datos. En otras palabras, es un valor que es mucho más alto o mucho más bajo que la mayoría de los demás puntos. Estos valores atípicos pueden surgir por varias razones, tales como:

- Picos de vibración reales
- Ruido del sensor
- Eventos operativos bruscos
- Problemas de calidad del dato

La presencia de outliers puede afectar drásticamente los resultados de un análisis y si no se tratan correctamente pueden producir conclusiones erróneas debido a que son valores extremos que se desvían significativamente de a tendencia general de los datos y tienden a sesgar las métricas estadísticas, como la media y la desviación estándar. [25].

Para la identificación de estos valores atípicos se aplicó el método IQR con el multiplicador 3 como criterio de detección de outliers. Este método es considerado adecuado para series sensoriales industriales donde la distribución no siempre es normal.

El método del rango intercuartílico (IQR) es una técnica estadística utilizada para identificar valores atípicos calculando la diferencia entre el tercer cuartil (Q3) y el primer cuartil (Q1). Los valores que se encuentran fuera de un rango ampliado de 1.5 veces el IQR por debajo de Q1 o por encima de Q3 se consideran outliers [26].

Los cuartiles dividen los datos en 4 partes iguales donde:

- Q1 (25%): Es el punto donde el 25% de los datos está por debajo de ese valor.
- Q2 (50%): O Mediana, está a mitad de los datos.
- Q3 (75%): Es el punto donde el 75% de los datos son menores o iguales a ese valor.

Los cuartiles sirven para verificar la distribución y detectar valores extraños.

El IQR entonces es una medida de dispersión que muestra qué tan extendidos están los datos centrales.

El IQR se calcula con la Fórmula 1:

$$\boldsymbol{IQR = Q3 - Q1} \tag{1}$$

Los outliers o valores extremos superiores se calculan con la Fórmula 2:

$$\boldsymbol{Límite\ superior = Q3 + 3 * IQR} \tag{2}$$

Los outliers o valores extremos inferiores se calculan con la Fórmula 3:

$$\boldsymbol{Límite\ inferior = Q1 - 3 * IQR} \tag{3}$$

En IQR el umbral tradicional para identificar outliers es usando el mutiplicador 1.5 que detecta valores que se alejan moderadamente del rango típico. Sin embargo, es útil cuando se trabaja con datos relativamente limpios y se busca una detección sensible.

Este criterio asume que la distribución es relativamente estable y sin demasiada variabilidad extrema [26].

Para el análisis de variabilidad de datos de este estudio se usó el multiplicador 3 para detectar outliers severos, es decir, valores muy alejados de la distribución principal. Este criterio se justifica porque:

- La señal tiene alta variabilidad natural
- El proceso industrial puede generar picos reales de operación
- El sensor puede registrar ruido extremo
- No se desea eliminar valores moderadamente altos (1.5 * IQR) que podrían ser eventos reales del equipo.

Lo que permitió identificar únicamente los valores severamente alejados del comportamiento típico, reduciendo el riesgo de eliminar información relevante para la dinámica del equipo de bombeo hidráulico

Aplicar IQR en datos de sensores permite obtener series temporales más estables y representativas, eliminando puntos que no reflejan el comportamiento real del equipo, manejar adecuadamente los outliers mejora los análisis y modelos predictivos, lo cual se alinea perfectamente con el uso en señales industriales donde la estabilidad estadística es crucial

La variable Acceleration RMS (Radial) corresponde a una de las señales vibracionales clave para el monitoreo del estado dinámico de la bomba hidráulica. Su análisis mediante el método de detección de valores atípicos basado en el rango intercuartílico (IQR) permitió identificar desviaciones importantes respecto al comportamiento típico de la señal. A partir del análisis estadístico se obtuvieron los siguientes valores:

- Q1 (primer cuartil): 0.7896
- Q3 (tercer cuartil): 0.8906
- IQR (Q3 – Q1): 0.101
- Límite inferior: 0.4866
- Límite superior: 1.1936
- Número de outliers detectados: 124

Estos resultados muestran que el rango normal esperado para la vibración radial de la bomba se encuentra delimitado entre **0.4866** y **1.1936** g RMS. Todo valor fuera de este intervalo se considera atípico bajo el criterio de 3 × IQR.

El hallazgo más relevante fue la presencia de **124** valores atípicos, lo cual indica una proporción considerable de mediciones fuera del comportamiento vibracional habitual. Este hallazgo contiene información valiosa para el diagnóstico del estado de la bomba, pero también evidencia una mayor dispersión y presencia de datos atípicos que deben ser tratados para evitar sesgos en etapas posteriores del modelado.

La Tabla 19 muestra el resumen de valores atípicos con IQR

Tabla 19. Resumen de valores atípicos en variables continuas

Variable	**Q1**	**Q3**	**IQR**	**Limite inferior**	**Límite superior**	**Outliers**
Acceleration RMS (Axial)	1.5244	1.7402	0.2158	0.877	2.3876	113
Acceleration RMS (Radial)	0.7896	0.8906	0.101	0.4866	1.1936	124
Acceleration RMS (Tangential)	1.2354	1.443625	0.208225	0.610725	2.0683	116
Motor Supply Frequency	49.72	50.35	0.63	47.83	52.24	122
Output Power	95.7625	122.6375	26.875	15.1375	203.2625	106
Overall Vibration	3.305	3.822	0.517	1.754	5.373	116
Peak to Peak	3.803725	4.133775	0.33005	2.813575	5.123925	121
Peak to Peak (Axial)	7.457	8.6953	1.2383	3.7421	12.4102	106
Peak to Peak (Radial)	3.803725	4.133775	0.33005	2.813575	5.123925	121
Peak to Peak (Tangential)	6.5586	7.75	1.1914	2.9844	11.3242	103
Skin Temperature	30	35.2	5.2	14.4	50.8	0
Speed	1483.3	1504.2	20.9	1420.6	1566.9	122
Total Number of Starts	593	1525.75	932.75	-2205.25	4324	0
Total Running Time	687.9	720.1	32.2	591.3	816.7	0
Vibration (Axial)	3.283	3.814	0.531	1.69	5.407	119
Vibration (Radial)	2.223	3.049	0.826	-0.255	5.527	0
Vibration (Tangential)	1.795	2.318	0.523	0.226	3.887	87

El resumen evidencia la presencia significativa de valores atípicos en las variables analizadas lo que resalta su sensibilidad o variación mecánica y la importancia de realizar un preprocesamiento adecuado antes usar modelos predictivos.

La Figura 9 muestra un gráfico boxplot donde se visualiza de manera clara la magnitud y distribución de estos valores en la variable principal de este estudio.

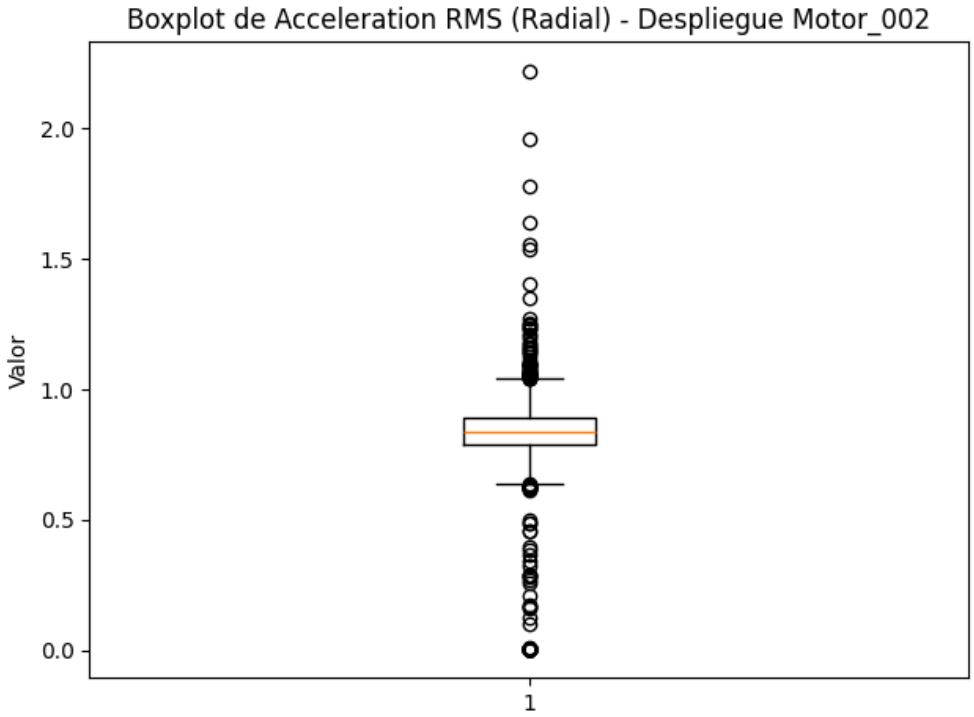


Figura 9. Boxplot de Acceleration RMS (Radial)

En este gráfico, el recuadro central representa la distribución principal de los datos: la mediana se encuentra ligeramente por debajo del valor **0.9**, mientras que el rango intercuartílico (IQR) se concentra aproximadamente entre **0.8** y **0.9**, indicando que la mayoría de las mediciones están dentro de ese intervalo estrecho.

Los bigotes del boxplot se extienden desde valores cercanos a **0.6** hasta aproximadamente **1.0**, mostrando el rango donde aún no se consideran valores atípicos. Sin embargo, el gráfico presenta una gran cantidad de outliers, tanto por debajo del bigote inferior como por encima del superior. Entre los valores bajos se observan puntos desde alrededor de **0.0** hasta **0.6**, mientras que en la parte superior aparecen valores dispersos que superan **1.0**, llegando incluso por encima de **2.0**, lo que evidencia eventos extremos en las mediciones de vibración radial.

En conjunto, el gráfico refleja una distribución con alta concentración de datos centrales, pero también con múltiples valores atípicos, lo cual es típico en señales de sensores industriales donde pueden existir variaciones bruscas, ruido o condiciones inusuales de operación.

La Figura 10 representa un histograma de Acceleration RMS (Radial)

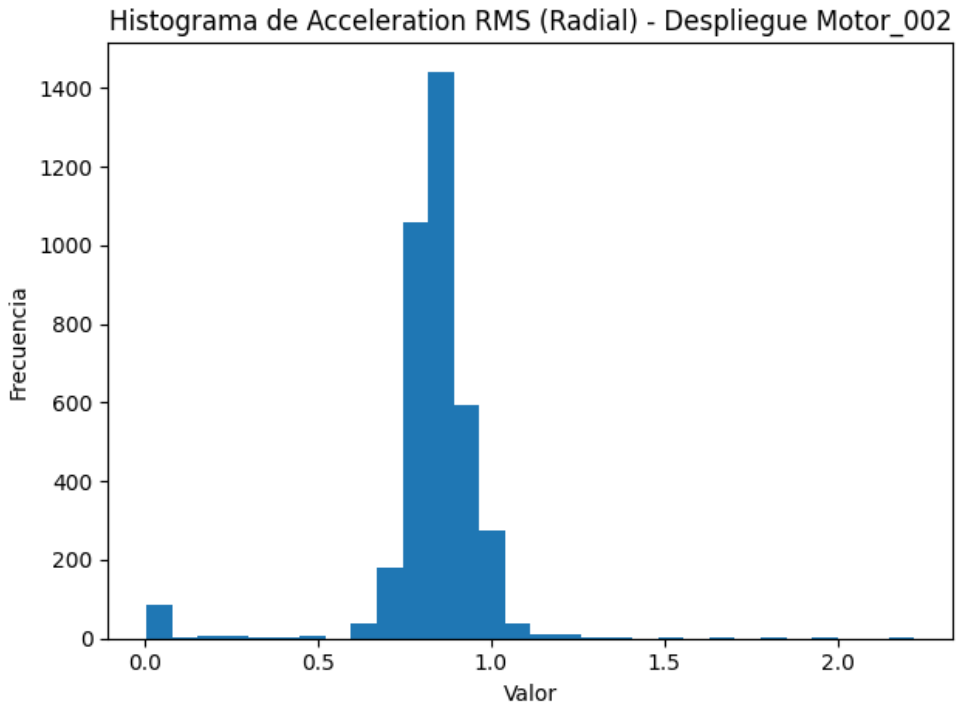


Figura 10. Histograma de Acceleration RMS (Radial)

La forma del histograma revela que la mayoría de las mediciones se concentran en un intervalo relativamente estrecho, aproximadamente entre **0.8** y **1.0**, formando un pico pronunciado que indica una alta frecuencia en ese rango. También se observan valores aislados por debajo de **0.5** y por encima de **1.5**, lo que sugiere la presencia de posibles outliers o eventos atípicos. En general, la distribución es unimodal y está ligeramente sesgada hacia la derecha, mostrando algunos valores dispersos de mayor amplitud. Este tipo de análisis permitió visualizar la variabilidad de la señal vibratoria del motor y detectar comportamientos fuera de lo habitual.

Se pudo verificar que en ciertos intervalos se presentaron picos abruptos en múltiples variables. Estos picos no corresponden al comportamiento real de la maquinaria, sino a vibraciones y aceleraciones generadas por la manipulación física del dispositivo.

La Figura 11 muestra un gráfico lineal de la variable principal.

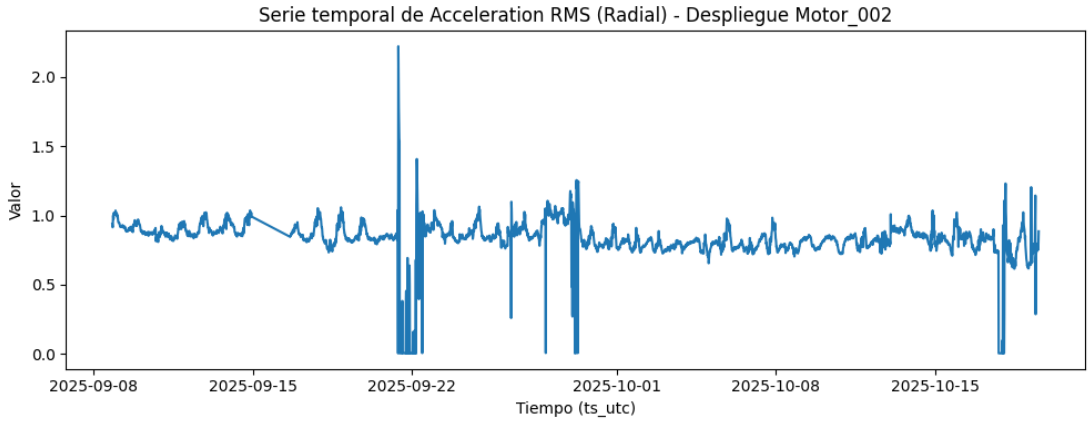


Figura 11. Gráfico lineal de Acceleration RMS (Radial)

Se aprecian tramos estables, tramos de mayor amplitud vibracional y presencia de valores extremadamente altos que no corresponden a fallas mecánicas sino a variabilidad del dataset y posibles errores residuales. Este comportamiento confirmó que la señal debe analizarse y limpiarse adecuadamente.

- ***Análisis de variables categóricas u ordinales***

Además de las variables numéricas continúas asociadas a la vibración, el conjunto de datos incluye un conjunto de variables categóricas u ordinales generadas internamente por el firmware del sensor ABB Ability™ Smart Sensor. Estas variables representan indicadores de condición calculados por el propio dispositivo a partir del análisis de vibraciones en distintos ejes, y no corresponden a magnitudes físicas medibles de forma directa.

Con el fin de evaluar la coherencia, distribución y variabilidad de estas señales, se realizó un análisis exploratorio agrupando cada variable categórica para identificar:

- Número total de mediciones
- Valores únicos presentes en el dataset
- Valor más frecuente reportado por el sensor

La Tabla 20 resume de los resultados de análisis de variables categóricas.

Tabla 20. Resumen de valores categóricos u ordinales

Variable	Conteo	Valores únicos	Valor más frecuente
Bearing Condition	3674	[0, 1, 2, 3, 4]	2
Looseness (BETA)	3764	[0, 1]	0
Misalignment (BETA)	3764	[0, 1]	0
Unbalance (BETA)	3764	[0, 1]	1
Number of Starts Between Measurements	3778	[0, 1, 2]	0

Bearing Condition*:* Esta variable presenta cinco valores distintos: 0, 1, 2, 3 y 4. La predominancia del valor 2 indica que, en la mayoría de las mediciones, el sensor clasifica la condición del rodamiento como un estado intermedio y estable, sin indicios de degradación severa. La presencia del valor 4, aunque poco frecuente, sugiere estados esporádicos donde el sensor podría haber detectado patrones vibratorios potencialmente anómalos o límites de saturación en sus algoritmos internos.

Looseness (BETA): Con únicamente dos valores posibles (0 y 1), esta variable indica si el algoritmo del sensor detecta evidencia de aflojamiento estructural en la máquina. El valor más frecuente es 0, lo que sugiere que la mayoría de los despliegues no presentan este patrón vibracional, o bien la señal medida no cumple las condiciones para activar este indicador.

Misalignment (BETA): Al igual que la anterior, esta variable opera como un indicador binario. El valor dominante es 0, lo cual es consistente con ausencia de desalineación detectable, o mediciones insuficientes para que el algoritmo considere el estado "1".

Unbalance (BETA): Esta variable también es binaria, pero a diferencia de las anteriores, su valor más frecuente es **1**, lo que indica la presencia relativamente constante de patrones vibratorios compatibles con desbalance, o una mayor sensibilidad de este indicador en los motores monitoreados.

Este resultado es coherente con lo observado en los tramos de alta vibración del EDA, donde los niveles superiores de aceleración RMS pueden activar este tipo de indicadores internos del sensor.

- ***Análisis de estructura temporal***

El análisis de estructura temporal consiste en examinar cómo están organizadas las mediciones en el tiempo, verificando si los datos siguen una secuencia correcta, si existen irregularidades, saltos, duplicaciones o pérdidas de información. En sensores IoT e industriales, la temporalidad es crítica porque los modelos de mantenimiento predictivo asumen que la señal representa un comportamiento continuo y correctamente espaciado en el tiempo.

El comportamiento temporal de las mediciones se analizó mediante el cálculo del intervalo entre mediciones consecutivas (Δt).

El cálculo de intervalo entre mediciones se realiza con la Fórmula 4:

$$\boldsymbol{\Delta t = ts_i - ts_{i-1}} \qquad \textbf{(4)}$$

La expresión describe el intervalo de tiempo entre dos mediciones consecutivas en una serie temporal donde:

ts_i: es el timestamp (marca de tiempo) de la medición actual, representa el instante en que se registró la lectura número i.

ts_{i-1}: es el timestamp de la medición anterior a la actual, es decir, el instante previo.

Δt: Es la diferencia entre ambos timestamps, y representa el tiempo transcurrido entre dos mediciones consecutivas del sensor.

Este cálculo permitió medir el número de segundos entre cada lectura consecutiva del sensor, idealmente, el sensor debe registrar un dato cada 900 segundos (15 minutos).

La Figura 12 muestra el análisis de estructura temporal del despliegue seleccionado.

```
Resumen estadístico de Δt (segundos):

count      3763.000000
mean        935.585171
std        2368.553924
min          22.000000
25%         899.000000
50%         899.000000
75%         901.000000
max      146151.000000
```

Figura 12. Resumen estadístico de estructura temporal

Los resultados del análisis de los intervalos temporales entre mediciones (Δt) evidencia que la mayoría de las lecturas siguen un patrón regular cercano a 900 segundos (15 minutos). Esto se observa en los percentiles 25%, 50% y 75%, todos alrededor de 899–901 segundos, lo cual coincide con la frecuencia esperada para este tipo de sensores de vibración industrial.

Sin embargo, la presencia de valores extremos significativamente mayores destacando un Δt máximo de 146151 s (40.6 horas) indica periodos prolongados sin recepción de datos, posiblemente asociados a fallas de comunicación, interrupciones energéticas o desconexiones del dispositivo. El valor mínimo de 22 segundos sugiere lecturas atípicamente cercanas, que podrían corresponder a duplicados o reenvíos del sistema.

La elevada desviación estándar (2368.55 s) refleja esta mezcla entre intervalos regulares y vacíos prolongados, confirmando una estructura temporal irregular. Esto es importante porque implica la necesidad de realizar procesos de detección de gaps, resampling o imputación para asegurar una serie temporal uniforme y apta para análisis posteriores o modelado predictivo.

Además, se observó ruido de sincronización (timestamp misalignment) lo que provoca desalineación entre señales, ya que las 22 variables no siempre comparten el mismo timestamp. Algunas señales se registran con adelantos o retrasos de 1 a 450s, lo que produce Δt atípicos dentro de la misma variable y discrepancias cuando se analizan todas en conjunto.

- ***Gaps y discontinuidades***

En el contexto de series temporales sensoriales, un gap es un intervalo anormalmente grande entre dos mediciones consecutivas. Esto ocurre cuando el sistema no registra datos durante un periodo en el que sí se esperaba una muestra [20].

Los gaps detectados corresponden a periodos donde el sensor dejó de registrar mediciones, ocasionando saltos en la secuencia temporal. Estos eventos pueden deberse a:

- Desconexiones del sensor

- Fallas de comunicación
- Sobrecarga del sistema SCADA/PLC
- Pérdida de paquetes
- Reinicios del equipo

Los gaps comprometen el análisis porque generan huecos en la continuidad temporal, afectando a cálculos estadísticos, análisis de tendencias, modelos de predicción, imputación o reconstrucción de series.

Para evaluar la estructura temporal de la serie sensorial se analizó el intervalo entre mediciones consecutivas (Δt).

El sistema debe registrar datos cada 900 segundos; sin embargo, debido al jitter del hardware, latencias de comunicación o retrasos propios del SCADA/PLC, existen variaciones menores en la periodicidad real.

Con base en prácticas de adquisición digital, se definió una tolerancia del 150% del periodo esperado, equivalente a 1350 segundos. Este valor permite absorber variaciones operativas moderadas sin clasificar falsos gaps.

Por lo tanto, cualquier Δt que supere 1350 segundos fue considerado un gap temporal, indicando una interrupción real en la transmisión de datos o una pérdida de muestras.

La Figura 13 detalla el resumen de gaps detectados en el despliegue seleccionado.

```
Número de gaps detectados: 2
Ejemplos de gaps detectados:
                          ts_utc   delta_s
72788  2025-09-16 15:00:01+00:00  146151.0
111828 2025-10-05 01:49:00+00:00    2650.0
```

Figura 13. Resumen de gaps detectados

El resultado de análisis de gaps temporales se identificaron 2 gaps, es decir, intervalos entre mediciones que excedieron la tolerancia definida de 1350 segundos, lo cual indica interrupciones reales en la adquisición de los datos.

Los 146151 segundos (aproximadamente 40.6 horas) es un gap severo, equivalente a casi dos días sin registro de datos. Este tipo de interrupción sugiere una falla

prolongada en la transmisión o almacenamiento del sistema sensorial, por ejemplo: desconexión del dispositivo, caída de red, reinicios del sistema o detención del equipo.

Los 2650 segundos (aproximadamente 44.1 minutos) es un gap moderado, pero aun así supera ampliamente la tolerancia de 1350 s. Indica un periodo donde las mediciones dejaron de llegar durante aproximadamente tres ciclos de muestreo (cada uno de 900 s). Puede deberse a congestión temporal de red, saturación del sistema SCADA o pérdida de paquetes.

La identificación de estos gaps confirma que el sistema presenta pérdidas reales de información en determinados momentos, lo cual afecta la continuidad temporal y debe considerarse en el proceso de preprocesamiento y en la generación de la nueva serie temporal para el modelo de mantenimiento predictivo.

- ***Conclusiones EDA***

El análisis exploratorio permitió identificar que el dataset presenta múltiples problemas de calidad que deben ser tratados antes de aplicar cualquier modelo de aprendizaje automático:

- Presencia significativa de outliers en variables vibracionales.
- Desincronización temporal entre mediciones.
- Gaps distribuidos a lo largo de la serie.
- Variabilidad extrema en variables críticas.

Estos hallazgos justifican plenamente la necesidad del sistema de preprocesamiento desarrollado en este trabajo, así como la fase de imputación, remuestreo, sincronización y etiquetado de calidad implementada posteriormente.

3.5 Fase 3: Preparación de Datos (Data Preparation)

El preprocesamiento de datos sensoriales constituye una etapa fundamental para garantizar la calidad y fiabilidad de cualquier modelo analítico o predictivo aplicado posteriormente. Los datos provenientes de sistemas de monitoreo industrial suelen presentar problemas como ruido, valores atípicos, inconsistencias temporales, pérdidas de información o variaciones en la frecuencia de muestreo, los cuales pueden distorsionar el comportamiento real de las variables. Un sistema de preprocesamiento

permite identificar, corregir o mitigar estas irregularidades antes de alimentar un modelo de aprendizaje automático, asegurando que este reciba una representación fiel y coherente del estado operativo real del equipo. Una adecuada preparación de los datos no solo mejora la capacidad predictiva de los modelos, sino que también reduce el riesgo de interpretaciones erróneas y aumenta la fiabilidad del sistema de mantenimiento predictivo.

3.5.1 Diseño del sistema de preprocesamiento de datos

La fase de preparación de los datos constituye el núcleo del sistema desarrollado en esta investigación. Aquí se implementa el conjunto de procedimientos técnicos que permiten transformar las mediciones crudas, heterogéneas e irregulares provenientes del sensor industrial ABB Ability™ Smart Sensor en un dataset limpio, sincronizado y apto para modelos de mantenimiento predictivo basados en aprendizaje automático.

Este proceso se diseñó en dos fases independientes pero complementarias:

- Fase 1: Identificación y etiquetado de anomalías
- Fase 2: Corrección, imputación, reconstrucción temporal y generación del dataset limpio

Ambas fases se integran en una arquitectura modular que garantiza trazabilidad, reproducibilidad y uso futuro en el proyecto institucional de mantenimiento predictivo.

La Figura 14 muestra el flujo de datos del sistema de preprocesamiento.

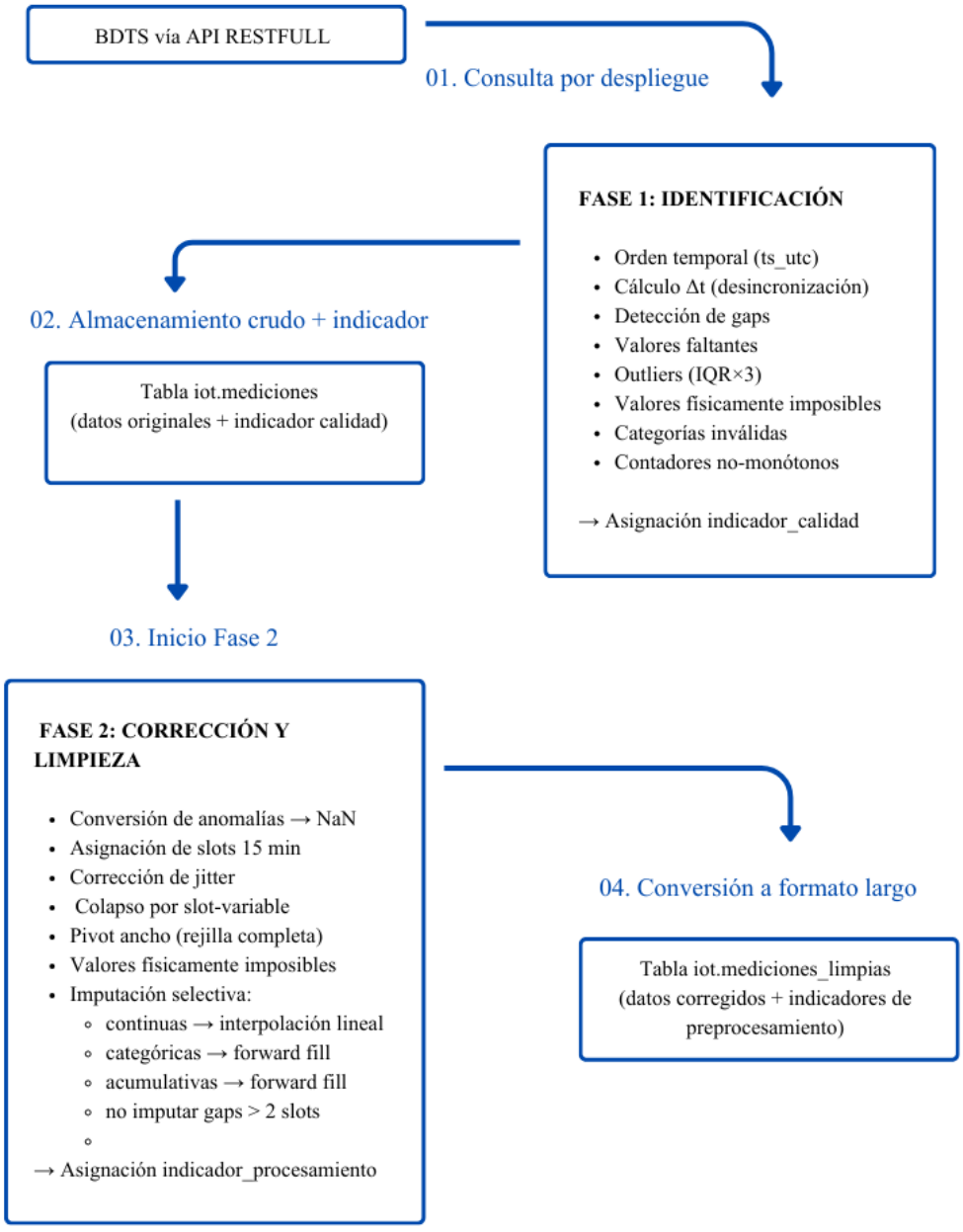


Figura 14. Arquitectura del sistema de preprocesamiento de datos

3.5.2 Herramientas de software y librerías utilizadas

El sistema de preprocesamiento de datos sensoriales se implementó utilizando Python 3.12.10 como lenguaje principal de programación. La elección de Python se debe a su amplio ecosistema de librerías para análisis de datos, manejo de series temporales e integración con APIs y bases de datos. A continuación, se describen las herramientas y librerías utilizadas en cada fase del sistema.

- *Python 3.12.10*: Lenguaje de programación interpretado, de alto nivel y multiparadigma. Es ampliamente utilizado en ciencia de datos, análisis de series temporales, machine learning e integración de sistemas
- *Pandas*: Librería estándar para análisis y manipulación de datos tabulares en Python. Permite manejar datos estructurados, ordenar, agrupar, pivotar, calcular estadísticos, convertir fechas y más.
- *NumPy*: Librería central de Python para cálculo numérico con arreglos multidimensionales. Ofrece operaciones matemáticas vectorizadas, manejo eficiente de memoria, funciones estadísticas rápidas.
- *Requests (cliente HTTP)*: Librería más utilizada en Python para consumo de APIs REST. Permite realizar solicitudes HTTP de forma sencilla y segura.
- *Datetime*: es la librería estándar de Python para fechas y horas, zonas horarias, operaciones temporales (sumas, diferencias, comparaciones).
- *Typing*: Permite agregar anotaciones de tipo en Python. Ayuda en documentación, autocompletado y claridad del código.
- Visual Studio Code (VSC) como editor.

3.5.3 Fase 1: Detección y clasificación de anomalías

La primera fase consiste en detectar anomalías sin modificar los valores originales, pero asignando un indicador de calidad que califica al dato como anómalo o un dato bueno. Esto permite preservar la integridad del dato, realizar auditoría completa, comprender patrones de falla del sensor, apoyar la validación del despliegue, y facilitar el análisis futuro por parte de otros investigadores.

Esta fase se ejecuta por despliegue, y se inicia con una consulta a la BASE DE DATOS TIMES SERIES a la tabla INGESTAS a través de la API RESTFUL para obtener las 22 variables de un despliegue seleccionado en formato largo. Se utiliza la librería requests de Python para interactuar con los endpoints de la API RESTful provista por el equipo colaborativo. Esta llamada devuelve los datos en formato JSON, los cuales son inmediatamente transformados en un DataFrame de Pandas.

La Figura 15 muestra los endpoints que consume el sistema de preprocesamiento.

```
BASE_INGESTAS       = f"{API_ROOT}{API_PREFIX}/ingestas"
BASE_MEDICIONES     = f"{API_ROOT}{API_PREFIX}/mediciones"
BASE_MEDICIONES_LIMPIAS = f"{API_ROOT}{API_PREFIX}/mediciones_limpias"
BASE_CARACTERISTICAS = f"{API_ROOT}{API_PREFIX}/caracteristicas"
BASE_DESPLIEGUES    = f"{API_ROOT}{API_PREFIX}/despliegues"
BASE_ASSETS         = f"{API_ROOT}{API_PREFIX}/assets"
BASE_MOTORES        = f"{API_ROOT}{API_PREFIX}/motores"
```

Figura 15. Endpoints de API Resfull

La conexión a la API se realiza mediante un objeto requests.Session() con cabeceras HTTP que incluyen un token Bearer almacenado en variables de entorno. Las llamadas al endpoint se parametrizan con los campos: asset_codigo, motor_codigo, ts_from y ts_to, y se ejecutan con un timeout de 60 segundos para evitar bloqueos prolongados.

Se valida la presencia de las columnas esperadas y se registra en logs la cantidad de filas descargadas y el rango temporal cubierto por el despliegue, así como información sobre al sensor y motor monitoreado.

La Figura 16 muestra la consulta y respuesta del despliegue.

```
Ingrese el ID de despliegue a procesar: 2

 CÓDIGO DE ASSET          : 37156
 CÓDIGO DE MOTOR          : Motor_002
 DESDE (ts_from)          : 2025-09-08T14:33:01-05:00
 HASTA (ts_to)            : 2025-10-19T08:29:48-05:00

=== DATOS CRUDOS DEL DESPLIEGUE ===

   ingesta_id                    ts_utc                                 variable   valor  despliegue_id
0       59742 2025-09-08 14:33:01-05:00              Acceleration RMS (Axial)  1.9863              2
1       59743 2025-09-08 14:33:01-05:00             Acceleration RMS (Radial)  0.9385              2
2       59744 2025-09-08 14:33:01-05:00          Acceleration RMS (Tangential)  1.6855              2
3       59745 2025-09-08 14:33:01-05:00                    Bearing Condition     3.0              2
4       59746 2025-09-08 14:33:01-05:00                     Looseness (BETA)     0.0              2
5       59747 2025-09-08 14:33:01-05:00                  Misalignment (BETA)     0.0              2
6       59748 2025-09-08 14:33:01-05:00                Motor Supply Frequency   49.56              2
7       59749 2025-09-08 14:33:01-05:00  Number of Starts Between Measurements     0.0              2
8       59750 2025-09-08 14:33:01-05:00                          Output Power  141.71              2
9       59751 2025-09-08 14:33:01-05:00                       Overall Vibration   3.926              2

TOTAL DE FILAS CARGADAS: 82702
```

Figura 16. Consulta por despliegue

- ***Diagnóstico temporal***

Una vez obtenido el conjunto de mediciones desde la API, el sistema realiza un diagnóstico temporal exhaustivo con el fin de evaluar la consistencia de la línea de tiempo y detectar anomalías asociadas al muestreo irregular característico de sensores industriales. Esta actividad es crítica porque la calidad temporal afecta directamente la

validez de cualquier análisis basado en series de tiempo y determina, además, las posibilidades de imputación y sincronización en fases posteriores.

Se evalúa la estructura cronológica de las ingestas de un despliegue con el fin de detectar:

- Duplicados en la misma clave lógica (asset, motor, ts_utc, variable),
- Jitter (micro-variaciones en timestamps),
- Gaps temporales relevantes que impidan la imputación o que requieran un tratamiento especial.

Estos resultados guían las reglas de sincronización e imputación de Fase 2.

El sistema ordena las mediciones por su marca de tiempo (ts_utc), elimina duplicados y calcula el intervalo temporal entre cada par consecutivo (Δt), con ello se verifica la irregularidad correspondiente a los timestamps del despliegue.

El sistema define un umbral dinámico para la detección de discontinuidades temporales (gaps). Dado que la frecuencia esperada de muestreo es de 900 s (15 min), se adoptó un factor de tolerancia de **1.5**×, ampliamente utilizado en literatura PHM para sensores industriales con jitter moderado. Por tanto, se considera gap toda separación entre mediciones consecutivas mayor a 1350 s (22.5 min). Este valor permite distinguir entre variaciones normales de muestreo y pérdidas reales de datos.

El sistema clasifica un gap como relevante cuando su duración supera un múltiplo del periodo nominal establecido.

Además, se identifica jitter como variaciones menores a 7.5 min respecto al tiempo ideal de muestra. El valor se eligió para detectar micro-variaciones sin confundirlas con ausencia real de datos que corresponde a una tolerancia de **0.5**× por la timestamp esperado. Esta información será relevante más adelante cuando se realice la sincronización y asignación a rejilla temporal uniforme.

El diagnóstico temporal produce tres banderas que acompañan a cada registro y describen exclusivamente la calidad temporal de los datos del despliegue.

La Tabla 21 muestra las banderas (flags) generadas en el diagnostico temporal.

Tabla 21. Banderas generadas en diagnostico temporal

Flag	Descripción
delta_s	Diferencia temporal en segundos respecto al timestamp anterior.
is_gap	Verdadero cuando el intervalo supera el umbral definido para gaps.
is_small_delta	Verdadero cuando el intervalo es menor al jitter permitido.

Estas banderas constituyen la base para la lógica de calidad aplicada en fases posteriores, pero en esta etapa solo describen la condición temporal de cada registro sin modificar los valores originales.

La Figura 17 muestra el resumen de diagnóstico temporal.

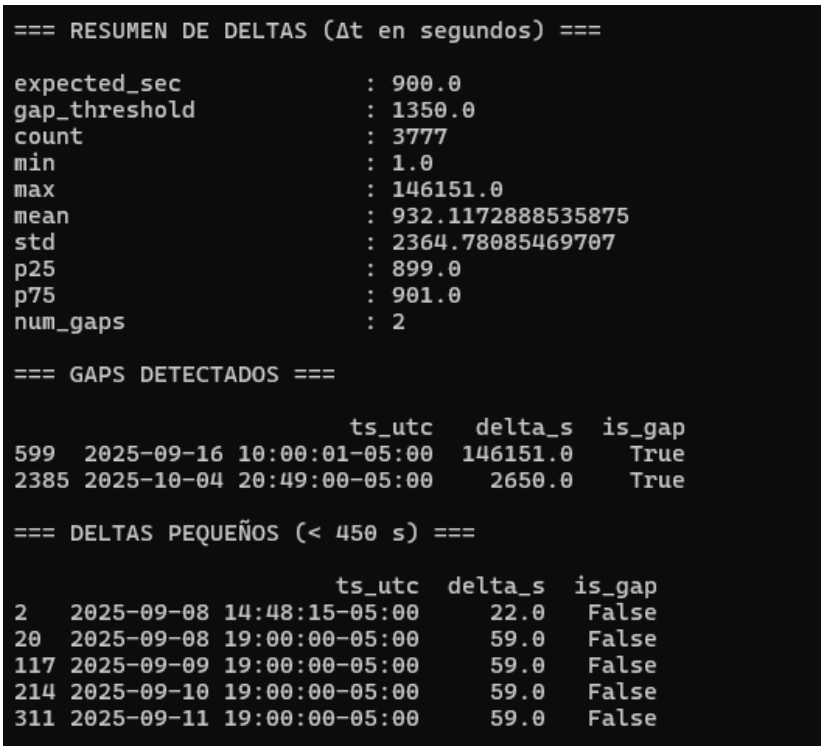


Figura 17. Resumen de diagnóstico temporal del despliegue

La detección temprana de estas anomalías temporales fue fundamental para interpretar correctamente los patrones de medición y planificar la reconstrucción temporal posterior.

- ***Detección de anomalías por tipo de variable***

El módulo de validación por variable inspecciona cada medición según su naturaleza clasificadas en la etapa de análisis exploratorio de datos, se asigna banderas que

reflejan problemas de valor o de consistencia lógica. Estas banderas alimentan el indicador de calidad y guían acciones posteriores. A continuación, se detallan las decisiones, criterios y funciones principales empleadas.

Se clasificaron las variables en cuatro grupos ya que cada grupo requiere reglas de validación diferentes las mismas que son: vibración, físicas continuas, categóricas y acumulativas

Valores faltantes: Se identifican mediciones sin valor registrado (NaN). La función mark_missing(df) distingue explícitamente valores faltantes y crea la bandera is_missing usando df["valor"].isna()

La Figura 18 muestra el método en Python para la detección de valores faltantes.

```python
def mark_missing(df: pd.DataFrame) -> pd.DataFrame:
    df_out = df.copy()
    df_out["is_missing"] = df_out["valor"].isna()
    return df_out
```

Figura 18. Método de detección de valores faltantes

Valores físicamente imposibles: Toda medición continua (vibración y físicas) o acumulativa con valores menores a cero se clasifica como físicamente imposible, dado que viola restricciones inherentes al proceso físico. La función mark_invalid_physical(df) convierte el valor a numérico y marca la bandera is_invalid_physical = True cuando el valor es numérico y < 0.

La Figura 19 muestra el método en Python para la detección de valores imposibles.

```python
def mark_invalid_physical(df: pd.DataFrame) -> pd.DataFrame:
    df_out = df.copy()
    df_out["is_invalid_physical"] = False

    mask_num = df_out["variable"].isin(VIBRATION_VARS + PHYSICAL_VARS + ACCUMULATIVE_VARS)
    df_num = df_out[mask_num].copy()

    df_num["valor"] = pd.to_numeric(df_num["valor"], errors="coerce")

    invalid_idx = df_num.index[df_num["valor"] < 0]
    df_out.loc[invalid_idx, "is_invalid_physical"] = True

    return df_out
```

Figura 19. Método de detección de valores físicamente imposibles

Las lecturas negativas en estas variables son físicamente imposibles o indicativas de error de instrumentación.

Outliers en variables continuas: Valores extremos que no representan el comportamiento real del equipo. El método compute_vibration_stats(df) calcula para cada variable de vibración, estadísticos descriptivos como la media, mínimo, máximo y los cuartiles Q1 y Q3, así como el rango intercuartílico (IQR) como método de detección de outliers superiores e inferiores.

La Figura 20 muestra el resumen estadístico de variables de vibración.

```
=== Estadísticas de vibración ===
                        variable  count      mean     min      max        q1        q3       iqr
0         Acceleration RMS (Axial)   3764  1.605418  0.0049   3.3516  1.524400  1.740200  0.215800
1        Acceleration RMS (Radial)   3764  0.828722  0.0052   2.2207  0.789600  0.890600  0.101000
2     Acceleration RMS (Tangential)   3764  1.333856  0.0052   3.3047  1.235400  1.443625  0.208225
3                 Overall Vibration   3764  3.462995  0.0470   6.7970  3.305000  3.822000  0.517000
4                      Peak to Peak   3764  3.897339  0.0372   9.1562  3.803725  4.133775  0.330050
5             Peak to Peak (Axial)   3764  7.908393  0.0347  13.5078  7.457000  8.695300  1.238300
6            Peak to Peak (Radial)   3764  3.897339  0.0372   9.1562  3.803725  4.133775  0.330050
7         Peak to Peak (Tangential)   3764  7.027141  0.0366  13.7812  6.558600  7.750000  1.191400
8                Vibration (Axial)   3764  3.432943  0.0370   6.7970  3.283000  3.814000  0.531000
9               Vibration (Radial)   3764  2.642742  0.0430   5.2150  2.223000  3.049000  0.826000
10           Vibration (Tangential)   3764  2.037858  0.0400   4.7890  1.795000  2.318000  0.523000
```

Figura 20. Estadísticos de variables de vibración e IQR

Usando las estadísticas de la función anterior se aplican dos umbrales basados en IQR para marcar ouliers en el método mark_vibration_outliers(df, stats):

- is_outlier = True si *valor* < Q1 - 3 * IQR o *valor* > Q3 + 3 * IQR lo que representa valores extremos en el límite superior e inferior.
- is_high = True si valor < Q1 – 1.5 * IQR o *valor* > Q3 + 1.5 * IQR para valores atípicos moderados.

Este método se eligió por su versatilidad ante distribuciones no necesariamente normales o no gussianas y su amplio uso en análisis industrial.

is_outlier tiene prioridad sobre is_high (si es outlier no se marca is_high).

Categorías fuera de dominio: Valores fuera del dominio permitido en variables de diagnóstico o categórica. El método mark_categorical_invalid(df) convierte el valor a número, compara con el dominio esperado (CATEGORICAL_DOMAINS) y marca is_invalid_category = True si el valor no pertenece o no puede convertirse a entero. Esto asegura que los diagnósticos y flags diagnósticos respeten el dominio provisto por el fabricante.

La Figura 21 muestra el método de Python para la detección de categorías no registradas.

```python
for var, group in df_cat.groupby("variable"):
    raw_domain = CATEGORICAL_DOMAINS.get(var)
    if raw_domain is None:
        continue

    valid_vals = {int(v) for v in raw_domain}

    for idx, row in group.iterrows():
        val_num = row["valor_num"]
        if pd.isna(val_num):
            continue
        try:
            val_int = int(val_num)
        except Exception:
            df_out.at[idx, "is_invalid_category"] = True
            continue

        if val_int not in valid_vals:
            df_out.at[idx, "is_invalid_category"] = True
```

Figura 21. Método de detección de categorías fuera de dominio

Integridad en variables acumulativas: Se verifica que los contadores sean monótonos crecientes. Cualquier decremento se consideró una falla de lectura o reinicio del dispositivo. El método mark_accumulative_integrity(df) ordena por variable y ts_utc, convierte a numérico e inspecciona que la secuencia sea no decreciente sino marca is_invalid_monotonic = True.

La Figura 22 muestra el método de Python para la detección de monotonía invalida.

```python
df_acc = df_acc.sort_values(["variable", "ts_utc"])
df_acc["valor"] = pd.to_numeric(df_acc["valor"], errors="coerce")

for var, group in df_acc.groupby("variable"):
    prev_val: Optional[float] = None
    for idx, row in group.iterrows():
        val = row["valor"]
        if pd.isna(val):
            prev_val = val
            continue
        if prev_val is not None and not pd.isna(prev_val) and val < prev_val:
            df_out.at[idx, "is_invalid_monotonic"] = True
        prev_val = val
```

Figura 22. Método de detección de integridad de variables acumulativas

Antes de asignar el indicador de calidad se asegura que un mismo registro que cumple múltiples condiciones se clasifique por la más severa, este código esta almacenado en la base de datos en el esquema iot.indicador_calidad.

La Figura 23 muestra los indicadores de calidad registrados en las base de datos.

codigo_calidad [PK] smallint	nombre_estado character varying (50)	severidad smallint	descripcion character varying (200)
0	OK	0	Medición válida, sin anomalías de valor ni temporales
1	Valor faltante	1	La medición no tiene valor registrado (NaN o nulo)
2	Gap temporal	2	Medición posterior a un gap grande en la serie temporal
3	Valor extremo	3	Valor superior al percentil 99 de su variable dentro del despliegue
4	Categoría inválida	4	Valor fuera del dominio permitido para una variable categórica
5	Valor físicamente imposible	5	Valor que viola límites físicos plausibles (negativo u otro imposible)
6	Error de integridad en contador	6	Variable acumulativa que disminuye cuando debería ser monótona creciente
9	Error desconocido	9	Falla inesperada o estado no clasificado

Figura 23. Registros de la tabla indicador_calidad

- ***Indicador de calidad asignado***

Es importante destacar que en esta fase no se modifican los valores originales. Únicamente se agregan las banderas y el código de calidad para garantizar trazabilidad de los datos históricos.

La Tabla 1 muestra el resumen de los indicadores de calidad de la Fase 1.

Tabla 22. Resumen de indicadores de calidad

Código	Nombre	Descripción	¿Cuándo se asigna?
0	OK	Medición válida	No presenta ninguna anomalía de valor ni temporal
1	Valor faltante	La medición no tiene valor numérico registrado	valor = NaN
2	Gap temporal	La medición ocurre inmediatamente después de un gap grande	$\Delta s\ 0 > 1.5 \times 900$
3	Valor extremo	Valor mayor al límite superior de IQR con multiplicador 3	valor > Q3+3·IQR
4	Categoría inválida	Valor no pertenece al dominio definido de la variable categórica	Valor categórico fuera del dominio
5	Valor físicamente imposible	Valor viola límites físicos (negativo, o menor al mínimo plausible)	valor < 0
6	Error de integridad (contadores)	Violación de monotonía en contadores acumulativos	El contador disminuye
9	Error desconocido	Falla inesperada durante procesamiento	Excepciones no clasificadas

La bandera is_high se mantiene informativa y no se incluye como indicador de calidad porque representa atipicidad moderada corresponde a un valor válido.

Entonces, el módulo de validación por variable clasifica las mediciones según su grupo funcional y aplica reglas específicas por tipo. Para variables de vibración se calculan cuartiles (Q1, Q3) y el rango intercuartílico (IQR) por variable; a partir de estos estadísticos se definen umbrales para detectar atipicidad moderada (is_high, IQR×1.5) y valores extremos (is_outlier, IQR×3). Para variables físicas y acumulativas se valida no-negatividad mediante coerción numérica; valores negativos se marcaron como is_invalid_physical. Las variables categóricas se validan contra un dominio finito, marcando is_invalid_category cuando el valor no pertenece al dominio esperado. En los contadores acumulativos se verifica la monotonicidad temporal, aplicando is_invalid_monotonic ante decrementos. Finalmente, las banderas severas se consolidaron en un quality_code jerárquico que prioriza los errores de integridad y los convierte en criterios para la fase de preprocesamiento.

La Figura 24 muestra el resultado del resumen de calidad de datos de un despliegue.

```
TOTAL DE MEDICIONES: 82702

=== CONTEO POR INDICADOR DE CALIDAD ===

Código 0 (OK)             : 81531 filas (98.58 %)
Código 2 (Gap temporal)   : 44 filas (0.05 %)
Código 3 (Valor extremo)  : 1126 filas (1.36 %)
Código 4 (Categoría inválida) : 1 filas (0.00 %)

=== CONTEO POR BANDERAS ==

  is_missing              : 0 filas (0.00 %)
  is_invalid_physical     : 0 filas (0.00 %)
  is_high                 : 955 filas (1.15 %)
  is_outlier              : 1126 filas (1.36 %)
  is_invalid_category     : 1 filas (0.00 %)
  is_invalid_monotonic    : 0 filas (0.00 %)
  is_gap                  : 44 filas (0.05 %)
  is_small_delta          : 501 filas (0.61 %)
```

Figura 24. Resumen de calidad de datos

- ***Almacenamiento en la tabla mediciones con indicadores de calidad***

La salida final de esta fase es un dataset sin modificaciones, pero con un indicador de calidad por cada fila. Estos datos se almacenan en:

iot.mediciones (formato largo) con referencia a iot.indicador_calidad

La tabla iot.mediciones almacena la versión curada pero no modificada de las ingestas sensoriales, junto con el indicador de calidad asignado, esta estructurada de la siguiente manera:

- **medicion_id (BIGSERIAL)**: identificador interno único de la fila; permite trazabilidad y referencia independiente del timestamp.
- **despliegue_id (INTEGER, FK)**: referencia al despliegue asociado; garantiza que cada medición pueda relacionarse con los metadatos del despliegue y que las mediciones se borren si se elimina el despliegue (ON DELETE CASCADE)
- **ts_utc (TIMESTAMPTZ)**: marca de tiempo de la medición en UTC; es obligatoria y representa la dimensión temporal primaria del registro
- **ts_local_tz (TIMESTAMPTZ GENERATED STORED)**: columna calculada que convierte ts_utc a la zona horaria local (America/Guayaquil); facilita consultas y visualizaciones en hora local sin perder la referencia UTC. Al definirse como STORED, el valor se conserva en disco evitando recalculaciones frecuentes.
- **variable (VARCHAR, FK)**: nombre original de la variable tal como existe en la base de datos de variables; la FK asegura consistencia del catálogo de variables y facilita joins para metadatos adicionales (unidad, cname).
- **valor (NUMERIC)**: valor medido (puede ser NULL si falta o si la medición fue convertida a missing por las banderas). Usar tipo NUMERIC permite preservar precisión numérica para mediciones físicas.
- **indicador_calidad (SMALLINT, FK)**: código que representa la calidad asignada. Esta columna liga cada fila con la tabla iot.indicador_calidad, que documenta la semántica y severidad de cada código.
- **importado_en (TIMESTAMPTZ DEFAULT now())**: registro automático del momento en que la fila es insertada en la tabla iot.mediciones; permite auditoría de la operación de ingestión.

La persistencia de las mediciones originales junto con un indicador_calidad es una decisión de diseño crítica que responde a múltiples requerimientos funcionales y operativos. Mantener los datos originales separados de las versiones corregidas garantiza que la fase de evaluación (Fase 1) sea no destructiva: ninguna transformación

aplicada posteriormente sobrescribe la evidencia inicial. Esto permite reprocesar, ajustar umbrales o mejorar algoritmos sin perder la fuente.

El guardado de datos se hace a través de API RESFULL por inserción masiva.

La Figura 25 muestra el resumen de carga de mediciones ingresadas a la base de datos.

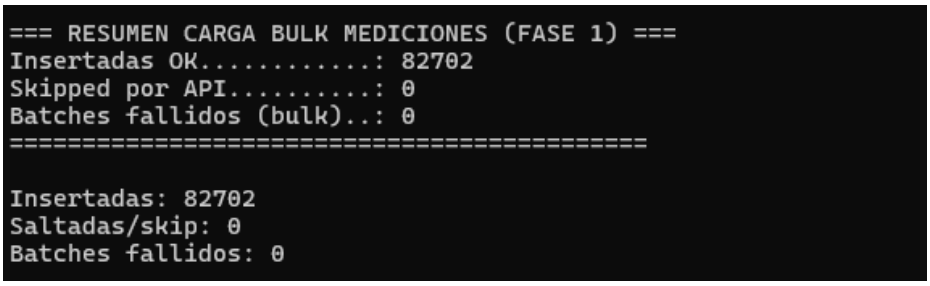


Figura 25. Almacenamiento de datos con indicador de calidad

Con la inserción de los registros en iot.mediciones, la Fase 1 del pipeline se considera finalizada: los datos originales quedan almacenados, validados y etiquetados con un indicador_calidad que resume la severidad y naturaleza de las anomalías detectadas. Esta tabla constituye la base inmutable para las fases posteriores de preprocesamiento y para cualquier proceso de auditoría, análisis exploratorio o reprocesamiento que se requiera realizar.

3.5.4 Fase 2: Corrección, imputación y reconstrucción temporal

La segunda fase del sistema tiene como objetivo generar un dataset limpio, uniforme y completamente alineado temporalmente, adecuado para ingeniería de características y entrenamiento de modelos de mantenimiento predictivo. A diferencia de la fase anterior, esta etapa sí modifica los valores, aplicando imputaciones controladas y normalizando la estructura temporal.

- ***Conversión de anomalías severas a NaN***

A partir de los indicadores obtenidos en la Fase 1, todas las mediciones consideradas severamente anómalas outliers extremos, valores físicamente imposibles, categorías inválidas, contadores no monótonos y valores faltantes se transformaron a NaN.

El sistema convierte sistemáticamente todas estas observaciones en NaN, garantizando que ningún valor incorrecto participe en la imputación (evitando "propagar errores"),

se preserve la información de dónde el sensor falló, necesaria para evaluaciones de confiabilidad y que los métodos de imputación operen solo sobre huecos plausiblemente recuperables.

Esta etapa corresponde al procedimiento implementado en la función convert_flags_nan() del sistema. La función recibe el dataset de la fase 1 que contiene las banderas de calidad, crea una copia de trabajo para evitar modificaciones accidentales in-place. Construye una máscara booleana mask_severa que agrupa varias banderas severas definidas en FLAG_AS_MISSING_COLS (is_invalid_physical, is_invalid_category, is_invalid_monotonic, is_outlier, is_missing). Devuelve un dataframe modificado con valores NaN en las observaciones (flags) consideradas no confiables detectadas en la fase anterior.

La Figura 26 muestra el método de Python para la conversión de anomalías a NaN.

```python
def convert_flags_nan(df_limpio: pd.DataFrame) -> pd.DataFrame:
    if df_limpio.empty:
        return df_limpio.copy()
    df = df_limpio.copy()
    if "valor" not in df.columns:
        raise ValueError("Falta la columna 'valor' en df_limpio.")
    mask_severa = False
    for col in FLAG_AS_MISSING_COLS:
        if col in df.columns:
            mask_severa = mask_severa | df[col].astype(bool)
    df.loc[mask_severa, "valor"] = pd.NA

    return df
```

Figura 26. Método de conversión de anomalías a NaN

Convertir estos registros a NaN evita que la imputación utilice valores erróneos como base para generar estimaciones, reduciendo la probabilidad de introducir datos ficticios que puedan enmascarar fallas reales del activo.

La operación es no destructiva respecto a la evidencia original: los registros sin modificar permanecen en la tabla iot.mediciones si el usuario u operador decidió guardarlos antes.

- ***Asignación de slots temporales y corrección del jitter***

Los sensores ABB no garantizan una marca temporal exacta para todas las variables: algunas mediciones se retrasan o adelantan algunos segundos o incluso minutos. Para

resolver esta desalineación, todas las mediciones son asignadas al slot temporal más cercano en una rejilla fija de 15 minutos, utilizando un parámetro de tolerancia (*jitter*).

Esto permite:

- Alinear correctamente mediciones provenientes del mismo instante físico
- Reducir las inconsistencias entre variables
- Genera un dataset perfectamente sincronizado
- Evita duplicados o eventos fuera de fase entre variables
- Preparar el dataset para convertirlo a formato ancho

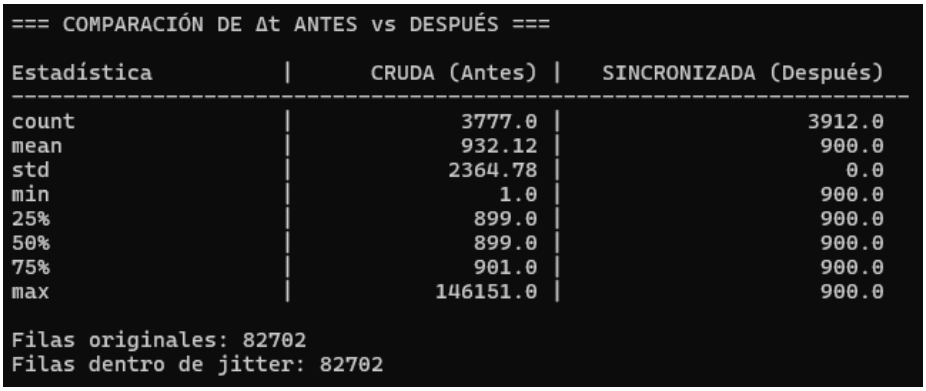
Por lo cual, para corregir la serie temporal, el sistema aplica una rejilla temporal de 15 minutos y un proceso de asignación al slot más cercano. Para cada medición propone un ts_slot redondeando ts_utc al múltiplo más cercano de frequencia y calcula la diferencia absoluta en segundos entre ts_utc y ts_slot. Si esa diferencia excede jitter_max_seconds, la medición se considera "fuera de tolerancia" y se excluye del conjunto que continuará el preprocesado. Se aplica floor/ceil para evitar perder la primera o última observación cuando éstas no caen exactamente en el múltiplo de la rejilla. La rejilla es el eje temporal canónico sobre el que se construye el dataset ancho.

La tolerancia máxima en segundos (jitter_max_seconds) que se considera corregible por redondeo. Se ajusta según EDA, valores típicos en pruebas fueron 60 – 450 s.

Sin embargo, la elección de tolerancia fue de 60 s que permite absorber variaciones pequeñas sin perder integridad temporal.

El redondeo corrige jitter leve del sensor y permite que mediciones que corresponden al mismo estado físico queden alineadas en la misma fila del dataset ancho. Excluir mediciones fuera de tolerancia evita introducción de ruido temporal.

La Figura 27 muestra el resumen de estadísticas descriptivas comparando el antes y después de timestamp del despliegue.



```
=== COMPARACIÓN DE Δt ANTES vs DESPUÉS ===

Estadística          |      CRUDA (Antes) |   SINCRONIZADA (Después)
-----------------------------------------------------------------------
count                |             3777.0 |                     3912.0
mean                 |              932.12 |                      900.0
std                  |             2364.78 |                        0.0
min                  |                1.0 |                      900.0
25%                  |              899.0 |                      900.0
50%                  |              899.0 |                      900.0
75%                  |              901.0 |                      900.0
max                  |           146151.0 |                      900.0

Filas originales: 82702
Filas dentro de jitter: 82702
```

Figura 27. Estadístico descriptivo temporal antes y después de corrección de jitter

El resultado de esta corrección temporal mejora significativamente la estadística descriptiva de la serie temporal.

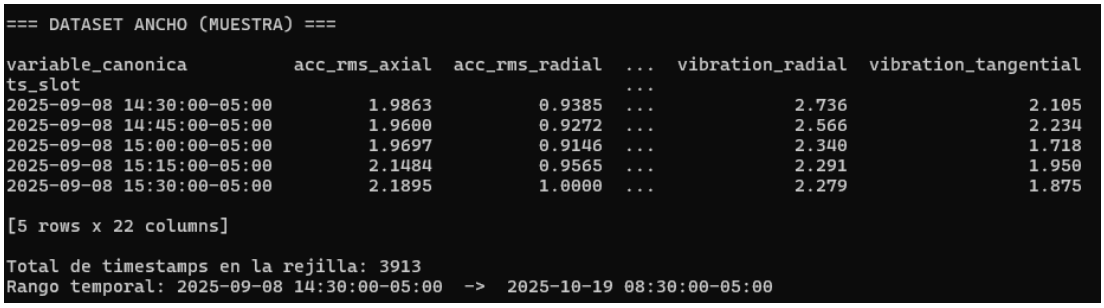
- ***Construcción de la rejilla temporal y formato ancho***

Una vez obtenidos los ts_slot, el sistema genera la rejilla temporal completa entre el primer y último timestamp válido del despliegue. Esta rejilla contiene todas las marcas esperadas a intervalos constantes de 15 minutos.

Posteriormente, el dataset se reorganiza a formato ancho, donde:

- Cada fila representa un instante de tiempo
- Cada columna representa una variable sensorial
- Cada celda corresponde al valor (original o imputado) de dicha variable en ese instante.

Cuando una variable no reporta un valor en un slot determinado, se asigna un NaN. Esto permite revelar huecos reales de información que podrían pasar inadvertidos en formato largo.Si una variable no reporta valor en cierto slot, se genera un NaN, lo que permitie identificar huecos reales de información no evidentes en el análisis crudo.



```
=== DATASET ANCHO (MUESTRA) ===

variable_canonica          acc_rms_axial  acc_rms_radial  ...  vibration_radial  vibration_tangential
ts_slot                                                   ...
2025-09-08 14:30:00-05:00         1.9863          0.9385  ...             2.736                 2.105
2025-09-08 14:45:00-05:00         1.9600          0.9272  ...             2.566                 2.234
2025-09-08 15:00:00-05:00         1.9697          0.9146  ...             2.340                 1.718
2025-09-08 15:15:00-05:00         2.1484          0.9565  ...             2.291                 1.950
2025-09-08 15:30:00-05:00         2.1895          1.0000  ...             2.279                 1.875

[5 rows x 22 columns]

Total de timestamps en la rejilla: 3913
Rango temporal: 2025-09-08 14:30:00-05:00  ->  2025-10-19 08:30:00-05:00
```

Figura 28. Dataset en formato ancho y reconstrucción de rejilla temporal

- ***Imputación limitada***

Para esta fase se aplica métodos comunes de imputación de datos con el objetivo de conservar series temporales continuas, cruciales para el cálculo de features (promedios, tendencias, valores máximos, etc.). También evita saltos o lagunas que podrían afectar la performance de modelos de clasificación o predicción.

Los métodos aplicados son:

Interpolación lineal: Un método numérico que permite estimar un valor desconocido de una función cuando este se encuentra dentro de un intervalo definido por dos puntos tabulados, interpolar consiste en calcular un valor aproximado de una función para un valor dado de la variable independiente cuando dicho valor se halla comprendido en un intervalo. El método se basa en la ecuación de la recta que une dos puntos conocidos y asume que el comportamiento entre ellos sigue una relación lineal [27].

Calcula la pendiente entre dos valores conocidos y proyecta un valor estimado para el punto faltante siguiendo una relación lineal simple. Se aplica únicamente entre dos puntos válidos, por lo que es un método confiable cuando los datos tienen tendencia suave.

Forward Fill (relleno hacia adelante): es una técnica de imputación de valores faltantes donde cada valor ausente se reemplaza por la última observación válida anterior. En pandas, esto se implementa mediante el método ffill(). Se usa para mantener la continuidad temporal de las series cuando los sensores no reportan nuevos valores. Permite evitar pérdida de registros y mantener secuencias completas. Cuando

encuentra un valor NaN, el método copia el valor no nulo más reciente hacia adelante hasta que aparece un nuevo valor real. El resultado es que los huecos se llenan con valores repetidos de la última medición válida [28].

Muchos sensores industriales mantienen su última lectura (ej. estado, temperatura, presión) hasta que hay un cambio real. Forward Fill respeta esta dinámica, evitando introducir estimaciones artificiales que puedan distorsionar señales, especialmente si los cambios reales son esporádicos.

La imputación fue diseñada para no distorsionar la serie y mantener un equilibrio entre completitud y veracidad de los datos. En el sistema la función imputar_dataset_ancho() convierte cada columna a numérico, preservando NaN. Posteriormente inicializa códigos de preprocesamiento del DataFrame con ceros. Las imputaciones se implementan mediante interpolación de Pandas y reglas específicas para cada grupo de variables.

La Figura 29 muestra los métodos de Interpolación lineal y Forward Fill en Python.

```
codes = pd.Series(0, index=s.index, dtype="int8")

is_na = s.isna()
if not is_na.any():
    return s, codes
if metodo == "linear":
    s_aux = s.interpolate(method="time")
    code_imputed = code_linear
elif metodo == "ffill":
    s_aux = s.ffill()
    code_imputed = code_ffill
else:
    return s, codes
```

Figura 29. Métodos de imputación

Para cada columna decide la estrategia de imputación solo donde es estadísticamente razonable y operacionalmente seguro.

La Tabla 23 muestra los métodos de imputación por tipo de variable y justificación de su aplicabilidad en este sistema.

Tabla 23. Métodos de imputación por tipo de variable

Tipo de variable	Método de imputación	Justificación
Continuas	Interpolación lineal	Cambian suavemente; los valores intermedios tienen significado físico
Categóricas	Forward-fill	El estado del equipo se mantiene hasta que ocurra un cambio
Acumulativas	Forward-fill	Preservan la naturaleza monótona

Por otro lado, el sistema solo imputa cuando los gaps tienen longitud máxima de dos intervalos consecutivos (30 minutos). Cualquier brecha mayor se conserva como NaN y se clasifica como missing_not_imputed.

- Un gap o hueco grande implica un período de ausencia real de mediciones, no una variación suave.
- Imputar grandes intervalos podría introducir información ficticia incompatible con el comportamiento operativo de la bomba.
- Para mantenimiento predictivo, es preferible conservar la evidencia del fallo del sensor.

- ***Asignación del indicador de preprocesamiento***

Una vez finalizada la imputación limitada en el dataset ancho, el sistema genera un indicador explícito de preprocesamiento (processing_code) para cada celda. Este indicador resume la acción realizada sobre el dato durante la Fase 2 y permite garantizar trazabilidad y reproducibilidad en etapas posteriores.

El processing_code se asigna a nivel de cada combinación (*ts_slot*, *variable*).

La Tabla 24 muestra el resumen de indicadores de preprocesamiento.

Tabla 24. Resumen de indicadores de preprocesamiento

Código	Nombre	Significado
0	Original	Valor original utilizado sin modificaciones
1	imputed_linear	Imputado mediante interpolación lineal
2	imputed_ffill	Imputado mediante propagación hacia adelante
3	missing_not_imputed	No imputado debido a gap grande o anomalía severa

Estos códigos provienen de la tabla **iot.indicador_preprocesamiento**, equivalente funcional a la tabla de indicadores de calidad utilizada en la Fase 1. Cada fase posee su propio catálogo de códigos, lo que permite distinguir claramente las acciones de detección (Fase 1) y las acciones de corrección/imputación (Fase 2).

Este indicador permitió documentar completamente el impacto del preprocesamiento sobre cada medición.

La decisión es asignada mediante una función jerárquica, garantizando que:

- Un valor severamente dañado nunca se considere imputado
- Las imputaciones queden diferenciadas por método
- Sea posible medir la calidad del dataset final.

La Figura 30 muestra el resultado de mediciones preprocesadas en el despliegue.

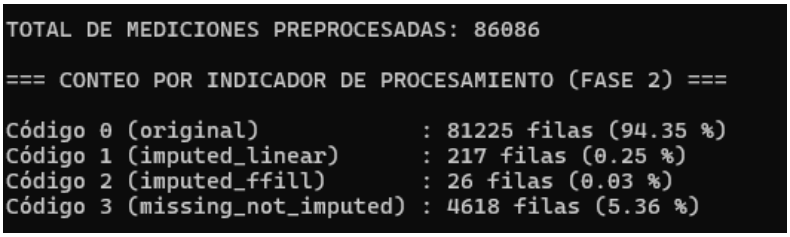


Figura 30. Resumen de mediciones preprocesadas

El resultado de esta etapa es un dataset ancho con valores anómalos corregidos y sincronizados con su respectivo indicador de preprocesamiento.

- ***Conversión final a formato largo y almacenamiento***

Una vez completada la imputación y asignación de códigos, el dataset ancho se convierte nuevamente a formato largo para cumplir con el modelo lógico de la base de datos. Durante esta conversión se preservan explícitamente los valores NaN (incluidos aquellos correspondientes a gaps grandes).

La conversión se realiza de esta forma:

- Desapila el dataset ancho usando stack(dropna=False) para mantener explícitamente todos los valores, incluidos NaN y sus códigos asociados.

- Integra el indicador de preprocesamiento correspondiente a cada celda, preservando la distinción entre datos originales, imputados y no imputados.
- Restaura el nombre original de la variable, revirtiendo los nombres canónicos utilizados para pivotear.
- Agrega el identificador del despliegue y demás campos requeridos para mantener consistencia con la estructura del sistema.

El resultado es un DataFrame en formato largo listo para ser persistido en la tabla.

Finalmente, las mediciones limpias se insertan en la tabla iot.mediciones_limpias

Esta tabla mantiene la misma estructura base que iot.mediciones (Fase 1), incluyendo campos como:

- despliegue_id
- ts_utc / ts_slot
- variable
- valor
- indicador_preprocesamiento
- importado_en

Sin embargo, a diferencia de la tabla de la Fase 1, todos los datos almacenados aquí ya se encuentran sincronizados a una rejilla uniforme de 15 minutos, normalizados por jitter sin duplicados, con huecos representados explícitamente, con imputaciones controladas y su indicador de preprocesamiento correspondiente

Es decir: iot.mediciones_limpias representa el primer dataset completamente estandarizado, listo para ingeniería de características y entrenamiento de modelos predictivos.

La inserción final se realiza mediante la API Resfull permitiendo registrar en la base de datos tanto los valores como sus metadatos de procesamiento de forma eficiente.

La Figura 31 muestra el almacenamiento de datos limpios por lotes de la Fase 2.

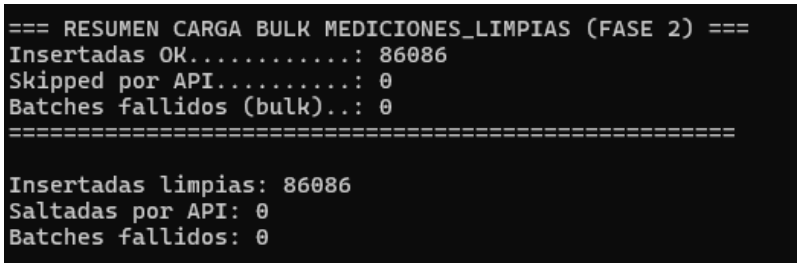


Figura 31. Almacenamiento de mediciones limpias

Esta tabla constituye la versión final del dataset limpio y estructurado, apto para su uso en fases posteriores, como la ingeniería de características y el entrenamiento de modelos predictivos.

Este dataset constituye la versión final y depurada del despliegue, lista para ser utilizada en:

- Ingeniería de características
- Modelos de Machine Learning
- Análisis comparativo con las mediciones originales.

La Fase 2 del pipeline sincroniza las mediciones a una rejilla regular de 15 minutos, corrige el jitter mediante redondeo controlado y colapsa múltiples lecturas por slot conservando la última observación. A continuación, se pivotea el dataset a formato ancho y se remuestrea la rejilla para asegurar la presencia explícita de todos los slots esperados. La imputación se aplica de forma limitada y diferenciada: interpolación lineal para variables continuas y vibracionales, y propagación hacia adelante para variables categóricas y acumulativas; además, solo se imputaron gaps de hasta dos slots consecutivos (30 minutos). Todas las operaciones preservaron la traza del procesamiento mediante un processing_code por celda que describe si el valor fue original, imputado (lineal/ffill) o no imputado.

3.5.5 Fase 3: Construcción de Características (Feature Engineering)

Esta etapa tiene como objetivo transformar los datos sensoriales ya preprocesados en la Fase 2 en un conjunto de características representativas que permitieran describir el comportamiento de las señales medidas en el tiempo. A diferencia de enfoques tradicionales de análisis vibracional donde se aplican transformadas como FFT o

espectrogramas en este proyecto se optó por no utilizar transformadas en frecuencia, debido a que las variables disponibles en la base de datos eran principalmente valores RMS (Root Mean Square) ya calculados por el sistema de adquisición. Al tratarse de medidas ya procesadas, no se cuenta con las señales en el dominio temporal necesarias para aplicar transformadas espectrales. Por esta razón, la generación de características se centró en métodos estadísticos, temporales y de consistencia interna.

El propósito de esta fase fue construir un conjunto de características que cumplieran con:

- Describir de manera robusta el comportamiento de cada variable sensorial en intervalos temporales definidos, incluso cuando las señales presentan ruido o valores faltantes.
- Permitir que los modelos de machine learning trabajen con información más compacta y significativa, reduciendo variabilidad dentro de cada ventana.
- Mantener trazabilidad con el preprocesamiento, incorporando un atributo que indique el nivel de intervención o imputación de cada ventana (indicador_preprocesamiento de Fase 2).
- Asegurar que cada característica sea computada únicamente cuando exista suficiencia de datos válidos, evitando que el modelo aprenda patrones erróneos

La etapa de generación de características toma como entrada el dataset resultante de la Fase 2 y produce un conjunto de indicadores temporales resumen por ventana para variables continuas de vibración y físicas. Las features se calculan por despliegue y por variable, y se almacenan en la tabla iot.caracteristicas para su uso posterior en el entrenamiento de modelos.

Parámetros y reglas operativas:

- Ventanas de 3 horas (10800 segundos)
- Tamaño de slot de 15 minutos (900 segundos)
- Ventana contiene 12 muestras por variable (ventana/slot)
- Umbral mínimo de muestras válidas, 50% de las esperadas, al menos 6 valores por ventana (MIN_SAMPLES_RATIO = 0.5).

- Códigos de preprocesamiento válidos para considerar una muestra en el cálculo: 0, 1, 2 (Fase 2: original, imputed_linear, imputed_ffill). El código 3 (missing_not_imputed) se excluye de cálculos estadísticos.

Estas decisiones aseguran que los cálculos estadísticos sean estables, que ventanas incompletas no distorsionen los modelos ML.

Se trabajó exclusivamente con las variables continúas de vibración y continuas físicas, detalladas en la Tabla 17.

- ***Características estadísticas básicas***

  - Media (mean): Representa el nivel promedio de la señal dentro de la ventana. Es útil para modelos que identifican cambios sostenidos en el comportamiento operativo.

  - Desviación estándar (std): Captura la variabilidad interna. Una ventana con alta dispersión puede indicar ruido, inestabilidad o mala calidad del dato.

  - Mínimo y máximo (min, max): Permiten detectar picos irregulares o saturaciones.
  - Rango (range): Diferencia directa entre máximo y mínimo. Útil para identificar ventanas con movimientos abruptos.
  - Rango intercuartil (IQR): Métrica robusta frente a outliers que describe la dispersión central.

- ***Características temporales***

  - Pendiente o tendencia (slope): Se obtiene mediante una regresión lineal simple entre el valor y el tiempo relativo dentro de la ventana. Captura si la señal presenta una tendencia ascendente o descendente, útil para anticipar cambios graduales.

- ***Características de completitud***

  - Conteo de muestras válidas (count): Indica cuántos registros dentro de la ventana fueron considerados correctos según el indicador_preprocesamiento.

Esta característica es especialmente relevante para modelos que integran calidad del dato en la predicción.

- ***Derivación vectorial (Vibración RMS)***

  – Cuando las tres componentes RMS (axial, radial y tangencial) están disponibles, se generó adicionalmente: RMS Vectorial:

  El calculo de RMS vectorial es con la Fórmula 5:

$$\mathbf{acc_rms_vector} = \sqrt{\mathbf{x2} + \mathbf{y2} + \mathbf{z2}} \qquad (5)$$

  Esta característica resume la energía vibratoria total del equipo y es comúnmente usada en monitoreo industrial.

La función de generación de características incorporó controles adicionales de integridad en la cual se consideraron características si la ventana cumplía el mínimo de datos válidos. Si alguna ventana contenía menos del 50% de muestras correctas, simplemente no se generaba. Las características mantienen el indicador máximo de preprocesamiento dentro de la ventana, garantizando trazabilidad y se descartaron cálculos cuando los valores eran en su mayoría NaN, permitiendo que las características obtenidas sean estables, confiables y útiles para los modelos. La Figura 32 muestras los métodos de creación de características en base a las variables seleccionadas en Python.

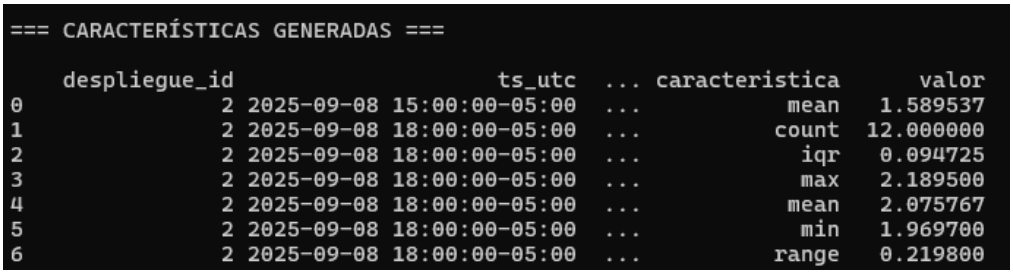
```
mean_val = float(np.nanmean(vals))
std_val = float(np.nanstd(vals, ddof=1)) if n_valid > 1 else 0.0
min_val = float(np.nanmin(vals))
max_val = float(np.nanmax(vals))
range_val = float(max_val - min_val)
iqr_val = _iqr(vals[~np.isnan(vals)]) if n_valid > 0 else float("nan")

times = (win_valid.index.view("int64") // 10**9).astype(float)
if len(times) > 0:
    t0 = times.min()
    times_rel = times - t0
    slope_val = _slope_time(vals.astype(float), times_rel) if n_valid > 1 else 0.0
else:
    slope_val = 0.0

count_val = int(n_valid)
indicator_window = int(win_valid["indicador_preprocesamiento"].max()) if "indicador_preprocesamiento" in win_valid.columns else 0
```

Figura 32. Métodos de creación de características

El producto final de esta etapa es un dataframe en formato largo con las columnas: despliegue_id, ts_utc (timestamp representativo del cierre de la ventana), variable, característica, valor, ventana_s (segundos), indicador_preprocesamiento. Cada fila representa una característica puntal generada de una variable en una ventana temporal especifica.



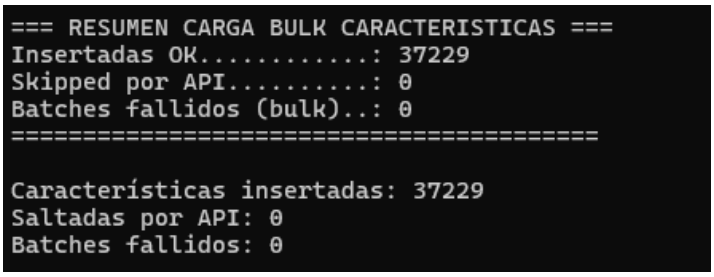
```
=== CARACTERÍSTICAS GENERADAS ===

   despliegue_id                   ts_utc  ... caracteristica      valor
0              2 2025-09-08 15:00:00-05:00  ...           mean   1.589537
1              2 2025-09-08 18:00:00-05:00  ...          count  12.000000
2              2 2025-09-08 18:00:00-05:00  ...            iqr   0.094725
3              2 2025-09-08 18:00:00-05:00  ...            max   2.189500
4              2 2025-09-08 18:00:00-05:00  ...           mean   2.075767
5              2 2025-09-08 18:00:00-05:00  ...            min   1.969700
6              2 2025-09-08 18:00:00-05:00  ...          range   0.219800
```

Figura 33. Dataset con características nuevas

Posteriormente las características generadas se guardan en la base de datos en la tabla iot.caracteristicas, para el campo indicador_preprocesamiento sé verifica que, si todos los puntos dentro de la ventana son originales, la característica se marca con código 0. Si la ventana contiene valores imputados, el indicador toma el código predominante entre 1 (interpolación lineal) o 2 (forward-fill), según el tipo de imputación aplicada. Para la inserción se usa el endpoint masivo de la Api Restful.

La Figura 34 muestra el resultado de guardar las características en formato largo de un despliegue determinado.



```
=== RESUMEN CARGA BULK CARACTERISTICAS ===
Insertadas OK.............: 37229
Skipped por API...........: 0
Batches fallidos (bulk)..: 0
==========================================

Características insertadas: 37229
Saltadas por API: 0
Batches fallidos: 0
```

Figura 34. Almacenamiento de características en la base de datos.

Tras la inserción del dataset en iot.caracteristicas, el sistema completa la etapa de preparación: los datos originales y su indicador de calidad permanecen en iot.mediciones, los datos preprocesados en iot.mediciones_limpias y las features en iot.caracteristicas.

3.6 Fase 4: Modelado (Modeling)

En esta fase se desarrolló un modelo de clasificación binaria capaz de identificar automáticamente si una medición sensorial es “BUENA” (0) o “MALA” (1). A diferencia de otros proyectos donde el modelo se entrena sobre datos completamente preprocesados, en este caso se trabajó intencionalmente con el dataset crudo, proveniente directamente de la tabla iot.mediciones, al cual únicamente se añadió el campo indicador_calidad construida en la Fase 1.

Esta decisión se tomó porque el objetivo del clasificador no es utilizar datos limpios, sino aprender a detectar patrones que indiquen mala calidad directamente desde el dato original, replicando la evaluación manual-experticia del indicador de calidad. Por lo tanto, se conservaron los valores reales, incluyendo ruido, posibles anomalías, variabilidad natural y registros imperfectos, con el fin de asegurar que el modelo representara fielmente las condiciones operativas de captura.

Este modelo no busca predecir fallas, sino únicamente evaluar la calidad del dato entrante para:

- Analizar cuantitativamente el impacto de la calidad del dato en el dataset.
- Permitir, a futuro, un mecanismo rápido de clasificación en tiempo real de mediciones nuevas.

Primero se toma la tabla de la Fase 1 que contiene las mediciones en formato largo junto con el indicador de calidad generado por el sistema. Se convierte ese indicador multiclase a una etiqueta binaria, 0 si la medición fue considerada válida, 1 si se detectó cualquier anomalía relevante. Para cada registro se mantuvieron las columnas numéricas (continuas) seleccionadas y se descarta temporalmente variables categóricas y acumulativas porque el objetivo del clasificador es probar cuánto aportan los valores numéricos por sí solos.

3.6.1 Selección de Variables

Se seleccionaron únicamente las variables continúas relacionadas con la condición del motor y el comportamiento, clasificadas como Variables Continuas físicas relacionadas con vibración y no relacionadas con vibración, detallada en la Tabla 17.

Estas variables fueron tomadas directamente tal como aparecen en el dataset original, sin aplicar imputaciones ni reconstrucciones temporales, ya que el interés del modelo es clasificar los registros tal como llegan desde el sensor.

Estas variables reflejan condiciones físicas y vibratorias del motor, y son directamente afectadas por anomalías reales del sensor o el proceso. Se excluyeron variables categóricas y acumulativas. El modelo se centra únicamente en el valor del dato, no en su contexto temporal.

3.6.2 Construcción del Dataset

Variable objetivo (Y):

- 0 = dato BUENO
- 1 = dato MALO

Convertido desde el indicador de calidad multiclase (6 códigos) a binario.

Variables predictoras (X):

- Las variables seleccionadas anteriormente.

Partición del Dataset

Se utilizó:

- 80% para entrenamiento (training)
- 20% para prueba (test)

Con estratificación, para mantener proporcionalidad entre buenos y malos.

3.6.3 Modelos de clasificación entrenados

Se entreno dos modelos simples y comparables: Regresión Logística caracterizada como una línea base interpretable y un Random Forest como un modelo no lineal, ideal frente al ruido y proveniente de la práctica estándar en problemas industriales.

- ***Regresión Logística***

La regresión logística es un modelo estadístico y de aprendizaje automático utilizado para resolver problemas de clasificación, especialmente cuando el objetivo es predecir si un evento pertenece o no a una categoría específica. Según IBM, este modelo permite estimar probabilidades y asignar observaciones a clases como sí y no, fallo y no fallo o normal y anómalo siendo ampliamente usado en múltiples industrias debido a su simplicidad e interpretabilidad. La regresión logística se basa en la relación entre variables independientes y una variable dependiente categórica, convirtiéndola en una herramienta eficaz para análisis predictivo orientado a la toma de decisiones[29] .

En cuanto a su funcionamiento, la regresión logística aplica la función sigmoide, que transforma una combinación lineal de variables de entrada en un valor entre 0 y 1, interpretado como probabilidad. IBM destaca que este proceso permite construir modelos que predicen la probabilidad de un resultado binario, mientras que GeeksforGeeks describe que el algoritmo ajusta los parámetros a través de técnicas como maximum likelihood estimation para mejorar la precisión. En aplicaciones prácticas, se entrena con datos previamente limpiados y preparados, creando un modelo capaz de clasificar eventos y detectar patrones basados en variables numéricas o derivadas del procesamiento de señales [30].

- ***Random Forest***

El algoritmo Random Forest es un modelo de aprendizaje automático basado en el uso de múltiples árboles de decisión que trabajan conjuntamente para resolver problemas de clasificación y predicción. Su objetivo principal es mejorar la precisión y robustez de los modelos individuales mediante un enfoque de ensembling, donde cada árbol aporta una parte de la decisión final. Este modelo es ampliamente utilizado en el análisis predictivo porque permite manejar grandes cantidades de variables, detectar patrones complejos y ofrecer resultados estables incluso en contextos donde existen datos ruidosos o no lineales.

En cuanto a su funcionamiento este algoritmo construye numerosos árboles de decisión utilizando subconjuntos aleatorios de datos y características, proceso conocido como bootstrap aggregation o bagging. IBM explica que cada árbol vota por

una clase y el resultado final se determina mediante mayoría, lo que reduce el riesgo de sobreajuste característico de un único árbol. Esta técnica se aplica entrenando el bosque con datos previamente limpiados, reduciendo la presencia de outliers y usando variables derivadas mediante ingeniería de características. Una vez entrenado, el modelo es capaz de clasificar nuevas observaciones con alta precisión y de identificar qué variables influyen más en la decisión [31].

En el procesamiento de datos sensoriales, Random Forest influye al analizar simultáneamente múltiples mediciones como vibración, presión, temperatura y otros indicadores de condición del equipo. Debido a su capacidad para manejar ruido y variabilidad, resulta especialmente útil en mantenimiento predictivo, donde los sensores pueden generar patrones complejos o inconsistentes. Este tipo de modelo permite detectar anomalías y predecir fallas con mayor exactitud, integrando señales diversas y priorizando las más relevantes. Esto convierte a Random Forest en una herramienta ideal para interpretar datos sensoriales y generar diagnósticos confiables sobre el estado operativo de los sistemas industriales [31].

Para la regresión logística se usó regularización por defecto, se estableció como 2000 el límite máximo de pasos que puede dar para aprender, teniendo que cuenta que, si los datos son complejos, necesita más pasos para ajustarse bien con esto se evita que el modelo se quede corto, pueda encontrar una buena solución y terminar su proceso de aprendizaje. Se aplica el parámetro de RANDOM_STATE para obtener los mismos resultados para experimentación. Para evitar que el modelo solo aprenda de la clase mayoritaria, se ajusta el peso según cuantos registros hay de cada clase (0 y 1) con class_weight='balanced'.

En Random Forest se estableció como 200 el numero de árboles que tendrá el bosque para mayor estabilidad y precisión, la cantidad fue ajustada para tener balance entre calidad y tiempo. Al igual que Regresión logística se asegura que los resultados sean reproducibles para fines de experimentación. Dado a que la clase objetivo está desequilibrada es decir que existe muy pocos en la clase 1 que corresponde a los datos clasificados como MALOS, se estableció que cada árbol tome en cuenta las clases de forma proporcional y evite aprender solo de la clase mayoritaria y considera que la

clase minoritaria es importante ya que en este estudio se pretender detectar cuando un dato se considera MALO o tiene fallo.

```
scaler = StandardScaler()
X_train_scaled = X_train.copy()
X_test_scaled = X_test.copy()
X_train_scaled.iloc[:, 0] = scaler.fit_transform(X_train.iloc[:, [0]]).ravel()
X_test_scaled.iloc[:, 0]  = scaler.transform(X_test.iloc[:, [0]]).ravel()

lr = LogisticRegression(max_iter=2000, random_state=RANDOM_STATE, class_weight="balanced")
lr.fit(X_train_scaled, y_train)

rf = RandomForestClassifier(n_estimators=200, random_state=RANDOM_STATE, class_weight="balanced")
rf.fit(X_train, y_train)
```

Figura 35. Parámetros principales de los modelos

Posteriormente se entrenaron los modelos con el fin de comparar su desempeño, Regresión Logística, como enfoque lineal base, y Random Forest, como modelo no lineal capaz de manejar relaciones más complejas entre variables. Después del entrenamiento, ambos modelos se evaluaron empleando métricas estándar como accuracy, precision, recall, F1-score, matriz de confusión, con el objetivo de analizar con detalle el comportamiento de cada clasificador.

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=TEST_SIZE, random_state=RANDOM_STATE, stratify=y
)
print("Train:", X_train.shape, "Test:", X_test.shape)
print("Proporción y_train:", np.bincount(y_train) / len(y_train))
print("Proporción y_test:", np.bincount(y_test) / len(y_test))
```

✓ 0.0s

```
Train: (45121, 16) Test: (11281, 16)
Proporción y_train: [0.97949957 0.02050043]
Proporción y_test: [0.97952309 0.02047691]
```

Figura 36. Entrenamiento de modelos

3.6.4 Entrenamiento y validación

Una vez entrenados los dos modelos de clasificación Regresión Logística y Random Forest se procedió a evaluar su desempeño sobre el conjunto de prueba (20 % del dataset total). Las métricas utilizadas fueron accuracy, precisión, recall, F1-score y la matriz de confusión, con el fin de analizar el rendimiento en la detección de mediciones de baja calidad (clase 1).

```python
print("=== LOGISTIC REGRESSION ===")
print(f"Accuracy: {acc_lr:.4f}  Precision: {prec_lr:.4f}  Recall: {rec_lr:.4f}  F1: {f1_lr:.4f}")
print(classification_report(y_test, y_pred_lr, zero_division=0))
print("Confusion Matrix (LR):")
print(confusion_matrix(y_test, y_pred_lr))

print("\n=== RANDOM FOREST ===")
print(f"Accuracy: {acc_rf:.4f}  Precision: {prec_rf:.4f}  Recall: {rec_rf:.4f}  F1: {f1_rf:.4f}")
print(classification_report(y_test, y_pred_rf, zero_division=0))
print("Confusion Matrix (RF):")
print(confusion_matrix(y_test, y_pred_rf))
```

Figura 37. Evaluación de métricas claves de modelos entrenados

El modelo de Regresión Logística mostró un comportamiento claramente insuficiente para el problema. Aunque obtuvo un recall extremadamente alto para la clase 1 (0.9913), lo hizo a costa de una precisión muy baja (0.0303) y un accuracy general reducido (0.3492). Esto significa que el modelo logró identificar casi todos los datos malos, pero generando una gran cantidad de falsos positivos, clasificando erróneamente como "MALOS" muchos datos que en realidad eran correctos.

La Figura 38 muestra el resultado del resumen de las métricas utilizadas para su evaluación.

```
=== LOGISTIC REGRESSION ===

Accuracy: 0.3492  Precision: 0.0303  Recall: 0.9913  F1: 0.0587
              precision    recall  f1-score   support

           0       1.00      0.34      0.50     11050
           1       0.03      0.99      0.06       231

    accuracy                           0.35     11281
   macro avg       0.51      0.66      0.28     11281
weighted avg       0.98      0.35      0.49     11281

Confusion Matrix (LR):
[[3710 7340]
 [   2  229]]
```

Figura 38. Resumen de métricas de Regresión logística

La matriz de confusión evidencia este fenómeno: de los 11050 datos buenos, solo 3710 fueron clasificados correctamente, mientras que 7340 fueron etiquetados de forma incorrecta como malos. Esto implica que el modelo tiende a sobre clasificar la clase

minoritaria, inutilizándolo operativamente para un sistema de diagnóstico temprano de la calidad en tiempo real.

Tabla 25. Matriz de confusión de Regresión Logística

VALORES PREDICCIÓN	VALORES REALES	
	Verdaderos Positivos 3710	**Falsos Positivos** 7340
	Falsos Negativos 2	**Verdaderos Negativos** 229

En resumen, la Regresión Logística no fue adecuada para este tipo de datos, ya que no logró capturar la estructura no lineal presente en las señales sensoriales.

El modelo Random Forest presentó un rendimiento notablemente superior. Alcanzó un accuracy de 0.9957, una precisión de 0.8396, un recall de 0.9740 y un F1-score de 0.9018, lo cual indica un equilibrio sólido entre detección de datos malos y minimización de falsos positivos.

```
=== RANDOM FOREST ===

Accuracy: 0.9957  Precision: 0.8396  Recall: 0.9740  F1: 0.9018
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     11050
           1       0.84      0.97      0.90       231

    accuracy                           1.00     11281
...
[[11007    43]
 [    6   225]]
```

Figura 39. Resumen de métricas claves de Random Forest

La matriz de confusión confirma su desempeño: de 11050 datos buenos, 11007 fueron clasificados correctamente y solo 43 fueron falsamente etiquetados como malos. Para la clase minoritaria (231 datos malos), el modelo identificó correctamente 225 y solo dejó escapar 6. Estas cifras demuestran que Random Forest no solamente logra

detectar la mayoría de las anomalías reales, sino que también preserva la confiabilidad del sistema evitando alarmas innecesarias.

Tabla 26. Matriz de confusión de Random Forest

VALORES PREDICCIÓN	VALORES REALES	
	Verdaderos Positivos 11007	**Falsos Positivos** 43
	Falsos Negativos 6	**Verdaderos Negativos** 225

Este rendimiento superior se explica porque Random Forest captura relaciones no lineales entre variables sensoriales, maneja adecuadamente interacciones entre magnitudes físicas y vibracionales y es robusto ante ruido y variabilidad en señales industriales.

Con base en los resultados cuantitativos y en su estabilidad frente a ruido y desbalance de clases, Random Forest fue seleccionado como el modelo final para el clasificador binario de calidad de datos. Sus métricas reflejan un desempeño consistente tanto para datos buenos como para datos malos, lo que lo hace adecuado para ser integrado en un flujo de evaluación rápida en tiempo real dentro del sistema de preprocesamiento.

3.7 Fase 5: Evaluación (Evaluation)

La fase de evaluación tiene como finalidad determinar si los resultados obtenidos por el sistema de preprocesamiento y por el modelo de clasificación satisfacen los objetivos planteados en el proyecto. En este caso, la evaluación se realizó atendiendo a tres ejes principales: calidad del preprocesamiento, consistencia de los datos resultantes, y desempeño del modelo encargado de analizar el impacto de la calidad de los datos sensoriales.

La evaluación del sistema se realizó comparando el estado de los datos antes y después de cada fase de procesamiento de un despliegue en específico, detallada en Figura 24.

En la Fase 1, centrada en la detección de problemas de calidad, se identificó que el **98.58%** de las mediciones no presentaban fallas y podían considerarse correctas. Sin embargo, el sistema detectó un conjunto relevante de anomalías: **1.36%** correspondió a valores extremos, **1.15%** a valores anormalmente altos, **0.05%** a brechas temporales y menos del **0.01%** a categorías inválidas. Aunque estos porcentajes parecen pequeños, su presencia sostenida puede comprometer la confiabilidad de un modelo de mantenimiento predictivo, lo cual evidencia la necesidad de un preprocesamiento estructurado.

En cuanto a la evaluación de la estructura temporal, detallada en la Figura 27, los datos crudos presentaban una alta variabilidad en el intervalo entre mediciones (Δt), con una desviación estándar de **2364.78 s** y un valor máximo atípico de **146,151 s**, confirmando desincronización severa. Tras la sincronización, el Δt se estabilizó completamente en **900 s**, eliminando jitter y garantizando una base temporal uniforme. Esto representa una mejora del 100% en la regularidad temporal, permitiendo que los modelos posteriores trabajen sobre una serie coherente y comparable en el tiempo.

En la Fase 2, enfocada en el tratamiento de datos faltantes, el sistema produjo un total de **86,086** mediciones preprocesadas. De ellas, el **94.35%** permaneció como datos originales, mientras que solo el **0.25%** fue imputado mediante interpolación lineal y el **0.03%** mediante forward-fill. Adicionalmente, un **5.36%** correspondió a datos faltantes no imputados, los cuales se preservaron para evitar alterar el comportamiento real del sensor. Este resultado se detalla en la Figura 30 y confirma que el sistema aplica imputación únicamente cuando es justificable, manteniendo la integridad estadística de la señal.

Se compararon las métricas de desempeño. Los resultados de Regresión Logística en la Figura 38, mostró alta sensibilidad (recall), pero muy baja precisión con el **3.03%,** generando muchos falsos positivos y confirmando que este modelo no es adecuado para identificar patrones de mala calidad en datos sin preprocesar. En cambio, el modelo Random Forest presentó valores muy altos **de 99.5%** de accuracy, **83%** de precisión**, 97.4 %** de recall **y 90.1%** de F1-score, así como una matriz de confusión balanceada, estor resultados están detallada en la Figura 39. Esto demuestra su capacidad para aprender patrones complejos en datos crudos sin requerir imputaciones ni transformaciones avanzadas.

La evaluación también permitió analizar el impacto de la calidad del dato en el modelo predictivo. Los resultados evidenciaron que:

- El ruido y los valores irregulares presentes en el dataset crudo afectan fuertemente a modelos lineales (como la Regresión Logística).
- El indicador de calidad construido en la Fase 1 capturó adecuadamente la presencia de mediciones atípicas o erróneas.
- El Random Forest logró identificar estas mediciones incluso sin un dataset limpio, confirmando que algunas técnicas de machine learning son naturalmente más resistentes al ruido.

Finalmente, se verificó que las métricas obtenidas se alinean con los objetivos del estudio, permitiendo demostrar de manera empírica cómo la calidad del dato afecta el desempeño de los modelos y cómo un enfoque adecuado puede mejorar su capacidad de clasificación en mantenimientos predictivos.

3.8 Fase 6: Despliegue (Deployment)

En el marco de este trabajo, el despliegue se orienta bajo el principio de Transferencia de Conocimiento y Transferencia de Producto, lo que significa que el sistema de preprocesamiento no se instala directamente en un servidor operativo, sino que se entrega como un componente funcional completamente documentado para ser incorporado por el equipo encargado de la infraestructura general del proyecto institucional "Desarrollo e Implementación de un Sistema de Mantenimiento Predictivo Basado en Inteligencia Artificial para Optimizar Equipos de Bombeo Hidráulico en la Producción de Crudo".

3.8.1 Interoperabilidad con los módulos existentes

El sistema fue diseñado para integrarse sin conflictos con los módulos de software construidos por el grupo de investigación. Las principales formas de integración actualmente son:

Integración con el módulo de comunicación de datos (API RESTful)

- El sistema consume directamente los datos a través de la API de comunicación, utilizada por los demás investigadores.
- No modifica el módulo de adquisición, sino que trabaja sobre sus salidas, lo que garantiza compatibilidad con despliegues futuros.

Integración con el módulo de almacenamiento (Base de Datos Times Series)

El sistema escribe sus resultados en las tablas estructurales ya existentes:

- iot.mediciones para datos originales con indicador de calidad (Fase 1)
- iot.mediciones_limpias para datos preprocesados (Fase 2)
- iot.caracteristicas para las caracteristicas generadas (Fase 3)

Esta estructura común permite que los módulos ajenos a esta investigación (diagnóstico, dashboard, análisis energético) puedan consumir los datos eficientemente.

Dado que el sistema desarrollado se encuentra integrado con los módulos institucionales del proyecto incluyendo componentes que manejan claves de autenticación, tokens de comunicación y estructuras internas de la base de datos su publicación en un repositorio público no es permitida, a fin de resguardar la integridad, seguridad y propiedad intelectual del proyecto. Por esta razón, el código fuente completo, junto con su documentación técnica, será entregado exclusivamente al equipo responsable de la infraestructura del proyecto institucional para su incorporación en el repositorio privado correspondiente. Si embargo este estudio garantiza la reproducibilidad del sistema mediante la documentación exhaustiva durante todas las fases de CRISP-DM.

El sistema queda entonces a disposición del equipo institucional, quien será el responsable de incorporarlo al entorno definitivo de integración y operación. De esta forma, la fase de despliegue se considera cumplida, ya que el producto desarrollado es funcional, interoperable, reproducible y se encuentra alineado con la arquitectura general del proyecto de mantenimiento predictivo.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- El análisis exploratorio reveló que los datos sensoriales presentaban problemas críticos de calidad, entre ellos presencia de outliers en variables vibracionales, desincronización temporal entre señales, gaps distribuidos a lo largo de la serie y variabilidad en mediciones clave, lo que impedía su uso directo en modelos predictivos. Para abordar estas inconsistencias se clasificaron las variables en continuas físicas de vibración, continuas físicas no relacionadas con vibración, categóricas diagnósticas y acumulativas, ya que cada grupo mostraba fallos de calidad distintos y requería métodos de limpieza e imputación específicos. Durante este análisis también se determinó que Acceleration RMS Radial constituía la variable principal dentro del comportamiento vibroacústico debido a su sensibilidad a anomalías y coherencia estructural frente al resto de señales.

- El sistema de preprocesamiento desarrollado permitió transformar datos sensoriales crudos, irregulares y heterogéneos en un flujo estructurado y confiable, adecuado para las necesidades del proyecto de mantenimiento predictivo. La integración de las tres fases, diagnóstico y etiquetado de calidad, reconstrucción temporal e imputación controlada, y generación de características estadísticas resultó en un pipeline que garantiza integridad, consistencia temporal y pertinencia analítica. Además, la arquitectura de almacenamiento distribuida en tablas independientes fortaleció la trazabilidad del proceso: en la Fase 1 se conservaron las mediciones originales en su estado crudo, incorporando únicamente el indicador de calidad para identificar tipo y origen de los errores; en la Fase 2, los datos ya sincronizados, filtrados e imputados se almacenaron como un dataset limpio separado; y en la Fase 3, las características derivadas se registraron en una tabla propia para su posterior consumo por los modelos de aprendizaje automático.

- El análisis del impacto de la calidad del dato sensorial evidenció que las irregularidades presentes en el dataset crudo tales como ruido, valores atípicos y variaciones temporales tienen efectos diferenciados sobre los modelos predictivos. La incorporación del indicador de calidad generado en la Fase 1 permitió identificar adecuadamente mediciones anómalas sin alterar los valores originales, y su uso como insumo en el entrenamiento reveló que modelos lineales como la Regresión Logística son altamente sensibles a dichos problemas, reflejando un rendimiento deficiente con una exactitud de apenas 34.92%, precisión de 3.03% y un F1-score de 5.87%, además de una matriz de confusión fuertemente sesgada. En contraste, el modelo Random Forest mostró un desempeño significativamente superior, alcanzando una exactitud de 99.57%, precisión de 83.96%, recall de 97.40% y un F1-score de 90.18%, demostrando capacidad para aprender patrones relevantes incluso sobre datos crudos y parcialmente ruidosos con un indicador de calidad confirmando empíricamente que la calidad del dato sensorial influye directamente en la capacidad de aprendizaje de los modelos de machine learning.

- La integración del sistema al proyecto de investigación "Desarrollo e Implementación de un Sistema de Mantenimiento Predictivo Basado en Inteligencia Artificial para Optimizar Equipos de Bombeo Hidráulico en la Producción de Crudo" se realizó de forma satisfactoria, ya que la arquitectura generada se acopló al ecosistema existente, proporcionando datasets limpios y características temporales que pueden ser consumidos directamente por los módulos desarrollados por otros investigadores del equipo. El pipeline aporta versiones controladas del dato (original, limpio y derivado), facilitando experimentación consistente, reproducibilidad y colaboración interdisciplinaria

4.2 Recomendaciones

- Se recomienda incorporar características avanzadas como métricas espectrales y medidas de entropía temporal para capturar patrones que las estadísticas de ventana no reflejan, especialmente en vibración.

- Se sugiere probar métodos de imputación que utilicen relaciones entre variables como KNN multivariante, MICE o modelos autoregresivos para mejorar coherencia en gaps o huecos mayores y reducir sesgo frente a la imputación univariante actual.

- Se recomienda realizar pruebas de rendimiento (benchmark) controlado entre diferentes modelos como árboles, redes recurrentes, TCNs o Transformers temporales para identificar la arquitectura que mejor aprovecha las series preprocesadas y las nuevas características.

REFERENCIAS BIBLIOGRÁFICAS

[1] "World Energy Outlook 2023 – Analysis - IEA." Accessed: Dec. 05, 2025. [Online]. Available: https://www.iea.org/reports/world-energy-outlook-2023#downloads

[2] "Mantenimiento predictivo: beneficios para la industria del petróleo y gas – Mobility Work." Accessed: Dec. 05, 2025. [Online]. Available: https://mobility-work.com/es/blog/mantenimiento-predictivo-petroleo-gas/

[3] "BCE - Banco Central del Ecuador." Accessed: Dec. 05, 2025. [Online]. Available: https://www.bce.fin.ec/

[4] "Ministerio de Ambiente y Energía – Ecuador." Accessed: Dec. 05, 2025. [Online]. Available: https://www.ambienteyenergia.gob.ec/

[5] "Gemelos digitales y mantenimiento predictivo en petróleo y gas - Foundtech." Accessed: Dec. 05, 2025. [Online]. Available: https://foundtech.me/mantenimiento-predictivo-industria-petrolera/

[6] "(3) Uso de Machine Learning para el Mantenimiento Predictivo: Transformando la Industria con Datos e IA | LinkedIn." Accessed: Dec. 05, 2025. [Online]. Available: https://www.linkedin.com/pulse/uso-de-machine-learning-para-el-mantenimiento-la-con-elies-fmqtf/

[7] S. Cofre-Martel, E. L. Droguett, and M. Modarres, "Big machinery data preprocessing methodology for data-driven models in prognostics and health management," *Sensors*, vol. 21, no. 20, 2021, doi: 10.3390/s21206841.

[8] J. H. Yoo, Y. K. Park, and S. S. Han, "Predictive Maintenance System for Wafer Transport Robot Using K-Means Algorithm and Neural Network Model," *Electronics (Switzerland)*, vol. 11, no. 9, 2022, doi: 10.3390/electronics11091324.

[9] S. A. Kutubuddin and D. V. A. Rama Sastry, "A Python-Based Approach for Predictive Maintenance Condition Monitoring of Lubricants," 2024. [Online]. Available: https://www.jisem-journal.com/

[10] M. Wang, X. Su, H. Song, Y. Wang, and X. Yang, “Enhancing predictive maintenance strategies for oil and gas equipment through ensemble learning modeling,” *J Pet Explor Prod Technol*, vol. 15, no. 3, Mar. 2025, doi: 10.1007/s13202-025-01931-x.

[11] S. F. Chevtchenko *et al.*, “Anomaly Detection in Industrial Machinery Using IoT Devices and Machine Learning: A Systematic Mapping,” *IEEE Access*, vol. 11, 2023, doi: 10.1109/ACCESS.2023.3333242.

[12] O. Serradilla, E. Zugasti, J. Rodriguez, and U. Zurutuza, “Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects,” *Applied Intelligence*, vol. 52, no. 10, 2022, doi: 10.1007/s10489-021-03004-y.

[13] S. Raschka, J. Patterson, and C. Nolet, “Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence,” 2020. doi: 10.3390/info11040193.

[14] R. Aliyu, A. A. Mokhtar, and H. Hussin, “Prognostic Health Management of Pumps Using Artificial Intelligence in the Oil and Gas Sector: A Review,” 2022. doi: 10.3390/app122211691.

[15] S. M. Almazrouei, F. Dweiri, R. Aydin, and A. Alnaqbi, “An Ensemble Machine Learning Model for Predictive Maintenance on Water Injection Pumps in the Oil and Gas Industry,” 2024. [Online]. Available: www.ijacsa.thesai.org

[16] A. Aminzadeh *et al.*, “A Machine Learning Implementation to Predictive Maintenance and Monitoring of Industrial Compressors,” *Sensors*, vol. 25, no. 4, Feb. 2025, doi: 10.3390/s25041006.

[17] T. Abbasi, K. H. Lim, and K. S. Yam, “Predictive Maintenance of Oil and Gas Equipment using Recurrent Neural Network,” in *IOP Conference Series: Materials Science and Engineering*, 2019. doi: 10.1088/1757-899X/495/1/012067.

[18] T. Langer, R. Meyes, and T. Meisen, “Guided Exploration of Industrial Sensor Data,” *Computer Graphics Forum*, vol. 43, no. 1, 2024, doi: 10.1111/cgf.15003.

[19] T. Langer and T. Meisen, “Visual Analytics for Industrial Sensor Data Analysis,” in *International Conference on Enterprise Information Systems, ICEIS - Proceedings*, 2021. doi: 10.5220/0010399705840593.

[20] K. Sahoo, A. K. Samal, J. Pramanik, and S. K. Pani, “Exploratory data analysis using python,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 12, 2019, doi: 10.35940/ijitee.L3591.1081219.

[21] M. Wang, X. Su, H. Song, Y. Wang, and X. Yang, “Enhancing predictive maintenance strategies for oil and gas equipment through ensemble learning modeling,” *J Pet Explor Prod Technol*, vol. 15, no. 3, Mar. 2025, doi: 10.1007/s13202-025-01931-x.

[22] “What is CRISP DM? - Data Science PM.” Accessed: Dec. 01, 2025. [Online]. Available: https://www.datascience-pm.com/crisp-dm-2/

[23] R. Wirth and J. Hipp, “CRISP-DM: Towards a Standard Process Model for Data Mining.”

[24] “Estadística descriptiva: Qué es, objetivo, tipos y ejemplos.” Accessed: Dec. 06, 2025. [Online]. Available: https://www.questionpro.com/blog/es/estadistica-descriptiva/

[25] “¿Qué son los Outliers? | MIOTI.” Accessed: Dec. 05, 2025. [Online]. Available: https://mioti.es/es/que-son-los-outliers/

[26] Ekta Maini, “Rango intercuartil para detectar valores atípicos en los datos - GeeksforGeeks.” Accessed: Dec. 01, 2025. [Online]. Available: https://www.geeksforgeeks.org/machine-learning/interquartile-range-to-detect-outliers-in-data/

[27] Suraj Joshi, “¿Qué es la interpolación de datos? - GeeksforGeeks.” Accessed: Dec. 01, 2025. [Online]. Available: https://www-geeksforgeeks-org.translate.goog/data-analysis/what-is-data-interpolation/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=wa

[28] Malli, “Método ffill() de Pandas DataFrame - Spark por {Examples}.” Accessed: Dec. 01, 2025. [Online]. Available: https://sparkbyexamples.com/pandas/pandas-dataframe-ffill-method/

[29] “¿Qué es la regresión logística? | IBM.” Accessed: Dec. 01, 2025. [Online]. Available: https://www.ibm.com/es-es/think/topics/logistic-regression

[30] kartik, “Logistic Regression in Machine Learning - GeeksforGeeks.” Accessed: Dec. 01, 2025. [Online]. Available: https://www.geeksforgeeks.org/machine-learning/understanding-logistic-regression/

[31] “¿Qué es Random Forest? | IBM.” Accessed: Dec. 01, 2025. [Online]. Available: https://www.ibm.com/mx-es/think/topics/random-forest