Escuela Superior Politécnica del Litoral

Facultad de Ingeniería en Mecánica y Ciencias de la Producción

Asistente virtual basado en Inteligencia Artificial para la obtención de indicadores de desempeño

INGE-2850

Proyecto Integrador

Previo a la obtención del Título de:

Ingeniero en Mecatrónica

Presentado por:

Grace Angela Diaz Perez

Josue Raphael Moreno Charco

Guayaquil - Ecuador

Año: 2025

Dedicatoria

Dedico este trabajo a Dios, por darme vida, sabiduría y la fuerza necesaria para superar cada obstáculo.

A mis padres, ejemplo de perseverancia y fortaleza, quienes me enseñaron con amor el valor del esfuerzo y la humildad.

A mis hermanos, por ser mis compañeros de vida, mis amigos y mis mayores alentadores en este camino.

Grace Diaz

Dedicatoria

Dedico este trabajo a Dios, por regalarme la vida, la sabiduría y la fortaleza para avanzar con esperanza en cada desafío.

A mis padres, cuyo amor, ejemplo y sacrificio han sido la base de todo lo que soy y la inspiración constante para nunca rendirme.

A mis hermanas, compañeras de vida y cómplices de sueños, por su apoyo incondicional y la alegría que siempre me brindan.

A mis abuelos, pilares de cariño y sabiduría que iluminan mi camino. En especial a mi abuela Teresa, por su apoyo incansable y su fe inquebrantable en mí, que han sido un motor invaluable en este proceso.

Josue Moreno

Agradecimientos

Mi más sincero agradecimiento a la Universidad por brindarme la oportunidad de formarme profesionalmente.

A mi tutor de tesis, por su orientación, paciencia y valiosos aportes durante todo el proceso.

Finalmente, a todas aquellas personas que, de manera directa o indirecta, hicieron posible la realización de este trabajo

Grace Diaz

Agradecimientos

Expreso mi más sincero agradecimiento a la Universidad por brindarme la oportunidad de formarme profesionalmente y por ser el espacio donde he podido crecer académica y personalmente.

A mi tutor de tesis, por su paciencia, dedicación y valiosos aportes que orientaron cada etapa de este trabajo.

A mis compañeros, por su apoyo, colaboración y por compartir conmigo este camino de aprendizajes y desafíos, que hicieron más enriquecedor este proceso.

Y especial agradecimiento, a todas las personas que, de manera directa o indirecta, contribuyeron a la culminación de este proyecto.

Josue Moreno

Declaración Expresa

Nosotros Grace Angela Diaz Perez y Josue Raphael Moreno Charco acordamos y reconocemos que:

La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores.

La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por nosotros durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que nos corresponda de los beneficios económicos que la ESPOL reciba por la explotación de nuestra innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique a los autores que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Guayaquil, 28 de mayo del 2025.

Grace Angela Diaz Perez

Josue Raphael Moreno Charco

Evaluadores

Firmado electrónicamente por:
BRYAN JOAO
PURUNCAJAS MAZA
Validar únicamente con FirmaEC

Bryan Puruncajas, Ph.D.

Profesor de Materia

Firmado electrónicamente por:
CHRISTOPHER JAVIER
VACCARO CEDILLO
Validar únicamente con FirmaEC

Christopher Vaccaro, M.Sc.

Tutor de proyecto

Resumen

En la industria ecuatoriana, el acceso a indicadores clave de desempeño (KPIs) es limitado debido al uso de registros manuales y plataformas aisladas que generan demoras en la toma de decisiones y aumentan los costos operativos. El presente proyecto plantea el diseño de una plataforma web que integra un asistente virtual basado en inteligencia artificial para consultar métricas de confiabilidad y disponibilidad, así como predecir fallos en equipos críticos. El objetivo es reducir los tiempos de consulta y fortalecer la toma de decisiones estratégicas mediante información confiable y en tiempo real.

El sistema fue desarrollado combinando una base de datos centralizada, un entorno web intuitivo y un motor de procesamiento de lenguaje natural que interpreta preguntas de los usuarios. Asimismo, se incorporó un modelo de aprendizaje automático Random Forest entrenado con datos históricos de operación, el cual se integró al asistente para generar predicciones oportunas.

Los resultados demostraron una reducción del tiempo promedio de consulta de cerca de una hora a once minutos y un desempeño predictivo confiable con métricas F1=0.9262 y ROC AUC=0.9567.

Se concluye que la solución representa una alternativa viable para la digitalización de procesos industriales, al mejorar la eficiencia, la trazabilidad y la productividad en planta.

Palabras clave: Industria 4.0, digitalización, machine learning, productividad

Abstract

In the Ecuadorian industry, access to key performance indicators (KPIs) is often limited due to the use of manual records and isolated platforms, which cause delays in decision-making and increase operating costs. This project proposes the design of a web-based platform with a virtual assistant powered by artificial intelligence, capable of responding to queries about reliability and availability metrics while also predicting failures in critical equipment. The objective is to reduce consultation times and provide reliable real-time information to strengthen the plant's strategic management.

The system was developed by combining a centralized database, an intuitive web environment, and a natural language processing engine that interprets user questions. In addition, a Random Forest machine learning model was trained with historical operating data and integrated into the assistant to generate timely predictions.

The results showed a reduction in the average consultation time from nearly one hour to eleven minutes, together with a reliable predictive performance, achieving F1=0.9262 and ROC AUC=0.9567.

It is concluded that the solution is a viable alternative for the digitalization of industrial processes, enhancing efficiency, traceability, and productivity in manufacturing environments.

Keywords: Industry 4.0, digitalization, machine learning, productivity

Índice general

Abreviaturas

EMCali	Empresas Municipales de Cali
ERP	Planificación de Recursos Empresariales
GenAI	Inteligencia Artificial Generativa
IA	Inteligencia Artificial
IIoT	Internet Industrial de las Cosas
IoT	Internet de las Cosas
IT	Tecnología de la Información
KPIs	Indicadores Clave de Desempeño
LLM	Modelos de Lenguaje de Gran Escala (Large Language Models)
ML	Aprendizaje Automático
MTBF	Tiempo promedio entre fallas
MTTR	Tiempo promedio de reparación
OEE	Eficiencia General de los Equipos
OKRs	Objetivos y Resultados Clave
OT	Tecnología Operacional
PLN	Procesamiento de Lenguaje Natural
SCADA	Supervisión, Control y Adquisición de Datos
SMART	Específico, Medible, Alcanzable, Relevante y con Tiempo
VVA	Asistente Virtual de Voz

Simbología

min	Minutos
%	Porcentaje
bar	Bar (unidad de presión)

Índice de figuras

Índice de tablas

Capítulo 1

1. Introducción

A lo largo de la historia, la revolución industrial ha transformado la forma en que las sociedades producen, consumen y se relacionan con la tecnología. En este marco, la industria 4.0 marca una nueva etapa, caracterizada por la digitalización e interconexión de sistemas mediante tecnologías como el Internet de las Cosas (IoT), la Inteligencia Artificial (IA), el análisis de datos masivos y la computación en la nube. Esta integración redefine la productividad y mejora la calidad, eficiencia, sostenibilidad y toma de decisiones estratégicas [1].

A partir de esta evolución digital, las organizaciones industriales enfrentan la necesidad de adoptar herramientas innovadoras para mantenerse competitivas en un entorno global cada vez más dinámico. Conceptos como la integración horizontal y vertical garantizan el flujo continuo de información entre niveles y áreas funcionales, generando fábricas inteligentes donde los dispositivos IoT permiten supervisión continua, mantenimiento predictivo y mejora del rendimiento [2].

Como consecuencia de estas capacidades tecnológicas, uno de los pilares de esta transformación es el enfoque basado en datos [3]. Las decisiones empresariales hoy se sustentan en análisis cuantitativos y visualizaciones que facilitan la comprensión y comunicación de información compleja [4]. Herramientas como dashboards interactivos o plataformas como Tableau permiten identificar patrones y oportunidades de mejora en múltiples áreas, desde operaciones hasta marketing [5]. En este contexto, los Indicadores Claves de Desempeño (KPIs) cumplen un rol central al proporcionar métricas claras y medibles para evaluar procesos industriales. Su monitoreo continuo permite establecer metas, detectar desviaciones y tomar decisiones basadas en evidencia, fortaleciendo la mejora continua y la transparencia [6].

Por otro lado, factores humanos como la resistencia al cambio, la falta de capacitación y una débil cultura organizacional limitan la adopción plena de la industria 4.0. Superar estas

barreras requiere no solo tecnología, sino también rediseño de procesos, formación del personal y una mentalidad orientada a la innovación [7].

Bajo este panorama, la automatización de la medición de KPIs industriales representa una herramienta clave para mejorar el desempeño global. Mediante la recolección y procesamiento inteligente de datos, y su visualización accesible, las organizaciones pueden anticipar desviaciones, optimizar recursos y alinear sus decisiones con objetivos estratégicos. Por ello, este trabajo propone el diseño de una solución tecnológica orientada a mejorar la visualización de KPIs en un entorno industrial real de la industria ecuatoriana. De esta forma, se identifican oportunidades concretas de mejora en los procesos actuales de monitoreo y análisis de indicadores, con el objetivo de reducir tiempos de consulta, automatizar la gestión de datos y fortalecer la toma de decisiones basada en principios de la industria 4.0.

1.1 Descripción del problema

En la planta en estudio, la consulta y seguimiento de KPIs, como Tiempo Medio de Reparación (MTTR) y Tiempo Medio entre Fallas (MTBF), se realiza mediante registros físicos, hojas de cálculo y plataformas aisladas que no están diseñadas para un análisis eficiente. Estas herramientas limitan la filtración, comparación y visualización de información de forma clara, especialmente al analizar datos históricos, promedios mensuales o comparaciones entre diferentes periodos.

Esta falta de integración retrasa la toma de decisiones, incrementa el margen de error humano y reduce la capacidad de reacción ante desviaciones operativas. La dependencia de formatos dispersos también complica el trabajo colaborativo, impide una trazabilidad efectiva y genera cuellos de botella al momento de auditar o evaluar el rendimiento de la planta.

Además, no existe un sistema que les permita acceder a la información de forma remota, ni que se adapte fácilmente a cambios o incorporación de nuevos KPIs conforme evolucionan los

objetivos de la empresa. Esta rigidez tecnológica afecta la capacidad de escalar mejoras y limita el potencial de digitalización de procesos claves dentro del marco de la industria 4.0.

En pocas palabras, el problema radica en la carencia de una herramienta automatizada, centralizada y flexible que permita visualizar los KPIs en tiempo real, desde cualquier lugar dentro de la empresa, y que facilite el análisis operativo con base en datos confiables, actualizados y organizados.

1.2 Justificación del problema

El presente proyecto busca resolver la limitada disponibilidad de información operativa confiable, centralizada y de rápida consulta en planta, un factor crítico que impacta directamente en la capacidad de la organización para ejecutar una gestión productiva eficiente y basada en datos.

Resolver este problema es importante porque afecta no solo a la operatividad diaria, sino también al desempeño estratégico de la planta. Más de 50 colaboradores, entre supervisores y técnicos, enfrentan dificultades frecuentes para acceder a métricas clave. En la plataforma actual, obtener una sola métrica puede tomar más de una hora, y consultar datos históricos o hacer comparaciones entre períodos resulta aún más complejo y lento, lo que representa no solo un retraso operativo, sino también una pérdida económica acumulada al año.

Por otro lado, esta falta de agilidad impide detectar desviaciones a tiempo, genera cuellos de botella en la toma de decisiones y contribuye a un aumento de mantenimientos correctivos evitables. Estas reacciones tardías ante los eventos en línea impactan directamente en los costos y disponibilidad de los equipos.

Además, la ausencia de una herramienta moderna limita la generación de propuestas de mejora continua, obstaculiza la innovación tecnológica y debilita la cultura de excelencia operativa. Incluso ciertas métricas clave no se encuentran disponibles en la plataforma actual, lo que fragmenta la visión integral del proceso y deteriora la trazabilidad y control de calidad.

Solucionar este problema permitirá transformar la gestión de la información en planta mediante un sistema que automatice la recolección, visualización y análisis de indicadores clave de desempeño. Esto no solo mejorará la eficiencia operativa y reducirá los tiempos de respuesta, sino que también alineará a la organización con las buenas prácticas de empresas líderes como Nestlé y La Fabril, donde la transformación digital ha demostrado generar impactos positivos medibles en productividad, calidad y sostenibilidad [8], [9], [10].

De manera general, la implementación de una solución tecnológica adecuada representa una oportunidad tangible para reducir tiempos entre consultas, permitiendo al personal tener a la mano información oportuna y sentar las bases para una industria más inteligente, proactiva y competitiva.

1.3 Objetivos

1.3.1 Objetivo general

Diseñar un sistema inteligente a través de una plataforma web que integre un asistente virtual con inteligencia artificial, capaz de consultar indicadores de desempeño y predecir si una máquina fallará mediante la interacción en lenguaje natural, con el propósito de reducir el tiempo de consulta de KPIs estratégicos en la industria ecuatoriana de estudio.

1.3.2 Objetivos específicos

1. Elaborar una plataforma web multiplataforma con integración a una base de datos PostgreSQL, que permita la interacción de los usuarios a través de una interfaz intuitiva y accesible.
2. Integrar un asistente virtual en la plataforma web, basado en inteligencia artificial y procesamiento de lenguaje natural, capaz de responder consultas sobre métricas de confiabilidad y disponibilidad actuales e históricas.

3. Desarrollar modelo de aprendizaje automático entrenado con datos históricos que permita predecir el fallo de una máquina en una determinada condición de operación, facilitando la toma de decisiones operativas.

1.4 Marco teórico

Estudios más recientes han evidenciado cómo la automatización de KPIs han transformado los procesos operativos en la industria [11]. En la industria ecuatoriana, los KPIs son fundamentales para monitorear la eficiencia, la calidad y la sostenibilidad del proceso productivo. Tradicionalmente, estos indicadores se gestionaban de forma manual, dificultando el acceso a datos en tiempo real y limitando la toma de decisiones informadas. Grandes empresas dedicadas a la producción en Ecuador han implementado tecnologías como sistemas de monitoreo en tiempo real, gestión automatizada de inventarios y plataformas digitales mejorando significativamente la eficiencia y la toma de decisiones. La digitalización de KPIs representa un cambio hacia sistemas interconectados que permiten recopilar, visualizar y analizar datos de forma automática y en tiempo real. Este proceso integra tecnologías como sensores IoT y plataformas de visualización como Power BI o Qlik Sense, permitiendo un control más preciso de variables críticas como rendimiento de línea, consumo energético, desperdicios o cumplimiento de producción. Estas soluciones son coherentes con herramientas de planificación estratégica como el Balanced Scorecard, promoviendo una dirección más ágil, rigurosa y adaptable en contextos de alta competitividad [12] .

1.4.1 KPIs claves en la industria ecuatoriana

Los KPIs son métricas cuantificables que permiten evaluar el nivel de cumplimiento de objetivos estratégicos dentro de una organización. Su función es actuar como una brújula para la toma de decisiones basada en datos, alineando las operaciones diarias con la visión y objetivos globales de la empresa [13], [14]. A diferencia de los OKRs (Objectives and Key Results), que

articulan metas cualitativas y cuantitativas de forma más amplia, los KPIs se centran exclusivamente en medir el rendimiento asociado a metas específicas. Esta distinción es fundamental para no confundir los OKRs con los KPIs [15]. Para que un KPI sea efectivo, debe cumplir con el enfoque SMART, es decir debe ser especifico, medible, alcanzable, relevante y con plazo definido. Por ejemplo, “aumentar ventas” es una meta ambigua, mientras que “incrementar un 10% las ventas en el mercado europeo en el Q4” constituye un KPI SMART [16].

En el contexto de la planta industrial en estudio, dos KPIs fundamentales son el MTTR y el MTBF. El MTBF mide el tiempo promedio que un sistema o equipo opera sin fallas, reflejando directamente su confiabilidad [17]. Un valor alto de MTBF indica una menor frecuencia de fallos, lo que contribuye a una mayor estabilidad operativa, reducción de tiempos de inactividad no programada y mayor aprovechamiento de la capacidad instalada. En industria, donde la continuidad del proceso productivo es clave para evitar pérdidas por interrupciones, el MTBF se convierte en un indicador crítico para planificar mantenimientos preventivos efectivos [18].

Por su parte, el MTTR evalúa el tiempo promedio necesario para restablecer el funcionamiento de un equipo tras una falla. Este KPI está estrechamente relacionado con la eficiencia del mantenimiento correctivo, la disponibilidad de repuestos, la preparación del personal técnico y la gestión de recursos logísticos. Un MTTR bajo es deseable, ya que minimiza la duración de las paradas no planificadas, lo cual es esencial para cumplir con los programas de producción y mantener la competitividad del proceso [19].

La combinación estratégica de ambos indicadores permite a la organización tomar decisiones informadas sobre inversiones en mantenimiento, rediseño de procesos y asignación de recursos. En efecto, empresas líderes en el sector como AB InBev demuestran que la optimización del MTBF y la reducción del MTTR son elementos clave para garantizar altos niveles de disponibilidad operativa y sostenibilidad industrial en entornos de alta demanda [18], [20]

1.4.2 IIoT y sensores inteligentes

El Internet Industrial de las Cosas (IIoT) y los sensores son fundamentales para automatizar y optimizar la recopilación y el procesamiento de datos para los KPIs de manufactura. Resulta poco práctico intentar realizar un seguimiento manual de la enorme cantidad de datos que se generan en un entorno industrial. Las plataformas IIoT utilizan sensores y dispositivos inteligentes para recopilar datos críticos de maquinaria, líneas de producción e incluso de la fuerza laboral de forma automática. Una de las aplicaciones más valiosas del IIoT es la facilitación del mantenimiento predictivo, ya que recopila datos en tiempo real sobre el estado de los equipos, alertando sobre posibles fallas antes de que ocurran [21].

1.4.3 Inteligencia artificial y machine learning

La Inteligencia Artificial (IA) y el Machine Learning (ML) están jugando un papel cada vez más relevante en la digitalización y evolución de los KPIs. Estas tecnologías no solo permiten analizar grandes volúmenes de datos de forma rápida y precisa, sino que también ofrecen capacidades predictivas y prescriptivas que mejoran significativamente la toma de decisiones estratégicas. En lugar de limitarse a medir lo que ya ocurrió, como hacen los KPIs tradicionales, la IA y el ML permiten anticipar comportamientos, prever fallos, detectar patrones y recomendar acciones óptimas antes de que surjan problemas [22].

En la industria ecuatoriana, por ejemplo, estas herramientas tecnológicas mejoran la precisión con cada dato registrado, acercándose al concepto del “lote dorado”: máxima calidad con mínimo uso de recursos. Esto permite a los maestros manufactureros enfocarse en aspectos creativos de su trabajo, mientras los sistemas inteligentes se encargan del análisis de datos [23].

Además, la IA se incorpora en plataformas de análisis para asistir a los usuarios en tiempo real. Un caso concreto es el de Coca-Cola E&ME, que ha implementado un KPI enfocado en medir el aumento de ventas atribuido al uso de IA generativa (GenAI) [24]. Esta integración tecnológica

representa una evolución fundamental en la forma en que las empresas gestionan sus operaciones: ya no reaccionan a los problemas una vez ocurridos, sino que los anticipan y actúan antes de que afecten la producción o el negocio.

1.4.4 Plataformas de análisis y visualización de datos

Las plataformas para el análisis y visualización de datos, conocidas como dashboards, son programas que automatizan el procesamiento de la información recolectada, convirtiéndola en análisis e insights útiles sin necesidad de intervención manual constante. Estas herramientas permiten reunir datos de distintas fuentes y crear reportes de desempeño, además de emitir alertas cuando algún KPI no cumple con las expectativas. También brindan opciones para personalizar las visualizaciones, adaptándose a las necesidades específicas de cada equipo o área dentro de la empresa. Algunas de las soluciones más populares en este ámbito son Zoho Analytics, Tableau, Rapid Miner, Knime, Qlik Sense, Apache Spark, SAS, Talend, Databox, Klipfolio, Grafana, Domo y Microsoft Power BI [25].

En la industria manufacturera, la verdadera digitalización de los KPIs se logra al integrar las tecnologías operativas (OT), como los sistemas de supervisión, control y adquisición de datos (SCADA), con las tecnologías de la información (IT), como los sistemas de gestión de datos [26]. Esta integración no es solo la adopción de una tecnología nueva, sino la unión de todo el ecosistema de datos de la fábrica. Así, las empresas pueden tener una visión integral del rendimiento, desde el nivel de máquina hasta la estrategia corporativa, algo que antes no era posible con sistemas desconectados. Por lo tanto, las organizaciones deben apostar por una arquitectura de datos integrada que potencie sus KPIs en lugar de limitarse a adquirir software aislado [27].

1.4.5 Estado del arte

Como se mencionó previamente, la evolución tecnológica ha permitido que muchas industrias pasen de métodos manuales a sistemas automatizados para gestionar y monitorear sus KPIs. En el caso de Ecuador, no se encuentran registros documentados de iniciativas que integren asistentes virtuales basados en IA como consultores de métricas en plantas industriales, lo que evidencia una brecha tecnológica. No obstante, existen referencias internacionales de empresas que se han enfocado en desarrollo de soluciones similares con resultados positivos, las cuales se detallan a continuación para establecer un marco comparativo relevante.

GEA InsightPartner Brewery

GEA InsightPartner Brewery es un sistema avanzado desarrollado por la empresa alemana GEA, diseñado para la monitorización inteligente en tiempo real de procesos en la industria. Este sistema integra múltiples fuentes de datos en una única plataforma tipo dashboard, incluyendo sensores, variadores de frecuencia, mediciones de motores y otros parámetros críticos. Su objetivo es centralizar la visualización de métricas clave, permitiendo tanto la consulta de datos como el ingreso directo de indicadores por parte del usuario [28].

Figura 1

Dashboard plataforma GEA para línea de producción [29]

Una de las características más destacadas de InsightPartner es su reciente incorporación de modelos de inteligencia artificial que permiten la predicción de KPIs y la detección de fallos en máquinas críticas seleccionadas, a través de análisis de tendencias y modelos de previsión (véase Figura 1). Este enfoque predictivo se materializa en tarjetas inteligentes que alertan al operador sobre escenarios futuros que podrían afectar la calidad o sostenibilidad del lote, fomentando así una toma de decisiones anticipada e informada, muy por encima del sistema tradicional de alarmas. Este enfoque ha sido bien recibido en la industria debido a su impacto directo en la eficiencia operativa, el mantenimiento predictivo y el control de calidad [30].

AZLOGICA® y asistentes virtuales industriales

Por su parte, la empresa AZLOGICA®, empresa pionera en soluciones de Internet de las Cosas (IoT) e inteligencia artificial desde 2008, ha desarrollado múltiples implementaciones de asistentes virtuales de voz (VVA, por sus siglas en inglés), integrando plataformas como Amazon Alexa para facilitar la interacción entre usuarios, sistemas empresariales y activos digitales [31]. Uno de sus casos de éxito se dio en colaboración con la Cooperativa Coomeva en Colombia, donde se implementó Alexa como asistente de voz para atender requerimientos de usuarios durante y después de la pandemia de COVID-19. Los afiliados podían, mediante comandos de voz, consultar eventos institucionales, descargar certificados y ejecutar pagos de servicios, todo sin contacto físico, promoviendo así el concepto de "oficina sin contacto" [32].

Otro ejemplo destacado es el de EMCali (Empresas Municipales de Cali), donde AZLOGICA implementó una central inteligente para el monitoreo de una flota de más de 650 unidades móviles. Mediante comandos de voz dirigidos a Alexa, los directores podían consultar información operativa en tiempo real, como el estado de recursos, consumo de combustible por vehículo y reportes de riesgo en ruta. Esta implementación permitió optimizar operaciones, reducir la carga operativa del personal y mejorar la capacidad de respuesta ante eventos críticos. Además,

facilitó la interacción con grandes volúmenes de datos sin necesidad de interfaces gráficas complejas, mejorando la accesibilidad del sistema [32].

A partir de estos casos se identifican elementos clave que pueden ser adaptados a nuevas soluciones. En el caso de GEA InsightPartner, uno de los retos encontrados es que la visualización en dashboards puede volverse excesivamente compleja conforme crecen los volúmenes de datos, lo cual dificulta la rápida interpretación por parte de los operadores. En el caso de AZLOGICA, aunque los asistentes de voz han demostrado ser eficaces en distintos contextos, su implementación en ambientes industriales presenta desafíos adicionales, como el ruido en líneas de producción, que puede afectar la precisión del reconocimiento de voz.

No obstante, hay lecciones valiosas. Por ejemplo, AZLOGICA no desarrolló su asistente desde cero, sino que adaptó una plataforma ya existente (Alexa), lo que redujo tiempos y costos de desarrollo. Además, sus sistemas no almacenan datos operativos de forma local, lo cual abre una oportunidad para diseñar una solución que sí incorpore almacenamiento y análisis histórico de métricas, aportando valor adicional a los usuarios industriales.

Estos elementos serán considerados en la siguiente sección, donde se planteará el diseño de una solución adaptada a las condiciones de una planta industrial ecuatoriana, integrando lo mejor de ambos enfoques: la inteligencia predictiva de GEA y la interacción natural de AZLOGICA.

Capítulo 2

2. Metodología

En esta sección se describe el proceso seguido para diseñar la arquitectura del asistente virtual, considerando los requerimientos del cliente y explorando distintas alternativas técnicas. El enfoque se basó en etapas secuenciales que permitieron estructurar el desarrollo, desde la definición de los requerimientos hasta la validación del sistema, priorizando la seguridad, la eficiencia operativa y la autonomía tecnológica.

2.1 Análisis de requerimientos

Dado que los requerimientos fueron previamente definidos por el cliente, se procedió a realizar una revisión detallada de los elementos claves que contribuyen a una experiencia inmersiva entre el asistente virtual y el usuario. A partir de esta revisión, se identificaron los aspectos más relevantes, los cuales se presentan en la siguiente Tabla 1.

Tabla 1

Requerimientos del cliente

Producto	
Asistente virtual basado en Inteligencia Artificial para la consulta de indicadores claves de desempeño	
Requerimientos	
Concepto	Descripción
Procesamiento de lenguaje natural (PLN)	El sistema debe comprender y procesar consultas del usuario escritas en lenguaje natural, con preguntas como: "¿Cuál fue la productividad ayer?".
Conexión a base de datos PostgreSQL	El sistema debe establecer conexión con una base de datos PostgreSQL que contenga los datos necesarios.

Consulta de KPIs	El sistema debe permitir consultar valores mensuales de indicadores claves de confiabilidad y disponibilidad
Predicción del fallo de una máquina	El sistema debe integrar modelos de aprendizaje automático para predecir posibles fallos en una máquina, utilizando datos actuales e históricos de operación.
Interfaz web accesible	El sistema debe estar disponible a través de una interfaz web funcional en navegadores, accesible desde dispositivos conectados a la red empresarial.
Usabilidad	La interfaz debe ser minimalista, clara e intuitiva, permitiendo que usuarios no técnicos realicen consultas sin necesidad de capacitación previa.
Portabilidad	El sistema debe ser accesible desde distintos navegadores (Chrome, Firefox, Edge, Safari) sin instalaciones adicionales.
Escalabilidad funcional	El sistema debe permitir en el futuro la incorporación de nuevos indicadores o visualizaciones gráficas sin alterar significativamente la arquitectura.

Los requerimientos definidos corresponden a los aspectos fundamentales que debe cumplir el asistente virtual implementado en la página web. Cada uno de ellos representa una funcionalidad o característica clave, cuya importancia será considerada al momento de seleccionar tecnologías y soluciones. En su mayoría, estos requerimientos están enfocados en garantizar una interacción fluida, comprensible y efectiva entre el usuario y el sistema, priorizando la experiencia de uso dentro del entorno web.

2.2 Soluciones

2.2.1 Soluciones alternativas

Considerando los requerimientos establecidos por el cliente y el análisis exhaustivo realizado en el capítulo anterior, se definieron diversas soluciones potenciales para la implementación de un asistente virtual orientado a la consulta de KPIs y a la predicción de fallos en maquinaria, basándose en condiciones operativas específicas. Estas alternativas fueron evaluadas cuidadosamente para determinar su viabilidad técnica, escalabilidad y facilidad de integración dentro del entorno empresarial. Entre las diferentes alternativas se tiene:

- **Alternativa 1: Solución web local y modelo predictivo en archivo serializado**
  Esta opción propone desplegar el asistente virtual directamente en la infraestructura local de la empresa, integrando un modelo PLN de Meta, ejecutado localmente mediante Ollama y un modelo de predicción previamente entrenado, almacenado en un archivo serializado. Todo el sistema opera desde una interfaz web accesible en la red interna, sin depender de servicios externos. Esta alternativa prioriza la seguridad, el control total de la información y una baja latencia en las consultas.
- **Alternativa 2: Solución web basada en la nube**
  En esta alternativa, el asistente virtual se implementa en una página web alojada en la nube, utilizando servicios externos como la API de OpenAI para el procesamiento del lenguaje natural y modelos predictivos desplegados en servidores remotos. Esta opción ofrece alta escalabilidad, disponibilidad desde cualquier ubicación con conexión a internet, y menor carga operativa para el equipo de TI local, facilitando su mantenimiento y actualización.
- **Alternativa 3: Desarrollo web con modelo de PLN propio desde cero**
  Esta opción consiste en desarrollar una solución web completamente personalizada,

incluyendo la construcción de un modelo de Procesamiento de Lenguaje Natural propio, sin depender de servicios o modelos pre-entrenados externos como OpenAI. Esto implica entrenar el modelo con datos específicos del dominio, ajustarlo a las necesidades del negocio y diseñar una arquitectura conversacional interna.

2.2.2 Criterios de selección

Para la selección adecuada de la solución más conveniente para el desarrollo del asistente virtual, se definieron una serie de criterios que consideran tanto aspectos técnicos como operativos y de experiencia del usuario. Estos criterios permiten evaluar las alternativas de manera objetiva y estructurada, alineándose con los requerimientos y recursos disponibles. Los criterios identificados son:

- **Viabilidad técnica:** Facilidad para implementar la solución de manera efectiva en el entorno web.
- **Escalabilidad:** Capacidad para adaptarse y crecer en función del volumen de datos y nuevas funcionalidades.
- **Tiempo de desarrollo:** Duración estimada para diseñar, desarrollar y desplegar la solución completa.
- **Costo de implementación y operación:** Evaluación de los costos iniciales y recurrentes, incluyendo infraestructura, licencias y mantenimiento.
- **Recursos computacionales:** Nivel de hardware y software necesario para ejecutar la solución (servidores, GPUs, etc.).
- **Fluidez de interacción:** Calidad y naturalidad de la experiencia conversacional, incluyendo velocidad y precisión de respuestas.

- **Accesibilidad y edición del código fuente:** Facilidad para acceder, modificar y mantener el código fuente de forma autónoma.
- **Protección de datos:** Capacidad para garantizar la seguridad, privacidad y control local de la información sensible manejada por el sistema.

Para ponderar estos criterios se utilizó una escala del 1 al 6, donde 6 representa el criterio con mayor peso o importancia, y 1 el de menor. Esta escala permite asignar una relevancia cuantificable a cada aspecto, facilitando un análisis comparativo entre las alternativas propuestas.

Además, para facilitar la asignación de estos valores, los criterios se agruparon en tres niveles de relevancia:

- **Alta relevancia:** criterios con mayor impacto en la decisión (valor 3).
- **Media relevancia:** criterios importantes pero secundarios (valor 2).
- **Baja relevancia:** criterios complementarios o de menor impacto (valor 1).

Esta metodología busca garantizar una selección equilibrada y alineada con los objetivos del proyecto y las capacidades disponibles. La siguiente Tabla 2 muestra la asignación de relevancia y la ponderación numérica asignada a cada criterio:

Tabla 2

Criterios de selección

Criterio	Relevancia	Ponderación
Viabilidad técnica	3	6
Escalabilidad	2	3
Tiempo de desarrollo e implementación	2	3
Costo de implementación y operación	3	6
Recursos computacionales	2	4
Fluidez de interacción	1	2
Accesibilidad y edición del código fuente	1	2
Protección de datos	1	2

2.2.3 Matriz de decisión

Los criterios establecidos proporcionaron una base sólida para enfocar la evaluación en los requerimientos reales del cliente, considerando al mismo tiempo las limitaciones técnicas y las particularidades del entorno de implementación. Esta estructura permitió analizar de forma comparativa las distintas alternativas y orientar la decisión hacia la opción más conveniente, asignando niveles de relevancia en una escala del 1 al 3, donde el valor 3 representa la mayor prioridad.

Tabla 3

Matriz de decisión

Criterio	Ponderación	Alternativa 1	Alternativa 2	Alternativa 3
Viabilidad técnica	6	2	3	1
Escalabilidad	3	2	3	2
Tiempo de desarrollo e implementación	3	3	3	1
Costo de implementación y operación	6	3	2	1
Recursos computacionales	4	3	1	3
Fluidez de interacción	2	2	3	2
Accesibilidad y edición del código fuente	2	3	1	3
Protección de datos	2	3	1	3
Puntaje sin ponderación		21	17	15
Puntaje con ponderación		79	62	49

Con base en la matriz de decisión elaborada (Tabla 3), la alternativa 1 obtuvo el mayor puntaje total (79 puntos), posicionándose como la opción más equilibrada según los criterios

definidos. Esta alternativa destaca por su alta protección de datos, bajo costo de implementación y accesibilidad al código fuente, ofreciendo un control robusto sobre la información y autonomía tecnológica, lo cual es clave para el entorno empresarial.

La alternativa 2 alcanzó un puntaje de 62 puntos, situándose como una opción viable gracias a su alta escalabilidad y viabilidad técnica. Sin embargo, presenta limitaciones en aspectos críticos como la protección de datos y el acceso al código fuente, dado que depende de servicios externos, lo que puede afectar la confidencialidad y la capacidad de personalización.

Finalmente, la alternativa 3 obtuvo el puntaje más bajo, con 49 puntos. Aunque ofrece mayor control y flexibilidad en el código, representa una desventaja significativa por su alta demanda de recursos computacionales, además de requerir mayor tiempo y costo de desarrollo, lo que puede impactar negativamente la viabilidad del proyecto en términos de infraestructura y presupuesto.

2.3 Consideraciones en base a normativas

En un entorno industrial, la incorporación de normativas es una condición fundamental para garantizar que el sistema cumpla con los niveles de calidad, seguridad operativa y compatibilidad exigidos por el sector. Dado que esta solución se basa en el despliegue local de una plataforma web que integra procesamiento de lenguaje natural mediante un modelo PLN en Ollama y un modelo predictivo serializado, fue imprescindible alinear su diseño con los estándares técnicos aplicables a nivel industrial.

La Tabla 4 recoge las normativas clave consideradas durante el desarrollo. Estas abarcan desde lineamientos para la gestión segura de la información en sistemas locales, hasta principios para asegurar la interoperabilidad, el mantenimiento a largo plazo y la fiabilidad en condiciones de operación continua. De este modo, el sistema no solo responde a los requerimientos funcionales,

sino que también se prepara para una posible integración en procesos industriales donde la robustez y el cumplimiento normativo son determinantes.

Tabla 4

Normativas aplicadas con sus respectivas consideraciones

Normativas	Consideraciones
LOPDP – Artículo 7	La Ley Orgánica de Protección de Datos Personales, vigente desde 2021, exige que los sistemas que procesen datos internos o sensibles cuenten con medidas claras de consentimiento, almacenamiento seguro y trazabilidad. En este caso, el sistema mantiene los datos dentro de la red local, sin transmisión a terceros, cumpliendo los principios de confidencialidad y minimización[33].
Ley de comercio electrónico	Regula el valor legal y la integridad de documentos electrónicos y procesos automatizados. La solución asegura la trazabilidad de consultas, registros de actividad y almacenamiento interno cifrado, garantizando que los datos tengan validez jurídica frente a auditorías o procesos legales[34].
NTE INEN-ISO/IEC 27001	Norma internacional adoptada como norma técnica ecuatoriana que establece un sistema de gestión de seguridad de la información (SGSI). Se aplican medidas de control de segmentación de red, alineadas con esta norma para proteger datos operativos críticos[35].
UNESCO	Promueve principios éticos en el uso de IA como equidad, transparencia, inclusión y no discriminación. El sistema, al ejecutar modelos

	predictivos locales, permite inspección técnica, revisión del dataset de entrenamiento y evita sesgos por dependencia de proveedores externos[36].
ISO/IEC 25010:2011	Define el modelo de calidad para sistemas de software. Se priorizan atributos como usabilidad (interfaz clara para técnicos de planta), fiabilidad (respuesta consistente bajo carga), y mantenibilidad (arquitectura modular). Se incorporan pruebas funcionales unitarias y de estrés para garantizar estos aspectos[37].
Principios OCDE	Recomendaciones internacionales que enfatizan la necesidad de supervisión humana en sistemas con toma de decisiones. El sistema cuenta con un registro de decisiones y permite reversión manual de consultas o resultados, habilitando control humano efectivo[38].

2.4 Consideraciones éticas y legales

El diseño de la solución web local propuesta en el contexto industrial implica un alto compromiso ético, legal y estratégico, especialmente debido al tratamiento de datos sensibles asociados a procesos productivos, rendimiento operativo y decisiones empresariales críticas. Para garantizar la seguridad de la información, el cumplimiento normativo y la responsabilidad en el desarrollo, se consideran los siguientes aspectos:

Desde el punto de vista legal, la solución se adhiere a lo dispuesto en la Ley Orgánica de Protección de Datos Personales (LOPDP) del Ecuador, particularmente en su Artículo 7, que

regula la legitimidad en el tratamiento de datos mediante el consentimiento del titular o cuando este se derive de una obligación legal o un interés legítimo claramente fundamentado [33]. En este caso, la empresa mantiene la titularidad de los datos industriales, por lo que el sistema asegura que su procesamiento se realice exclusivamente en servidores internos, restringiendo el acceso externo y eliminando la dependencia de servicios en la nube. Asimismo, el Artículo 8 establece que dicho tratamiento debe realizarse con consentimiento informado y bajo criterios de especificidad y claridad, lo cual se cumple mediante políticas internas de seguridad de la información [33]. Si los datos incluyen información considerada estratégica o confidencial —como parámetros de producción o mantenimiento predictivo—, el sistema se ajusta además al Artículo 30 de la LOPDP, garantizando el tratamiento bajo condiciones de alta confidencialidad y mecanismos avanzados de protección contra accesos no autorizados [33].

En cuanto a la propiedad intelectual, el sistema se construye con herramientas de código abierto como Ollama para el procesamiento del lenguaje natural, y se integra con modelos predictivos previamente entrenados, almacenados localmente en archivos serializados. Esta arquitectura permite evitar compromisos contractuales con proveedores externos y mantiene el control total sobre la lógica del negocio y los algoritmos aplicados. El uso de software libre con licencias compatibles garantiza la legalidad de la implementación, al tiempo que promueve una cultura de innovación abierta y sostenible, sin riesgos legales asociados a restricciones de uso o distribución.

Desde una perspectiva ética, el proyecto busca potenciar la eficiencia operativa mediante un sistema inteligente de consulta y análisis que asista a los equipos técnicos en la toma de decisiones. Al ejecutarse en un entorno controlado, se prioriza la seguridad, la integridad de la información y la privacidad de las operaciones industriales, lo cual refleja un compromiso con los

principios de ética empresarial. El enfoque no solo busca maximizar el rendimiento y la trazabilidad de los procesos, sino también preservar el capital intelectual de la organización, evitando filtraciones o usos indebidos de datos críticos. De esta forma, el sistema contribuye a una transformación digital responsable, alineada con los valores de soberanía tecnológica, autonomía y protección del conocimiento estratégico.

2.5 Etapas del diseño

El proceso de diseño adoptado para implementar la solución se estructura en nueve etapas secuenciales e interconectadas. Estas fases garantizan que el desarrollo sea coherente con las necesidades específicas del cliente, asegurando la seguridad de los datos, el control total de la infraestructura y una respuesta eficiente del sistema en tiempo real, sin depender de servicios en la nube.

En la Figura 2, se muestra el proceso que guio el desarrollo de la solución. La primera etapa consistió en la entrevista con el cliente, en la cual se identificaron las necesidades centrales del proyecto, destacando la importancia de mantener los datos en una infraestructura local por motivos de privacidad, así como la necesidad de ofrecer un asistente virtual eficiente y autónomo para consultas internas.

La segunda etapa, definición de la problemática, permitió delimitar claramente los objetivos del sistema: implementar un asistente virtual basado en inteligencia artificial que pudiera responder consultas y generar predicciones, todo ello desde una red cerrada.

En la tercera etapa, se realizó un análisis de requerimientos, tanto funcionales como no funcionales. A nivel funcional, se incluyó el procesamiento del lenguaje natural mediante Ollama, la integración del modelo predictivo y la disponibilidad del sistema a través de una interfaz web.

En cuanto a los requerimientos no funcionales, se priorizó la baja latencia, la seguridad y la escalabilidad interna.

La cuarta etapa se centró en la evaluación y selección de la solución técnica. Se estudiaron diferentes alternativas de implementación, incluyendo soluciones en la nube, híbridas y locales. La alternativa seleccionada fue la ejecución de una solución 100% local, integrando un modelo PLN en Ollama para el manejo del lenguaje natural, y un modelo de predicción entrenado y almacenado en un archivo serializado, permitiendo una ejecución rápida sin depender de procesamiento externo.

En la quinta etapa, se definió la arquitectura conceptual del sistema. Esta incluye los módulos principales: el módulo de interfaz web, el módulo de lenguaje natural (Ollama ejecutándose localmente), el módulo de inferencia del modelo predictivo (cargado desde archivo). Se establecieron las entradas (consultas del usuario), los procesos internos (interpretación, análisis y predicción) y las salidas (respuestas del asistente virtual).

La sexta etapa abarcó el diseño backend y frontend. En el backend se diseñaron las rutas API que conectan con Ollama y el modelo serializado, así como los servicios para el manejo de sesiones y autenticación de usuarios. En esta fase también se realizó la selección del modelo de PLN desarrollado por Meta, adecuado para el sistema. Por su parte, en el frontend se desarrolló una interfaz limpia e intuitiva accesible desde navegadores dentro de la red local. Esta interfaz permite al usuario interactuar directamente con el asistente y visualizar los resultados de las predicciones y consultas.

Durante la séptima etapa, se realizó la selección del modelo de ML más adecuado. Se evaluaron distintos algoritmos supervisados según el tipo de predicción requerida, priorizando aquellos con alta precisión y bajo tiempo de inferencia. El modelo seleccionado fue entrenado

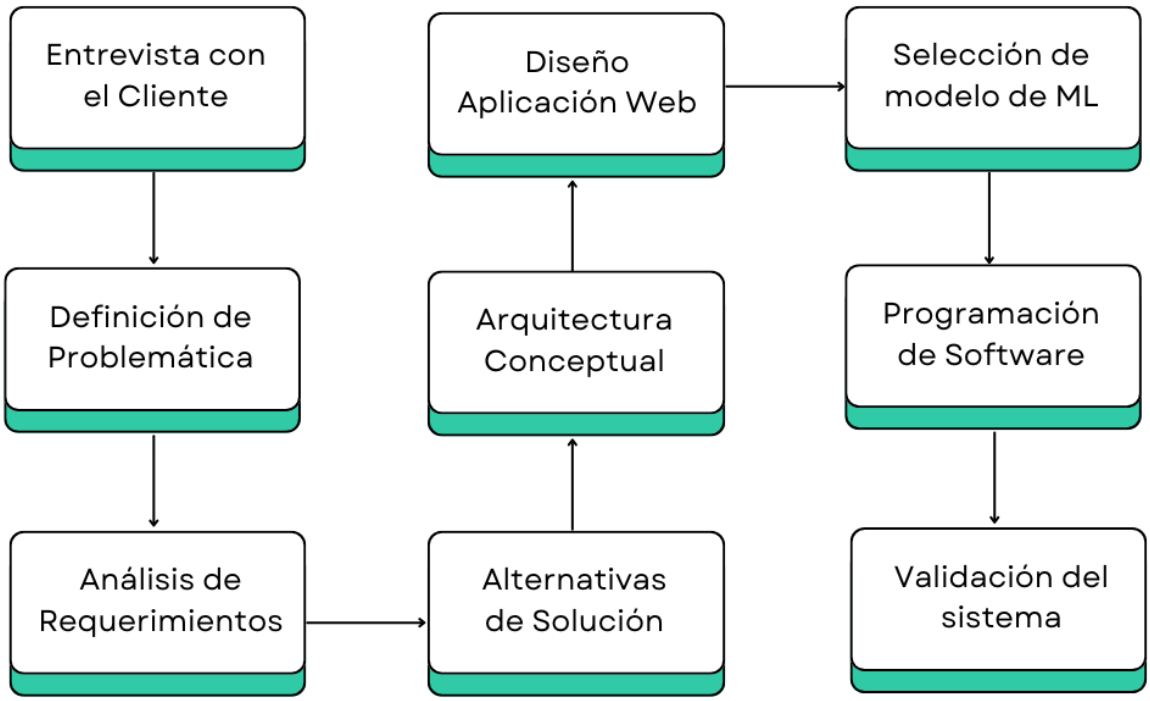
previamente, validado con datos, y posteriormente serializado para su integración rápida en el sistema.

La octava etapa consistió en la programación del software, integrando todos los módulos desarrollados. Se codificó la lógica de comunicación entre la interfaz web, el motor de lenguaje Ollama y el modelo predictivo serializado. Además, se configuraron los componentes internos para garantizar una ejecución estable y funcional en el entorno local de la empresa.

Finalmente, en la novena etapa, se realizaron pruebas del sistema y validación funcional. Las pruebas incluyeron la evaluación del tiempo de respuesta del asistente, la precisión del modelo predictivo, y el correcto funcionamiento de la interfaz web bajo distintos escenarios. Se utilizó un entorno local de pruebas para simular condiciones reales.

Figura 2

Diagrama de etapas de diseño



2.6 Arquitectura conceptual

La arquitectura conceptual define la organización general del sistema, estructurada en tres elementos fundamentales: entradas, procesos internos y salidas. Esta estructura es clave para garantizar que la solución basada en inteligencia artificial sea funcional, escalable y segura dentro de un entorno industrial. Las entradas al sistema están constituidas por las preguntas realizadas por los usuarios, por lo general, supervisores o técnicos de planta, a través de la interfaz web. Estas preguntas pueden redactarse en lenguaje natural, tales como: "¿Cuál fue el MTBF del mes anterior?" o "¿Cuál es el estado de la máquina L hoy?".

Previo al procesamiento de cualquier entrada, el sistema requiere la autenticación del usuario mediante credenciales válidas. Este mecanismo de inicio de sesión garantiza que solo usuarios autorizados, como técnicos o supervisores registrados, puedan acceder a las funcionalidades del asistente virtual. Este componente refuerza los criterios de seguridad de la arquitectura, asegurando la integridad del sistema y restringiendo el uso a personal acreditado.

Una vez autenticado el usuario, las preguntas se interpretan según su intención. Si la intención corresponde a una consulta, se genera automáticamente una instrucción hacia la base de datos de KPIs, devolviendo una respuesta en lenguaje natural que se muestra en la interfaz. Si la intención es una predicción, se consultan los registros de la base de datos de máquinas y los datos se procesan mediante un modelo predictivo, el cual genera una respuesta basada en probabilidades o proyecciones operativas. En caso de que la intención no sea ni consulta ni predicción, el sistema entrega una respuesta predeterminada que indica al usuario que su solicitud no se ajusta a las funcionalidades disponibles.

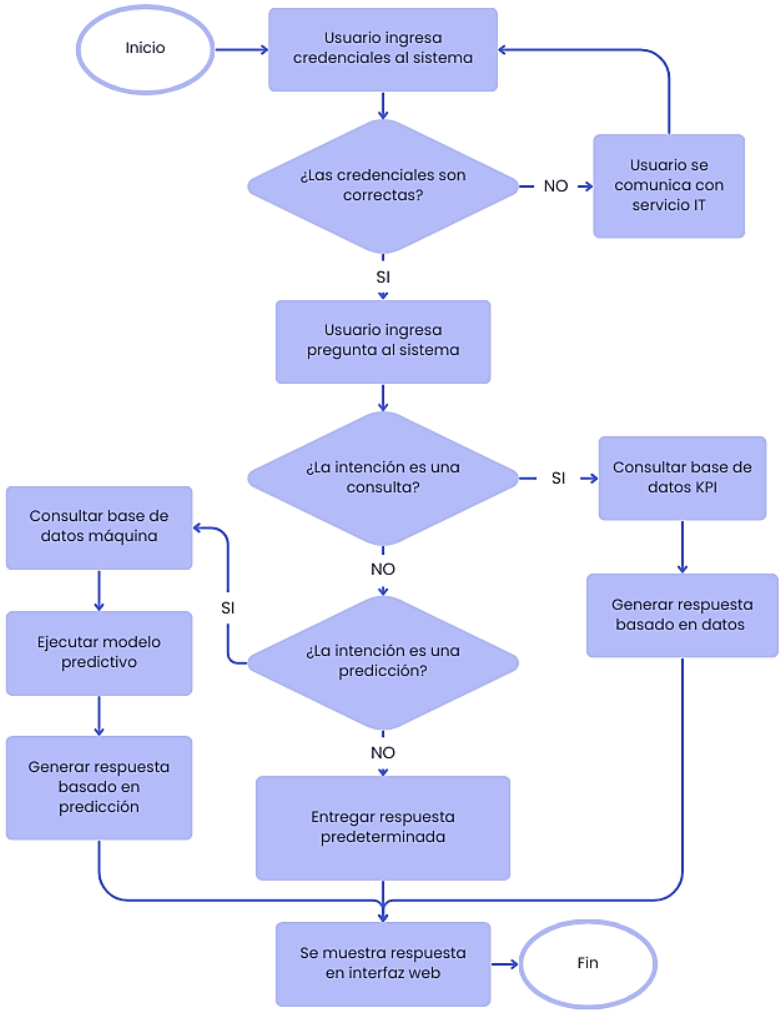
Las salidas del sistema son siempre respuestas textuales desplegadas en la interfaz web, ya sea en forma de consultas resueltas, predicciones o mensajes informativos.

En la Figura 3 se presenta el diagrama funcional de esta arquitectura conceptual, donde se evidencia el flujo de información desde la validación de credenciales hasta la entrega de la respuesta final, con el respaldo de la base de datos y los modelos de IA.

Finalmente, esta arquitectura considera tres principios clave: seguridad, al operar en un entorno local con acceso restringido; escalabilidad, al permitir la incorporación de nuevos indicadores o módulos sin alterar el núcleo del sistema; y mantenibilidad, gracias a un diseño modular y desacoplado que facilita actualizaciones independientes.

Figura 3

Flujo interacción usuario-asistente virtual



2.7 Diseño aplicación web

El diseño de la aplicación web se organiza en tres componentes principales: frontend, backend y API de comunicación con el modelo de inteligencia artificial. Esta arquitectura modular permite una integración fluida entre la interfaz de usuario, el procesamiento lógico y el acceso al modelo de predicción.

El frontend corresponde a la capa visual del sistema, con la cual interactúa directamente el usuario. En esta interfaz se lleva a cabo el proceso de autenticación de usuarios, se reciben las preguntas y se muestran las respuestas de forma clara e intuitiva. Este módulo está diseñado para ser responsivo y accesible, facilitando una experiencia de usuario eficiente.

El backend actúa como núcleo lógico del sistema. Su función principal es interpretar las preguntas del usuario, identificar su intención y realizar las consultas necesarias al modelo de IA. Este componente está construido en un entorno de ejecución optimizado que permite la gestión eficiente de solicitudes y respuestas.

La API cumple el rol de intermediario entre el backend y el modelo de predicción. Dado que el modelo está implementado en python, se desarrolló una API que expone endpoints específicos para la comunicación. Esta estructura permite que el backend, sin importar el lenguaje o framework utilizado, pueda interactuar de forma segura y eficiente con el modelo de machine learning.

2.7.1 Interfaz de usuario

Para el diseño de la interfaz de usuario se tomaron como referencia modelos de asistentes virtuales ampliamente reconocidos, como ChatGPT, Google Gemini y Deepseek. Se analizó cómo estos sistemas presentan visualmente sus interfaces para facilitar una interacción cómoda, fluida y accesible para los usuarios.

A partir del análisis, se identificaron patrones comunes en su diseño: una apariencia visual simple, con una página inicial que presenta una barra superior con el título del sistema y, al centro, una caja de texto donde el usuario puede escribir su pregunta. Al enviar el primer mensaje, la interfaz se transforma en un entorno tipo chat, con una apariencia centrada y limpia. El área de mensajes tiene desplazamiento independiente (scroll), manteniendo siempre fija la caja de entrada. También se incluyen elementos visuales como burbujas de texto y puntos animados que indican que el sistema está procesando una respuesta, lo cual mejora la experiencia de interacción.

Aunque no se dispone de información oficial sobre las tecnologías exactas utilizadas por estos sistemas, React es una de las bibliotecas más populares para el desarrollo de interfaces web modernas, por lo que se utilizó en este proyecto. Junto con React, se implementó la librería Tailwind CSS, que permite aplicar estilos visuales de manera rápida y consistente.

La aplicación web está compuesta por dos páginas principales: la página de inicio de sesión (login) y la interfaz de chat. Al iniciar el sistema, el usuario es recibido por la pantalla de login, diseñada con un enfoque centrado. Esta contiene dos campos de entrada: uno para el correo electrónico y otro para la contraseña, como se muestra en la Figura 4. Al ingresar las credenciales, estas son enviadas al backend, donde se realiza la validación correspondiente. Si los datos son correctos, el usuario es redirigido automáticamente a la interfaz principal del asistente virtual.

Por otro lado, además del diseño visual, en la interfaz de chat se incorporaron elementos funcionales para mejorar la experiencia de uso, como restricciones para evitar el envío accidental de mensajes vacíos, soporte para mensajes extensos sin desbordar el diseño, y la animación de respuesta mientras el sistema procesa la información. La estructura desarrollada incluye un encabezado (header), un pie de página (footer) y una zona central dinámica, como se muestra en las Figuras 5 y 6. Esta interfaz está conectada con el backend del sistema, el cual se encarga de gestionar las consultas y respuestas del asistente virtual. Durante el desarrollo y las pruebas, la

aplicación web fue ejecutada en un entorno local, permitiendo una integración fluida con el modelo del backend para validar el correcto funcionamiento del sistema.

Figura 4

Interfaz de autenticación del asistente virtual

Figura 5

Interfaz inicial del asistente virtual

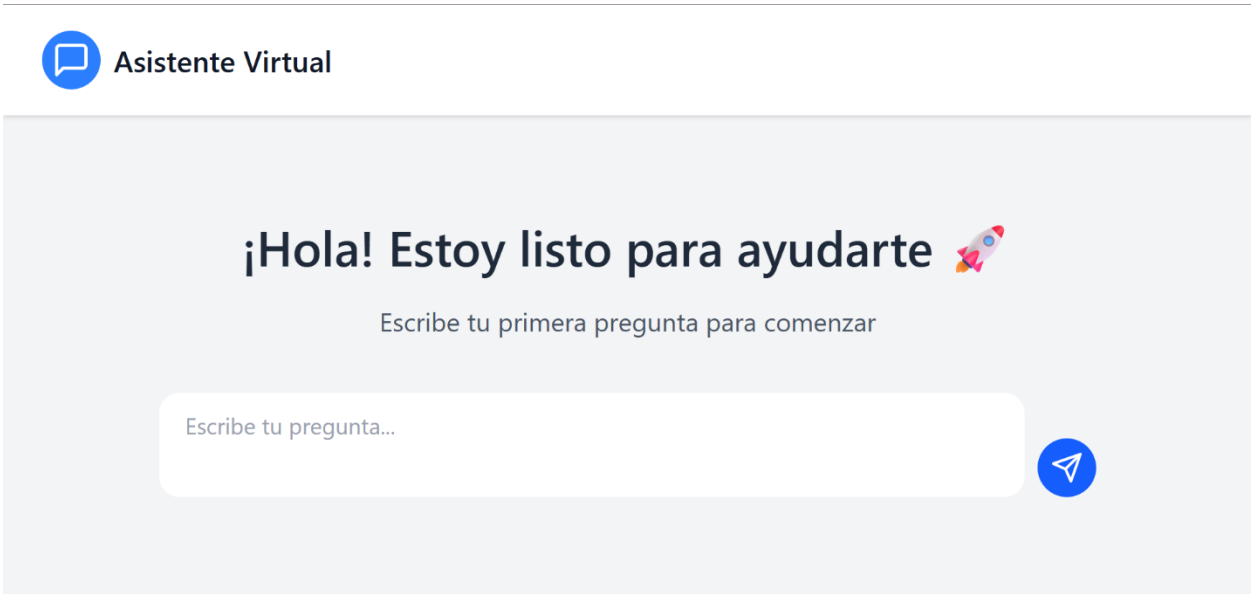
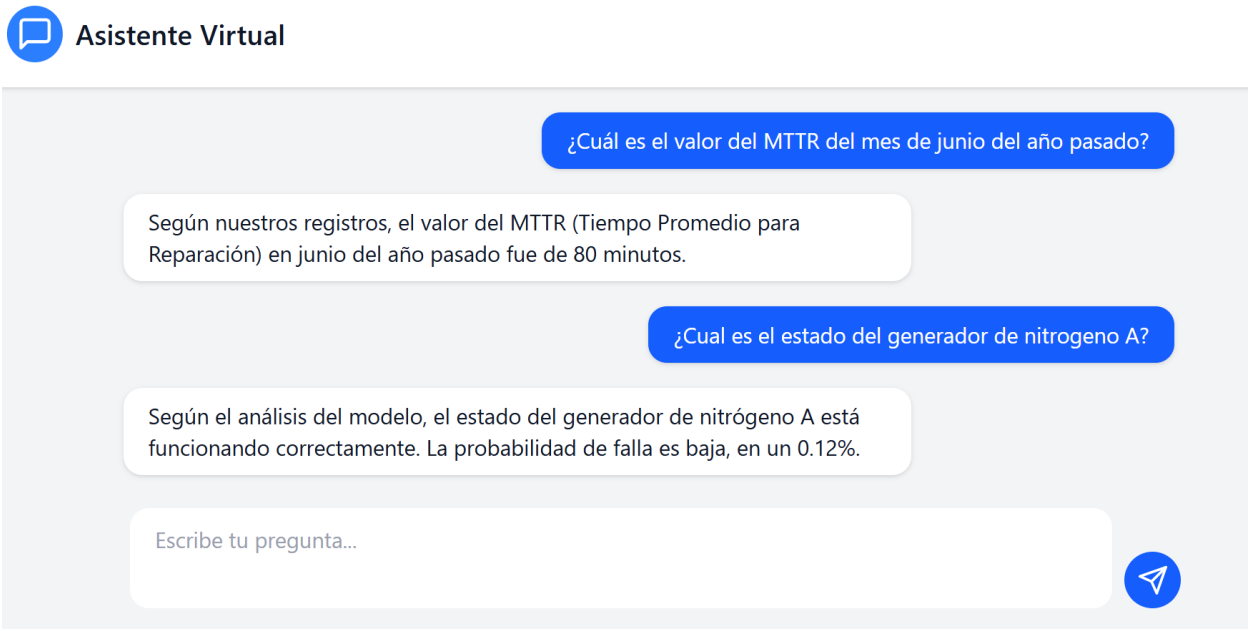


Figura 6

Interfaz de conversación al enviar primer mensaje



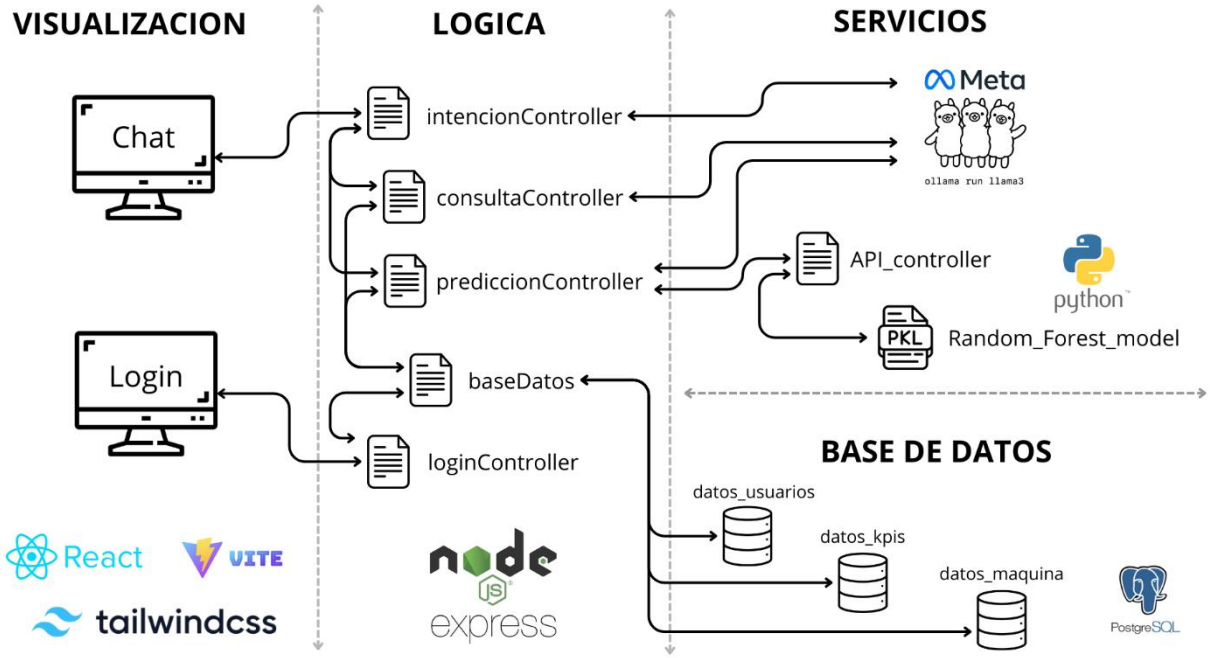
2.7.2 Lógica del sistema

Por su lado, el backend del sistema se desarrolló utilizando Node.js con el framework Express, lo que permitió crear un servidor ligero y eficiente, ideal para realizar pruebas en entorno local. Su principal función es recibir las preguntas formuladas por el usuario desde la interfaz web, procesarlas y entregar respuestas adecuadas, ya sea mediante una consulta a la base de datos o una predicción del estado de las máquinas. El servidor escucha peticiones en el puerto 3001, siendo accesible a través de http://localhost:3001.

Este flujo general de funcionamiento puede observarse en la Figura 7, que muestra la arquitectura interna del sistema, desde la interfaz web hasta la interacción con los distintos módulos de procesamiento.

Figura 7

Diagrama de arquitectura general del sistema



El sistema inicia en el frontend con la página de inicio. Cuando el usuario pulsa el botón de autenticación, sus credenciales se envían al backend y son gestionadas por el módulo loginController. Este módulo consulta la base de datos a través del módulo baseDatos, encargado de la comunicación con el sistema de almacenamiento. Con base en el resultado, el loginController valida si las credenciales corresponden a un usuario registrado. En caso de que los datos no sean válidos, el sistema devuelve un mensaje de no autorizado e indica al usuario que debe contactar al equipo de tecnologías de la empresa para la verificación de sus credenciales. Si los datos son correctos, el sistema concede el acceso a la interfaz de chat.

A partir de este momento, se inicia el flujo principal del sistema conversacional. Cada vez que el usuario envía un mensaje desde la interfaz, este se procesa mediante una solicitud POST al endpoint /preguntar. Esta solicitud contiene el mensaje del usuario en formato JSON. Una vez recibido, el sistema aplica un primer paso de procesamiento: la detección de intención, la cual es gestionada por el módulo intencionController. Este módulo utiliza el modelo PLN, al que se le

proporciona un prompt que describe tres posibles categorías de intención: consulta, predicción u ninguna. El modelo responde en formato JSON con la categoría correspondiente, y es aquí donde el módulo intencionController delega el flujo hacia los diferentes módulos dependiendo de la respuesta obtenida.

Si la intención detectada es consulta, el sistema delega el procesamiento al módulo consultaController. En este componente, se construye dinámicamente una instrucción SQL con apoyo del modelo PLN, utilizando un prompt que incluye ejemplos predefinidos y un formato de salida estructurado, compuesto por claves como sql_query (que contiene la consulta generada) y original_query (una versión resumida de la consulta en lenguaje natural). La instrucción SQL se envía al módulo baseDatos para obtener la información sobre indicadores clave como MTTR y MTBF. Los resultados obtenidos son enviados nuevamente al modelo PLN, que genera una explicación breve y comprensible para el usuario, resumiendo el resultado en lenguaje natural.

En caso de que la intención detectada sea predicción, el procesamiento es gestionado por el módulo prediccionController. Al igual que en el caso anterior, se genera primero una consulta SQL, esta vez sobre registros operativos de la máquina. La consulta recupera los datos más recientes de la máquina. Posteriormente, estos datos son enviados al modelo de aprendizaje automático expuesto a través de un servidor FastAPI en el puerto 8000. Esta actividad la maneja el módulo API_controller, encargado de enviar la información al modelo y recibir una respuesta binaria (0 o 1) indicando si es probable que la máquina falle o no. Finalmente, el resultado junto con los datos utilizados es enviados al modelo PLN, que elabora una explicación natural del pronóstico sin mencionar términos técnicos como “cero” o “uno”, sino empleando frases interpretables como “el modelo predice que no se espera una falla”.

Finalmente, si la intención no corresponda a ninguna de las categorías anteriores, el sistema genera una respuesta indicando que no se ha realizado ni una consulta ni una predicción, y sugiere al usuario cuál debería ser la acción esperada. Esta funcionalidad asegura que el asistente mantenga

su enfoque en su propósito principal, incluso cuando el usuario no formule una consulta estructurada.

En todas las rutas, se incorporaron validaciones básicas (por ejemplo, evitar mensajes vacíos), mensajes de log que permiten el seguimiento del procesamiento en consola, y manejo de errores que asegura que el servidor no se detenga ante entradas inesperadas o problemas en la conexión con otros servicios. Todo el backend se conecta localmente con una instancia de PostgreSQL para las consultas, con un servidor local con el modelo PLN alojado en localhost:11434 para generación e interpretación de lenguaje natural, y con el servicio de predicción basado en FastAPI para el modelo de aprendizaje automático.

Esta arquitectura modular y basada en microservicios locales permite realizar pruebas rápidas y un desarrollo ágil, manteniendo cada componente desacoplado y enfocado en una tarea específica: clasificación de intención, generación de SQL, ejecución de consulta, predicción con modelo externo y generación de respuestas en lenguaje natural.

Dado que gran parte de la lógica del sistema se apoya en el procesamiento de lenguaje natural para interpretar preguntas del usuario y generar respuestas comprensibles, fue fundamental seleccionar un modelo PLN que ofreciera un equilibrio entre precisión, velocidad de respuesta y eficiencia de recursos. En este contexto, se evaluaron distintos modelos locales desarrollados por Meta.

2.7.3 Selección de modelo PLN

Para la implementación del sistema de procesamiento de lenguaje natural (PLN), se optó por utilizar un modelo desarrollado por Meta, conocido como LLaMA, acrónimo de Large Language Model Meta AI (Gran Modelo de Lenguaje de Meta Inteligencia Artificial). Esta familia de modelos se caracteriza por ofrecer diferentes versiones con distintos tamaños, medidos en la

cantidad de parámetros (en miles de millones o “billones”), que determinan su capacidad y potencia para entender y generar texto {referencia}.

Meta ha lanzado varias generaciones y variantes de LLaMA, que se diferencian principalmente en el número de parámetros, lo que influye en su rendimiento y velocidad. A continuación, se presenta la Tabla 5 donde se enumeran los modelos evaluados para la selección del sistema:

Tabla 5

Modelos de LLaMA evaluados para selección del sistema

Modelo	Tamaño (parámetros)
LLaMA 2	7 billones
LLaMA 2	13 billones
LLaMA 3	8 billones
LLaMA 3.1	8 billones
LLaMA 3.2	1 billón
LLaMA 3.2	3 billones
TinyLLaMA	1.1 billones

A partir de la evaluación empírica de cada modelo, utilizando un total de 30 preguntas formuladas al sistema por modelo, se obtuvo el tiempo promedio de respuesta y el porcentaje de respuestas correctas. Este último indicador se refiere a la proporción de respuestas coherentes y pertinentes generadas por el modelo, en relación con lo solicitado por el usuario. A continuación, se presentan los gráficos que resumen estos resultados por modelo evaluado.

Figura 8

Porcentaje de respuestas correctas por modelo LLaMA

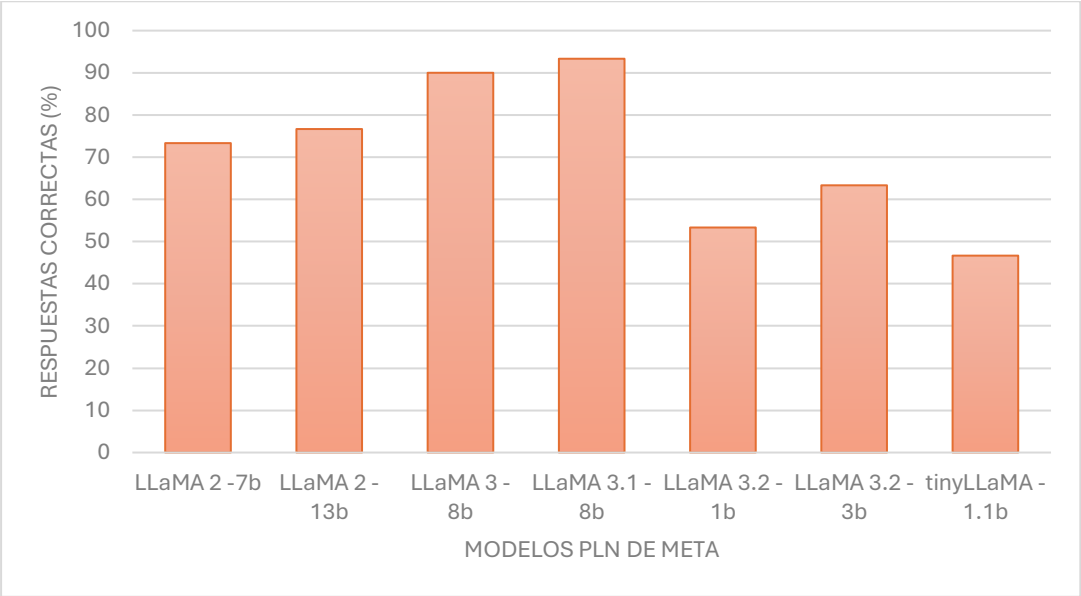
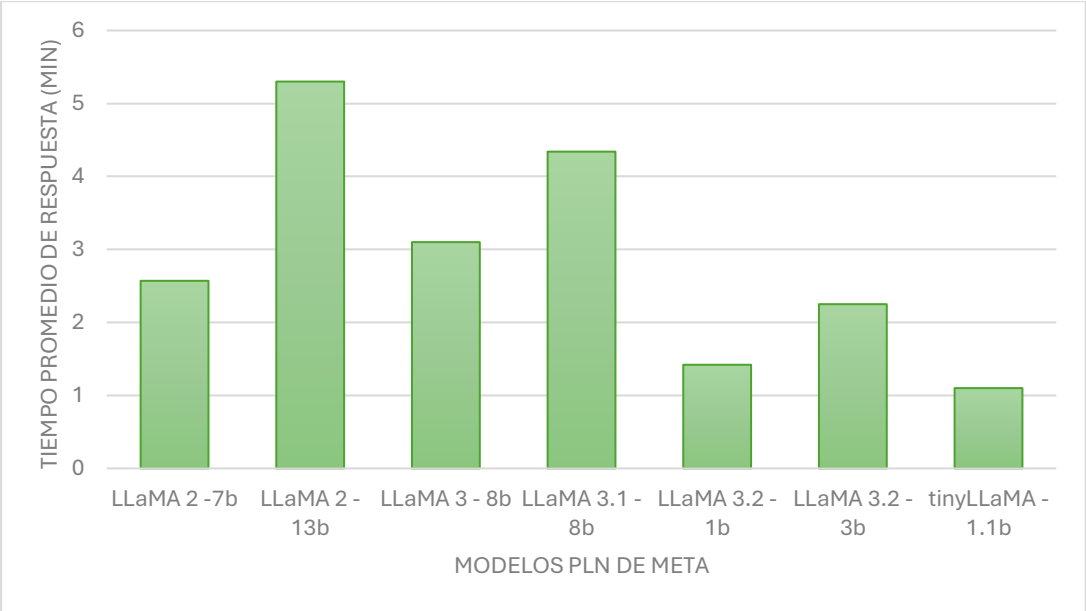


Figura 9

Tiempo promedio de respuesta por modelo LLaMA



Como se observa en las Figuras 8 y 9, los modelos más precisos son LLaMA 3 y LLaMA 3.1, ambos con 8 billones de parámetros. No obstante, la diferencia clave radica en el tiempo promedio de respuesta. Mientras que la precisión de ambos modelos es similar, LLaMA 3 demuestra ser considerablemente más rápido que su contraparte.

Este factor resulta fundamental, ya que, como se ha mencionado previamente, cada minuto cuenta al momento de tomar decisiones críticas en entornos industriales. Por ello, a pesar de la leve ventaja en precisión de LLaMA 3.1, el menor tiempo de respuesta de LLaMA 3 lo posiciona como la opción más equilibrada y adecuada para el sistema implementado, al ofrecer un rendimiento robusto sin comprometer la eficiencia.

2.7.4 API de comunicación con modelo predictivo

Para la implementación del modelo de predicción, se desarrolló una API utilizando el framework FastAPI, una herramienta moderna y eficiente para construir servicios web en Python. Esta API funciona como un microservicio que expone un único endpoint destinado a recibir datos numéricos desde el backend del sistema, procesarlos con un modelo de machine learning previamente entrenado y retornar una respuesta binaria que indica la probabilidad de falla de una máquina.

El servicio se inicia con la carga del modelo desde un archivo serializado en formato pickle (.pkl). Este modelo, entrenado previamente, se almacena en la ruta local “models/Random_Forest_model.pkl” y es cargado en memoria al momento de inicializar la aplicación. Esto permite que el modelo esté disponible de forma persistente durante toda la ejecución del servidor, evitando retrasos innecesarios por cargas repetitivas. En la siguiente sección, se describiera con más detalle el modelo predictivo.

La API define una estructura de entrada mediante la clase DatosEntrada, basada en pydantic, la cual valida que la solicitud POST contenga una lista de valores numéricos (list[float]). Esta lista representa las características que describen el estado actual de una máquina o componente, tales como presión, temperatura, ciclos de operación, entre otras.

Cuando el backend del sistema necesita realizar una predicción, envía una solicitud HTTP POST al endpoint /predecir, incluyendo en el cuerpo del mensaje los datos requeridos.

Internamente, la API convierte estos datos en una estructura compatible con el modelo (una lista de listas) y ejecuta el método predict del modelo cargado. El resultado es un valor entero (0 o 1) que representa la predicción del sistema: 0 si no se espera una falla, o 1 si es probable que ocurra una.

La respuesta se envía en formato JSON, facilitando la integración con otros componentes del sistema. Este microservicio está diseñado para funcionar en un entorno local, escuchando en el puerto 8000, y su arquitectura simple pero robusta permite realizar pruebas ágiles durante la etapa de desarrollo. Su integración con el backend desarrollado en Node.js asegura una comunicación fluida y desacoplada entre los módulos de procesamiento lógico y el modelo de inteligencia artificial.

Esta separación de responsabilidades entre backend y API de predicción mejora la mantenibilidad del sistema y permite futuras actualizaciones o sustituciones del modelo de predicción sin afectar el resto de la arquitectura.

2.8 Procedimiento de predicción de fallos en maquinaria industrial

El presente apartado describe de forma detallada el procedimiento implementado para desarrollar y evaluar un modelo de aprendizaje automático de predicción de fallos en un generador de nitrógeno, equipo crítico en la línea de envasado de las plantas dedicadas a la producción de bebidas. El objetivo de dicho sistema es anticipar la ocurrencia de fallos, permitiendo la intervención de mantenimiento preventivo y evitando paradas no planificadas que afectan la producción.

2.8.1 Adquisición de datos

En el presente estudio se utilizó datos de un caso real de operación de un generador de nitrógeno [39], monitoreado durante un periodo de dos meses (1 de agosto – 29 de septiembre de 2023). El sistema fue instrumentado con sensores IoT que transmitieron en tiempo real, mediante

el protocolo de Transporte de Telemetría mediante Cola de Mensajes (MQTT), mediciones continuas de las principales variables físicas involucradas en el proceso de separación de gases del generador de nitrógeno.

Para el análisis se consideraron exclusivamente las variables medidas de manera directa:

- Presión de aire del tamiz molecular de carbono (CMS).
- Concentración base de oxígeno.
- Presión de nitrógeno.
- Oxígeno sobre el umbral.

Durante el periodo de observación únicamente se registró un tipo de condición de fallo: "Fallo de presurización CMS", evento que compromete la eficiencia en la separación de oxígeno y nitrógeno.

2.8.2 Construcción y etiquetado del conjunto de datos

Con el fin de preparar los datos para el entrenamiento de modelos de aprendizaje automático, los autores del dataset implementaron un esquema de segmentación temporal basado en ventanas.

- Ventana de lectura (RW, Reading Window): intervalo temporal de entre 10 y 30 minutos, dentro del cual se recopilan las secuencias de mediciones de las variables seleccionadas.
- Ventana de predicción (PW, Prediction Window): intervalo de 30 minutos inmediatamente posterior a la RW, utilizado para verificar la ocurrencia o ausencia de un evento de fallo **(**Fallo de presurización CMS**).**

El proceso de etiquetado se realizó conforme al siguiente criterio: Si dentro de la PW se detecta al menos un código de alerta asociado al "Fallo de presurización CMS", la RW

inmediatamente anterior se etiqueta con la clase “fallo” y el valor 1. En caso contrario, la RW se etiqueta con la clase “normal” y el valor 0.

2.8.3 Preprocesamiento de datos

Los datos fueron preprocesados por los autores del dataset siguiendo estos pasos:

- Re-muestreo a frecuencia fija de 1 minuto para obtener series uniformes.
- Relleno de valores faltantes mediante la técnica relleno hacia adelante (forward fill), que consiste en sustituir un valor ausente con el último dato válido registrado, bajo el supuesto de que la variable se mantiene estable en ausencia de nuevas mediciones.
- Normalización min-máx. en rango [0,1] basada en parámetros del conjunto de entrenamiento.

Dado que los datos ya están preprocesados, en este proyecto se trabajó directamente con esta información preparada.

2.8.4 Partición del conjunto de datos

El dataset original consta de 85,860 registros y 6 columnas, las cuales son:

- Timestamp: Marca temporal de la medición (fecha y hora).
- CMS air pressure: Presión de aire en CMS (bar).
- Concentrazione ossigeno base: Concentración base de oxígeno (%).
- Nitrogen pressure: Presión de nitrógeno (bar).
- Oxygen over threshold: Indicador binario si la concentración de oxígeno supera el umbral.
- PW_0.5h: Variable objetivo-binaria que indica la presencia (1) o ausencia (0) de “Fallo de presurización CMS” en la ventana de predicción de 0.5 horas.

Para evitar fugas de información y respetar la secuencia temporal, se dividió el dataset en:

- Entrenamiento completo (80%): primeros 68,688 registros para entrenamiento y validación.
- Prueba (20%): últimos 17,172 registros para evaluación final.

Dentro del conjunto de entrenamiento completo:

- Entrenamiento final (80%): primeras 54,950 filas para ajuste y entrenamiento.
- Validación (20%): últimas 13,738 filas para validación y ajuste de hiperparámetros.

Esta estrategia garantiza que el modelo aprenda solo de datos históricos y se evalúe con datos futuros.

2.8.5 Almacenamiento de los datasets

Las particiones se almacenaron en formato binario pickle para facilitar su acceso y reutilización.

2.8.6 Análisis exploratorio de datos

Con el fin de fundamentar la selección de los modelos predictivos, se realizó un análisis exploratorio de datos (AED) considerando tres dimensiones principales: variabilidad y complejidad de las variables, dependencia temporal y relaciones multivariadas.

La variabilidad de cada variable se midió mediante la varianza σ^2, calculada como:

$$\boldsymbol{\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{2.1}$$

donde $x_i$es el valor en la observación i, $\bar{x}$ es la media y N el número total de observaciones. La complejidad se evaluó con la entropía de Shannon H sobre los valores discretizados en k bins:

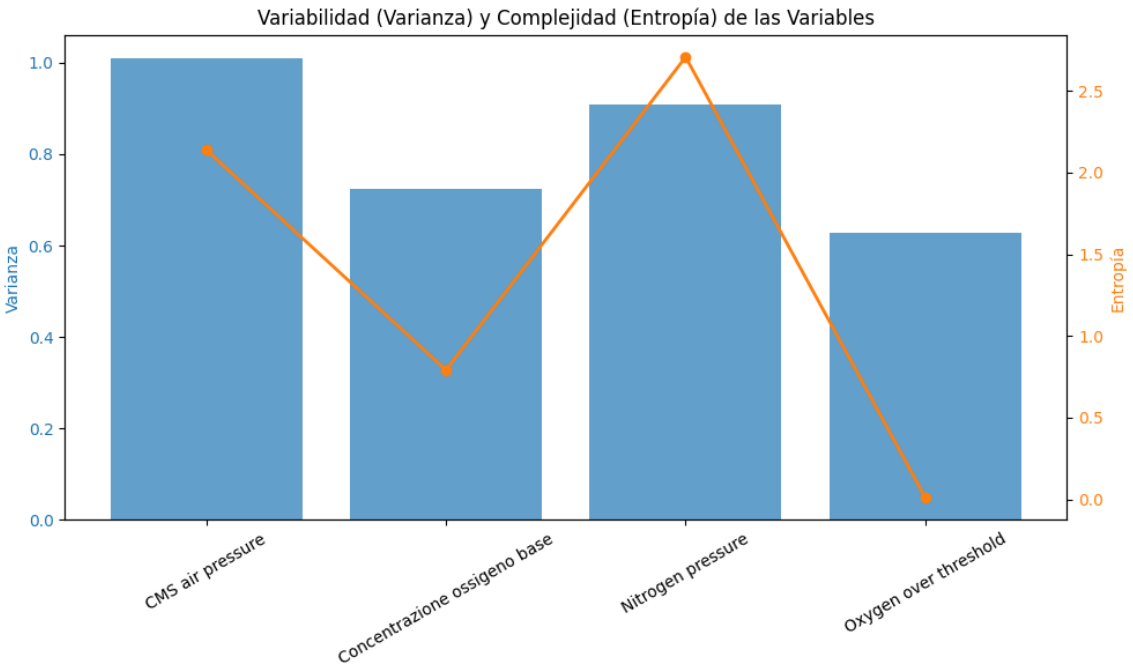
$$\boldsymbol{H = -\sum_{j=1}^{k} p_j \log_2 p_j} \tag{2.2}$$

donde p_j es la probabilidad estimada (frecuencia relativa) del valor en el bin j. Una mayor entropía indica distribuciones más complejas y menos predecibles.

En este estudio, variables como CMS air pressure y Nitrogen pressure presentaron varianzas elevadas (≈0.9–1.0) y entropías altas, reflejando gran dispersión y complejidad. En contraste, Oxygen over threshold mostró valores bajos en ambos indicadores, lo que evidencia una distribución más concentrada y sencilla (véase la Figura 10).

Figura 10

Varianza y entropía de las variables



Se evaluó la dependencia temporal usando la función de autocorrelación (ACF), que calcula la correlación de cada serie con sus versiones desplazadas en el tiempo (lags). Formalmente, para lag k:

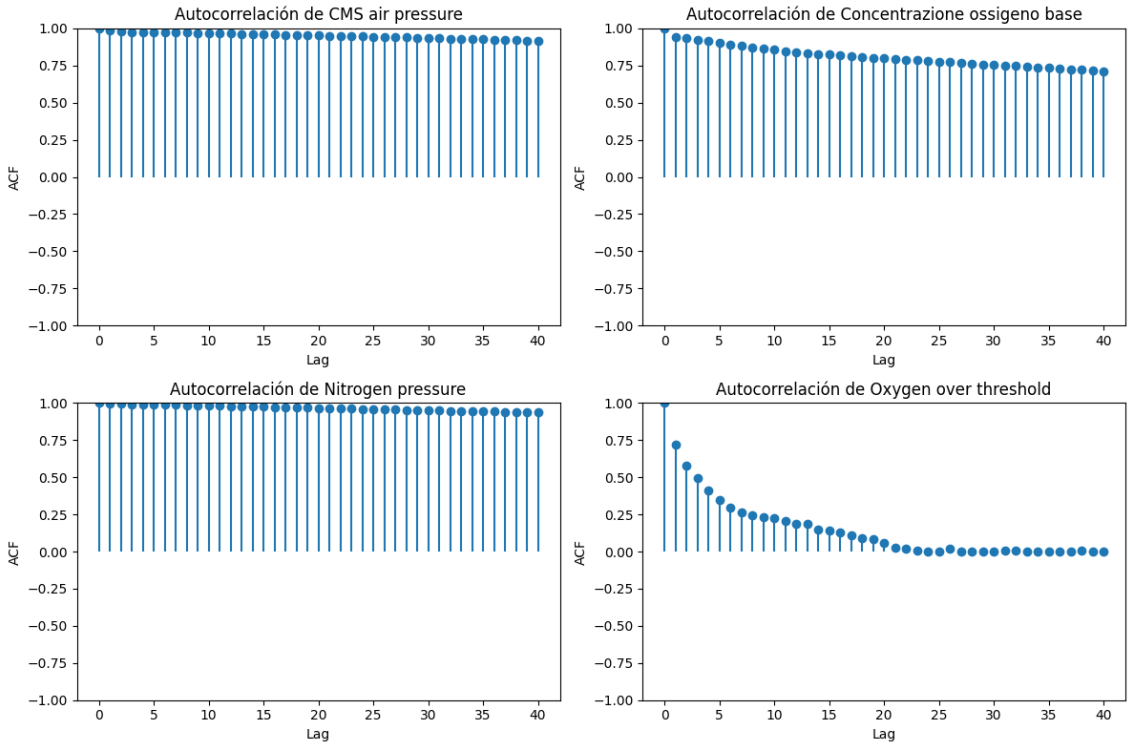
$$\boldsymbol{p_k = \frac{Cov(X_t, X_{t-k})}{\sigma^2}} \tag{2.3}$$

donde $X_{t,}$ es el valor en el tiempo t y σ^2 su varianza.

Se observó que CMS air pressure, Concentrazione ossigeno base y Nitrogen pressure presentaron autocorrelaciones elevadas y persistentes, mientras que Oxygen over threshold decayó rápidamente, mostrando dependencia de corto plazo (véase Figura 11).

Figura 11

Autocorrelación de las variables



Para identificar relaciones entre las variables, se calculó la matriz de correlación de Pearson:

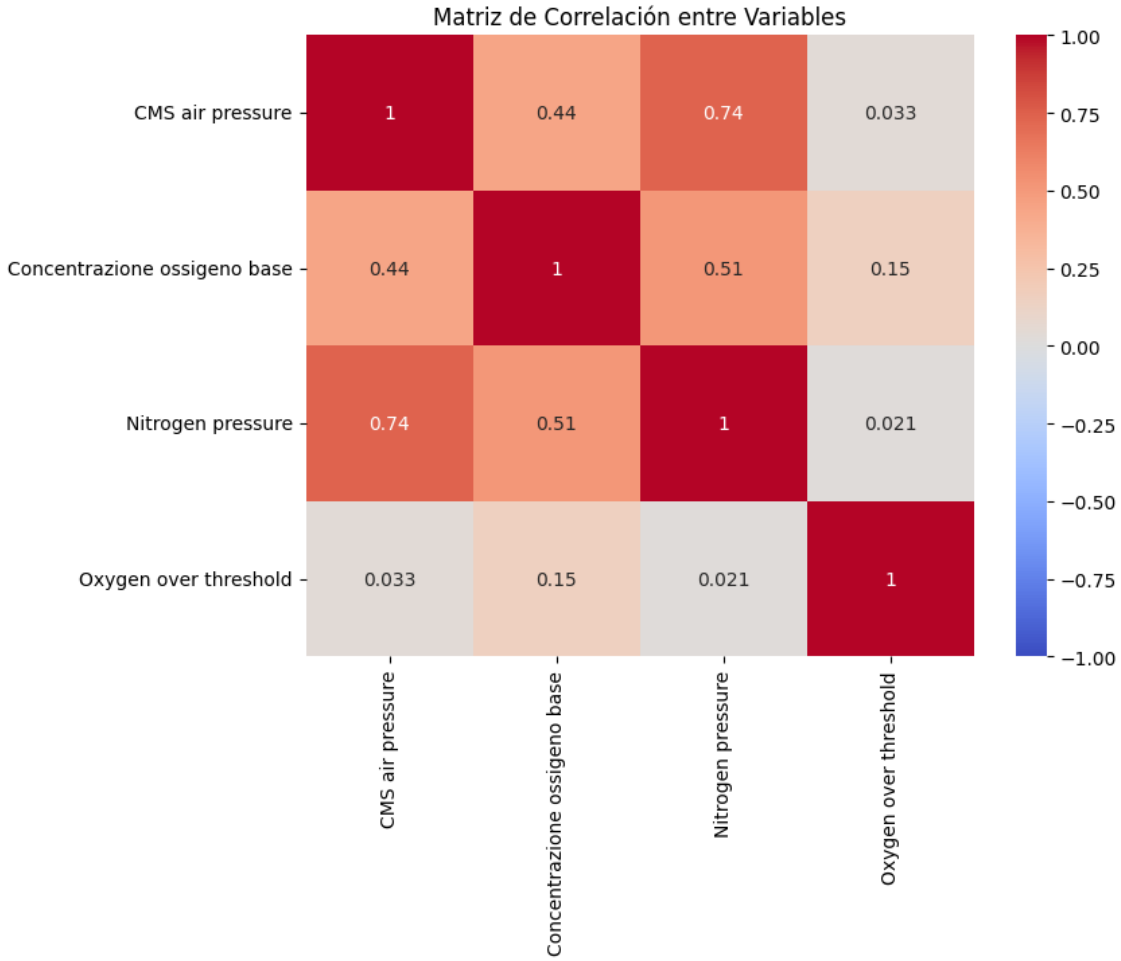
$$r_{xy} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}} \quad (2.4)$$

donde x_i y y_i son los valores de las dos variables en la observación i, $\bar{x}$ y $\bar{y}$ sus medias respectivas, y N el número total de observaciones.

La Figura 12 muestra correlaciones significativas (>0.4) entre CMS air pressure, Nitrogen pressure y Concentrazione ossigeno base, mientras que Oxygen over threshold presentó correlaciones cercanas a cero con las demás.

Figura 12

Matriz de correlación entre las variables



2.8.7 Selección del modelo predictivo

La elección del modelo Random Forest se justifica a partir de varios hallazgos del análisis exploratorio de datos. En primer lugar, algunas variables, como la presión de aire en el CMS y la presión de nitrógeno, mostraron una alta variabilidad y distribuciones complejas, con varianzas y entropías elevadas; en este contexto, la combinación de múltiples árboles de decisión que caracteriza a Random Forest contribuye a reducir el riesgo de sobreajuste y a mantener la

estabilidad frente a datos dispersos. En segundo lugar, se identificó una dependencia temporal de moderada a fuerte en distintas series, con autocorrelaciones persistentes; si bien no exige independencia estricta entre observaciones, Random Forest puede capturar dichos patrones temporales mediante la inclusión de retardos como atributos explicativos. Además, el análisis reveló correlaciones positivas entre la presión de aire en el CMS, la presión de nitrógeno y la concentración de oxígeno base, lo cual refuerza la pertinencia de un modelo capaz de representar interacciones no lineales y jerárquicas entre variables, como lo hace Random Forest. Finalmente, el modelo ofrece la ventaja de calcular la importancia relativa de cada variable, lo que no solo facilita la interpretación de los resultados, sino que también complementa el análisis exploratorio realizado. En conjunto, estas propiedades convierten a Random Forest en una alternativa sólida para este estudio, al combinar precisión predictiva, robustez frente al ruido y a la complejidad, y capacidad de modelar relaciones multivariadas en línea con los patrones detectados en el AED**.**

2.8.8 Preparación de datos para el entrenamiento

Para la preparación de los datos de series temporales, se implementó un enfoque basado en ventanas deslizantes, que permite transformar la serie original en un conjunto de datos estructurado para el entrenamiento del modelo (véase Figura 13 y Figura 14).

Cada muestra corresponde a:

- Una ventana de lectura (Reading Window, RW): contiene un histórico temporal de observaciones multivariadas.
- Una ventana de predicción (Prediction Window, PW): etiqueta determinada de acuerdo con la ocurrencia o no de un fallo.

Figura 13

Dataset de entrenamiento

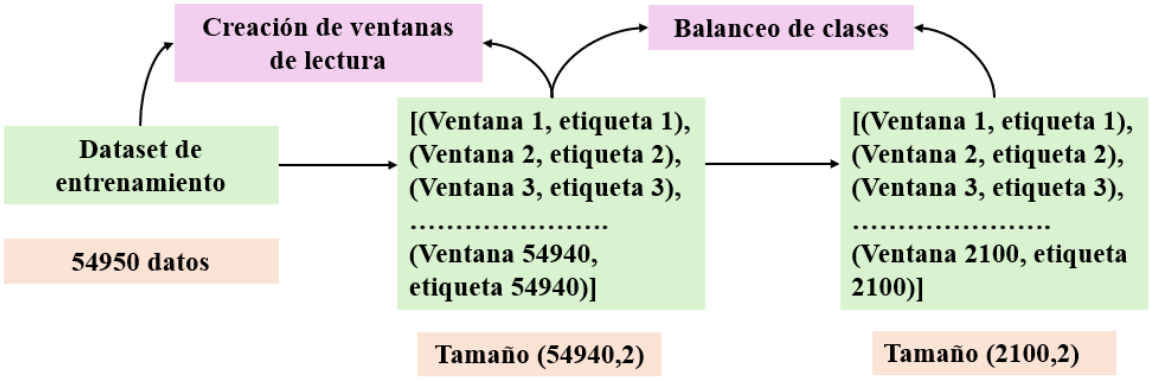


Figura 14

Ejemplo de una ventana de lectura de 10 minutos

```
=== Ventana de Entrenamiento ===
2023-08-01 03:07:00: CMS=-0.58, O2 Base=-0.99, N2=-1.39, O2 Thresh=-0.04
2023-08-01 03:08:00: CMS=-0.58, O2 Base=-0.99, N2=-1.39, O2 Thresh=-0.04
2023-08-01 03:09:00: CMS=-0.58, O2 Base=-0.99, N2=-1.39, O2 Thresh=-0.04
2023-08-01 03:10:00: CMS=-0.58, O2 Base=-0.99, N2=-1.39, O2 Thresh=-0.04
2023-08-01 03:11:00: CMS=-0.58, O2 Base=-0.99, N2=-1.39, O2 Thresh=-0.04
2023-08-01 03:12:00: CMS=-0.58, O2 Base=-0.99, N2=-1.39, O2 Thresh=-0.04
2023-08-01 03:13:00: CMS=-0.58, O2 Base=-0.99, N2=-1.39, O2 Thresh=-0.04
2023-08-01 03:14:00: CMS=-0.58, O2 Base=-0.99, N2=-1.39, O2 Thresh=-0.04
2023-08-01 03:15:00: CMS=-0.58, O2 Base=-0.99, N2=-1.39, O2 Thresh=-0.04
2023-08-01 03:16:00: CMS=-0.58, O2 Base=-0.99, N2=-1.39, O2 Thresh=-0.04
2023-08-01 03:17:00: CMS=-0.58, O2 Base=-0.99, N2=-1.39, O2 Thresh=-0.04

Etiqueta de la ventana: 0
```

Los datos multivariados fueron segmentados en ventanas temporales consecutivas, manteniendo la estructura secuencial y multicanal, lo que permite preservar las dependencias temporales y las relaciones entre variables.

Dentro de cada ventana, se realizó una separación explícita entre características y etiqueta:

- Características: corresponden a las variables independientes presentes en la RW.
- Etiqueta: se definió según la presencia o ausencia de falla (Fallo de presurización CMS) en la PW.

Este enfoque transforma cada ventana en un par ordenado de entrada-salida, representando un bloque histórico y su resultado futuro, facilitando al modelo la identificación de patrones temporales relevantes.

2.8.9 Estrategia de balanceo de clases

El conjunto de entrenamiento presentaba un marcado desbalance entre las clases, donde la clase de falla representaba una proporción significativamente menor comparada con la clase no falla. Este desbalance puede generar un sesgo en el entrenamiento del modelo, que tendería a predecir mayoritariamente la clase mayoritaria, disminuyendo la capacidad para detectar eventos críticos.

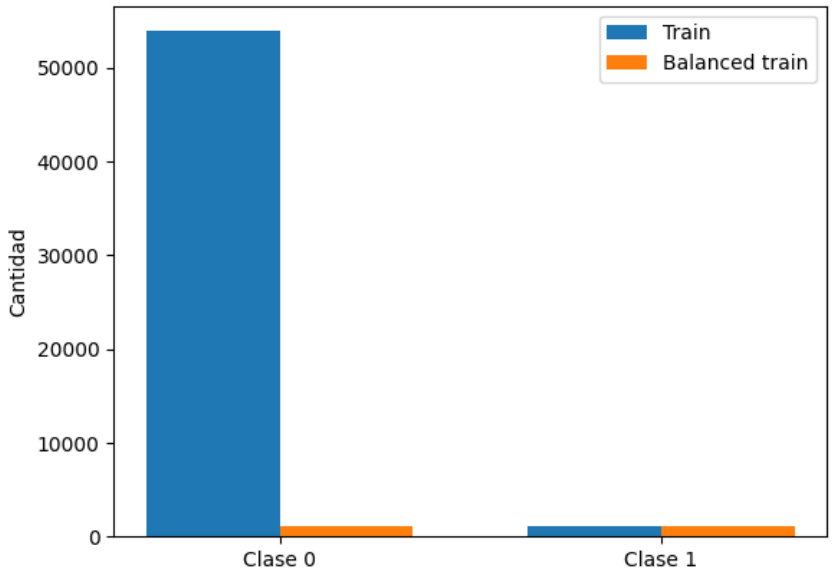
Para mitigar este efecto, se aplicó una técnica de muestreo aleatorio sin reemplazo conocida como random undersampling, cuyo procedimiento fue el siguiente:

1. Identificación de ventanas por clase: Se separaron todas las ventanas etiquetadas como clase minoritaria (falla) y clase mayoritaria (no falla).
2. Determinación de la cantidad mínima: Se estableció la cantidad mínima de ventanas entre ambas clases.
3. Selección aleatoria sin reemplazo: Se seleccionaron aleatoriamente, sin repetición, tantas ventanas como la cantidad mínima para cada clase.
4. Construcción del conjunto balanceado: La unión de estas muestras formó un conjunto con igual número de ventanas para ambas clases.
5. Aleatorización de muestras: Finalmente, se mezclaron aleatoriamente las ventanas para eliminar cualquier sesgo de orden en la presentación al modelo durante el entrenamiento.

Esta estrategia garantiza que el modelo aprenda a identificar patrones de ambas clases con igual importancia, mejorando la capacidad para detectar fallos sin sacrificar el desempeño global. En la Figura 15 se muestra el desbalance de clases antes y después de aplicar la técnica.

Figura 15

Comparación de las clases antes y después de la técnica



2.8.10 Representación de características para el modelo

Dado que Random Forest requiere vectores unidimensionales, cada ventana, originalmente una matriz bidimensional (tiempo × variables), se convirtió en un vector plano mediante la concatenación secuencial de sus elementos. Este procedimiento preserva la información temporal y multicanal, asegurando que el modelo reciba todos los patrones relevantes de la ventana histórica**.**

2.8.11 Entrenamiento del modelo

Se entrenó el clasificador Random Forest con el objetivo de modelar la ocurrencia de fallos en el sistema. Durante el proceso de ajuste se exploraron distintos valores de hiperparámetros clave. En particular, se evaluó el número de árboles en el bosque, considerando configuraciones

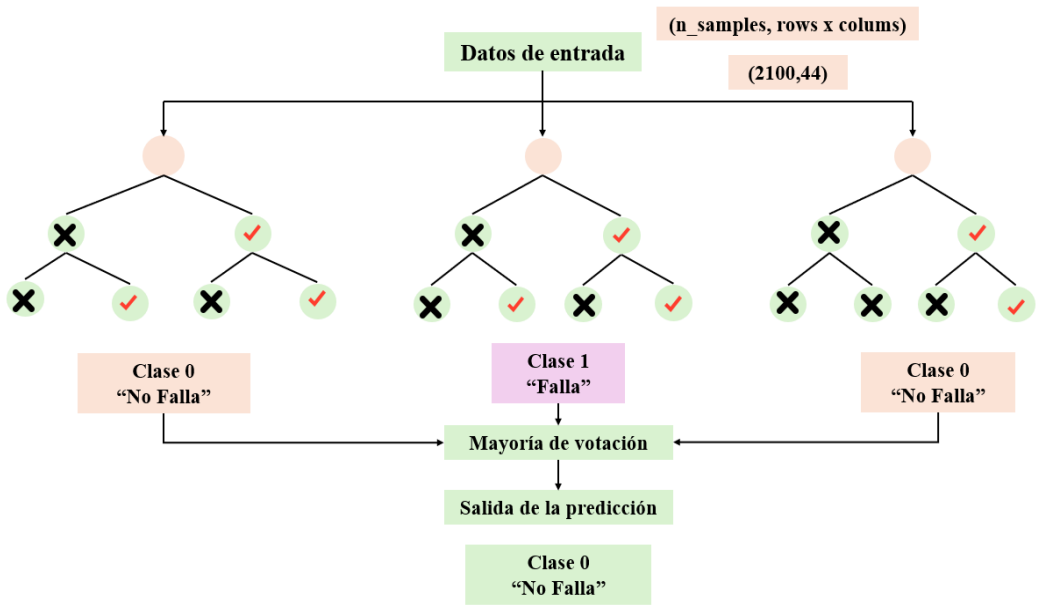
de 100, 150 y 200 árboles. Asimismo, se analizó el número máximo de variables utilizadas en cada división de nodo, probando las opciones {0.33, None, “sqrt”, “log2”}.

Adicionalmente, se incorporó un esquema de ponderación de clases con pesos inversamente proporcionales a la frecuencia de aparición de cada una. De esta manera, las muestras correspondientes a la clase minoritaria (falla) tuvieron mayor relevancia durante el entrenamiento, compensando posibles desequilibrios residuales en los datos y mejorando la sensibilidad del modelo hacia eventos críticos.

La lógica de funcionamiento del algoritmo se ilustra en la Figura 16, donde se muestra cómo múltiples árboles de decisión, entrenados con diferentes subconjuntos de variables y datos, generan predicciones parciales que luego se combinan mediante una regla de mayoría de votos. Este enfoque permite que Random Forest integre la diversidad de los árboles individuales para ofrecer una predicción más robusta y estable, reduciendo el riesgo de sobreajuste y mejorando la capacidad de generalización del modelo.

Figura 16

Entrenamiento modelo Random Forest



2.8.12 Validación y selección del mejor modelo

El desempeño de los modelos se evaluó sobre un conjunto de validación independiente, utilizando métricas estándar de clasificación:

- Precisión: proporción de verdaderos positivos sobre el total de predicciones positivas.
- Sensibilidad: proporción de verdaderos positivos sobre el total real de positivos.
- Exactitud: proporción de predicciones correctas sobre el total de muestras.
- Coeficiente F1 macro ponderado: media armónica de precisión y sensibilidad calculada para cada clase, ponderada según el tamaño relativo de cada clase.

El F1 macro ponderado se definió como criterio principal de selección por las siguientes razones:

1. Considera por igual ambas clases, evitando que la clase mayoritaria domine la evaluación.
2. Equilibra la compensación entre falsos positivos y falsos negativos, crucial en la predicción de fallos donde la detección correcta es tan importante como evitar falsas alarmas.
3. Proporciona una medida integral de desempeño, reflejando la capacidad real del modelo para identificar fallos y evitar predicciones erróneas.

Matemáticamente, el F1 score para cada clase i se define como:

$$\boldsymbol{F1_i = 2\frac{Precision_i x\ Sensibilidad_i}{Precision_i + \ Sensibilidad_i}} \tag{2.5}$$

Y el F1 macro ponderado se calcula como:

$$F1_{macro} = \sum_{i} w_i x F1_i \quad (2.6)$$

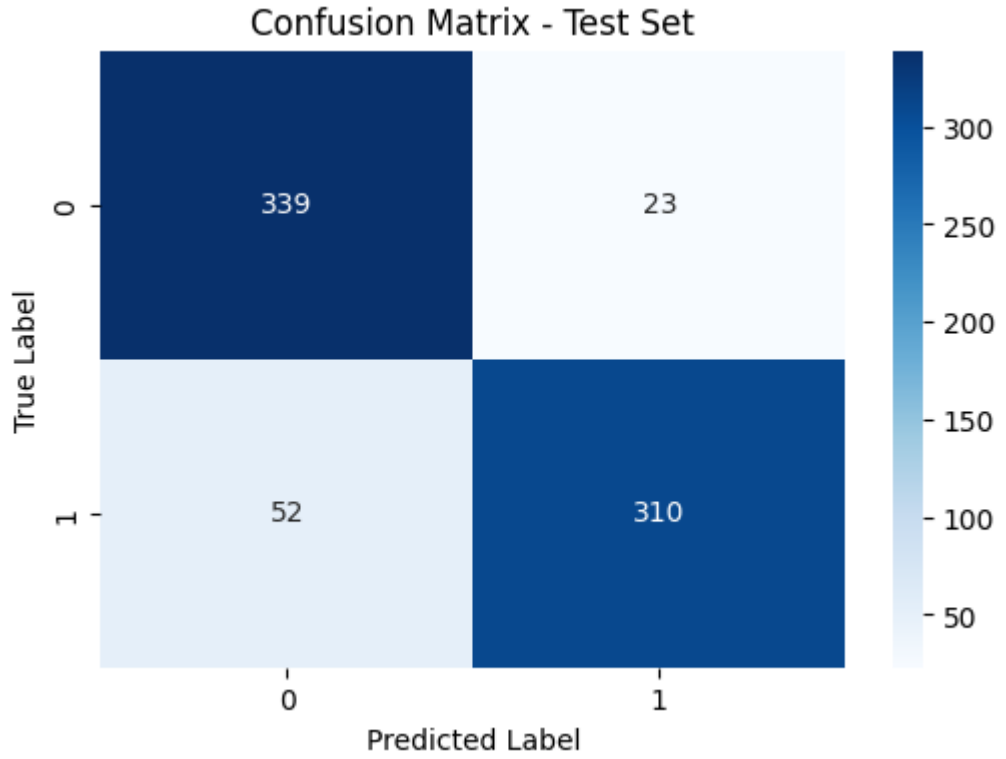
donde $w_i$es el peso relativo de la clase i basado en la proporción de muestras en el conjunto de validación.

2.8.13 Evaluación final

El modelo seleccionado fue evaluado utilizando un conjunto de prueba independiente, lo que permitió obtener una estimación imparcial de su rendimiento. Se calcularon todas las métricas estándar y se examinó la matriz de confusión (véase Figura 17), proporcionando un análisis detallado de la distribución de aciertos y errores por clase. Los resultados mostraron una exactitud del 89.6 %, mientras que la precisión alcanzó el 93.1 %, indicando que las predicciones positivas fueron correctas en la mayoría de los casos, minimizando falsos positivos. La sensibilidad fue de 89.6 %, evidenciando que el modelo identificó correctamente la mayoría de los casos positivos reales. El coeficiente F1 de 0.896 reflejó un equilibrio adecuado entre precisión y sensibilidad. La matriz de confusión presentó 339 verdaderos negativos, 310 verdaderos positivos, 23 falsos positivos y 52 falsos negativos, confirmando que el modelo mantiene un buen balance entre la capacidad de evitar errores. En conjunto, estos resultados indican que el modelo generaliza correctamente a datos no vistos, mostrando estabilidad y un desempeño confiable para la clasificación de nuevas instancias.

Figura 17

Matriz de confusión



Finalmente, el modelo óptimo fue serializado con pickle y almacenado para su posterior uso o despliegue, asegurando la reproducibilidad y la eficiencia en fases futuras del proyecto.

Capítulo 3

3. Resultados y análisis

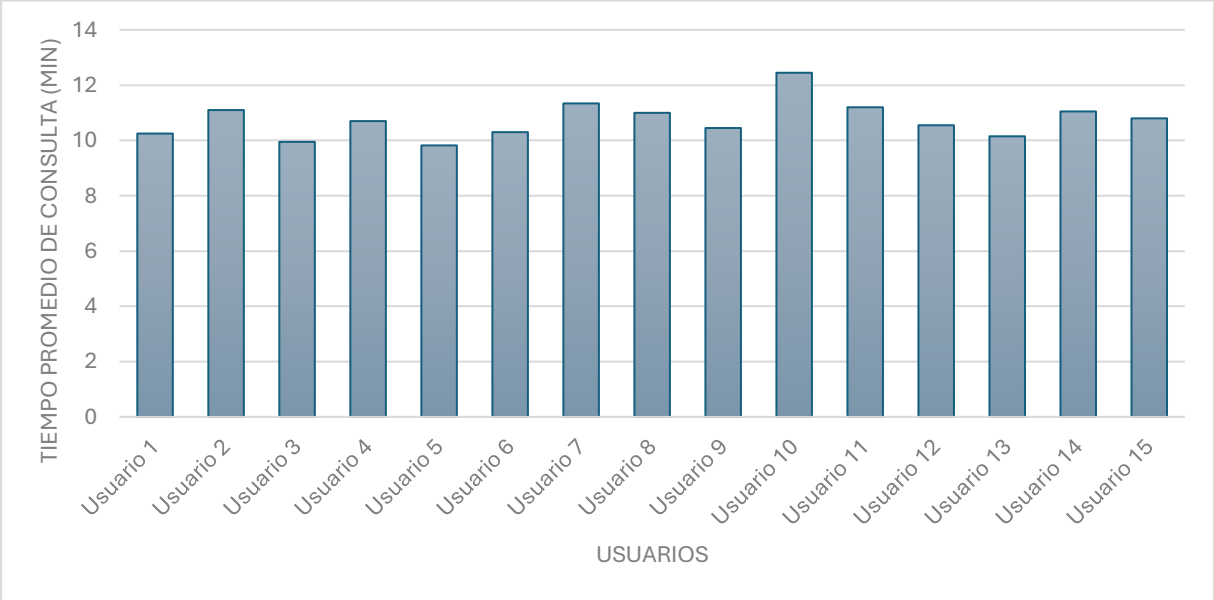
En esta sección se presentan los resultados obtenidos de la arquitectura de software diseñada para la consulta de indicadores clave de desempeño y la predicción de fallas de máquina. Cada subsección expone de manera detallada los resultados correspondientes, incluyendo gráficos estadísticos de la interacción usuario–interfaz, así como las métricas de evaluación del modelo predictivo y el análisis de costos.

3.1 Resultados de las pruebas de usabilidad y eficiencia del aplicativo web

Para evaluar el desempeño, la usabilidad e eficiencia del sistema, se realizó una prueba con 15 participantes, entre ellos técnicos, operadores, personal administrativo y de gerencia. Todos interactuaron con la herramienta para realizar una consulta o predicción. Se registró empíricamente el tiempo que cada usuario, sin conocimiento previo del sistema, necesitó para acceder, formular una consulta y obtener una respuesta. Esta medición permitió validar la funcionalidad de la herramienta en un contexto cercano al uso real. A continuación, se presenta una figura de barras con los resultados obtenidos.

Figura 18

Tiempo promedio de consulta por usuario



A partir de los resultados mostrados en la Figura 18, se observa que, en promedio, los usuarios tardaron 10.74 minutos en realizar una consulta o predicción sin conocimiento previo del sistema ni capacitación, recibiendo únicamente la instrucción de consultar el MTTR, MTBF o visualizar la predicción de fallas de la máquina. La variabilidad en los tiempos fue baja, con la mayoría de los usuarios completando la tarea entre 9 y 12 minutos, lo que indica una experiencia consistente y predecible. En contraste con el tiempo inicial estimado de aproximadamente 60 minutos por consulta, esto representa una reducción del 82.1% en el tiempo requerido para completar la tarea.

La retroalimentación recogida también revela que, aunque al principio algunos usuarios percibieron el sistema como complejo, rápidamente reconocieron su simplicidad y facilidad de uso gracias a un diseño simple que facilita realizar la tarea. En comparación con sus métodos actuales de búsqueda, los participantes destacaron que la herramienta resulta mucho menos tediosa y más eficiente, principalmente por el ahorro de tiempo y la eliminación de pasos adicionales innecesarios. Además, señalaron que la funcionalidad de predicción, inicialmente poco comprendida, resultó ser una ventaja valiosa. Tras interactuar con el sistema y analizar las respuestas, manifestaron que la herramienta puede integrarse perfectamente en sus rutas de inspección e incluso en sus inspecciones diarias, acelerando significativamente estos procesos y mejorando la planificación preventiva.

Con base en los resultados expuestos y la retroalimentación directa de los principales usuarios, se puede concluir que el sistema cumple efectivamente con su objetivo principal de reducir el tiempo de consulta. Asimismo, satisface los requerimientos planteados, ya que su interfaz simple e intuitiva permite que personas sin experiencia previa en el sistema puedan llevar a cabo consultas y predicciones de manera autónoma y eficiente.

Esta mejora significativa demuestra la eficiencia y usabilidad del sistema, permitiendo a los usuarios acceder, formular preguntas y obtener respuestas de forma rápida y sencilla. Así, se

acerca la experiencia al uso real, reduce la curva de aprendizaje y facilita la obtención eficiente de métricas clave, impactando directamente en la toma de decisiones basada en datos.

3.2 Evaluación del impacto de los hiperparámetros en Random Forest

Durante el entrenamiento con el modelo Random Forest se evaluaron diferentes configuraciones de los hiperparámetros número de estimadores y número máximo de características consideradas en cada división, con el fin de analizar su efecto en el desempeño del clasificador. Se probaron valores de número de estimadores de 100, 150 y 200 árboles, junto con tres estrategias de selección de variables: 0.33, sqrt y log2. Los resultados obtenidos, representados en la Figura 19, muestran que la configuración con número máximo de características de 0.33 alcanzó los valores más altos de la métrica F1 al pasar de 100 a 150 estimadores, aunque presentó una ligera disminución al llegar a 200 árboles.

En el caso de número máximo de características de log2, se observó un comportamiento constante entre 100 y 150 estimadores, con un aumento notable al alcanzar 200, logrando valores cercanos a 0.916. Por otro lado, la configuración con número máximo de características de sqrt presentó un desempeño relativamente estable entre 100 y 150 estimadores, con un incremento al llegar a 200, pero sin superar el rendimiento de la opción 0.33. Asimismo, se aprecia que el aumento en el número de estimadores no generó mejoras lineales en el valor de F1, indicando que incrementar el tamaño del ensamble no necesariamente se tradujo en un mejor balance entre precisión y sensibilidad.

Estos comportamientos se explican por la naturaleza de los hiperparámetros analizados. El hiperparámetro número de estimadores controla la cantidad de árboles que conforman el bosque. Al incrementar este valor, el modelo tiende a reducir su varianza y a producir predicciones más estables, aunque los beneficios adicionales se vuelven marginales a partir de cierto umbral, como se refleja en los resultados, donde pasar de 100 a 200 árboles no produjo mejoras sustanciales en

F1 para todas las configuraciones. Por otro lado, el hiperparámetro número máximo de características determina el número de variables candidatas para dividir cada nodo. Un valor relativamente alto como 0.33 incrementa la capacidad de los árboles individuales al disponer de un mayor conjunto de predictores, favoreciendo divisiones más informativas y un mejor ajuste local. En contraste, un valor reducido como log2 limita el acceso a variables, aumentando la diversidad entre árboles, pero a costa de una menor capacidad predictiva individual. La estrategia sqrt mostró un comportamiento intermedio, con estabilidad relativa, pero sin alcanzar el máximo rendimiento observado en 0.33.

En cuanto a las métricas, los efectos de estos hiperparámetros se reflejaron principalmente en la F1, que alcanzó su valor máximo de 0.918989 con la combinación óptima. Esta configuración correspondió a número de estimadores de 150 y número máximo de características de 0.33, logrando un desempeño robusto y equilibrado del modelo. Las demás métricas también reflejan un rendimiento destacado: exactitud de 0.91, precisión de 0.92, sensibilidad de 0.91 y área bajo la curva ROC 0.95, confirmando la capacidad del modelo para diferenciar entre clases y mantener un equilibrio adecuado entre falsos positivos y verdaderos positivos.

En este contexto, la mejor configuración identificada correspondió a número de estimadores de 150 y número máximo de características de 0.33, logrando el mayor valor de F1 entre todas las combinaciones evaluadas y demostrando un desempeño robusto, equilibrado y con alta capacidad discriminativa.

Figura 19

Comparación de parámetros

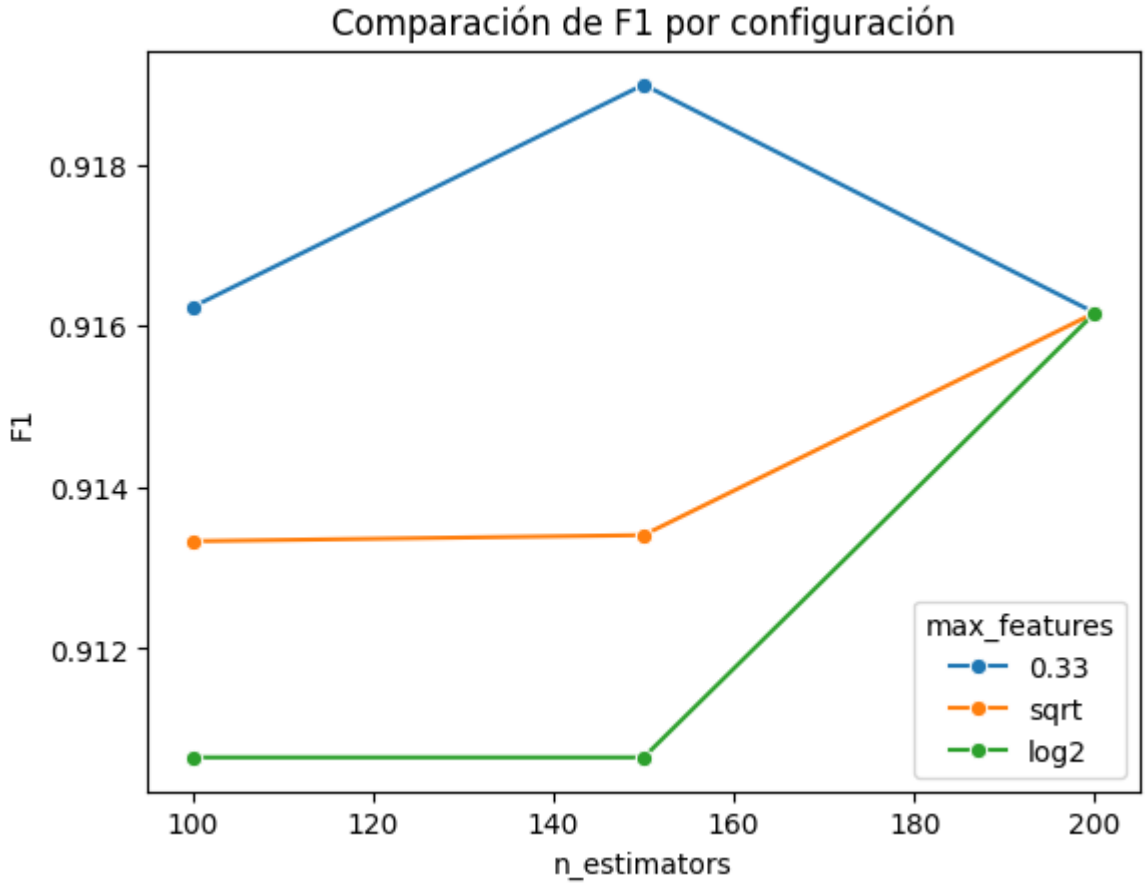
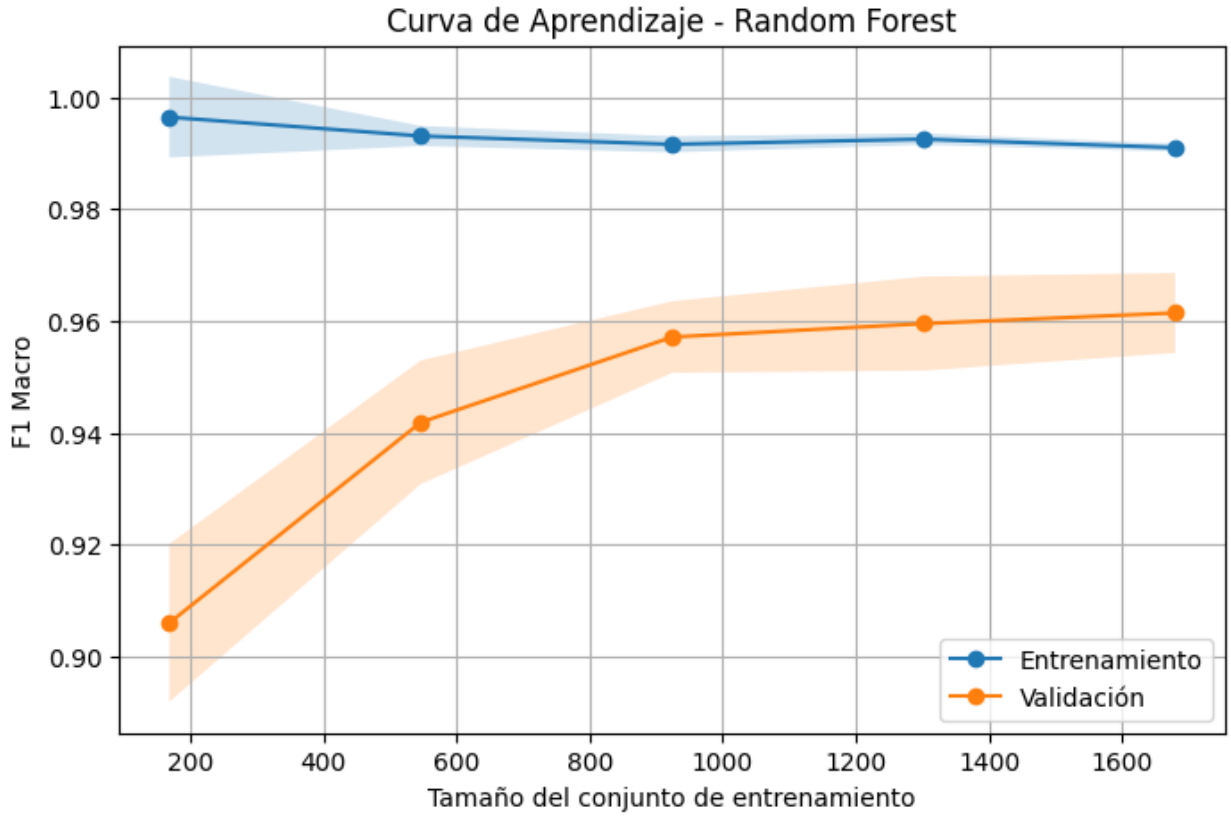


Figura 20

Curva de aprendizaje



La Figura 20 presenta la curva de aprendizaje del modelo Random Forest, mostrando la métrica F1 Macro en función del tamaño del conjunto de entrenamiento para los conjuntos de entrenamiento (línea azul) y validación (línea naranja). La curva de entrenamiento comienza en un valor muy alto, cercano a 0.996, y disminuye ligeramente hasta estabilizarse alrededor de 0.991 al incrementarse el tamaño del conjunto, lo que refleja un bajo sesgo, pero un alto sobreajuste inicial, típico cuando el modelo memoriza patrones debido a la escasez de datos. En contraste, la curva de validación inicia en aproximadamente 0.904 y aumenta progresivamente hasta cerca de 0.961, evidenciando que la varianza del modelo disminuye a medida que se incorporan más datos, mejorando su capacidad de generalización y reduciendo la brecha entre entrenamiento y validación. La banda sombreada alrededor de cada línea representa la dispersión del F1 Macro, mostrando que la variabilidad disminuye con conjuntos de entrenamiento más grandes, lo que

indica mayor estabilidad estadística y consistencia en el desempeño del modelo frente a distintos subconjuntos de datos.

3.3 Análisis de costos

La arquitectura de software desarrollada en el presente trabajo constituye una propuesta de código abierto, lo que permite a la empresa cliente y a otras organizaciones del mismo sector utilizarla, personalizarla y mejorarla de acuerdo con sus necesidades. No obstante, para lograr una implementación adecuada resulta indispensable contar con recursos humanos especializados y con equipos de cómputo que garanticen el correcto funcionamiento del sistema.

El cálculo de horas-hombre se estableció en 300 horas para el perfil FullStack y 425 horas para el perfil en inteligencia artificial, sumando un total de 725 horas de trabajo especializado. Esta distribución responde a la carga de trabajo inherente a cada rol. Mientras que el desarrollo web abarca principalmente la implementación de funcionalidades y el diseño de la interfaz, el trabajo en inteligencia artificial implica procesos más extensos y complejos, como la recolección y limpieza de datos, la experimentación con diferentes configuraciones de entrenamiento y la evaluación de desempeño del modelo, lo que justifica la mayor asignación horaria.

Para la valoración económica de estas horas se fijó una tarifa de 8,00 dólares por hora, tomando como referencia el monto promedio que perciben los perfiles de programadores junior en desarrollo web FullStack e inteligencia artificial. De este modo, la inversión estimada en recurso humano asciende a 2 400 dólares para el programador web FullStack y 3 400 dólares para el programador en inteligencia artificial, respectivamente.

En cuanto a los recursos materiales, se considera necesario contar con un servidor local o una computadora con características adecuadas para ejecutar y probar el sistema. En un escenario mínimo viable, se plantea el uso de un equipo con 16 GB de memoria RAM y sin unidades de procesamiento gráfico (GPU) dedicadas, cuyo costo estimado es de 900 dólares. Por otro lado, en

un escenario óptimo, se recomienda un equipo de similares características, pero con GPU dedicada, lo que elevaría el costo aproximado a 2 000 dólares, incrementando con ello la capacidad de procesamiento y reduciendo los tiempos de entrenamiento de los modelos de IA.

En la Tabla 6 se resumen los costos estimados tanto de recurso humano como de infraestructura computacional para la implementación del sistema.

Tabla 6

Inversión inicial para implementación de sistema

Descripción	Cantidad	Precio Total (USD)
Programador Web FullStack (300 h x $8/h)	1	$ 2400
Programador con conocimiento de Inteligencia Artificial (425 h x $8/h)	1	$ 3400
Servidor Local (16 GB RAM, sin GPU)	1	$ 883
Total (mínimo viable)		$ 6683
Servidor Local (16 GB RAM, con GPU dedicada)	1	$ 2000
Total (con GPU óptima)		$ 8683

De esta manera, la inversión total estimada para la implementación del sistema asciende a 6 683 dólares en el escenario mínimo viable, y a 8 683 dólares en el escenario óptimo con GPU dedicada. Estos valores representan únicamente los recursos básicos necesarios para el desarrollo y puesta en marcha del software, sin considerar otros gastos indirectos, dado que el enfoque principal del presente trabajo radica en el desarrollo de la arquitectura de software.

Partiendo de la inversión inicial presentada, se continúa con el modelo de negocio básico, donde el costo inicial o fijo corresponde a la instalación del software en la empresa, contemplando un valor aproximado de $740. Este monto incluye la personalización del sistema, como la integración de logotipos y elementos corporativos, así como la capacitación básica al personal encargado de operar la plataforma, siendo este un pago único.

Adicionalmente, se establece un costo por uso del sistema, que contempla el acceso al modelo predictivo entrenado y la utilización continua del software. El plan base considera un

máximo de 50 usuarios, con un valor mensual de $300, lo que permite la operación normal del sistema. Asimismo, se ofrece un servicio postventa, destinado a la resolución de problemas comunes y consultas técnicas relacionadas con el uso de la plataforma, garantizando así un soporte adecuado y la continuidad operativa del sistema.

Capítulo 4

4. Conclusiones y recomendaciones

El desarrollo del asistente virtual basado en inteligencia artificial, con capacidad para interpretar lenguaje natural, resultó satisfactorio al reducir significativamente los tiempos de consulta de KPIs y demostrar alta precisión en las predicciones de fallos de máquina. A continuación, se presentan las conclusiones y recomendaciones de este proyecto.

4.1 Conclusiones

En el contexto industrial actual, donde la eficiencia operativa y la disponibilidad de información en tiempo real son factores determinantes para la competitividad, se desarrolló un sistema web inteligente orientado a transformar la forma en que se consultan y analizan los indicadores clave de desempeño (KPIs) en planta. Este proyecto integró tecnologías de procesamiento de lenguaje natural, generación automática de consultas y modelos de machine learning para facilitar la toma de decisiones basada en datos confiables y actualizados. A continuación, se presentan las conclusiones más relevantes:

- Se logró desarrollar un aplicativo web con interfaz tipo chatbot capaz de comprender preguntas en lenguaje natural y generar consultas SQL para una base de datos PostgreSQL, integrando además un modelo de machine learning entrenado para procesamiento predictivo. Esta arquitectura permitió que el sistema devuelva respuestas claras y comprensibles para los usuarios, facilitando la interacción con los datos y la obtención de información relevante de manera eficiente. La integración de estas tecnologías evidencia la viabilidad de implementar soluciones de inteligencia artificial en aplicaciones web orientadas a la gestión de información y soporte a la toma de decisiones.
- El sistema mostró una mejora significativa en el tiempo de respuesta de las consultas, reduciendo el promedio de aproximadamente una hora, con métodos tradicionales, a cerca de 11 minutos por consulta mediante el aplicativo. Además, la retroalimentación de los

usuarios finales indicó que la herramienta es intuitiva y las respuestas generadas, incluso en casos de comparación o predicción de datos, son fácilmente comprendidas. Esto confirma que la plataforma no solo es técnicamente efectiva, sino que también proporciona una experiencia de usuario satisfactoria, cumpliendo con los objetivos de accesibilidad y eficiencia planteados inicialmente.

- El modelo Random Forest demostró capacidad para predecir fallos de la máquina con alta precisión y estabilidad en una ventana de predicción de 30 minutos (véase figura 21 y 22 en Apéndice). La configuración óptima de 150 estimadores y un número máximo de 0.33 características alcanzó un F1 de 0.896 y un ROC AUC de 0.93, mostrando un buen equilibrio entre precisión y sensibilidad con una sólida capacidad discriminativa, lo que respalda su uso para la toma de decisiones operativas.

4.2 Recomendaciones

El trabajo futuro en la arquitectura del aplicativo web y en el modelo predictivo abre la posibilidad de ampliar significativamente el alcance, la escalabilidad y la aplicabilidad del sistema en escenarios reales de gestión de datos e interacción con usuarios. Bajo esta perspectiva, se plantean las siguientes recomendaciones.

- Se sugiere incorporar funcionalidades para la generación automática de gráficos a partir de los datos procesados. Esto permitiría representar tendencias, comparaciones entre múltiples variables y predicciones de manera visual, facilitando la interpretación de los resultados por parte del usuario. La integración de gráficos en la interfaz enriquecería la experiencia de interacción y proporcionaría un apoyo visual de gran utilidad.
- Es aconsejable ejecutar el sistema en equipos que dispongan de GPU Aunque el modelo ha mostrado un desempeño aceptable utilizando únicamente con un servidor con memoria RAM, la aceleración mediante GPU puede mejorar notablemente los tiempos de inferencia

y la capacidad de manejar un mayor volumen de consultas en paralelo, lo que resulta esencial para aplicaciones a gran escala.

- Se recomienda extender las pruebas considerando otros sistemas de gestión de bases de datos además de PostgreSQL, incorporando otros sistemas de gestión de bases de datos como MySQL, SQLite o incluso soluciones NoSQL. Esto permitiría evaluar la portabilidad, compatibilidad y rendimiento del sistema en distintos entornos, lo cual es relevante dado que en aplicaciones reales se emplean diversas tecnologías de bases de datos según los requerimientos de cada organización.
- Se aconseja reentrenar periódicamente el modelo archivado en formato pickle con los nuevos datos recopilados, en lugar de entrenar un modelo desde cero. Esto permite mantener y mejorar la capacidad predictiva, aprovechar el aprendizaje previo del modelo y adaptarlo continuamente a cambios en las condiciones de operación de la máquina, garantizando predicciones precisas y confiables para la toma de decisiones operativas.

Referencias

[1] "Cuarta Revolución Industrial o Industria 4.0: concepto e historia," REPSOL. Accessed: May 24, 2025. [Online]. Available: https://www.repsol.com/es/energia-futuro/tecnologia-innovacion/cuarta-revolucion-industrial/index.cshtml

[2] J. Murcia Rodríguez and S. A. López Martínez, "Realidad aumentada: tecnología emergente para la formación empresarial," *European Public & Social Innovation Review*, vol. 9, pp. 1–18, Jul. 2024, doi: 10.31637/epsir-2024-402.

[3] "Cómo ayuda el análisis de datos en la toma de decisiones empresariales," *Gartner*, Accessed: May 24, 2025. [Online]. Available: https://www.gartner.es/es/tecnologia-de-la-informacion/insights/analisis-de-datos

[4] "Outperforming competitors as a data-driven organization," *MIT Technology Review*, Jan. 16, 2024. Accessed: May 24, 2025. [Online]. Available: https://www.technologyreview.com/2024/01/15/1086461/outperforming-competitors-as-a-data-driven-organization/

[5] Equipo de edición de Psico-smart, "Herramientas de análisis de datos para la optimización de procesos en empresas.," Vorecol. Accessed: May 24, 2025. [Online]. Available: https://psico-smart.com/articulos/articulo-herramientas-de-analisis-de-datos-para-la-optimizacion-de-procesos-en-empresas-168967

[6] "Indicadores clave de rendimiento (KPI): medición del éxito empresarial," Indicadores clave de rendimiento (KPI): medición del éxito empresarial. Accessed: May 24, 2025. [Online]. Available: https://payfit.com/es/contenido-practico/indicadores-clave-de-rendimiento/

[7] "Business Process Management Challenges And Solutions: A Comprehensive Guide: Osource," Osource Global. Accessed: May 24, 2025. [Online]. Available: https://osourceglobal.com/business-process-management-challenges-and-solutions/

[8] ITahora: La revista de Lider de Tecnologia, "La Fabril consolida su modelo de ciberresiliencia con CSIRT de última generación." Accessed: Jun. 14, 2025. [Online]. Available: https://itahora.com/2025/05/06/la-fabril-consolida-su-modelo-de-ciberresiliencia-con-csirt-de-ultima-generacion/

[9] "Big companies adopting Power BI," Bespoke.xyz. Accessed: Jun. 14, 2025. [Online]. Available: https://www.bespoke.xyz/big-companies-adopting-power-bi/

[10] Oyinloye Ibukunoluwa, “Nestlé’s Growth Path: Insights from Product Data with Power BI.” Accessed: Jun. 14, 2025. [Online]. Available: https://medium.com/@ibkoyinloye/nestl%C3%A9s-growth-path-insights-from-product-data-with-power-bi-2b197c37f2bf

[11] Stuart Kinsey, “How AI and Machine Learning Are Transforming KPI Tracking.” Accessed: Jun. 26, 2025. [Online]. Available: https://www.simplekpi.com/Blog/how-AI-and-machine-learning-are-transforming-kpi-tracking

[12] J. A. Quiceno Quiroz, Y. Echeverry Briceño, and B. S. Bedoya Montoya, “Centro de distribución cervecero: eficacia en tiempo real en Bavaria S.A en Girardota Antioquia,” Thesis, Corporación Universitaria Minuto de Dios, Girardota Antioquia, 2024. Accessed: Jun. 01, 2025. [Online]. Available: https://repository.uniminuto.edu/handle/10656/19590

[13] María Del Solar, “El impacto de la digitalización en la mejora continua,” Tecnologías ágiles y Cultura Lean se combinan para impulsar la eficiencia y la mejora continua a través de la digitalización industrial. Accessed: Jun. 10, 2025. [Online]. Available: https://zeotechnology.com/blog/digitalizacion-industrial-mejora-continua/

[14] R. Forradellas and S. Náñez Alonso, *IMPACTO DE LA DIGITALIZACIÓN EN LOS NUEVOS MODELOS DE NEGOCIO*. 2024.

[15] Amazon Ads, “Indicadores de rendimiento clave (KPI).” Accessed: Jun. 10, 2025. [Online]. Available: https://advertising.amazon.com/es-es/library/guides/key-performance-indicator

[16] Julia Martins, “Qué es un KPI, para qué sirve y cómo utilizarlo en tu proyecto.” Accessed: Jun. 10, 2025. [Online]. Available: https://asana.com/es/resources/key-performance-indicator-kpi

[17] ATLASSIAN, “MTBF, MTTR, MTTA, and MTTF.” Accessed: Jun. 24, 2025. [Online]. Available: https://www.atlassian.com/incident-management/kpis/common-metrics?utm_source=chatgpt.com

[18] Jacob Sawai Ben, “Implementation of Autonomous Maintenance and its Effect on MTBF, MTTR, and Reliability of a Critical Machine in a Beer Processing Plant,” *International Journal of Progressive Sciences and Technologies*, vol. 31, 2022, doi: 10.52155.

[19] eWorkOrders, “Understanding MTTR, MTBF, MTTF And Other Failure Metrics.” Accessed: Jun. 24, 2025. [Online]. Available: https://eworkorders.com/cmms-industry-articles-eworkorders/mttr-mtbf-mttf-metrics-explained/?utm_source=chatgpt.com

[20] IBM, “MTTR vs. MTBF: What’s the difference?” Accessed: Jun. 24, 2025. [Online]. Available: https://www.ibm.com/think/topics/mttr-vs-mtbf?utm_source=chatgpt.com

[21] Yague Kenny, Hernández Jairo, Trujillo Carlo, and Delgado Daniel, “Internet industrial de las cosas, evolución y desafíos.,” Trabajo de grado - Pregrado, Universidad Cooperativa de Colombia, 2020. Accessed: Jun. 08, 2025. [Online]. Available: https://repository.ucc.edu.co/entities/publication/95c71087-230c-4aae-b27d-c68708480286

[22] P. K. Misra, R. P. Melgarejo-Bolivar, C. I. Mendoza-Mollocondo, F. Torres-Cruz, M. E. Lourens, and S. K. Shukla, “Predicting Key Performance Indicators (KPI) in smart manufacturing through Machine Learning (ML),” 2024, p. 020022. doi: 10.1063/5.0218192.

[23] “GEA hace controlables los KPI de sostenibilidad de las cervecerías mediante IA - Se están llevando a cabo proyectos piloto para mejorar el rendimiento en conocidas fábricas de cerveza.” Accessed: Jun. 08, 2025. [Online]. Available: https://www.yumda.com/es/noticias/1181834/gea-hace-controlables-los-kpi-de-sostenibilidad-de-las-cervecerias-mediante-ia.html

[24] “La Revolución de la IA de Coca-Cola: Refinando Operaciones y Elevando Conexiones con Clientes - NextBrain AI | Aprendizaje Automático sin Código.” Accessed: Jun. 08, 2025. [Online]. Available: https://nextbrain.ai/es/blog/coca-colas-ai-revolution-refining-operations-and-elevating-customer-connections

[25] “Las 8 herramientas de análisis de datos que todo desarrollador debería conocer.” Accessed: Jun. 08, 2025. [Online]. Available: https://nexusintegra.io/es/8-herramientas-analisis-datos-todo-desarrollador-deberia-conocer/

[26] “Sistemas de ejecución de manufactura en la fabricación integrada por computador y prácticas de laboratorio de sistemas Scada.” Accessed: Jun. 08, 2025. [Online]. Available: https://repository.upb.edu.co/handle/20.500.11912/918

[27] L. A. Aimaretto, “Propuesta de algoritmo para mejorar la protección de enlaces en redes determinísticas con técnicas cross-layer,” p. 1, 2024, Accessed: Jun. 08, 2025. [Online]. Available: https://dialnet.unirioja.es/servlet/tesis?codigo=374725&info=resumen&idioma=ENG

[28] GEA, “OUR MISSION 30.” Accessed: Jun. 13, 2025. [Online]. Available: https://www.gea.com/en/company/about-us/our-strategy/

[29] GEA, “GEA launches real-time monitoring solution for food processing technology at Anuga FoodTec.” Accessed: Jun. 13, 2025. [Online]. Available: https://www.gea.com/es/news/trade-press/2024/gea-launches-insightpartner/

[30] YUMDA FOOD AND DRINK BUSINESS INFO, “GEA hace controlables los KPI de sostenibilidad de las cervecerías mediante IA.” Accessed: Jun. 13, 2025. [Online]. Available: https://www.yumda.com/es/noticias/1181834/gea-hace-controlables-los-kpi-de-sostenibilidad-de-las-cervecerias-mediante-ia.html

[31] AZLOGICA INTELLIGENCE OF THINGS, “LÍDERES EN DESARROLLO DE SOLUCIONES DE INTERNET DE LAS COSAS E INTELIGENCIA ARTIFICIAL.” Accessed: Jun. 13, 2025. [Online]. Available: https://azlogica.com/es/nosotros/

[32] AZLOGICA, “INDUSTRIA LATINOAMERICANA EN PRO DE LA ASISTENCIA VIRTUAL DE VOZ.”

[33] Ab. María Fernanda Delgado, “Conoce la Ley Orgánica de Protección de Datos (LOPDP).” Accessed: Jul. 18, 2025. [Online]. Available: https://smsecuador.ec/conoce-la-ley-organica-de-proteccion-de-datos/#gsc.tab=0

[34] CONGRESO NACIONAL, “LEY DE COMERCIO ELECTRONICO, FIRMAS Y MENSAJES DE DATO,” 2002.

[35] Dirección de Comunicación Social Servicio Ecuatoriano de Normalización – INEN, “Descubre la esencia de la seguridad de la información con la norma NTE INEN-ISO/IEC 27001.”

[36] “Ética de la inteligencia artificial.” Accessed: Jul. 18, 2025. [Online]. Available: https://www.unesco.org/es/artificial-intelligence/recommendation-ethics

[37] “ISO/IEC 25010.” Accessed: Jul. 18, 2025. [Online]. Available: https://iso25000.com/index.php/normas-iso-25000/iso-25010

[38] “Principios de Gobierno Corporativo de la OCDE y del G20 2023.” Accessed: Jul. 18, 2025. [Online]. Available: https://www.oecd.org/es/publications/2023/09/g20-oecd-principles-of-corporate-governance-2023_60836fcb.html

[39] N. O. Pinciroli Vago, F. Forbicini, and P. Fraternali, “Predicting Machine Failures from Multivariate Time Series: An Industrial Case Study,” *Machines*, vol. 12, no. 6, 2024, doi: 10.3390/machines12060357.

Apéndice

Figura 21

Comparación de etiquetas reales y predicciones del modelo en una ventana de predicción de 30 minutos normal

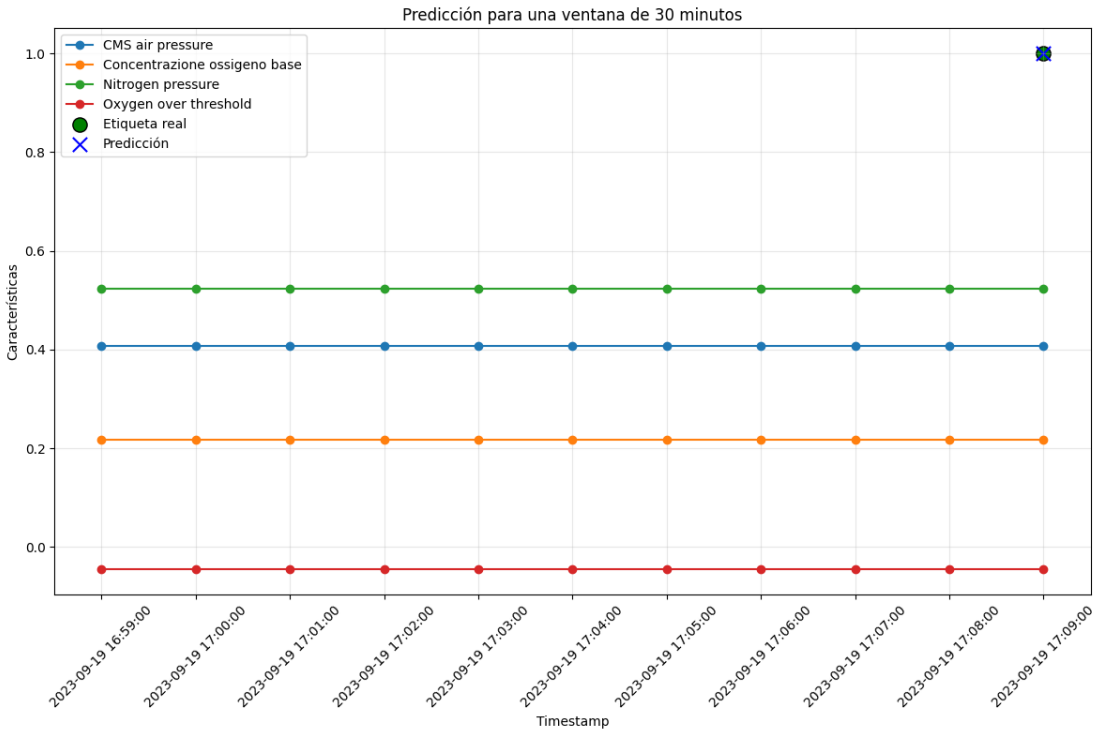


Figura 22

Comparación de etiquetas reales y predicciones del modelo en una ventana de predicción de 30 minutos en fallo