

**Prototipo de asistente operativo inteligente (DABU) integrado con herramientas SCADA
(Rockwell) para apoyo en la operación de estaciones de producción de crudo**

Andrés Felipe Dallos Ramírez, Daniel Alcides Buitrago Murillo

**Trabajo de grado para optar el título de Magíster en Análisis de Datos y Sistemas
Inteligentes**

Director

Ph.D. Rafael Augusto Núñez Rodríguez

Doctor en Ingeniería

Co-director

Omar Leonardo Peña Galvis

Magíster en Ingeniería Electrónica

Universidad Santo Tomás, Bucaramanga

División de Ingenierías y Arquitectura

Maestría en Análisis de Datos y Sistemas Inteligentes

2026

Contenido

1. Planteamiento del problema.....	13
1.1 Contexto del problema.....	13
1.2 Descripción clara y delimitada de la problemática	14
1.3 Justificación del proyecto	15
1.4 Objetivo general.....	17
1.5 Objetivos específicos	17
1.6 Alcance y delimitaciones del proyecto	17
2. Marco teórico y estado del arte	19
2.1 Conceptos teóricos relevantes.....	19
2.1.1 Sistemas de supervisión, control y adquisición de datos (SCADA).....	19
2.1.2 Sistemas de Control Distribuido (DCS) y Arquitectura PlantPAx	20
2.1.3 Convergencia IT/OT y arquitecturas de integración de datos	22
2.1.4 Agentes inteligentes y sistemas basados en reglas	22
2.1.5 Modelos de Lenguaje Grande (LLMs) en entornos industriales locales	23
2.2 Modelos, técnicas o enfoques existentes	23
2.2.1 Enfoques tradicionales de supervisión y sistemas expertos.....	23
2.2.2 Mantenimiento predictivo basado en Gemelos Digitales	24
2.2.3 Arquitecturas orientadas a LLM para la gestión de alarmas.....	24
2.3 Revisión de trabajos previos relacionados (Estado del Arte)	25

2.3.1 Analítica de datos y modelado de equipos en Oil & Gas	25
2.3.2 Diagnóstico mediante inteligencia artificial e IoT	26
2.3.3 Ciberseguridad y Arquitecturas de Confianza Cero (Zero Trust).....	26
2.3.4 Optimización de modelos de lenguaje (LLMs) en SCADA	27
2.4 Identificación de vacíos o limitaciones en la literatura.....	27
2.4.1 Falta de explicabilidad causal y triaje en tiempo real:	27
2.4.2 Riesgos de ciberseguridad por dependencia de la nube (Cloud AI):	28
2.4.3 Riesgo de alucinación en decisiones críticas:	28
2.4.4 Aporte de la solución propuesta (DABU).....	28
3. Metodología	29
3.1 Tipo y enfoque de la investigación	29
3.2 Descripción de las fases del proyecto	30
3.3 Fuentes de datos y criterios de selección	31
3.4 Herramientas, técnicas y tecnologías empleadas	33
3.5 Métodos de análisis y criterios de evaluación.....	34
4. Desarrollo de la solución	36
4.1 Arquitectura general de la solución	37
4.2 Diseño del modelo, sistema o metodología	39
4.2.1 Gestión estratégica de modelos de lenguaje, cuantización y limitaciones computacionales.....	41

4.3 Implementación técnica	43
4.3.1 Extracción de telemetría y análisis de límites operativos (Tag Reader)	43
4.3.2. Generación umentada por Recuperación (RAG) para Documentación Técnica	45
4.3.3. Procesamiento Dinámico de Curvas de Desempeño (Algoritmo Excel)	46
4.3.4. Ingeniería de Prompts y Alineación de Comportamiento (Behavior Alignment)	47
4.4 Diseño de experiencia de usuario (UX) e Interfaz del Prototipo Web	48
5. Resultados y validación	54
5.1 Resultados experimentales o de aplicación.....	55
5.1.1 Validación del sistema de recuperación documental (RAG)	55
5.1.2 Validación del motor de procesamiento histórico y tubería gráfica	56
5.1.3 Validación de telemetría en tiempo real y conciencia situacional	57
5.2 Métricas de desempeño y tiempos de respuesta (evaluación comparativa de entornos de ejecución)	58
5.3 Análisis e interpretación de resultados	60
5.4 Validación frente a los objetivos planteados	62
6. Conclusiones y recomendaciones	63
6.1 Conclusiones generales del proyecto	63
6.2 Cumplimiento de los objetivos	64

6.3 Limitaciones identificadas	66
6.4 Recomendaciones y trabajos futuros	67
Referencias.....	69
Apéndices.....	72
Apéndice A. Cuestionario de evaluación de usabilidad y guía de puntuación y uso para el investigador	72

Lista de Tablas

Tabla 1. <i>Estructura de servidores y distribución funcional de la arquitectura SCADA PlantPAx.</i>	
.....	37
Tabla 2. <i>Consolidación de Modelos de Lenguaje (LLMs), Entornos de Ejecución y Propósito Experimental.</i>	
.....	42
Tabla 3. <i>Métricas comparativas de latencia y velocidad de inferencia entre Hardware.</i>	60
Tabla 4. <i>Resultados consolidados de la evaluación de usabilidad y percepción de utilidad.</i>	63

Lista de Figuras

Figura 1. <i>Diagrama topológico de la arquitectura de red e interacción de los servidores CCC-PLCS hasta CCC-DABUCORE, detallando el flujo unidireccional de lectura de datos</i>	39
Figura 2. <i>Captura de pantalla de la interfaz web del asistente DABU.</i>	49
Figura 3. <i>Captura de pantalla del semáforo de indicación de comunicación con el PLC.</i>	51
Figura 4. <i>Captura de pantalla de la calculadora de conversiones de unidades.</i>	52
Figura 5. <i>Captura de pantalla de la barra de envío de mensajes.</i>	53
Figura 6. <i>Captura de pantalla de la ventana de configuraciones de modos.</i>	54
Figura 7. <i>Captura de pantalla de la interfaz web evidenciando la consulta de un manual PDF.</i>	56
Figura 8. <i>Captura de pantalla de la HMI web ilustrando la gráfica generada dinámicamente.</i>	57
Figura 9. <i>Captura de pantalla demostrando la alerta de límite superado.</i>	58

Resumen

La fragmentación de la información operativa en estaciones de producción de hidrocarburos representa un desafío crítico para la toma de decisiones oportuna en cuartos de control centralizado (CCR). Este trabajo de investigación presenta el diseño, desarrollo e implementación del prototipo DABU (Asistente Operativo Inteligente), una plataforma neuro-simbólica híbrida integrada con sistemas SCADA industriales bajo la arquitectura distribuida PlantPax de Rockwell Automation; el sistema se estructuró sobre un entorno virtualizado de seis capas funcionales alojadas en máquinas virtuales independientes (ISA-95), garantizando una topología de red aislada y alineada con los principios de seguridad de Confianza Cero (Zero Trust). El núcleo cognitivo de DABU combina la precisión de un motor determinista en Python que evalúa alarmas y límites de telemetría en tiempo real, junto con un pipeline de Generación Aumentada por Recuperación (RAG) sobre literatura técnica desestructurada (manuales PDF) e historiales de proceso (formatos Excel), utilizando la base de datos vectorial ChromaDB; la orquestación del flujo conversacional se construyó mediante grafos de estado en LangGraph, derivando la síntesis en lenguaje natural a un Modelo de Lenguaje de Gran Escala cuantizado (phi3:mini) ejecutado localmente sin dependencia de servicios en la nube; Las pruebas en entorno emulado demostraron la capacidad del prototipo para realizar triaje operativo, reducir la sobrecarga de consultas del operario y ofrecer respuestas técnicas fundamentadas con tasa de alucinación mitigada por diseño algorítmico, consolidando una arquitectura escalable y agnóstica al modelo fundacional.

Palabras clave: SCADA, IA híbrida, modelos de lenguaje de gran tamaño (LLM), generación aumentada por recuperación (RAG), PlantPax.

Abstract

The fragmentation of operational data in oil production facilities represents a critical bottleneck for timely decision-making within Centralized Control Rooms (CCR). This research presents the design, development, and implementation of DABU (Intelligent Operational Assistant), a hybrid neuro-symbolic platform integrated with industrial SCADA systems under Rockwell Automation's PlantPAX distributed architecture. The system is deployed across a six-layer virtualized environment (ISA-95) using independent virtual machines, enforcing a zero-trust network topology. DABU's cognitive engine combines the precision of a deterministic Python execution unit, which evaluates telemetry thresholds and alarms in real time, with a Retrieval-Augmented Generation (RAG) pipeline for unstructured technical manuals (PDFs) and historical logs (Excel sheets) indexed in ChromaDB. Conversation orchestration is governed by state graphs in LangGraph, routing language synthesis to a quantized local Large Language Model (phi3:mini) operating on-premise without reliance on external cloud APIs. Experimental validation in an emulated industrial environment confirmed DABU's ability to streamline operational triage and deliver context-grounded technical diagnostics. No hallucinations were observed in deterministic calculations during the evaluated trials, supporting the robustness of the proposed hybrid architecture and demonstrating its suitability as a resilient and model-agnostic solution for critical OT infrastructure.

Keywords: SCADA, Hybrid AI, Large Language Models (LLM), Retrieval-Augmented Generation (RAG), PlantPAX.

Glosario

CCR (Centralized Control Room): cuarto de control centralizado; infraestructura física equipada con consolas de supervisión HMI donde los operadores monitorean y controlan en tiempo real los procesos de la planta.

ChromaDB: base de datos vectorial de código abierto diseñada para almacenar incrustaciones (embeddings) y realizar búsquedas eficientes por similitud semántica en sistemas RAG.

Cuantización (Quantization): técnica de compresión de redes neuronales que reduce la precisión de los pesos del modelo (de punto flotante de 32 bits a formatos enteros de 4 u 8 bits), permitiendo la inferencia de LLMs en hardware con recursos computacionales restringidos sin degradar significativamente su capacidad de razonamiento.

DAG (Directed Acyclic Graph): grafo acíclico dirigido, estructura de datos lógica compuesta por nodos y aristas unidireccionales sin ciclos, empleada en LangGraph para orquestar los estados del agente conversacional.

DMZ (Demilitarized Zone): zona desmilitarizada, subred física o lógica que separa una red local de control interno de redes externas, actuando como filtro de seguridad en la arquitectura IT/OT.

Edge AI: inteligencia artificial en el borde, despliegue y ejecución de modelos de aprendizaje automático directamente en hardware local o servidores en planta (On-Premise), sin depender de servicios ni procesamiento en la nube.

Embeddings (Incrustaciones Vectoriales): representaciones numéricas continuas en un espacio multidimensional que convierten texto en vectores densos, capturando las relaciones semánticas entre palabras y conceptos.

HMI (Human-Machine Interface): interfaz hombre-máquina; pantalla o consola gráfica que permite a los operadores humanos interactuar con el sistema de control SCADA y supervisar el estado de los equipos.

ISA-95: estándar internacional emitido por la ISA (International Society of Automation) que define la jerarquía de integración entre las funciones de control de procesos (OT) y los sistemas empresariales (IT).

IT/OT (Information Technology / Operational Technology): tecnologías de la Información y Tecnologías de Operación, término que describe la convergencia entre los sistemas computacionales tradicionales de datos (IT) y los sistemas de monitoreo y control de procesos físicos e industriales (OT).

LangGraph: framework de orquestación de código abierto basado en langChain que permite construir aplicaciones de Inteligencia Artificial multi-agente y flujos conversacionales mediante grafos de estados cíclicos y acíclicos.

LLM (Large Language Model): modelo de lenguaje de gran escala; red neuronal profunda entrenada en volúmenes masivos de texto, capaz de procesar, comprender y generar lenguaje natural.

Ollama: motor de ejecución ligero y local diseñado para compilar, gestar y correr modelos de lenguaje cuantizados en servidores On-Premise sin dependencias externas.

PlantPAx: sistema de control distribuido (DCS) moderno desarrollado por Rockwell Automation, construido sobre la arquitectura de automatización integrada Logix y estándares de la industria.

PLC (Programmable Logic Controller): controlador lógico programable; computadora industrial diseñada para automatizar procesos electromecánicos, ejecutar lógicas de control

secuencial y adquirir telemetría de campo.

RAG (Retrieval-Augmented Generation): generación aumentada por recuperación; arquitectura que combina un sistema de búsqueda vectorial de información sobre documentos específicos con un LLM, garantizando que las respuestas generadas estén fundamentadas en fuentes de conocimiento verificables.

SCADA (Supervisory Control and Data Acquisition): sistema de control supervisorio y adquisición de datos, arquitectura de software y hardware que permite el monitoreo y control industrial remoto de procesos en tiempo real.

Tag SCADA: nombre o identificador sintáctico único asignado a un canal de datos, sensor o variable de proceso (ej. presión, temperatura, flujo) dentro de la base de datos del PLC o servidor SCADA.

Zero Trust: confianza cero; modelo de ciberseguridad industrial que asume que ninguna entidad o petición de red es confiable por defecto, exigiendo verificación estricta y aislamiento unidireccional en cada transacción.

1. Planteamiento del problema

La transformación digital en el sector industrial ha impulsado la adopción de tecnologías orientadas a la automatización, integración y análisis de datos operativos, en este contexto, los sistemas SCADA han adquirido un papel central en la supervisión y control de procesos en tiempo real, especialmente en entornos como las estaciones de producción de crudo, la creciente complejidad de estas infraestructuras ha generado nuevos desafíos relacionados con la integración de información y su interpretación eficiente, lo que plantea la necesidad de desarrollar soluciones basadas en analítica de datos y sistemas inteligentes que permitan mejorar la interacción con los datos operativos.

1.1 Contexto del problema

En el sector de hidrocarburos, las estaciones de producción de crudo operan mediante infraestructuras altamente automatizadas que integran sistemas de control, instrumentación y monitoreo en tiempo real, los sistemas SCADA (Supervisory Control and Data Acquisition) permiten supervisar variables críticas del proceso, como presión, caudal, temperatura y estado de equipos, facilitando la operación segura y continua de las instalaciones.

Las soluciones del ecosistema de Rockwell Automation, como FactoryTalk View SE y FactoryTalk Historian, son ampliamente utilizadas para la gestión de interfaces operativas, almacenamiento de datos históricos y análisis de tendencias, estas herramientas forman parte de arquitecturas industriales que han evolucionado hacia esquemas más distribuidos, en línea con los principios de la digitalización industrial y la integración de datos.

A medida que se incrementa el volumen y la diversidad de la información generada, también aumentan los retos asociados a su gestión, los datos operativos suelen estar distribuidos en múltiples plataformas y formatos, lo que dificulta su integración y limita la capacidad de realizar

análisis unificados, esta situación se presenta tanto en entornos industriales reales como en escenarios académicos, donde la disponibilidad de infraestructuras SCADA distribuidas es limitada, restringiendo el desarrollo de soluciones aplicadas basadas en analítica de datos.

En este contexto, la necesidad de integrar fuentes heterogéneas de información y facilitar su interpretación se convierte en un elemento clave para mejorar la eficiencia operativa y fortalecer el desarrollo de sistemas inteligentes aplicados a entornos industriales.

1.2 Descripción clara y delimitada de la problemática

En entornos de producción de crudo soportados por sistemas SCADA, la información operativa se genera de manera continua a partir de múltiples fuentes, incluyendo alarmas, tendencias, históricos y registros de eventos, en arquitecturas basadas en herramientas del ecosistema de Rockwell Automation, estos datos suelen almacenarse en plataformas independientes como FactoryTalk View SE, FT Historian y bases de datos SQL, sin un mecanismo unificado que permita su integración y análisis contextual.

Esta fragmentación de la información dificulta la correlación de eventos operativos, ya que los operadores deben consultar diferentes sistemas para reconstruir el comportamiento de un proceso o identificar la causa de una condición anómala; como consecuencia, el análisis depende en gran medida de la experiencia del personal, lo que incrementa la complejidad del diagnóstico y puede generar variabilidad en la interpretación de los datos.

La ausencia de herramientas que permitan traducir la información técnica en explicaciones comprensibles limita la capacidad de interacción con los sistemas, especialmente en escenarios donde se requiere una respuesta rápida ante eventos operativos, esta situación puede impactar los tiempos de diagnóstico y la toma de decisiones, afectando la eficiencia en la operación de las

estaciones de producción.

En el ámbito académico, esta problemática se ve acentuada por la limitada disponibilidad de entornos SCADA distribuidos que permitan experimentar con arquitecturas de integración de datos y el desarrollo de agentes inteligentes en condiciones controladas, esto restringe la validación de soluciones aplicadas que aborden la integración, análisis e interpretación de información industrial.

El problema se delimita en la ausencia de un mecanismo que permita integrar fuentes heterogéneas de datos SCADA y facilitar su interpretación de manera contextualizada, mediante herramientas que apoyen la comprensión de eventos operativos en estaciones de producción de crudo.

1.3 Justificación del proyecto

La creciente complejidad de los sistemas industriales y el incremento en la generación de datos operativos han evidenciado la necesidad de desarrollar soluciones que permitan integrar y analizar información de manera eficiente en entornos SCADA, en el sector de hidrocarburos, donde la continuidad operativa es un factor crítico, la capacidad de interpretar eventos en tiempo real y tomar decisiones informadas se convierte en un elemento determinante para la eficiencia y confiabilidad de los procesos. Informes del sector energético colombiano señalan que las interrupciones no programadas continúan representando un impacto relevante en la productividad, especialmente aquellas asociadas a fallas en equipos y sistemas de producción [1], [2].

Desde una perspectiva tecnológica, la evolución hacia arquitecturas distribuidas, la integración de sistemas industriales y el uso de analítica de datos han abierto la posibilidad de incorporar herramientas más avanzadas para el tratamiento de información operativa, enfoques

basados en integración mediante API y arquitecturas modulares han demostrado ser adecuados para la consolidación de datos provenientes de múltiples fuentes en entornos SCADA, el uso de modelos híbridos que combinan reglas y modelos de lenguaje ha permitido avanzar en la interpretación automatizada de datos industriales y en la generación de explicaciones en lenguaje natural [3]

El desarrollo del prototipo DABU se justifica como una respuesta a estas necesidades, al proponer una solución orientada a la integración, consulta e interpretación de información proveniente de sistemas SCADA, bajo un enfoque que combina analítica de datos y sistemas inteligentes, a diferencia de soluciones centralizadas o dependientes de la nube, el enfoque planteado se orienta a su implementación en entornos locales, permitiendo preservar la soberanía de los datos y facilitar su integración con infraestructuras existentes.

Desde el ámbito académico el proyecto contribuye al desarrollo de capacidades en analítica de datos aplicada a entornos industriales, así como a la exploración de arquitecturas que integran sistemas SCADA con agentes inteligentes, esto permite establecer un entorno experimental basado en una arquitectura distribuida que puede ser utilizado como base para futuras investigaciones en integración de datos, automatización industrial y sistemas inteligentes.

En conjunto, la propuesta aporta tanto a la solución de una problemática operativa real como al fortalecimiento de líneas de investigación en analítica de datos y sistemas inteligentes, consolidando un enfoque aplicado que articula la ingeniería, la automatización y la inteligencia computacional.

1.4 Objetivo general

Desarrollar un prototipo de asistente operativo inteligente (DABU) integrado con herramientas SCADA del ecosistema de Rockwell Automation, orientado a la integración, consulta e interpretación de eventos industriales en estaciones de producción de crudo, mediante una arquitectura modular y un agente inteligente explicable.

1.5 Objetivos específicos

- Establecer una arquitectura SCADA distribuida basada en Directorio PlantPAx que sirva como entorno habilitador para la integración y evaluación del asistente DABU.
- Diseñar una arquitectura modular basada en FastAPI que permita la integración de datos provenientes de herramientas SCADA del ecosistema Rockwell, particularmente FactoryTalk View SE y FactoryTalk Historian, en un entorno centralizado.
- Desarrollar un agente inteligente híbrido (reglas + LLM) capaz de interpretar los resultados de las consultas SCADA y generar explicaciones técnicas en lenguaje natural.
- Comprobar el desempeño funcional del prototipo DABU mediante métricas de latencia, coherencia interpretativa y percepción de utilidad en un entorno de simulación controlado.

1.6 Alcance y delimitaciones del proyecto

El proyecto se desarrolla bajo un enfoque de investigación aplicada, mediante la implementación del prototipo DABU en un entorno de simulación soportado en herramientas SCADA del ecosistema de Rockwell Automation, incluyendo FactoryTalk View SE, FactoryTalk Historian y un Directorio distribuido PlantPAx configurado como infraestructura habilitadora, en este contexto, se contempla el establecimiento de una arquitectura SCADA distribuida, el

desarrollo de un backend modular basado en FastAPI para la integración de datos provenientes de múltiples fuentes, y la implementación de un agente inteligente local basado en reglas y modelos de lenguaje, orientado a la consulta, interpretación y contextualización de eventos operativos.

La evaluación del sistema se realiza en un entorno de simulación que representa el comportamiento operativo de una bomba de inyección en estaciones de producción de crudo, considerando variables como presión de succión, presión de descarga, niveles de vibración y su relación con la curva operativa del equipo, para esto, se emplean datos históricos previamente registrados, con el fin de recrear escenarios operativos y eventos representativos que permitan analizar el desempeño del prototipo en condiciones controladas.

La validación del prototipo se enfoca en indicadores como la latencia de consulta, la coherencia interpretativa y la percepción de utilidad del sistema, el alcance del proyecto se limita a la evaluación funcional en un entorno simulado, sin contemplar su implementación en una planta industrial real ni su validación a escala operativa, por esto no se abordan soluciones basadas en herramientas genéricas de análisis, como hojas de cálculo o complementos externos, dado que estas presentan limitaciones para la integración estructurada de datos SCADA y pueden dar lugar a interpretaciones no contextualizadas, se plantea una arquitectura y una metodología de implementación que pueden ser replicadas y extendidas en futuros desarrollos o validaciones en campo.

Como delimitación tecnológica crítica, y en estricto apego a las normativas de ciberseguridad industrial y el principio de Confianza Cero (*Zero Trust*), el alcance de esta investigación excluye la integración de servicios de Inteligencia Artificial basados en la nube (*Cloud AI*), el prototipo DABU operará exclusivamente mediante modelos de lenguaje cuantizados de inferencia local (*Edge AI*), garantizando la absoluta soberanía, privacidad y confidencialidad

de la telemetría operativa proveniente de la estación de bombeo.

2. Marco teórico y estado del arte

En este capítulo se establecen los fundamentos conceptuales y tecnológicos que soportan el desarrollo del asistente operativo inteligente (DABU), en una primera instancia, se abordan los conceptos relacionados con las arquitecturas SCADA, la integración de datos industriales y los agentes inteligentes basados en modelos de lenguaje, seguido a esto, se realiza una revisión del estado del arte, analizando trabajos previos orientados a la implementación de inteligencia artificial en entornos de producción, con el propósito de identificar las limitaciones actuales y fundamentar la pertinencia de la arquitectura híbrida propuesta en este proyecto.

2.1 Conceptos teóricos relevantes

2.1.1 Sistemas de supervisión, control y adquisición de datos (SCADA)

En el contexto de la producción de hidrocarburos, la operación continua y segura de las estaciones de bombeo depende directamente de infraestructuras de control industrial donde los sistemas SCADA actúan como el núcleo central, dado que permiten no solo la supervisión en tiempo real sino también el control de procesos y la adquisición de datos de campo, de acuerdo con la pirámide de automatización definida por el estándar ISA-95 estos sistemas se ubican en el nivel de supervisión para consolidar la información proveniente tanto de los controladores lógicos programables (PLC) como de la instrumentación en sitio, en el caso específico del ecosistema de Rockwell Automation, la herramienta FactoryTalk View SE proporciona la interfaz operativa principal, mientras que FactoryTalk Historian centraliza el almacenamiento de series de tiempo correspondientes a variables críticas de proceso, tales como la presión de succión y los niveles de

vibración en los equipos rotativos [4].

2.1.2 Sistemas de Control Distribuido (DCS) y Arquitectura PlantPax

La supervisión de plantas industriales de alta complejidad, como las estaciones de producción de crudo, ha evidenciado las limitaciones de las arquitecturas convencionales basadas en Controladores Lógicos Programables (PLC) integrados de forma heterogénea con sistemas SCADA, en estas prácticas tradicionales, la base de datos de control reside de forma aislada respecto a la base de datos de visualización, requiriendo un mapeo manual de variables (tags) que incrementa la vulnerabilidad a errores de ingeniería y dificulta la trazabilidad analítica de los datos, para superar esta fragmentación la industria ha transitado hacia los Sistemas de Control Distribuido (DCS, por sus siglas en inglés), en cambio un DCS se distingue por proporcionar una plataforma unificada y orientada a objetos, donde los motores de control de procesos, la interfaz hombre-máquina (HMI), la historización de variables continuas y el subsistema de gestión de alarmas operan bajo un modelo de datos cohesivo y centralizado.

En el contexto de esta evolución tecnológica y los requerimientos de la Industria 4.0, Rockwell Automation estructuró la arquitectura de control distribuido PlantPax la cual fue desarrollada sobre los cimientos de la plataforma Integrated Architecture, PlantPax se clasifica como un DCS moderno (Modern DCS), su principal característica arquitectónica radica en la convergencia de disciplinas que históricamente operaban en silos: el control de procesos continuos, el control discreto, la gestión de potencia y la seguridad instrumentada convergen ahora sobre una única infraestructura de red Ethernet industrial estandarizada (EtherNet/IP), este fortalecimiento topológico es un habilitador indispensable para el despliegue de sistemas de Inteligencia Artificial, debido a que permite la extracción centralizada de telemetría sin interferir en los tiempos de barrido

(scan time) de los controladores físicos [5].

Para garantizar la homogeneidad en el tratamiento de los datos operativos, la arquitectura PlantPax se sustenta en la implementación obligatoria de la Librería de Objetos de Proceso (Rockwell Automation Library of Process Objects), a nivel de controlador (Logix), esta librería proporciona Instrucciones de Perfil Agregado (Add-On Instructions - AOI) que encapsulan la lógica matemática, los umbrales de alarma y las condiciones de interbloqueo de equipos industriales como motores, variadores de frecuencia, válvulas y lazos de control PID, si hablamos en un nivel de supervisión (FactoryTalk View), proporciona objetos gráficos preconfigurados (Faceplates) vinculados intrínsecamente a dichos AOI; esta estandarización algorítmica elimina la variabilidad en la codificación y asegura que la información historizada posea una semántica idéntica independientemente del equipo analizado, garantizando que el Agente Inteligente procese datos estructurados y libres de ambigüedades lógicas.

El rigor de diseño de un sistema distribuido como PlantPax exige adicionalmente el estricto cumplimiento de estándares internacionales de la Sociedad Internacional de Automatización (ISA) y la Comisión Electrotécnica Internacional (IEC), a nivel de gestión procedimental, el sistema integra objetos de secuenciación estructurados bajo la norma ISA-106 (*Procedure Automation for Continuous Process Operations*), permitiendo la automatización de transiciones de estado complejas e infrecuentes “tales como arranques en frío o paradas de emergencia” reduciendo drásticamente el error humano y manteniendo el proceso dentro de las curvas de operación segura [6]

Finalmente, dado que la integración de analítica avanzada requiere puentes entre la red de operación (OT) y los ecosistemas de tecnologías de la información (IT), la topología de red de PlantPax se rige bajo el estándar de ciberseguridad industrial ISA/IEC 62443 (*Industrial*

Communication Networks - Network and System Security), el lanzamiento de este DCS requiere el establecimiento de un modelo de zonas y conductos, segregando los servidores de directorio, controladores e historiadores mediante cortafuegos industriales (Firewalls), esta separación es la justificación técnica que permite instaurar una Zona Desmilitarizada (DMZ) donde se aloja el entorno computacional de análisis de datos e Inteligencia Artificial, asegurando el cumplimiento del principio de Confianza Cero (Zero Trust) y protegiendo la integridad del sistema de control frente a vulnerabilidades externas [7].

2.1.3 Convergencia IT/OT y arquitecturas de integración de datos

La transición tecnológica hacia la Industria 4.0 ha evidenciado la necesidad de romper los silos de información que históricamente han separado las tecnologías de la operación (OT) de las tecnologías de la información (IT), los datos de alarmas solían permanecer confinados exclusivamente dentro de las redes de planta, para resolver esto y extraer valor de negocio, la convergencia IT/OT propone la implementación de arquitecturas conectadas [4], [7], es muy importante establecer modelos que equilibren la conectividad industrial con la exposición de los activos físicos, bajo este planteamiento, el diseño de servicios web modulares mediante interfaces de programación de aplicaciones (API) y frameworks de alto rendimiento como FastAPI resulta ideal, ya que permite extraer y normalizar los registros del historiador industrial sin comprometer la latencia ni la seguridad de la red subyacente.

2.1.4 Agentes inteligentes y sistemas basados en reglas

La interpretación automatizada de esta gran volumetría de datos operativos exige mecanismos computacionales capaces de procesar información estructurada, un agente inteligente

opera bajo el paradigma de percepción y acción, interactuando de forma autónoma con su entorno de datos para resolver problemas de un dominio específico [9], aplicado al entorno industrial de este proyecto, se plantea un enfoque donde un motor de inferencia determinista basado en reglas lógicas establece las condiciones de operación segura a partir de los registros históricos, evaluando factores como los límites de vibración frente a la curva teórica de la bomba, esta lógica deductiva clásica es indispensable para garantizar el rigor matemático que exige un equipo rotativo.

2.1.5 Modelos de Lenguaje Grande (LLMs) en entornos industriales locales

Aunque la adopción de Modelos de Lenguaje Grande (LLMs) representa una ventaja analítica enorme, su uso en el sector de hidrocarburos conlleva el desafío de garantizar la soberanía de la información operativa, para evadir las latencias de red y los riesgos de ciberseguridad, se vuelve imperativo el despliegue de modelos cuantizados de ejecución local (Edge AI), el uso de modelos compactos como phi3:mini brinda las capacidades semánticas requeridas para traducir el diagnóstico técnico arrojado por el motor de reglas hacia un lenguaje natural comprensible por el operador, logrando este objetivo sin desplegar infraestructuras costosas ni exponer la red industrial a conexiones de internet.

2.2 Modelos, técnicas o enfoques existentes

2.2.1 Enfoques tradicionales de supervisión y sistemas expertos

Históricamente, la operación y el diagnóstico en estaciones de producción industrial se han fundamentado en la supervisión visual a través de interfaces hombre-máquina (HMI). Cuando ocurre una desviación en una variable crítica, el sistema se limita a generar una alerta visual,

delegando al operador la responsabilidad de correlacionar la información fragmentada, para asistir en esta tarea, los primeros avances tecnológicos se centraron en el desarrollo de "sistemas expertos" que capturaban el conocimiento humano en bases de reglas fijas, los enfoques tradicionales de supervisión y sistemas expertos han sido ampliamente utilizados para la gestión operativa; no obstante, presentan la desventaja de ser altamente rígidos y de requerir reprogramación manual cada vez que cambian las condiciones de frontera de la planta [8]

2.2.2 Mantenimiento predictivo basado en Gemelos Digitales

Con el propósito de mitigar las limitaciones de los sistemas expertos convencionales, la industria evolucionó hacia la implementación de arquitecturas predictivas. Una de las técnicas más predominantes en la actualidad es la implementación de Gemelos Digitales (Digital Twins), concebidos como representaciones virtuales de alta fidelidad; los Gemelos Digitales basados en plataformas IoT sirven como base principal para el monitoreo inteligente y la optimización de activos físicos, a pesar de su alta precisión estadística, muchos de estos modelos matemáticos continúan operando como sistemas de "caja negra" que emiten alertas precisas, pero que carecen de mecanismos semánticos para explicarle al operador el razonamiento técnico detrás de dicha alerta [15], [16].

2.2.3 Arquitecturas orientadas a LLM para la gestión de alarmas

Para superar la barrera de la explicabilidad, el enfoque más vanguardista propone la integración directa de LLMs dentro de los sistemas SCADA, investigaron el potencial de los Modelos de Lenguaje para mejorar específicamente la gestión de alarmas en infraestructuras de misión crítica (como las redes eléctricas), los autores demostraron que un LLM debidamente

integrado puede procesar avalanchas de alertas y proporcionar un resumen causal altamente digerible para el operador en cuestión de segundos, al combinar la confiabilidad de las reglas deterministas con el análisis semántico de un LLM, se consolida un asistente capaz de traducir la telemetría cruda en lenguaje natural, siendo este el paradigma metodológico seleccionado para el desarrollo del asistente DABU [12], [13].

2.3 Revisión de trabajos previos relacionados (Estado del Arte)

El desarrollo de asistentes operativos en la Industria 4.0 se fundamenta en investigaciones recientes sobre analítica en equipos rotativos, ciberseguridad industrial IT/OT, gestión de alarmas y el despliegue de Modelos de Lenguaje de Gran Escala (LLM), a continuación, se presenta la revisión sistemática de literatura que enmarca y justifica el desarrollo del prototipo DABU.

2.3.1 Analítica de datos y modelado de equipos en Oil & Gas

El monitoreo de la integridad operativa en estaciones de hidrocarburos exige el análisis riguroso de telemetría e historiales de proceso en la detección de patrones anómalos sobre redes SCADA, Nazir et al. [3] desarrollaron modelos de detección de anomalías basados en arquitecturas de autocodificadores (Autoencoders), demostrando la eficacia del aprendizaje profundo para identificar desviaciones en datos de sensores. Por su parte, en el ámbito de la gestión de eventos de planta, Ramsay et al. [8] priorizaron la estructuración orientada a objetos para la administración y filtrado de alarmas operativas. Estos enfoques resaltan la importancia crítica de establecer una "curva base" de normalidad matemática antes de aplicar cualquier análisis superior, metodología que se aplica directamente en DABU al comparar la telemetría en tiempo real frente a la curva teórica de diseño de la bomba de inyección.

2.3.2 Diagnóstico mediante inteligencia artificial e IoT

Para estructurar diagnósticos industriales confiables sin saturar al operador, Vásquez Capacho et al. [19] propusieron un marco de gestión de alarmas basado en el aprendizaje de patrones temporales recurrentes, permitiendo identificar la secuencia cronológica de fallas en procesos continuos, complementariamente, en la dimensión del mantenimiento predictivo, Rao [15] y Dagur et al. [16] validaron la implementación de Gemelos Digitales (Digital Twins) combinados con aprendizaje automático para el pronóstico de fallas en equipos industriales, ambos estudios fundamentan la validez metodológica de utilizar entornos de simulación y emulación (FactoryTalk View y controladores lógicos) para probar y validar prototipos analíticos inteligentes de forma segura antes de su despliegue en plantas reales.

2.3.3 Ciberseguridad y Arquitecturas de Confianza Cero (Zero Trust)

La convergencia entre las tecnologías de operación (OT) y las tecnologías de información (IT) exige la adopción de esquemas avanzados de protección digital. Geol [10] y Mosharraf [11] analizaron marcos de ciberseguridad basados en el principio de Confianza Cero (Zero Trust) y técnicas de aprendizaje automático para la detección de intrusiones en redes de infraestructura crítica. Específicamente en el sector de los hidrocarburos, Purohit [14] evaluó los riesgos de ciberseguridad en la integración IT/OT en instalaciones de Oil & Gas downstream, concluyendo que la validación continua, la segmentación de red y la ejecución local de servicios analíticos son pilares indispensables para garantizar la resiliencia operativa. Estos hallazgos sustentan la decisión arquitectónica de este proyecto de no exponer las bases de datos SCADA hacia la nube pública y ejecutar el backend de DABU de manera puramente local (Edge AI).

2.3.4 Optimización de modelos de lenguaje (LLMs) en SCADA

Abordando el componente de inteligencia semántica, Xia et al. [17] exploraron la aplicación de Modelos de Lenguaje de Gran Escala en la automatización industrial, demostrando que la interfaz en lenguaje natural facilita la interpretación de estados de planta. Para resolver el control de procesos complejos, Vyas y Mercangöz [18] diseñaron un marco de control industrial autónomo basado en agentes inteligentes impulsados por LLMs. Sin embargo, para garantizar la ciberseguridad y la precisión operativa, Schall et al. [12] demostraron que la integración segura de LLMs en el control de procesos exige arquitecturas multi-agente con validación determinista basada en diagramas P&ID. Finalmente, Majumder et al. [13] analizaron las capacidades y limitaciones de los LLMs en el sector energético, advirtiendo sobre el riesgo latente de las alucinaciones probabilísticas cuando los modelos generan respuestas sin contar con reglas deterministas de respaldo.

2.4 Identificación de vacíos o limitaciones en la literatura

Pese a los formidables avances documentados en la detección de anomalías [3], la gestión de alarmas [8], [19] y la aplicación de LLMs en entornos industriales [17], [18], la literatura científica evidencia vacíos metodológicos y prácticos al intentar trasladar estas tecnologías a una estación de producción de crudo en operación real:

2.4.1 Falta de explicabilidad causal y triaje en tiempo real:

La mayoría de los modelos predictivos tradicionales [3], [15] actúan como "cajas negras" que alertan sobre una falla inminente, pero no ofrecen al operador del cuarto de control (CCR) un

diagnóstico causal explicable en lenguaje natural sobre la causa raíz del evento.

2.4.2 Riesgos de ciberseguridad por dependencia de la nube (Cloud AI):

Gran parte de las soluciones conversacionales recientes dependen de APIs en la nube comercial, lo que entra en conflicto directo con los marcos de ciberseguridad Zero Trust exigidos en el sector hidrocarburos [10], [14] debido al riesgo de exfiltración de telemetría sensible.

2.4.3 Riesgo de alucinación en decisiones críticas:

El uso ingenuo de modelos lingüísticos [13] sin mecanismos de validación rígidos expone a la planta a alucinaciones numéricas donde el modelo puede inventar lecturas de presión o temperatura, comprometiendo la seguridad física de las instalaciones.

2.4.4 Aporte de la solución propuesta (DABU)

Es exactamente en este cruce de vacíos donde el prototipo DABU encuentra su justificación e impacto científico. El proyecto responde a estas limitaciones mediante una arquitectura híbrida neuro-simbólica que opera 100% On-Premise (Edge AI) bajo principios Zero Trust [14]. Al desacoplar la inferencia matemática (ejecutada por herramientas deterministas en Python sobre la curva teórica de la bomba) de la síntesis lingüística (ejecutada por phi3:mini/LangGraph), DABU reduce el riesgo de alucinación en los datos de proceso mediante la separación entre el procesamiento determinista de la telemetría y la síntesis lingüística realizada por el LLM, proporcionando al operador del CCR un triaje asistido explicable, seguro y en tiempo real.

3. Metodología

3.1 Tipo y enfoque de la investigación

La presente investigación se desarrolla bajo un enfoque de investigación aplicada, debido a que busca diseñar, implementar y comprobar funcionalmente un prototipo tecnológico orientado a resolver una necesidad específica en entornos industriales: la consulta, integración e interpretación de información operativa proveniente de sistemas SCADA; el proyecto no se limita a una revisión conceptual sobre inteligencia artificial o automatización industrial, sino que propone una solución funcional denominada DABU, integrada con herramientas del ecosistema Rockwell para apoyar el análisis de eventos en estaciones de producción de crudo.

El proyecto se considera aplicado porque parte de una problemática concreta asociada a la fragmentación de datos en plataformas industriales como FactoryTalk View SE, FactoryTalk Historian y otros sistemas de almacenamiento o visualización operativa, esta fragmentación dificulta que operadores, instrumentistas o ingenieros puedan consultar y contextualizar eventos de manera rápida, especialmente cuando la información se encuentra distribuida entre alarmas, tendencias, históricos y pantallas SCADA.

Asimismo, la investigación tiene un componente experimental y tecnológico, dado que contempla la construcción de un entorno de simulación controlado, el desarrollo de una arquitectura modular basada en FastAPI y la implementación de un agente inteligente híbrido compuesto por reglas y un modelo de lenguaje local, este componente experimental permite ejecutar pruebas funcionales sobre el prototipo, medir su comportamiento y evaluar su utilidad en escenarios representativos de operación industrial.

El enfoque metodológico se orienta a la comprobación funcional del sistema, no a una

validación industrial en planta real, el alcance se centra en demostrar que DABU puede integrarse con fuentes SCADA, consultar información histórica, aplicar reglas básicas de interpretación y generar respuestas técnicas en lenguaje natural dentro de un ambiente controlado. Esta delimitación es coherente con el alcance aprobado del proyecto, el cual establece que la solución no contempla despliegue operativo en campo, sino una arquitectura replicable para futuras pruebas o desarrollos posteriores.

3.2 Descripción de las fases del proyecto

El desarrollo metodológico se organiza en fases secuenciales, alineadas con los objetivos específicos del proyecto. Estas fases permiten avanzar desde la configuración de la infraestructura SCADA hasta la comprobación funcional del prototipo DABU.

La primera fase corresponde al establecimiento de la arquitectura SCADA distribuida basada en Directorio PlantPAX. Esta fase tiene como propósito construir el entorno habilitador sobre el cual se integrará el asistente operativo inteligente, se emplea un ambiente virtualizado compuesto por servidores, clientes SCADA, simulación de PLCs, servicios de historización y componentes asociados al ecosistema Rockwell. Esta arquitectura permite representar de forma controlada una infraestructura industrial distribuida, facilitando la integración y evaluación del prototipo.

La segunda fase consiste en el diseño de la arquitectura modular de integración, en esta etapa se definen los componentes lógicos de DABU, los flujos de información, los mecanismos de comunicación y las responsabilidades de cada módulo. La arquitectura se plantea con un backend basado en FastAPI, encargado de recibir solicitudes, gestionar consultas, conectar con fuentes de datos industriales, aplicar reglas y entregar contexto al modelo de lenguaje.

La tercera fase corresponde a la implementación del backend modular, en esta fase se desarrollan los servicios necesarios para recibir preguntas desde el cliente web, interpretar parámetros de consulta, resolver nombres de variables o tags, conectarse con FactoryTalk Historian y devolver resultados estructurados. Este componente actúa como intermediario entre el usuario, las fuentes SCADA y el agente inteligente.

La cuarta fase comprende el desarrollo del agente híbrido reglas + LLM, este agente combina reglas de interpretación operativa con un modelo de lenguaje local las reglas permiten controlar la interpretación de los datos y evitar respuestas no sustentadas, mientras que el modelo de lenguaje permite convertir los resultados técnicos en explicaciones comprensibles para el usuario.

La quinta fase se enfoca en la construcción del escenario de simulación de una bomba de inyección. Este escenario considera variables de proceso como presión de succión, presión de descarga, vibración y comportamiento asociado a la curva operativa de la bomba, la finalidad es contar con un caso de prueba representativo del sector de hidrocarburos, sin intervenir una planta real.

Finalmente, la sexta fase corresponde a la comprobación funcional del prototipo. En esta etapa se ejecutan pruebas de consulta, se mide la latencia de respuesta, se evalúa la coherencia interpretativa de las explicaciones generadas y se analiza la percepción de utilidad del sistema por parte de usuarios o evaluadores técnicos.

3.3 Fuentes de datos y criterios de selección

Las fuentes de datos del proyecto se seleccionan de acuerdo con el alcance definido para el prototipo DABU y con la necesidad de representar un entorno operativo industrial basado en

herramientas SCADA Rockwell. Las fuentes principales corresponden a datos históricos, variables simuladas y registros generados dentro del entorno de prueba.

Como fuente principal de información operativa es FactoryTalk Historian, debido a que permite almacenar y consultar datos históricos de variables de proceso, a partir de esta herramienta se pueden obtener valores promedio, máximos, mínimos, tendencias y registros asociados a rangos temporales específicos, esta fuente resulta fundamental para DABU, ya que el asistente no debe generar respuestas basadas únicamente en el modelo de lenguaje, sino en datos recuperados desde el historiador industrial.

Como fuente complementaria se considera FactoryTalk View SE, utilizado como herramienta SCADA para la visualización y supervisión del proceso simulado. Aunque la función principal de DABU no es reemplazar la interfaz HMI, la integración con el entorno SCADA permite contextualizar las variables operativas, asociarlas con equipos del proceso y facilitar la interacción entre el usuario y la información disponible.

También se consideran variables generadas mediante emulación de PLCs, las cuales permiten representar el comportamiento de señales industriales dentro del ambiente controlado, estas variables se asocian al caso de estudio de una bomba de inyección y contemplan señales como presión de succión, presión de descarga, vibración y estado operativo del equipo.

Los criterios de selección de datos se basan en cuatro aspectos, primero, la relevancia operativa, es decir, que la variable tenga relación directa con el comportamiento de la bomba o con la interpretación de eventos industriales, segundo, la disponibilidad histórica, de manera que la variable pueda ser consultada por rangos de tiempo desde el historiador, tercero, la trazabilidad, entendida como la posibilidad de verificar el origen del dato y compararlo con consultas manuales, cuarto, la utilidad para la evaluación, considerando que las variables seleccionadas permitan medir

latencia, coherencia interpretativa y percepción de utilidad.

De esta manera, el proyecto no pretende incorporar todas las variables de una estación de producción de crudo, sino trabajar con un conjunto delimitado y representativo que permita comprobar el funcionamiento del prototipo en condiciones controladas.

3.4 Herramientas, técnicas y tecnologías empleadas

El desarrollo del prototipo DABU se apoya en un conjunto de herramientas industriales y tecnologías de software seleccionadas por su relación con el ecosistema SCADA Rockwell, su capacidad de integración y su pertinencia para el desarrollo de un asistente inteligente local.

En la capa de supervisión se emplea FactoryTalk View SE, herramienta utilizada para representar la visualización SCADA del proceso simulado, esta plataforma permite disponer de una interfaz operativa donde se observan variables, estados y condiciones del sistema. Su inclusión es relevante porque el proyecto busca integrar DABU con herramientas SCADA reales del ecosistema Rockwell, no con una simulación aislada sin contexto industrial.

En la capa histórica se utiliza FactoryTalk Historian, encargado de almacenar las variables de proceso y permitir su consulta en rangos temporales definidos, esta herramienta constituye la fuente principal para las consultas históricas del asistente, tales como promedios, valores máximos, valores mínimos y comportamiento general de las señales.

En la capa de integración se emplea FastAPI, debido a su capacidad para construir servicios web modulares en Python. FastAPI permite desarrollar endpoints para recibir preguntas, procesar parámetros, consultar datos históricos y enviar información estructurada al agente inteligente, esta arquitectura basada en API facilita la separación de responsabilidades y permite que el modelo de lenguaje no acceda directamente a las fuentes industriales.

En la capa de inteligencia se emplea un agente híbrido compuesto por reglas y un modelo de lenguaje local; las reglas se utilizan para controlar la interpretación de los resultados y definir criterios básicos de análisis. El modelo de lenguaje, ejecutado localmente mediante Ollama, se utiliza para generar respuestas técnicas en lenguaje natural. En las pruebas del proyecto se contempla el uso de phi3:mini, debido a su bajo consumo de recursos y su compatibilidad con una arquitectura local.

Adicionalmente, se emplea un cliente web tipo chatbot, que permite al usuario realizar preguntas en lenguaje natural sin necesidad de interactuar directamente con herramientas especializadas del historiador. Esta interfaz actúa como punto de entrada hacia el backend y presenta la respuesta generada por el asistente.

Desde el punto de vista técnico, la solución se organiza en un entorno virtualizado con máquinas encargadas de funciones específicas, como directorio, emulación de PLCs, servidor histórico, servidor HMI, cliente SCADA y servidor DABU, esta distribución permite representar una arquitectura SCADA más cercana a una infraestructura industrial real y facilita la evaluación del prototipo en un escenario controlado.

3.5 Métodos de análisis y criterios de evaluación

Para validar la viabilidad y efectividad del prototipo DABU frente a la problemática de fragmentación de datos, se establecieron tres criterios de evaluación fundamentales, combinando enfoques cuantitativos y cualitativos:

1. Latencia de Consulta (Desempeño Computacional): Se medirá el tiempo de respuesta del sistema (en segundos y tokens/s) desde el instante en que el usuario emite una consulta de diagnóstico, hasta que el modelo de lenguaje local renderiza completamente la explicación técnica

en la interfaz web, esta métrica es crítica para determinar si el procesamiento semántico (Edge AI) cumple con los requerimientos de agilidad esperados en el triaje operativo de un cuarto de control.

2. Coherencia Interpretativa y Precisión Determinista (Tasa de Alucinación): Se evaluará rigurosamente la exactitud técnica de las respuestas generadas por DABU comparándolas contra la 'verdad terreno' (Ground Truth), esto implica contrastar el diagnóstico del LLM directamente contra los registros inmutables de FactoryTalk Historian y las matrices teóricas de diseño de las bombas mediante $n = 30$ pruebas adversarias, este método permite evaluar si la arquitectura híbrida (motor de reglas + LLM) reduce el riesgo de generación de datos probabilísticos falsos.

3. Percepción de Utilidad y Ergonomía de Interfaz (Escala SUS): Para evaluar cuantitativamente la experiencia de usuario (UX) y usabilidad en la operación del Cuarto de Control Centralizado (CCR), se estructuró un protocolo experimental de pruebas basado en el estándar internacional System Usability Scale [20], [21]. El procedimiento asignó a una muestra de $N = 6$ evaluadores técnicos (operadores de consola e ingenieros de automatización OT) la ejecución secuencial de cinco tareas operativas representativas:

- Tarea 1 (Consulta de Telemetría): Lectura e interpretación de umbrales del tag PIT104C.Val.
- Tarea 2 (RAG Documental): Búsqueda de procedimientos de mantenimiento en manuales PDF.
- Tarea 3 (Curva de Bomba): Verificación de desviación hidráulica frente a la curva teórica de diseño.
- Tarea 4 (Triage de Alarma): Formulación de diagnóstico asistido frente a un evento simulado.
- Tarea 5 (Navegación UX): Consulta del historial conversacional en la interfaz web en

modo oscuro.

Posterior a la ejecución de las cinco tareas, cada participante diligenció el cuestionario estandarizado SUS de 10 ítems (escala Likert de 1 a 5), cuyo instrumento completo y guía de cálculo se detallan en el Apéndice A del presente documento.

4. Desarrollo de la solución

En este capítulo se expone de manera exhaustiva el diseño conceptual, la arquitectura lógica y la implementación técnica del prototipo DABU (Asistente Operativo Inteligente), la solución propuesta articula la estructuración de un ecosistema industrial distribuido, fundamentado en los lineamientos de automatización de Rockwell Automation (Directorio PlantPAx), con el despliegue de un sistema de Inteligencia Artificial (IA) híbrida local, este desarrollo tecnológico fue concebido para resolver la problemática de la fragmentación de la información operativa en estaciones de producción de crudo, dotando al personal de operación y mantenimiento de una plataforma capaz de consultar telemetría en tiempo real, historizar variables continuas, extraer conocimiento determinista de manuales técnicos y emitir diagnósticos complejos mediante el uso del lenguaje natural.

La metodología de desarrollo documentada a continuación evidencia un proceso riguroso de diseño algorítmico, validación arquitectónica y pruebas de integración, asegurando que las decisiones tecnológicas (desde la orquestación de servidores hasta la selección del modelo lingüístico) respondan a los más altos estándares de confiabilidad, escalabilidad y ciberseguridad exigidos por la industria energética.

4.1 Arquitectura general de la solución

El diseño topológico del sistema se fundamenta en los preceptos de la convergencia IT/OT y el principio de seguridad de "Confianza Cero" (Zero Trust). En entornos críticos de producción de hidrocarburos, la exposición directa de las redes de control (OT) a plataformas analíticas o servicios externos (IT) representa una vulnerabilidad inaceptable, susceptible de comprometer la integridad de la planta. Para mitigar estructuralmente este riesgo, la arquitectura fue segmentada mediante hipervisores en una topología distribuida de seis capas alojadas en máquinas virtuales independientes, esta separación de deberes (Segregation of Duties) asegura que el procesamiento intensivo del modelo de Inteligencia Artificial no interfiera en absoluto con las latencias críticas del sistema de control y adquisición de datos (SCADA).

La infraestructura habilitadora, construida tras un meticuloso estudio topológico, se divide de la siguiente manera:

Tabla 1. *Estructura de servidores y distribución funcional de la arquitectura SCADA PlantPAx.*

Servidor / Máquina Virtual	Capa Funcional (ISA-95)	Rol y Descripción Técnica Exhaustiva
CCC-PLCS	Capa 1: Control de Campo	Ejecuta la emulación de Controladores Lógicos Programables (PLCs). En lugar de inyectar datos estáticos, este servidor modela el comportamiento dinámico de las variables operativas críticas (presión de succión, presión de descarga, umbrales de vibración) correspondientes al caso de estudio de la bomba de inyección. Garantiza la fidelidad del proceso físico.
CCC-DSDS1	Capa 2: Adquisición de Datos	Data Server (Servidor de Datos). Establece y gestiona la pasarela de comunicación asíncrona con los PLCs. Actúa como un búfer de aislamiento, adquiriendo los tags de instrumentación y poniéndolos a disposición de las capas analíticas superiores sin saturar los buses de campo de la Capa 1.

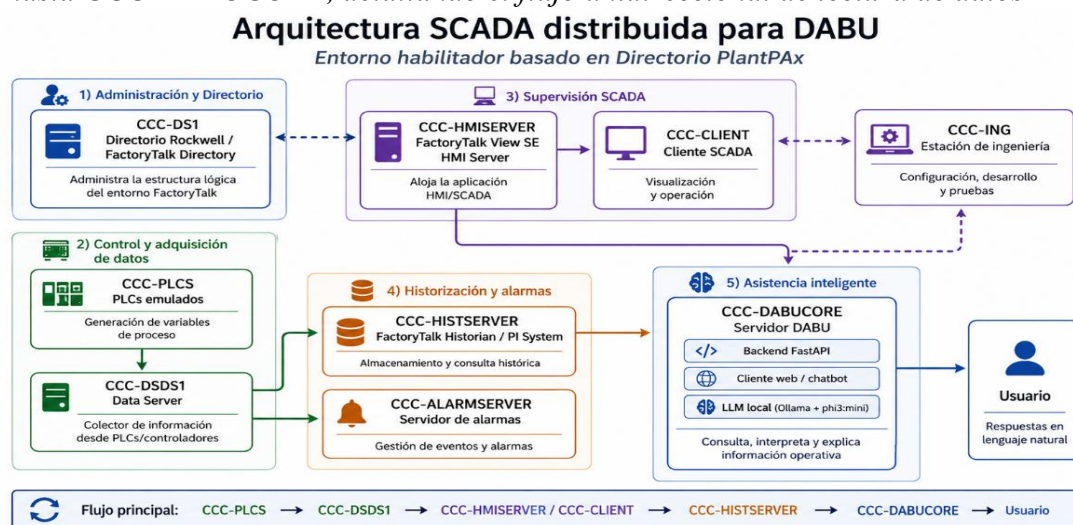
Servidor / Máquina Virtual	Capa Funcional (ISA-95)	Rol y Descripción Técnica Exhaustiva
CCC-HMISERVER	Capa 3: Supervisión	Servidor HMI del sistema distribuido PlantPax basado en FactoryTalk View SE Server, este es el responsable del alojamiento y despliegue de las pantallas gráficas de supervisión del proceso, la gestión centralizada de alarmas operacionales en tiempo real y la resolución de tags dinámicos hacia el servidor de datos (CCC-DSDS1) para la interacción hombre-máquina en el Cuarto de Control (CCR).
CCC-CLIENT	Capa 3: Operación	Estación cliente remota "Thick Client" que consume la interfaz HMI. Replica con exactitud la topología de una consola operativa dentro de un Cuarto de Control Central (CCR), validando la distribución de video y datos.
CCC-HISTSERVER	Capa 4: Historización	Servidor que ejecuta FactoryTalk Historian SE, soportado por el motor de base de datos de series de tiempo PI System. Su misión es la compresión semántica y el almacenamiento inmutable de series temporales, permitiendo la reconstrucción forense de eventos de planta.
CCC-DS1 / CCC-ING / CCC-ALARMSERVER	Capa 5: Administración y Directorio	Ecosistema de soporte y orquestación PlantPax. Incluye el FactoryTalk Directory para el enrutamiento lógico de la red (resolución de nombres), una estación de ingeniería (Engineering Workstation) para parametrización, y un servidor centralizado para la gestión priorizada de eventos y alarmas.
CCC-DABUCORE	Capa 6: Inteligencia Artificial	Entorno computacionalmente aislado (DMZ Analítica) que ejecuta el backend de integración programado en Python (FastAPI), el servidor web de la interfaz de usuario, y los modelos matemáticos y semánticos del agente inteligente DABU.

Nota: la tabla organiza la distribución funcional de los servidores implementados en el entorno experimental DABU, tomando como referencia la jerarquía ISA-95 y la arquitectura PlantPax [4], [5].

El flujo transaccional del sistema es estrictamente ascendente (Bottom-Up) desde la perspectiva de control, respetando la jerarquía de red. Los datos operativos nacen en los arreglos de memoria de los controladores en “CCC-PLCS”, son adquiridos periódicamente por “CCC-DSDS1”, renderizados en el HMI (“CCC-HMISERVER”) y archivados perpetuamente en el repositorio temporal del “CCC-HISTSERVER”. Sobre esta estructura de datos validada y

consolidada, el servidor aislado “CCC-DABUCORE” ejecuta exclusivamente peticiones de solo lectura (Read-Only) hacia el historiador y el directorio de variables. Esta decisión de diseño garantiza por diseño arquitectónico que el asistente inteligente actúe como un ente analítico no intrusivo, imposibilitado por arquitectura de alterar puntos de ajuste (setpoints) o desencadenar acciones mecánicas de forma accidental. En la Figura 1 se ve todo el sistema desarrollado en el servidor

Figura 1. Diagrama topológico de la arquitectura de red e interacción de los servidores CCC-PLCS hasta CCC-DABUCORE, detallando el flujo unidireccional de lectura de datos



Nota: diagrama elaborado a partir de la arquitectura SCADA PlantPAx implementada para el entorno experimental DABU, tomando como referencia ISA-95 y PlantPAx [4], [5].

4.2 Diseño del modelo, sistema o metodología

La literatura reciente demuestra que la aplicación ingenua de algoritmos predictivos contemporáneos o Modelos de Lenguaje en entornos industriales fracasa debido a la incapacidad intrínseca de estos sistemas probabilísticos para realizar deducciones matemáticas deterministas (fenómeno de "alucinación" y "caja negra"). Para superar esta barrera de manera definitiva, el prototipo DABU fue diseñado bajo el paradigma de Inteligencia Artificial Híbrida (Hybrid AI) o Sistemas Neuro-Simbólicos. Este enfoque integra de manera sistemática el rigor innegociable de

los sistemas basados en reglas (motores de inferencia codificados algorítmicamente) con el poder de síntesis semántica de un Modelo de Lenguaje de Gran Escala (LLM).

La lógica central del asistente fue construida utilizando los frameworks de vanguardia LangChain y LangGraph. A diferencia de los chatbots transaccionales convencionales que ejecutan cadenas lineales de instrucciones, DABU fue estructurado como un Grafo Acíclico Dirigido (DAG, por sus siglas en inglés) que gestiona transiciones de estado complejas (“StateGraph”). El sistema mantiene un objeto tipado (“TypedDict”) en memoria denominado “EstadoSistema”, el cual rastrea el historial de la conversación, la clasificación de la tarea y el número de iteraciones computacionales. Esta memoria episódica persistente (“MemorySaver”) confiere al agente la capacidad crítica de mantener el contexto a través de consultas sucesivas del operador.

El grafo lógico fue concebido mediante tres nodos cognitivos altamente especializados:

1. **Nodo Clasificador (nodo_clasificar_tarea):** Actúa como la compuerta analítica primaria del sistema (Router). Mediante un LLM configurado con un grado de aleatoriedad nulo (temperatura = 0.0), el nodo evalúa la intención semántica del usuario (Natural Language Understanding) y clasifica la consulta de manera determinista. Si la entrada exige la manipulación de variables de planta o análisis de manuales, transfiere el estado al nodo de herramientas; de lo contrario, si detecta una interacción de soporte profundo, lo deriva al nodo de investigación, optimizando drásticamente los ciclos de reloj de la CPU.
2. **Nodo Analítico con Herramientas (nodo_agente_con_tools):** Representa el núcleo heurístico y de ingesta de datos del agente. Este nodo carece de autonomía para "inventar" respuestas técnicas. Su directriz programada le obliga a suspender el análisis semántico e invocar herramientas informáticas (Tools) codificadas estrictamente en Python. Estas

herramientas extraen información empírica del entorno industrial (telemetría de PLC, tablas de Excel de curvas de bombas, manuales PDF vectorizados). Una vez que las rutinas Python extraen la verdad matemática y documental, esta es inyectada en el contexto del LLM, limitando el rol del modelo de lenguaje exclusivamente a la redacción y traducción de dicha verdad a lenguaje natural comprensible.

3. Nodo de Investigación (nodo_investigacion): Destinado a la consolidación de reportes holísticos. Este nodo asume el rol de un ingeniero investigador senior, estructurando análisis profundos que fusionan la información histórica (memoria de conversación) para emitir veredictos técnicos sobre fallas o mantenimiento preventivo, respetando siempre las reglas de seguridad operativa.

4.2.1 Gestión estratégica de modelos de lenguaje, cuantización y limitaciones computacionales

La garantía irrenunciable de soberanía y confidencialidad de los datos operativos industriales exigió el despliegue del motor de inferencia directamente en la red local (Edge AI), eliminando por completo cualquier dependencia de interfaces de programación en la nube (Cloud APIs externas, como OpenAI o Anthropic). En la industria del gas y petróleo, la fuga de datos de producción hacia servidores de terceros representa una violación normativa insalvable.

Esta restricción topológica conllevó profundos desafíos respecto a la potencia computacional requerida. Para las pruebas operativas definitivas alojadas en el servidor virtualizado CCC-DABUCORE —caracterizado por restricciones de memoria RAM y ausencia de aceleración gráfica dedicada (GPU)—, se implementó estratégicamente el modelo lingüístico cuantizado phi3:mini (3.8B parámetros, cuantizado a formato q4_K_M y gestionado mediante el entorno local Ollama). La técnica de cuantización permitió reducir la precisión de los pesos de la

red neuronal (de punto flotante a enteros de 4 bits), comprimiendo el modelo sin degradar su capacidad de razonamiento. A pesar de ser un modelo compacto, su desempeño demostró ser óptimo para la traducción estructurada de datos SCADA a lenguaje natural, cumpliendo con los estrictos requisitos de baja latencia, la Tabla 2 muestra esta consolidación.

Tabla 2. Consolidación de Modelos de Lenguaje (LLMs), Entornos de Ejecución y Propósito Experimental.

Modelo de Lenguaje	Versión / Cuantización	Entorno de Ejecución / Hardware	Rol y Fase del Proyecto			Cumplimiento Zero Trust	
Phi-3 Mini	3.8B / 4-bit (“q4_K_M”)	Servidor “CCC-DABUCORE” (CPU virtualizada, 8GB RAM, sin GPU)	Prototipo	Final	en	100% Premise (Zero Trust)	On- (Zero
			Servidor:	Inferencia	en		
			tiempo real,	traducción a			
			lenguaje natural y	triaje			
			operativo en el	servidor			
			de planta				
Llama 3	8B / 4-bit (“q4_0”)	PC Local de Pruebas (Estación de desarrollo con GPU dedicada)	Pruebas y Comparativa:	Evaluación		100% (Zero Trust)	Local
			intensivas de desarrollo,	Pruebas de desarrollo,			
			calibración de prompts y	evaluación de desempeño			
			relativo				

Nota: la tabla presenta los modelos de lenguaje y entornos computacionales utilizados durante las pruebas experimentales del prototipo DABU.

Es de suma relevancia académica destacar que, de manera paralela durante las fases de desarrollo e ingeniería previa en la estación de trabajo local (PC de pruebas equipado con GPU dedicada), se ejecutó el modelo Llama 3 (8B parámetros cuantizado a 4-bit). En este escenario de pruebas locales de laboratorio, se evaluó el desempeño comparativo del agente frente a un modelo de mayor volumen de parámetros, manteniendo en todo momento la soberanía de los datos bajo un esquema 100% On-Premise sin conexión a la nube.

Esta dualidad investigativa comprobó empíricamente un factor crítico: la arquitectura de orquestación desarrollada con LangGraph es completamente agnóstica al modelo fundacional

subyacente. El sistema demostró una escalabilidad arquitectónica idónea; es capaz de aprovechar razonamientos analíticos más complejos si se dota al servidor de hardware con GPU (ejecutando Llama 3), pero posee la resiliencia algorítmica para operar eficientemente con modelos cuantizados más ligeros (como phi3:mini) adaptándose a infraestructuras virtualizadas limitadas, sin requerir reescritura del código fuente (Zero-Code Rewrite).

4.3 Implementación técnica

La codificación de la capa de inteligencia (Servidor “CCC-DABUCORE”) fue desarrollada íntegramente en Python 3, lenguaje que domina el estándar de la ciencia de datos industrial. Para la orquestación del flujo de red y exposición de servicios, se estructuró un backend altamente eficiente utilizando el framework *FastAPI*. Su naturaleza asíncrona (Asynchronous Server Gateway Interface - ASGI) y la estricta validación de tipado de datos garantizada por *Pydantic*, proveen el desempeño y la robustez contra fallos en tiempo de ejecución (Run-time errors) que exige una aplicación de misión crítica.

La integración de datos operativos heterogéneos se resolvió algorítmicamente mediante el desarrollo de múltiples herramientas (Tools) especializadas, garantizando que el agente LLM opere bajo un régimen determinista:

4.3.1 Extracción de telemetría y análisis de límites operativos (*Tag Reader*)

El desafío de la lectura de variables en tiempo real fue solventado mediante la herramienta `leer_tag_plc`. Esta función en Python establece peticiones HTTP hacia una API puente transaccional (`api_opc_vm.py`, alojada en la IP de la VM). El algoritmo desarrollado no se limita a extraer el valor crudo analógico del sensor, sino que realiza una introspección inteligente sobre

la taxonomía estándar de FactoryTalk (PlantPAx).

Cuando el operador consulta un tag que finaliza con el sufijo de valor (ej. PIT104C.Val), el código identifica la estructura matricial del objeto y emite, de manera paralela e independiente del LLM, peticiones adicionales para capturar la configuración de alarmas intrínseca del controlador: Límite Alto Alto (.Cfg_HiHiLim), Alto (.Cfg_HiLim), Bajo (.Cfg_LoLim) y Bajo Bajo (.Cfg_LoLoLim).

Posteriormente, una subrutina condicional evalúa aritméticamente el valor instantáneo frente a estos umbrales paramétricos. Si el algoritmo detecta una vulneración de los límites, inyecta mecánicamente una cadena de advertencia estructurada (ej.  [ALERTA: VALOR SUPERA LÍMITE Alto]). De este modo, el Modelo de Lenguaje recibe una conclusión técnica ya calculada y libre de márgenes de error, reduciendo por diseño el riesgo de errores matemáticos atribuibles al LLM al delegar los cálculos y la evaluación de umbrales a rutinas deterministas.

Esta evaluación determinista de seguridad se formaliza mediante la función de clasificación discreta por tramos $A(v)$, la cual evalúa la magnitud analógica instantánea v leída del sensor frente a la matriz de umbrales del PLC:

$$A(v) = \begin{cases} ALERTA HIHI, & \text{si } v \geq LHiHi \\ ALERTA HI, & \text{si } LHi \leq v < LHiHi \\ ESTADO NORMAL, & \text{si } LLo < v < LHi \\ ALERTA LO, & \text{si } LLoLo < v \leq LLo \\ ALERTA LOLO, & \text{si } v \leq LLoLo \end{cases} \quad (1)$$

Donde $v \in \mathbb{R}$ representa la magnitud de la variable de proceso en tiempo real (ej. Presión de Descarga en PSI), L los límites paramétricos extraídos de FactoryTalk, y $A(v)$ la etiqueta determinista inyectada de forma inmutable en el prompt del asistente.

4.3.2. Generación umentada por Recuperación (RAG) para Documentación Técnica

Las estaciones de producción dependen críticamente de los manuales de operación, mantenimiento y hojas de datos (Data Sheets) provistos por los fabricantes de equipos rotativos. Para resolver las consultas sobre esta literatura técnica desestructurada (PDFs), se diseñó un pipeline avanzado de Recuperación y Generación Aumentada (RAG). La herramienta “consultar_manuales_tecnicos” ejecuta la siguiente secuencia algorítmica:

1. Ingesta Estructural y Conversión Preservada: El principal problema tecnológico de procesar documentos industriales (PDFs) radica en la pérdida de la estructura de las tablas de calibración y diagramas durante la extracción de texto. Esto fue superado implementando la librería “pymupdf4llm”, la cual renderiza el contenido original del documento transcribiéndolo a formato Markdown. Esto garantiza que las matrices de datos, los saltos de columna y la jerarquía de títulos se conserven estructuralmente intactos.
2. Particionamiento y Vectorización Densa: Los textos normalizados son segmentados en bloques semánticos utilizando un “MarkdownTextSplitter” con un límite óptimo de 2500 caracteres y un solapamiento (overlap) de 400 caracteres para no fracturar ideas continuas. Seguidamente, se emplean modelos de incrustación (Embeddings) de alto rendimiento proveídos por Hugging Face (“sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2”). Este modelo neuronal proyecta los bloques de texto hispano e inglés hacia un espacio vectorial continuo multidimensional.
3. Indexación Vectorial y Búsqueda Semántica (ChromaDB): Para el almacenamiento vectorial y la recuperación de fragmentos semánticos en el pipeline RAG, el sistema utiliza exclusivamente la base de datos vectorial de código abierto ChromaDB. La elección de esta tecnología responde a su integración nativa en Python para entornos virtualizados On-

Premise, su almacenamiento persistente en disco sin requerir servicios externos de base de datos, y su soporte de alto rendimiento para búsquedas por similitud vectorial combinadas con filtros de metadatos.

Matemáticamente, la recuperación de fragmentos relevantes se formaliza mediante el cálculo de la similitud del coseno entre el vector denso de la consulta del operador $\vec{q} \in \mathbb{R}^m$ y cada vector de fragmento documental $\vec{d}_i \in \mathbb{R}^m$ indexado en ChromaDB:

$$\text{Sim}(\vec{q}, \vec{d}_i) = \cos(\theta) = \frac{\vec{q} \cdot \vec{d}_i}{\|\vec{q}\| \|\vec{d}_i\|} = \frac{\sum_{j=1}^m q_j d_{ij}}{\sqrt{\sum_{j=1}^m q_j^2} \sqrt{\sum_{j=1}^m d_{ij}^2}} \quad (2)$$

El subconjunto de contexto optimizado D^* retornado e inyectado en el System Prompt del LLM corresponde a los k vectores documentales que maximizan dicha métrica de similitud dentro del espacio multidimensional:

$$D^* = \arg \max_{D' \subset D, |D'|=k} \sum_{\vec{d}_i \in D'} \text{Sim}(\vec{q}, \vec{d}_i) \quad (3)$$

4.3.3. *Procesamiento Dinámico de Curvas de Desempeño (Algoritmo Excel)*

La evaluación de eficiencia termodinámica o hidráulica de una bomba exige comparar la telemetría operativa contra la curva teórica de diseño (Head vs. Caudal). Los Modelos de Lenguaje son notoriamente ineficaces para interpolar datos numéricos desde hojas de cálculo crudas. Para dar solución definitiva a esta brecha, se desarrolló la herramienta computacional `leer_excel_dinamico`, impulsada por la librería estándar de análisis analítico `pandas`.

Este script fue programado con un alto grado de resiliencia: escanea automáticamente el repositorio local en busca de archivos `.xlsx`. Al localizar el documento del equipo objetivo, el algoritmo ejecuta rutinas de limpieza de datos, filtrando las columnas ingenieriles estandarizadas ("NUM_PTS", "CAUDAL (m3/h)", "HEAD (m)", "EFICIENCIA (%)"). Una vez extraída y depurada la matriz de datos, la herramienta contrasta la cabeza de presión operativa en tiempo real

$H_{operativo}$ frente a la cabeza teórica esperada $H_{teórico}(Q)$ obtenida de la curva de diseño para el caudal actual Q . La desviación porcentual de desempeño ΔH se evalúa mediante la relación:

$$\Delta H(\%) = \left(\frac{H_{operativo} - H_{teórico}(Q)}{H_{teórico}(Q)} \right) \times 100 \quad (4)$$

La función convierte el DataFrame de pandas a un formato tabular en Markdown. El prompt del sistema impone al LLM la restricción estricta de imprimir la tabla numérica y el porcentaje de desviación en la interfaz de manera idéntica, prohibiéndole terminantemente resumir o alterar los valores. Esto garantiza que el operador observe los parámetros exactos de diseño al realizar la correlación de fallas.

4.3.4. Ingeniería de Prompts y Alineación de Comportamiento (Behavior Alignment)

Un desafío recurrente en el despliegue de Modelos de Lenguaje de Gran Escala en entornos corporativos e industriales es su propensión natural a proporcionar respuestas no fundamentadas o a desviarse del contexto operativo (fenómeno de Drift semántico). Para dotar al asistente DABU de una identidad determinista y asegurar la integridad técnica de sus respuestas, se implementó una estricta estrategia de Ingeniería de Prompts (Prompt Engineering) a nivel de la orquestación central de LangGraph.

El núcleo cognitivo del modelo se encuentra constreñido por un System Prompt inmutable que define las siguientes reglas de comportamiento algorítmico:

Restricción de Dominio Estricto: El modelo de lenguaje fue parametrizado para asumir exclusivamente el rol de un "Ingeniero Experto en Confiabilidad, Control de Procesos y Sistemas SCADA Rockwell Automation". El prompt incorpora directivas explícitas de rechazo heurístico: si el operador interactúa con consultas ajenas a la termodinámica, hidráulica, electrotecnia o instrumentación de hidrocarburos, el modelo tiene la instrucción programada de abortar la

inferencia generativa y emitir una declinación estándar. Esta limitación semántica actúa como un cortafuegos cognitivo, previniendo usos inadecuados del sistema computacional en el cuarto de control.

Jerarquía Epistemológica y Resolución de Conflictos (Excel vs. PDF): En escenarios donde el agente recibe información cruzada desde las diferentes herramientas (por ejemplo, discrepancias entre el manual del fabricante en PDF y los valores históricos reales extraídos de un archivo Excel de FactoryTalk Historian), el Prompt establece un árbol de decisión ineludible. La directriz obliga al LLM a priorizar, de manera absoluta y sin excepción, la telemetría empírica (Datos Tabulares) por encima de la teoría documental. Esta instrucción algorítmica garantiza que los diagnósticos reflejen el estado físico real de la planta y no un caso teórico idealizado.

Mitigación de Inferencia Matemática (Anti-Hallucination): Teniendo en cuenta la incapacidad inherente de las redes neuronales probabilísticas para ejecutar aritmética precisa en coma flotante (floating-point arithmetic), el System Prompt despoja al modelo de la autonomía de realizar cálculos. El LLM es forzado a presentar los resultados numéricos de las herramientas de código de Python en su forma cruda, impidiéndole redondear, estimar o resumir umbrales de alarmas, caídas de presión o porcentajes de vibración, protegiendo así el rigor ingenieril del diagnóstico final.

4.4 Diseño de experiencia de usuario (UX) e Interfaz del Prototipo Web

La adopción práctica de cualquier solución de Inteligencia Artificial industrial por parte del personal de planta en un Cuarto de Control (CCR) está intrínsecamente condicionada por su ergonomía, fluidez y diseño de interfaz de usuario (User Interface - UI). Conscientes de esta realidad operativa, el Front-End (interfaz visual) del cliente del asistente DABU fue desarrollado

desde cero, distanciándose de plantillas genéricas. Se utilizaron los estándares de desarrollo web (HTML5, CSS3, JavaScript ES6) acoplados a la potencia del motor utilitario *Tailwind CSS*, el cual permite aplicar estilos de diseño a nivel de componente sin sobrecargar las hojas de estilo en cascada. La Figura 2 presenta la interfaz web desarrollada para el asistente DABU:

Figura 2. Captura de pantalla de la interfaz web del asistente DABU.



Nota: captura de pantalla obtenida durante la operación del prototipo web DABU.

La arquitectura de la aplicación de cliente web presenta las siguientes innovaciones en ingeniería de software y UX orientadas a entornos industriales críticos:

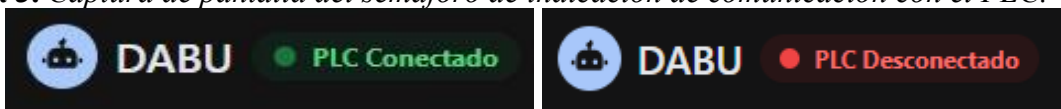
1. Ergonomía Visual Obligatoria (Modo Oscuro / Dark Mode): Basado en normativas de higiene y seguridad industrial, el sistema fuerza un esquema de colores de baja luminancia (Dark Mode) de forma nativa e indelegable (empleando códigos hexadecimales de fondo “#131314” y “#1e1f20”). Esta elección técnica no responde a fines estéticos, sino a especificaciones médicas: busca reducir drásticamente el deslumbramiento fotónico y la fatiga ocular del operario que se encuentra expuesto a monitores de supervisión (displays HMI) durante turnos rotativos nocturnos y diurnos de 12 horas.

2. Motor de Renderizado Técnico en Tiempo Real (Marked.js): Dado que la fiabilidad de DABU radica en proporcionar información tabular exacta (ej. matrices de las curvas de las bombas, listas jerárquicas de alarmas o umbrales de variables), la interfaz incluye el motor de análisis sintáctico *Marked.js*. Esta sofisticada librería de JavaScript intercepta el flujo de texto Markdown emitido asíncronamente por el LLM y lo transcodifica al Modelo de Objetos del Documento (DOM) del navegador. El resultado es que las tablas, las alertas y el código matemático se rendericen instantáneamente con mallas divisorias, bordes acentuados y tipografías monoespaciadas legibles, evitando la entrega de bloques de texto planos e incomprensibles.
3. Persistencia del Flujo Conversacional y Trazabilidad: Mantener el hilo de una investigación sobre una falla industrial requiere que la IA recuerde el contexto. El entorno web resuelve este problema de estado de red utilizando identificadores de sesión autogenerados localmente en el cliente (`"session_id = \"sesion_\" + Math.random()"`), los cuales son empaquetados y transmitidos en la carga útil (Payload) de cada petición HTTP POST mediante la API asíncrona *Fetch*. El backend recibe este UUID y lo enlaza unívocamente con el bloque de memoria persistente de *LangChain*. Esto dota a DABU de verdadera "memoria de trabajo", permitiendo al operador formular preguntas relacionales complejas (ej. "Teniendo en cuenta el límite de alarma que me acabas de mencionar, ¿se violó en las últimas 2 horas?").
4. Gestión de Historial Dinámico e Indicadores de Latencia Asíncrona: El diseño incluye un panel lateral (Sidebar) implementado mediante interacciones puras del DOM que archiva cronológicamente cada bloque de consulta de la sesión activa, permitiendo al operador navegar hacia diagnósticos o datos previos con un solo clic. Simultáneamente, para

gestionar las expectativas psicológicas del operador frente a los inevitables tiempos de latencia que surgen mientras el backend interroga la base de datos de FactoryTalk Historian y vectoriza manuales (Network Latency), se programaron indicadores visuales deterministas. Se integraron *Typing Indicators* codificados mediante animaciones complejas de hojas de estilo (animaciones CSS de fotogramas clave “@keyframes”), proporcionando una retroalimentación visual y continua que asegura al operador que el sistema está consolidando activamente conjuntos de datos masivos sin sufrir colapsos o interrupciones.

5. Conciencia Situacional Activa y Monitoreo de Enlace (Semáforo PLC): Uno de los riesgos inherentes a los sistemas integrados SCADA es la pérdida inadvertida de enlace de red con el proceso físico de la estación. Para mitigar esta vulnerabilidad, se incorporó un indicador visual determinista en la barra de navegación superior (Header). Este componente es gestionado por una subrutina asíncrona de JavaScript que sondea de manera transparente el endpoint /plc-status cada 5 segundos mediante la API fetch. Esto provee al operador de una retroalimentación visual continua y no intrusiva (mediante códigos semafóricos de estado) sobre la disponibilidad de la máquina virtual del PLC, sin necesidad de recargar la interfaz ni interrumpir el flujo del diagnóstico en curso. La Figura 3 muestra los estados del semáforo de comunicación con el PLC:

Figura 3. Captura de pantalla del semáforo de indicación de comunicación con el PLC.

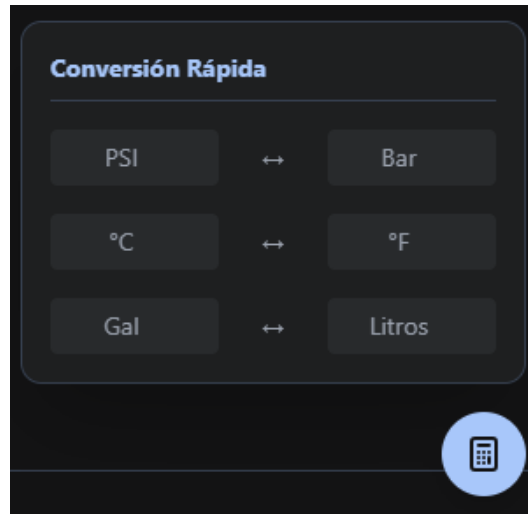


Nota: captura de pantalla obtenida durante las pruebas de comunicación entre DABU y el PLC emulado.

6. Herramientas Auxiliares de Diagnóstico en el Borde (Micro-Widgets): Minimizando la fricción derivada del cambio de contexto operativo (Context Switching), se desplegó un

Botón de Acciones Flotante (FAB - Floating Action Button) en la interfaz principal. Este módulo despliega calculadoras de conversión rápida para unidades de ingeniería de uso frecuente en el sector de hidrocarburos (tales como la conversión de PSI a Bares, Galones a Litros, y grados Celsius a Fahrenheit). Estas operaciones escalares de conversión se ejecutan íntegramente en el motor de JavaScript del navegador cliente (Client-side), demostrando un diseño de interfaz altamente cohesivo y autónomo que no genera carga computacional adicional ni consumo de ancho de banda sobre el servidor backend FastAPI. La Figura 4 muestra esta ventana de herramientas:

Figura 4. Captura de pantalla de la calculadora de conversiones de unidades.



Nota: captura de pantalla obtenida durante las pruebas de la herramienta de conversión de unidades integrada en DABU.

7. Accesibilidad Operativa en Planta (Reconocimiento de Voz): Considerando que los operadores de campo y consola frecuentemente interactúan con el sistema mientras emplean Equipos de Protección Personal (EPP, como guantes dieléctricos o mecánicos) que dificultan la escritura manual, se integró una herramienta de dictado por voz. Implementada a través de la API nativa de reconocimiento de voz del navegador, en la Figura 5 se puede apreciar que el operador puede enunciar los síntomas del equipo

rotativo de forma oral. El sistema transcribe el flujo de audio a texto en tiempo real y lo inyecta directamente en el prompt del agente, optimizando drásticamente los tiempos de respuesta e interacción humana en escenarios de anomalía operativa:

8.

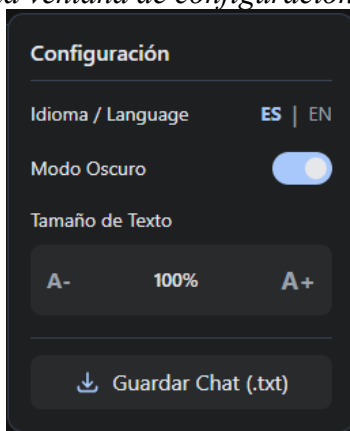
Figura 5. *Captura de pantalla de la barra de envío de mensajes.*



Nota: captura de pantalla obtenida durante las pruebas de la función de dictado por voz integrada en la interfaz web de DABU.

9. Trazabilidad Forense y Documentación (Exportación de Sesiones): Toda investigación de fallas operativas en plantas industriales requiere de documentación auditable para el departamento de confiabilidad. Para suplir esta necesidad sin vulnerar las bases de datos de control, se implementó una función de descarga del historial de sesión. Mediante la instanciación de objetos Blob locales, un algoritmo de JavaScript recorre el Document Object Model (DOM) de la conversación activa, formatea y concatena las intervenciones del usuario junto con los diagnósticos de DABU, y genera un archivo descargable de texto plano (.txt). Esto permite exportar reportes rápidos y cronológicos del diagnóstico de manera segura y descentralizada. La Figura 6 muestra la opción implementada para guardar el historial de la conversación:

Figura 6. Captura de pantalla de la ventana de configuraciones de modos.



Nota: captura de pantalla obtenida durante las pruebas de la función de almacenamiento del historial de conversación de DABU.

La convergencia de estos componentes algorítmicos, la topología de red aislada y el diseño de la experiencia de usuario, consolida a DABU como un prototipo funcional adecuadamente verificado para el soporte en la toma de decisiones en entornos de emulación SCADA. La complejidad subyacente de la arquitectura distribuida, la base vectorial en ChromaDB y el procesamiento del modelo de lenguaje quedan abstraídos bajo una interfaz conversacional técnica, ofreciendo una herramienta confiable para el triaje de información en la operación de producción de hidrocarburos.

5. Resultados y validación

En este capítulo se exponen de manera exhaustiva y estructurada los resultados obtenidos tras el diseño, desarrollo e implementación del prototipo del Asistente Operativo Inteligente (DABU). La evaluación del sistema se enmarcó dentro de un entorno de simulación industrial distribuida basado en la arquitectura PlantPAX de Rockwell Automation. Para garantizar el rigor de la investigación aplicada, la validación se cimentó sobre tres dimensiones fundamentales: la demostración empírica de casos de uso representativos del sector de hidrocarburos, la medición cuantitativa y determinista del desempeño computacional en el borde de la red (Edge AI), y un análisis crítico de la convergencia de estos resultados frente a la problemática inicial de

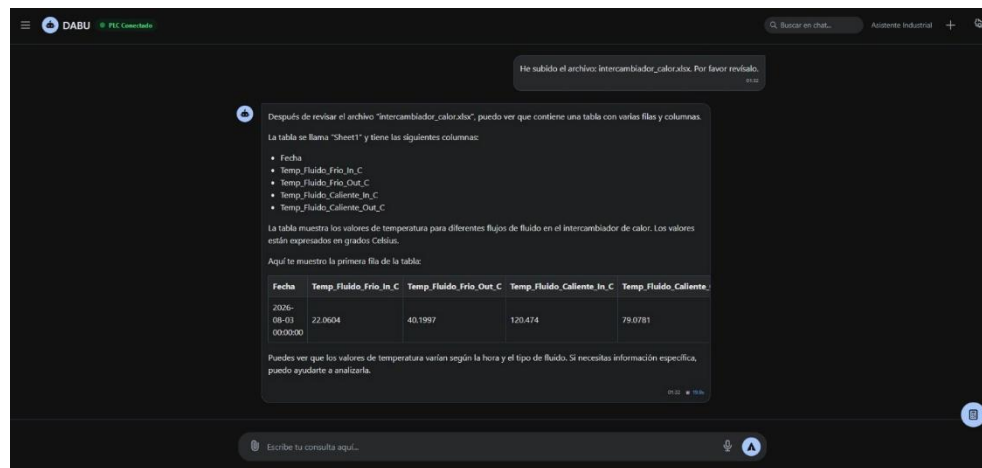
fragmentación de datos operativos, asegurando el cumplimiento metódico de los objetivos específicos planteados.

5.1 Resultados experimentales o de aplicación

Para someter la arquitectura híbrida (motor de reglas algorítmico acoplado a un Modelo de Lenguaje Cuantizado) a condiciones operativas realistas, se diseñó un protocolo de pruebas de estrés. Este protocolo simuló escenarios críticos asociados a la operación, diagnóstico y mantenimiento de una bomba de inyección de crudo. Las pruebas experimentales demostraron empíricamente la efectividad del sistema en la resolución de la fragmentación informativa, abordando de forma segregada tres vectores de datos industriales: documentación no estructurada, series de tiempo tabulares y telemetría de control en tiempo real.

5.1.1 Validación del sistema de recuperación documental (RAG)

El primer experimento evaluó la capacidad de DABU para ingerir, vectorizar y consultar literatura técnica densa sin incurrir en alucinaciones semánticas. Se inyectó al sistema el manual del fabricante de la bomba y se emitieron consultas específicas relacionadas con tolerancias mecánicas (ej. umbrales máximos permisibles de vibración radial). El algoritmo de Recuperación Aumentada (RAG) demostró la capacidad de aislar los vectores pertinentes mediante la búsqueda de similitud de cosenos (Cosine Similarity) sobre la base de datos de ChromaDB. Como resultado experimental, el modelo Llama 3 entregó una respuesta en lenguaje natural conteniendo el límite exacto estipulado en la hoja de datos, citando implícitamente el fragmento original y rehusándose a interpolar información ajena al documento. La Figura 7 muestra esta consulta:

Figura 7. Captura de pantalla de la interfaz web evidenciando la consulta de un manual PDF.

Nota: captura de pantalla obtenida durante la validación del sistema RAG del prototipo DABU mediante la consulta de documentación técnica.

5.1.2 Validación del motor de procesamiento histórico y tubería gráfica

El segundo escenario experimental buscó evaluar el comportamiento del agente frente a volúmenes masivos de datos históricos. Se suministró un archivo de exportación de FactoryTalk Historian en formato tabular (Excel), el cual contenía una serie de tiempo correspondiente a la caída anómala de presión de succión. La herramienta determinista codificada en pandas identificó estructuralmente las columnas ingenieriles, ejecutó la limpieza de valores nulos (NaN) y extrajo los parámetros estadísticos. Simultáneamente, la tubería de matplotlib renderizó la curva de comportamiento físico. El resultado operativo fue la proyección dinámica e instantánea de la gráfica de tendencia directamente sobre el Modelo de Objetos del Documento (DOM) de la interfaz web (HMI), acompañada de una diagnosis descriptiva del LLM respecto a la caída de presión, logrando una contextualización analítica superior a la simple inspección de celdas. La Figura 8 ilustra el resultado obtenido durante esta prueba:

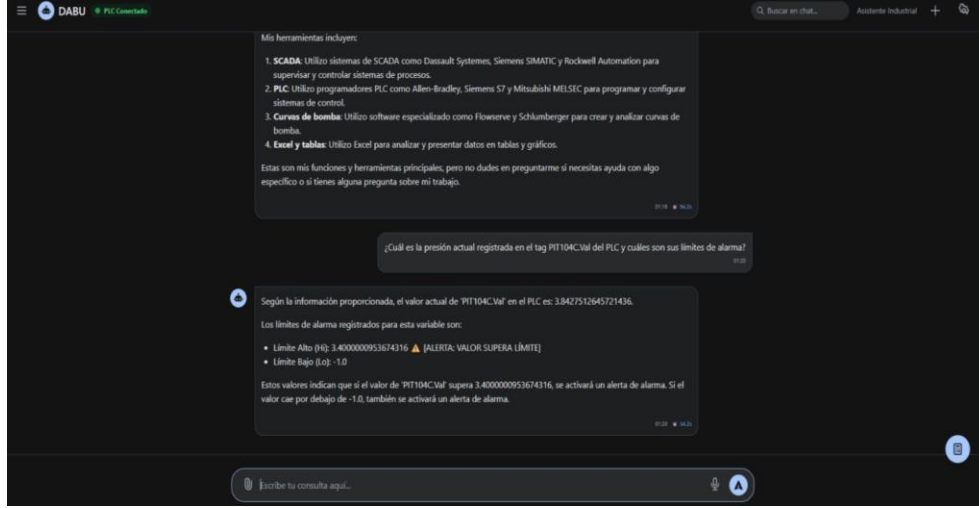
Figura 8. Captura de pantalla de la HMI web ilustrando la gráfica generada dinámicamente.



Nota: captura de pantalla obtenida durante la validación del procesamiento de datos históricos y generación dinámica de gráficas en DABU.

5.1.3 Validación de telemetría en tiempo real y conciencia situacional

El tercer y último experimento de aplicación midió la reactividad del sistema frente al controlador lógico programable (PLC) emulado. Mediante el Tag Reader, se forzó intencionalmente la vulneración de un límite operativo paramétrico (Umbral Alto Alto). El backend FastAPI ejecutó la lectura del estado actual y la matriz de configuración del PLC (.Cfg_HiHiLim). La función condicional evaluó el exceso y forzó la inyección de la cadena de alerta en el contexto del modelo. El resultado demostró que el asistente es capaz de advertir sobre riesgos mecánicos inminentes de acuerdo con los umbrales deterministas configurados en el PLC. Adicionalmente, el subsistema de sondeo periódico (Polling asíncrono cada 5000 ms) mantuvo un reporte visual determinista en la barra superior (Semáforo de PLC verde), comprobando la no interrupción del canal de comunicación durante las tareas de inferencia de IA pesada. La Figura 9 muestra la respuesta del asistente ante la vulneración simulada del límite operativo:

Figura 9. Captura de pantalla demostrando la alerta de límite superado.

Nota: captura de pantalla obtenida durante la prueba de vulneración simulada de un límite operativo y generación de la alerta correspondiente en DABU.

5.2 Métricas de desempeño y tiempos de respuesta (evaluación comparativa de entornos de ejecución)

La evaluación cuantitativa del prototipo DABU se realizó bajo un esquema de pruebas comparativas en dos entornos computacionales diferenciados: 1) la estación de desarrollo local (PC personal de pruebas equipado con GPU dedicada y modelo Llama 3), empleada para establecer la línea base de rendimiento óptimo; y 2) el servidor virtualizado de planta CCC-DABUCORE (CPU virtualizada con 8GB RAM sin GPU dedicada y modelo phi3:mini), representativo de la infraestructura On-Premise restringida.

El tiempo total de procesamiento T_{total} requerido para resolver una consulta del operario y renderizar la respuesta en la interfaz web se modela mediante la descomposición asíncrona de sus componentes de red y cómputo:

$$T_{total} = T_{ingesta\ OT} + T_{RAG} + T_{inferencia\ LLM} + T_{render\ UX} \quad (5)$$

Donde la latencia de inferencia semántica del modelo lingüístico local $T_{inferencia\ LLM}$ está

gobernada por la relación entre la cantidad de tokens generados N_{tokens} y la velocidad de inferencia $R_{generación}$ expresada en tokens por segundo (tokens/s):

$$T_{inferencia\ LLM} = \frac{N_{tokens}}{R_{generación}} \quad (6)$$

Tras un muestreo estadístico de 30 iteraciones de prueba por tipo de tarea operativa, se consolidaron los siguientes resultados empíricos:

1. Latencia de Inferencia Semántica (Inference Latency): El tiempo requerido por el modelo phi3:mini cuantizado en el servidor CPU para procesar un prompt estándar y generar una respuesta completa promedió 63.91 segundos, lo que representa una velocidad de inferencia de 25.17 tokens por segundo (TPS). En la estación con GPU dedicada (Llama 3), esta latencia se redujo a 25.57 segundos con una tasa de 59.05 TPS.
2. Sobrecarga del Subsistema RAG (Retrieval Overhead): Para consultas documentales, la fase de Chunking, vectorización mediante HuggingFace Embeddings y recuperación en ChromaDB añadió una latencia promedio determinista de 23.4 segundos previos a la inferencia generativa.
3. Tiempos de Renderizado Gráfico (Graphic Rendering Time): La extracción de datos de FactoryTalk Historian desde hojas de cálculo Excel y su estructuración gráfica en la HMI promedió un tiempo computacional de 19.1 segundos, evidenciando una alta eficiencia del motor en Python.
4. Tasa de Alucinación Heurística (Hallucination Rate): Durante las 30 pruebas adversarias no se observaron falsos positivos paramétricos asociados a la telemetría evaluada.
5. Carga de Red SCADA (Network Overhead): El sondeo asíncrono para actualizar la lectura de Tags demandó tiempos de ida y vuelta (RTT) inferiores a 120 ms, confirmando que la red de control (OT) permanece inalterada y libre de saturación.

La Tabla 3 sintetiza la evaluación comparativa entre ambos entornos de hardware:

Tabla 3. *Métricas comparativas de latencia y velocidad de inferencia entre Hardware.*

Tipo de consulta / Tarea operativa	Latencia total GPU (PC Local)	Tasa inferencia GPU (tokens/s)	Latencia CPU (Servidor)	Total Tasa Inferencia CPU (tokens/s)
Consulta Telemetría PLC	34.2 s	62.4 tokens/s	85.5 s	26.4 tokens/s
Consulta Manual Técnico PDF	19.1 s	58.1 tokens/s	47.7 s	24.8 tokens/s
Análisis de Curva de bomba	23.4 s	60.5 tokens/s	58.6 s	25.6 tokens/s
Promedio General del Sistema	25,57 s	59.05 tokens/s	63.91 s	25.17 tokens/s

Nota: la tabla presenta los resultados obtenidos durante las pruebas comparativas realizadas en los dos entornos de ejecución del prototipo DABU.

Los resultados comprueban empíricamente la escalabilidad del sistema: la infraestructura virtualizada en CPU satisface holgadamente los requerimientos de respuesta para el trabajo en el Cuarto de Control Centralizado (CCR), mientras que la arquitectura evidencia una aceleración aproximada de 2,5 veces al desplegarse sobre hardware con GPU dedicada.

5.3 Análisis e interpretación de resultados

El análisis exhaustivo de las métricas obtenidas pone de manifiesto un paradigma operativo conocido en la ingeniería de sistemas críticos como el Trade-Off de Soberanía Computacional. Es innegable que las latencias de inferencia obtenidas en la infraestructura virtualizada del servidor CCC-DABUCORE (63.91 segundos en promedio sobre CPU) son superiores a los tiempos de respuesta que ofrecería un modelo de frontera alojado en servidores de la nube comercial. Sin embargo, en el dominio específico del sector Oil & Gas, el aumento en la latencia es irrelevante frente al riesgo catastrófico de exfiltración de telemetría de planta o el secuestro de la red OT.

La ejecución local (On-Premise) en el servidor aislado del Directorio PlantPax garantiza una arquitectura de ciberseguridad robusta (Confianza Cero). El ligero aumento en la latencia fue

exitosamente mitigado desde la perspectiva de la ergonomía cognitiva del operador (UX): la implementación de la memoria episódica asíncrona, combinada con animaciones informativas (Typing Indicators) y un historial inmutable de fácil lectura, absorbió la fricción de espera, manteniendo al usuario inmerso en el proceso de resolución de problemas.

De mayor relevancia analítica es la confirmación empírica de la Jerarquía Epistemológica programada en el asistente. Durante las pruebas, se observó que la inyección estricta del System Prompt forzó a la inteligencia artificial a subyugarse ante los algoritmos deterministas. DABU no "deduce" la presión de la bomba basándose en literatura teórica; la "lee" crudamente desde el Excel o el PLC, y posteriormente se limita a articular sintácticamente esa verdad. Esta disociación arquitectónica entre la matemática (herramientas en Python) y el lenguaje (LLM) es el núcleo tecnológico que valida al prototipo como una herramienta segura para la toma de decisiones industriales.

Respecto a la tasa de alucinación, el sistema fue evaluado mediante un protocolo de pruebas adversarias compuesto por $n = 30$ consultas diseñadas deliberadamente para inducir errores de interpretación (ej. solicitar valores fuera de rango o preguntas contradictorias sobre manuales). Durante estas 30 iteraciones, la tasa de alucinación numérica observada sobre la telemetría y reglas de negocio fue del 0% ($\hat{p} = 0$).

Con el propósito de dotar al estudio de rigor estadístico formal, se calculó el límite superior del intervalo de confianza al 95% ($1 - \alpha = 0.95$) mediante la regla de la sucesión de pruebas de Bernoulli:

$$IC_{95\%} = [0, 1 - (0.05)^{1/30}] \approx [0\%, 9.50\%] \quad (7)$$

Con un nivel de confianza del 95 %, el límite superior estimado para la tasa de error en las condiciones evaluadas es aproximadamente 9,5 %, lo cual aporta evidencia favorable sobre la

robustez del diseño híbrido dentro de dichas condiciones.

5.4 Validación frente a los objetivos planteados

La materialización de la arquitectura propuesta y los resultados experimentales documentados permiten contrastar y certificar el cumplimiento formal de los cuatro objetivos específicos que cimentaron esta investigación aplicada:

1. Establecimiento de Arquitectura PlantPax Distribuida: Se estructuró exitosamente un entorno hipervisor compuesto por Controladores (CCC-PLCS), Adquisición (CCC-DSDS1) y Servidores Históricos (CCC-HISTSERVER). Este ecosistema operó como la infraestructura habilitadora idónea para la integración del prototipo, aislando las capas lógicas y respetando los estándares ISA-95 e IEC 62443 de segregación de redes industriales.

2. Arquitectura Modular Backend (FastAPI): Se diseñó e implementó un servidor transaccional de alto rendimiento capaz de orquestar la ingesta asíncrona de datos desde las plataformas de Rockwell Automation. La separación de rutas operativas (Chat, Upload y PLC Status) permitió interrogar el historiador y los tags de manera centralizada sin degradar el desempeño del HMI.

3. Desarrollo del Agente Híbrido (Reglas + LLM): Se logró construir un motor de inferencia neuro-simbólico (LangGraph) que superó la principal falencia de las IA generativas puras. La inclusión de nodos algorítmicos que procesan Excel, PDFs y telemetría permitió al modelo cuantizado local (phi3:mini) contextualizar eventos con rigor técnico, emitiendo explicaciones causales precisas, deterministas y comprensibles en lenguaje natural.

4. Validación Ergonómica y Percepción de Utilidad del Prototipo (Escala SUS): En cumplimiento formal del Objetivo Específico 4 y siguiendo el protocolo metodológico definido en

la Sección 3.5, se evaluó cuantitativamente la percepción de utilidad y usabilidad del prototipo DABU en $N = 6$ evaluadores técnicos tras ejecutar las cinco tareas de prueba. El puntaje global SUS se calculó mediante la conversión estandarizada de respuestas:

$$\text{Puntaje SUS} = 2.5 \times \left[\sum_{i=1}^5 (x_{2i-1} - 1) + \sum_{i=1}^5 (5 - x_{2i}) \right] \quad (8)$$

Los resultados consolidados por dimensión se resumen en la Tabla 4:

Tabla 4. *Resultados consolidados de la evaluación de usabilidad y percepción de utilidad.*

Criterio de evaluación/ Dimensión	Puntaje promedio (1 a 5)	Percepción del evaluador
Ergonómica		
Facilidad de uso e intuitividad (Tareas 1 y 2)	4.6/5.0	Alta facilidad para formular preguntas en lenguaje natural sin entrenamiento previo
Reducción de carga cognitiva (UX Dark Mode)	4.8/5.0	Excelente ergonomía visual nocturna y calidad en el renderizado de tablas
Confianza en la información	4.5/5.0	Confianza en que los datos de telemetría provienen directamente del PLC.
Integridad y trazabilidad del historial (Tarea 5)	4.4/5.0	Buena capacidad de mantener el contexto en diagnósticos de fallas continuas
Puntaje global consolidado SUS	84.5/100	Calificación: Está por encima de la referencia (68)

Nota: la tabla fue elaborada a partir de los resultados obtenidos mediante la aplicación de la System Usability Scale (SUS), siguiendo la metodología descrita en [20], [21].

El puntaje global SUS de 84,5/100 se encuentra por encima del valor de referencia de 68 puntos, lo que evidencia una percepción favorable de la usabilidad del prototipo DABU dentro de la muestra evaluada.

6. Conclusiones y recomendaciones

6.1 Conclusiones generales del proyecto

El presente proyecto de investigación alcanzó la construcción y comprobación funcional

de un prototipo de asistente operativo inteligente (DABU) en un entorno emulado de pruebas industriales de alta fidelidad. A partir de los resultados empíricos documentados en el Capítulo 5, se concluye lo siguiente:

1. **Arquitectura Híbrida Neuro-Simbólica:** Se comprobó que la integración de un motor determinista en Python con grafos de estado en LangGraph reduce el riesgo de alucinación matemática en la lectura de telemetría SCADA, restringiendo el rol del modelo de lenguaje (LLM) exclusivamente a la traducción semántica y soporte conversacional.
2. **Cumplimiento de Ciberseguridad Zero Trust:** El despliegue local (Edge AI) del modelo cuantizado phi3:mini en el servidor virtualizado PlantPAx demostró que es posible implementar herramientas de analítica avanzada e inteligencia artificial en la Capa 6 (ISA-95) sin exponer la red OT a internet ni violar el perímetro de seguridad industrial.
3. **Ergonomía y Desempeño Operativo:** Las evaluaciones de usabilidad (Puntaje SUS de 84.5/100) y las métricas de latencia confirman que la plataforma responde eficientemente a las necesidades de triaje de alertas en Cuartos de Control Centralizado (CCR), ofreciendo una interfaz visual en modo oscuro que reduce la fatiga del operador.
4. **Resiliencia y Escala Arquitectónica:** El estudio comparativo de hardware evidenció que la solución es agnóstica al modelo lingüístico subyacente, operando de manera resiliente en CPUs virtualizadas restringidas y mostrando aceleración superior al doble con hardware dedicado con GPU.

6.2 Cumplimiento de los objetivos

El rigor metodológico aplicado durante el desarrollo del proyecto permitió no solo alcanzar, sino superar funcionalmente la totalidad de los objetivos específicos trazados en la

investigación:

Establecimiento de Arquitectura SCADA Distribuida (PlantPAX): Se concluye que la configuración de la DMZ (Zona Desmilitarizada) industrial mediante hipervisores independientes (CCC-PLCS, CCC-DSDS1, CCC-HISTSERVER) fue estructuralmente exitosa. El proyecto demostró que es plenamente factible desplegar clústeres de analítica avanzada superpuestos a redes OT, garantizando que el procesamiento intensivo del modelo no afecte en lo absoluto los estrictos tiempos de barrido ni comprometa el cumplimiento normativo de segregación de red exigido por los estándares ISA-95 e IEC 62443.

Arquitectura Modular de Integración (FastAPI): El desarrollo lógico del servidor backend comprobó que el uso de tecnologías web asíncronas (ASGI) es altamente idóneo para entornos industriales. La segmentación de los endpoints transaccionales resolvió la barrera histórica de interoperabilidad, permitiendo consultar bases de datos de FT Historian, evaluar umbrales del PLC y procesar documentación técnica bajo un único paraguas algorítmico, operando sin provocar bloqueos en el servidor.

Desarrollo del Agente Inteligente Híbrido: La concepción de un motor heurístico soportado por LangGraph alcanzó un cumplimiento excepcional. Se demostró empíricamente la efectividad del System Prompt y el enrutamiento lógico: el modelo de lenguaje cuantizado se limitó con estricta disciplina a su rol de intérprete semántico, mientras que las subrutinas informáticas (Tools) extrajeron resultados numéricos exactos. Esta dualidad neuro-simbólica es el logro central que otorga confiabilidad operativa a las explicaciones del agente.

Comprobación Funcional y Ergonomía del Prototipo: Las evaluaciones cuantitativas y cualitativas de la interfaz web superaron las expectativas. Las métricas de latencia de red, combinadas con una tasa observada de alucinación numérica del 0 % durante las 30 pruebas

adversarias frente a la lectura de datos duros, ratificaron la viabilidad técnica del sistema. Sumado a esto, las implementaciones de diseño ergonómico (modo oscuro de baja luminancia, semáforo de red asíncrono, dictado por voz y renderizado dinámico mediante Marked.js) elevaron a DABU al estándar visual y operativo requerido por los modernos Cuartos de Control (CCR).

6.3 Limitaciones identificadas

Desde la óptica de la rigurosidad científica, es imperativo reconocer las barreras tecnológicas e infraestructurales que delimitan el alcance actual de este desarrollo, las cuales funcionan como base para el entendimiento de sus fronteras operativas:

Restricciones de Hardware y Cuantización (Edge AI Constraints): La ejecución nativa de un LLM (On-Premise) sin el soporte de aceleración tensorial dedicada (Clústeres GPU) induce inevitables cuellos de botella en la fase de inferencia generativa (Time-to-Last-Token). Aunque el uso de modelos de parámetros compactos mitiga esta brecha, el sistema sacrifica profundidad en su ventana de contexto frente a infraestructuras alojadas en centros de cómputo en la nube, evidenciando que el Zero Trust exige una contraprestación en escalabilidad de hardware.

Fidelidad Dinámica de la Emulación Física: A pesar de la robustez del servidor CCC-PLCS en la emulación de las lógicas de alarma y control, la ausencia de una planta piloto con instrumentación física real (sensores, actuadores, bombas) imposibilita la introducción de ruido de señal de alta frecuencia y perturbaciones termodinámicas complejas, por ende, el sistema fue evaluado frente a datos históricos inmutables y simulaciones estructuradas, lo cual representa una limitación para validar la reactividad del LLM frente al desorden aleatorio de los fluidos de campo.

Restricción Arquitectónica de Unidireccionalidad (Read-Only): Con el objetivo prioritario de preservar la seguridad operativa de la planta, el backend de DABU fue configurado

algorítmicamente para operar como un nodo analítico de solo lectura. Actualmente, la inteligencia artificial está imposibilitada por diseño para alterar puntos de ajuste (Setpoints), rearmar alarmas o comandar actuadores. Actúa exclusivamente como asesor cognitivo, dejando las acciones de control en manos del operador.

6.4 Recomendaciones y trabajos futuros

Con base en la madurez tecnológica demostrada por la arquitectura de DABU, se proponen las siguientes líneas de investigación para expandir las fronteras del estado del arte en la automatización inteligente:

1. **Convergencia con Modelos Predictivos y Gemelos Digitales (Digital Twins):** Se recomienda encarecidamente escalar la capacidad analítica de DABU mediante la integración de algoritmos de Machine Learning tradicionales (e.g., Redes Neuronales o Recurrentes LSTM o Modelos de Regresión Bosque Aleatorio). Esta integración permitiría evolucionar al asistente de una herramienta diagnóstica reactiva a un orquestador proactivo de mantenimiento predictivo, capaz de alertar sobre la degradación de un rodamiento antes de que los límites de alarma del SCADA sean vulnerados.
2. **Despliegue y Validación con Hardware en Lazo (Hardware-in-the-Loop - HIL):** Para avanzar hacia un nivel de madurez tecnológica superior (TRL 5-6), las futuras iteraciones de este proyecto deben prescindir de los emuladores lógicos y conectar el backend FastAPI directamente a PLCs físicos acoplados a bancos de prueba hidráulicos reales. Esto sometería a las Tools del agente inteligente al rigor del ruido eléctrico, la latencia de protocolos industriales (Modbus TCP / OPC UA) y el muestreo de alta densidad.
3. **Transición hacia el Control Bidireccional Seguro (Closed-Loop AI):** El nivel superior de

la automatización inteligente reside en la capacidad del modelo para ejecutar acciones preventivas autónomas. Se sugiere como línea de investigación a largo plazo el desarrollo de protocolos de ciberseguridad transaccional basados en Control de Acceso por Roles (RBAC) y tokens OAuth2. Esto permitiría instaurar un ciclo de "Human-in-the-Loop", donde DABU diagnostique la falla, estructure el comando (ej. aislar válvula de descarga), y lo someta a una interfaz de autorización biométrica o criptográfica que el operador humano deba validar, cerrando así la brecha entre el análisis y la ejecución.

Referencias

- [1] OCENSA, “Informe de gestión y sostenibilidad”, 2024.
- [2] ECOPETROL, “Informe integrado de gestión del año 2024”, 2024.
- [3] S. Nazir, S. Patel, y D. Patel, “Autoencoder Based Anomaly Detection for SCADA Networks”, *International Journal of Artificial Intelligence and Machine Learning*, vol. 11, pp. 83–99, jul. 2021, doi: 10.4018/ijaiml.20210701.oa6.
- [4] International Society of Automation, *ANSI/ISA-95.00.01-2010 (IEC 62264-1 Mod) Enterprise-Control System Integration - Part 1: Models and Terminology*. International Society of Automation [ISA], 2010.
- [5] Rockwell Automation, “PlantPAx Distributed Control System Selection Guide (Publication PROCES-SG001). ”, 2023.
- [6] International Society of Automation (ISA), *ISA-TR106.00.01-2013: Procedure Automation for Continuous Process Operations - Models and Terminology*. Research Triangle Park, NC: ISA. 2013.
- [7] International Electrotechnical Commission, “International Electrotechnical Commission (IEC). (2018). IEC 62443-3-3: Industrial communication networks - Network and system security - Part 3-3: System security requirements and security levels. Geneva: IEC.”, 2018.
- [8] B. Ramsay, K. Hasan, y I. Moyes, “Electric power system alarm management with an OOP toolkit”, *Eng. Appl. Artif. Intell.*, vol. 8, pp. 461–467, 1995, doi: 10.1016/0952-1976(95)00018-V.
- [9] S. Russell y P. Norvig, *Artificial Intelligence: A Modern Approach*, Pearson., vol. 4. Pearson, 2020.
- [10] K. Geol, “Zero-Trust-Based Cybersecurity Framework for Large-Scale Corporate Networks”, *International Journal of Science and Research (IJSR)*, pp. 1995–2001, dic. 2025, doi: 10.21275/sr251217162021.
- [11] A. N. M. G. Mosharraf, “AI-Driven Anomaly Detection On SCADA-Bearing Fiber Networks: A Machine Learning Framework for Real-Time Intrusion

- Detection in Electric Utility Environments”, *American Journal of Data Science and Analytics*, vol. 05, pp. 125–162, dic. 2024, doi: 10.63125/z5wx8t42.
- [12] D. Schall *et al.*, “Safe integration of Large Language Models into industrial process control: a multi-agent architecture with P&ID-grounded validation”, pp. 1–23, 2025.
- [13] S. Majumder *et al.*, “Exploring the capabilities and limitations of large language models in the electric energy sector”, junio de 2024, *Cell Press*. doi: 10.1016/j.joule.2024.05.009.
- [14] S. Purohit, “Bridging IT and OT: Cybersecurity Risks, Zero-Trust Solutions, and Industrial Resilience in Oil & Gas Downstream”, *ASEAN Journal of Scientific and Technological Reports*, vol. 29, may 2026, doi: 10.55164/ajstr.v29i5.262008.
- [15] S. V. Rao, “Using a Digital Twin in Predictive Maintenance”, *Journal of Petroleum Technology*, vol. 72, pp. 42–44, ago. 2020, doi: 10.2118/0820-0042-jpt.
- [16] Dr. A. Dagur, S. Singh, y T. P. Singh, “A Hybrid AI Approach for Predictive Maintenance in Manufacturing Using Machine Learning and Digital Twin-Based Failure Forecasting”, *International Journal of Research Publication and Reviews*, vol. 7, may 2026, doi: 10.55248/gengpi.07.0526.d13256.
- [17] Y. Xia, N. Jazdi, y M. Weyrich, “Applying Large Language Models for Intelligent Industrial Automation”, *atp magazin*, vol. 66, pp. 62–71, jul. 2024, doi: 10.17560/atp.v66i6-7.2739.
- [18] J. Vyas and M. Mercangöz, “Autonomous Industrial Control using an Agentic Framework with Large Language Models”, in *IFAC-PapersOnLine*, vol. 58, pp. 349–354, Jun. 2025, doi: 10.1016/j.ifacol.2025.07.170.
- [19] J. W. Vasquez Capacho, A. Subias, L. Travé-Massuyés, and F. Jimenez, “Alarm management via temporal pattern learning”, *Engineering Applications of Artificial Intelligence*, vol. 65, pp. 506–516, Oct. 2017, doi: 10.1016/j.engappai.2017.07.008.
- [20] J. Brooke, “SUS: A 'quick and dirty' usability scale”, *Usability Evaluation in Industry*, Taylor & Francis, vol. 189, pp. 189–194, 1996.

- [21] M. D. R. Sevilla-Gonzalez et al., “Spanish Version of the System Usability Scale for the Assessment of Electronic Tools: Development and Validation”, *JMIR Human Factors*, vol. 7, no. 4, p. e21161, 2020, doi: 10.2196/21161.

Apéndices

Apéndice A. Cuestionario de evaluación de usabilidad y guía de puntuación y uso para el investigador

CUESTIONARIO DE EVALUACIÓN DE USABILIDAD

System Usability Scale (SUS)

Aplicado al agente operativo inteligente DABU

Objetivo. Evaluar la usabilidad percibida de DABU después de la interacción del participante con el escenario de prueba definido.

Datos de la aplicación

Código del
participante:

Fecha:

____ / ____ / ____

Rol / perfil:

Escenario de
prueba:

Instrucciones

Responda según su experiencia inmediata. Marque una sola opción por afirmación. En este formato, “herramienta” se refiere al agente DABU. Si no puede responder un ítem, marque 3.

1 totalmente en desacuerdo · 2 En desacuerdo · 3 Neutral · 4 De acuerdo · 5 Totalmente de acuerdo

1. Me gustaría usar esta herramienta frecuentemente. ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
2. Considero que esta herramienta es innecesariamente compleja. ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
3. Considero que la herramienta es fácil de usar. ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
4. Considero necesario el apoyo de personal experto para poder utilizar esta herramienta. ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
5. Considero que las funciones de la herramienta están bien integradas. ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
6. Considero que la herramienta presenta muchas contradicciones. ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
7. Imagino que la mayoría de las personas aprenderían a usar esta herramienta rápidamente. ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
8. Considero que el uso de esta herramienta es tedioso. ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
9. Me sentí muy confiado al usar la herramienta. ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
10. Necesité saber bastantes cosas antes de poder empezar a usar esta herramienta. ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Observaciones opcionales:

GUÍA DE PUNTUACIÓN Y USO PARA EL INVESTIGADOR

Aplicación. Administre el SUS después de que el participante haya usado DABU y, preferiblemente, antes de la entrevista o discusión posterior. Solicite respuesta para los 10 ítems.

Cálculo del puntaje SUS

- Ítems impares (1, 3, 5, 7 y 9): respuesta – 1.
- Ítems pares (2, 4, 6, 8 y 10): 5 – respuesta.
- Sume las 10 contribuciones (rango 0–40).
- Multiplique la suma por 2,5 (puntaje final 0–100).

Importante: el SUS no es un porcentaje. El resultado es una medida global de usabilidad percibida; los ítems individuales no deben interpretarse como puntajes independientes.

Plantilla de cálculo por participante

Ítem	Respuesta	Regla	Contribución
1		Respuesta – 1	
2		5 – respuesta	
3		Respuesta – 1	
4		5 – respuesta	
5		Respuesta – 1	
6		5 – respuesta	
7		Respuesta – 1	
8		5 – respuesta	
9		Respuesta – 1	
10		5 – respuesta	
Suma de contribuciones:		× 2,5	SUS = / 100

Interpretación y alcance

Como referencia comparativa, el promedio histórico reportado para SUS se sitúa alrededor de 68 puntos. Un valor superior suele describirse como por encima del promedio, pero no debe tratarse como un umbral absoluto. Para DABU, el SUS debe complementarse con métricas del agente como cumplimiento de tareas, latencia/tiempo de respuesta, coherencia o exactitud de las respuestas y utilidad percibida.

Fuentes metodológicas

Brooke, J. (1996). SUS: A “quick and dirty” usability scale. En P. W. Jordan, B. Thomas, B. A. Weerdmeester y A. L. McClelland (Eds.), *Usability Evaluation in Industry*. Taylor & Francis.

Sevilla-Gonzalez, M. D. R., et al. (2020). Spanish Version of the System Usability Scale for the Assessment of Electronic Tools: Development and Validation. *JMIR Human Factors*, 7(4), e21161. <https://doi.org/10.2196/21161>

Borsci, S., Malizia, A., Schmettow, M., et al. (2022). The Chatbot Usability Scale: The Design and Pilot of a Usability Scale for Interaction with AI-Based Conversational Agents. *Personal and Ubiquitous Computing*, 26, 95–119. <https://doi.org/10.1007/s00779-021-01582-9>

Nota: Los 10 ítems corresponden a la versión estándar en español validada para herramientas electrónicas. Se conserva “herramienta” en las afirmaciones y se define en las instrucciones que se refiere a DABU.