

**UNIVERSIDAD TÉCNICA FEDERICO SANTA
MARÍA**

**DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO - CHILE**



**“INTEGRACIÓN DE MACHINE LEARNING
PARA LA INTERPRETACIÓN Y
EXPLICACIÓN DE LOS SISTEMAS DE
CONTROL EN DCS”**

DANIEL SEBASTIÁN LAZO LARRAÍN

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO
CIVIL ELECTRÓNICO MENCIÓN CONTROL AUTOMÁTICO**

**PROFESOR GUÍA: DR. JUAN CARLOS AGÜERO.
PROFESOR CORREFERENTE: IVAN RIVERO**

Marzo - 2026



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: Integración de Machine Learning para la interpretación y explicación de los sistemas de control DCS

Nombre del candidato(a): __Daniel Sebastian Lazo Larraín__

Carrera / Grado: ____Ingeniero Civil Electrónico____

Campus: ____Casa Central____ **Departamento:** ____Electrónica____

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, __Juan Carlos Agüero Vásquez, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: ____10/06/2026____ **Firma:** 

Estudiante o Candidato(a):

Fecha: ____10/06/2026____ **Firma:** 

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

AGRADECIMIENTOS

Me es necesario comenzar agradeciendo el apoyo incondicional de mi familia siempre en las buenas y en las malas, en especial a mi Madre Ana Larraín, la cual sin sus llamados constantes, su preocupación a todas horas del día y su gran amor que siempre me entrega sin importar la distancia, fueron el motor para continuar en todos los estudios. Obviamente agradecer también a mi Padre Christian Lazo el cual estuvo preocupado y dando animo a mis capacidades, a mis hermanos Javier Lazo y Maximiliano Lazo que con sus conversaciones banales me dieron el apoyo y crecimiento para seguir aprendiendo y saber descansar cuando mas lo necesitaba. No puedo no mencionar a mi tía yaya y lita por siempre estar preocupadas por mi sin importar donde esté.

Cambien quiero agradecer a todos mis amigos de Santiago Lelo, Félix, Joe y a todo mi grupo de amigos que cada uno apporto demasiado en todo el esfuerzo que conlleva irse lejos de la familia, son mas de 10 años compartiendo vivencias apoyándonos en el estudio, jugando y riendo juntos que sin su apoyo a la lejanía hubiera sido mucho mas difícil. Gracias por tomar cada una de mis llamadas pidiendo consejos y darme animo en las locuras que me forjaron a día de hoy. Gracias al Seba, Pablo, Maca, Isi, Domi, amigos y amigas de Plan Común y Electrónica, todos fueron una motivación y un gran crecimiento como persona conociendo distintas realidades y apoyándonos todos juntos, esto no es una carrera si no que un camino que recorreremos juntos como futuros colegas.

También quiero agradecer al área de control automático de Collahuasi por darme la oportunidad de poder realizar mi memoria allí con ellos, su gran sentido de compañerismo en el equipo y como me enseñaron que un pequeño grupo es un gran pilar fundamental en todo un proceso sensible y profundo en una gran compañía.

Finalmente agradecer Al Centro Avanzado de Ingeniería Eléctrica y Electrónica ANID AC3E CIA250006, y al Proyecto Interno USM 2025 PI_LIR_25_12

RESUMEN

La creciente complejidad de los sistemas de control distribuido (DCS) en la industria minera moderna ha generado una brecha significativa entre la lógica de control realmente implementada en los controladores y la documentación técnica disponible para su comprensión y mantenimiento. Esta problemática impacta directamente en los tiempos de diagnóstico, la seguridad operacional y los costos asociados a detenciones no programadas, especialmente en operaciones de gran escala como la Compañía Minera Doña Inés de Collahuasi (CMDIC).

El presente trabajo de titulación propone el diseño e implementación de un asistente virtual inteligente, denominado *Agente DCS*, orientado a la interpretación y explicación de la lógica de control en sistemas DCS, utilizando modelos de lenguaje grandes (LLMs) ejecutados de forma local y técnicas avanzadas de extracción de información estructurada. La solución se enfoca específicamente en archivos de exportación FHX de la plataforma Emerson DeltaV, los cuales son transformados desde su representación textual jerárquica hacia un modelo basado en grafos que permite capturar explícitamente las relaciones lógicas entre bloques funcionales, señales y enclavamientos.

Sobre esta base estructurada, se implementa una arquitectura de recuperación de información y generación aumentada (RAG), que permite realizar consultas en lenguaje natural y generar explicaciones técnicas coherentes, actualizadas y contextualizadas, sin requerir acceso directo a las estaciones de ingeniería ni a infraestructuras externas de cómputo. El sistema ha sido diseñado para operar completamente en modo offline, respetando las restricciones de ciberseguridad industrial y las limitaciones de hardware propias de entornos operacionales reales.

Los resultados obtenidos demuestran una reducción significativa en los tiempos

de análisis y diagnóstico de fallas, así como una mejora sustancial en la trazabilidad y comprensión de la lógica de control, incluso para personal no especialista. Asimismo, el análisis económico evidencia un impacto positivo en la reducción del OPEX operacional y del MTTR, validando la viabilidad técnica y económica de la solución propuesta. Finalmente, se identifican oportunidades de mejora orientadas a la escalabilidad del sistema, la integración multi-marca y la evolución hacia modelos de lenguaje de mayor capacidad.

Abstract

The increasing complexity of Distributed Control Systems (DCS) in the modern mining industry has generated a significant gap between the control logic implemented in controllers and the technical documentation available for its understanding and maintenance. This gap directly affects fault diagnosis times, operational safety, and costs associated with unplanned downtime, particularly in large-scale operations such as Compañía Minera Doña Inés de Collahuasi (CMDIC).

This thesis presents the design and implementation of an intelligent virtual assistant, named *DCS Agent*, aimed at interpreting and explaining control logic in DCS environments using locally deployed Large Language Models (LLMs) and advanced structured information extraction techniques. The proposed solution focuses on FHX export files from the Emerson DeltaV platform, which are transformed from their hierarchical textual representation into a graph-based model that explicitly captures logical relationships among functional blocks, signals, and interlocks.

Based on this structured representation, a Retrieval-Augmented Generation (RAG) architecture is implemented, enabling natural language queries and the generation of coherent, up-to-date, and context-aware technical explanations without requiring direct access to engineering workstations or external computing infrastructures. The system operates entirely in offline mode, complying with industrial cybersecurity policies and the hardware constraints typical of real operational environments.

The results demonstrate a significant reduction in analysis and fault diagnosis times, as well as a substantial improvement in the traceability and understanding of control logic, even for non-specialist personnel. Furthermore, the economic analysis indicates a positive impact on operational expenditure (OPEX) and a reduction in Mean Time to Repair (MTTR), validating the technical and economic feasibility of the proposed solution. Finally, opportunities for future improvements related to system scalability, multi-vendor integration, and the adoption of higher-capacity language models are identified.

GLOSARIO

Mayúsculas

CGR	Ciclo de Gestión de Riesgo aplicado en las operaciones de CMDIC.
CMDIC	Compañía Minera Doña Inés de Collahuasi.
CNN	Redes Neuronales Convolucionales (<i>Convolutional Neural Networks</i>).
DCS	Sistema de Control Distribuido (<i>Distributed Control System</i>), arquitectura de control para procesos industriales complejos.
FBD	Diagrama de Bloques de Funciones (<i>Function Block Diagram</i>), lenguaje gráfico definido en la norma IEC 61131-3.
FHX	Formato de texto jerárquico para la exportación de bases de datos de Emerson DeltaV.
FP16	Formato de coma flotante de media precisión (16 bits) que representa la precisión original de los pesos del modelo antes de ser cuantizados.
Gen-AI	Inteligencia Artificial Generativa.
GGUF	<i>GPT-Generated Unified Format</i> , formato de archivo optimizado para la carga y ejecución eficiente de LLMs en hardware local.
IEC	Comisión Electrotécnica Internacional (<i>International Electrotechnical Commission</i>).
INT4	Cuantización de modelos a enteros de 4 bits para optimización de memoria.

I/O	Señales de Entrada y Salida (<i>Input/Output</i>) de los controladores.
ISA	Sociedad Internacional de Automatización (<i>International Society of Automation</i>).
LLMs	Modelos de Lenguaje Grande (<i>Large Language Models</i>).
MTTR	Tiempo Medio de Reparación (<i>Mean Time To Repair</i>).
OPEX	Gastos Operativos (<i>Operational Expenditures</i>).
PID	Algoritmo de control Proporcional, Integral y Derivativo.
PLC	Controlador Lógico Programable (<i>Programmable Logic Controller</i>).
PROMPT	Instrucción, comando o contexto estructurado proporcionado a un modelo de lenguaje para guiar su proceso de razonamiento y generación.
RAFT	Ajuste Fino Aumentado por Recuperación (<i>Retrieval-Augmented Fine-Tuning</i>).
RAG	Generación Aumentada por Recuperación (<i>Retrieval-Augmented Generation</i>).
ROI	Retorno de Inversión (<i>Return on Investment</i>).
SFC	Grafos de Función Secuencial (<i>Sequential Function Chart</i>).
ST	Texto Estructurado (<i>Structured Text</i>), lenguaje de alto nivel para DCS.
VATS	Verificación y Autorización para Trabajo Seguro en CMDIC.

Índice general

<i>Abstract</i>	IV
<i>Glosario</i>	V
<i>1.. Introducción</i>	8
1.1. Estado del arte y Brecha Tecnológica	9
1.1.1. Resumen del Estado del Arte de los Sistemas de Control Dis- tribuido (DCS)	10
1.1.2. Modelos de Lenguaje para la Explicación de Código	11
1.1.3. Recuperación de información visual vía diagramas	12
1.1.4. Técnicas de Optimización de entrenamiento: RAG y RAFT	13
1.2. Objetivos del Proyecto	14
1.2.1. Objetivo General	14
1.2.2. Objetivo Específicos	14
1.3. Objetivos, Alcances y Limitaciones	14
<i>2.. Marco Teórico</i>	17
2.1. Contexto Operacional y Normativo	17
2.1.1. Descripción del Proceso Minero en CMDIC	17
2.2. Gestión de Riesgos	18
2.2.1. Ciclo de Gestión de Riesgo (CGR)	19
2.3. Sistemas de Control Distribuido (DCS)	34
2.3.1. Arquitectura Emerson DeltaV	35
2.3.2. Estándar IEC 61131-3 y Lenguajes de Programación	36

2.3.3.	Estructura de Archivos de Exportación (FHX)	37
2.4.	Inteligencia Artificial Generativa y LLMs	38
2.4.1.	Modelos de Lenguaje Grande (LLMs) y Transformadores . . .	39
2.4.2.	Desafíos: Alucinación y Ventana de Contexto	40
2.4.3.	Optimización y Cuantización para Hardware Limitado	42
2.5.	Recuperación de Información y Grafos	43
2.5.1.	RAG (Retrieval-Augmented Generation)	43
2.5.2.	Teoría de Grafos aplicada a Lógica de Control	44
3..	<i>Diseño de la Solución</i>	46
3.1.	Arquitectura General del Sistema	46
3.1.1.	Capas del Sistema y Procesos de Integración	47
3.1.2.	Infraestructura y Entorno de Ejecución	49
3.2.	Interfaz de Adquisición de Datos Crudos y Preprocesamiento Estructural	49
3.2.1.	Metodología de Extracción y Segmentación de Datos	50
3.2.2.	Transformación a Objeto Jerárquico Estructurado	51
3.2.3.	Manejo Crítico de Inconsistencias y Lógica Probabilística . .	52
3.3.	Modelado Estructural: Conversión a Grafo	53
3.3.1.	Caracterización Profunda de Nodos	53
3.3.2.	Aristas	54
3.3.3.	Modelado de Compuertas Lógicas como Puentes de Decisión	55
3.3.4.	Implementación Algorítmica con NetworkX	55
3.4.	Diseño de la Lógica del Agente y Orquestación RAG	56
3.4.1.	Estrategia de Búsqueda y Recuperación Basada en Grafos . .	56
3.4.2.	Diseño del Prompt Engineering	57
3.4.3.	Orquestación y Razonamiento Iterativo (Sub-consultas)	58
3.4.4.	Eficiencia en el Uso de Tokens y Cuantización	58
3.5.	Arquitectura de Visualización y Experiencia de Usuario (UI/UX) . . .	61
3.5.1.	Selección del Framework: Streamlit como Motor Visual . . .	62

3.5.2.	Diseño Funcional y Componentes de la Interfaz	62
3.5.3.	Galería de Implementación y Capturas de Diseño	63
3.6.	Síntesis Metodológica del Diseño	66
4..	<i>RESULTADOS Y CONCLUSIONES</i>	68
4.1.	Casos de Estudio	68
4.2.	Evaluación de Desempeño y Métricas	69
4.2.1.	Análisis de Reducción de Tiempos (Eficiencia)	70
4.2.2.	Precisión y Fidelidad Técnica (Exactitud)	71
4.2.3.	Impacto en la Continuidad Operacional (MTTR)	73
4.2.4.	Reducción de Tiempos de Diagnóstico y Análisis	74
4.3.	Análisis de Valor Agregado para CMDIC: Impacto Económico y Ope- racional	75
4.3.1.	Determinación del Valor de la Hora-Humanas (HH) y Respal- do de Mercado	75
4.3.2.	Comparativa de Costos por Análisis Integral (Caso Crítico) . .	76
4.3.3.	Proyección de Impacto en el OPEX Anual	76
4.3.4.	Análisis de Costo de Oportunidad y MTTR	77
4.3.5.	Justificación del Retorno de Inversión (ROI)	78
4.4.	Oportunidades de Mejora y Trabajo Futuro	79
4.4.1.	Escalamiento de Recursos Computacionales y Capacidad de Cómputo	79
4.4.2.	Evolución hacia Modelos de Lenguaje de Gran Escala	80
4.4.3.	Optimización de los Métodos de Entrenamiento y Recupera- ción (RAFT)	80
4.4.4.	Integración Multi-Marca y Estandarización de Datos	80
4.4.5.	Integración interna a Controladores DCS	81
4.4.6.	Derechos de Autor, Propiedad Intelectual y Restricciones de Uso	81

4.5. Conclusiones	82
-----------------------------	----

Índice de figuras

1.1. Descripción LEDEX	12
2.1. Ciclo de Gestión de Riesgo CMDIC.	19
2.2. Informe de aprendizaje.	22
2.3. Mapa de Procesos	25
2.4. Ejemplo de matriz de riesgo	27
2.5. Jerarquía de Controles.	28
2.6. Portada del Instructivo de procedimiento Estandarización de la ejecución.	30
2.7. Verificación y Autorización para el uso del sistema VATS.	34
3.1. Arquitectura General y Flujo de Datos del Agente DCS.	46
3.2. Flujo de identificación	52
3.3. Grafo generado de conexiones que regulan enclavamiento de Temperatura	56
3.4. Rendimiento de la GPU AL generar respuesta	61
3.5. Interfaz Inicial del Agente DCS con sus opciones	63
3.6. Archivos almacenados en la base de datos	64
3.7. Display de extracción de archivo y segmentación	65
3.8. Interfaz de consultas sobre tags y conexiones	65
4.1. Generación grafo y lista de tags relevantes ZAHH (Alarma de posicionamiento Alto Alto	70
4.2. Consulta de lógica ante MolinoDet	72

4.3. Respuesta ante consulta de Lógica y señal de salida	72
4.4. Pregunta ante Enclavamiento	73
4.5. Respuesta ante Enclavamiento	74

Índice de cuadros

4.1. Comparativa de Tiempos de Ejecución: Proceso Manual vs. Agente DCS.	71
4.2. Benchmark de Referencia para Costos de Ingeniería Senior en Minería.	76
4.3. Comparativa Económica de Diagnóstico: Manual vs. Agente DCS. . .	76

1. INTRODUCCIÓN

Durante el ultimo tiempo la industria minera se encuentra en un punto acelerado de transformación digital, buscando integrar tecnologías de la industria para optimizar procesos y aumentar la seguridad. En este contexto, la Compañía Minera Doña Inés de Collahuasi, ubicada en la región de Tarapacá, opera sistemas de control basados en plataformas DCS. Estos sistemas son el cerebro de toda la operación, siendo encargados de gestionar desde chancadores hasta procesos de flotación. Específicamente, la infraestructura de control se sustenta en la arquitectura ABB 800xA, Emerson Delta V y Siemens, con un estándar robusto en la minería mundial. Sin embargo, la escala de Collahuasi implica la gestión de miles de señales de entrada y salida (I/O) y cientos de lazos de control interconectados. Esta densidad de información convierte a la red de control en un sistema altamente acoplado, donde una modificación menor en un bloque funcional puede tener grandes repercusiones en subsistemas adyacentes, haciendo que la trazabilidad manual sea una tarea propensa al error y comprensión limitada.

Históricamente, la lógica de control de estos sistemas se documentaban en "Filosofías de control", documentos técnicos escritos en lenguaje natural que explica el funcionamiento esperado y descriptivo de los controladores. Sin embargo, con el paso de los años y las continuas modificaciones operativas, se ha generado una brecha significativa entre la documentación estática y el código real que se ejecuta en los controladores. El formato nativo de exportación de estos sistemas, en específico del Emerson Delta V, es conocido como FHX, contiene la definición textual de la lógica gráfica. No obstante, su estructura no está diseñada para ser legible por humanos de forma lineal, sino para ser importada por el software propietario. Esto genera una ba-

rrera de entrada técnica: la única forma de interpretar la lógica es teniendo acceso a la estación de ingeniería y navegando visualmente por los diagramas, lo cual no siempre es posible desde redes corporativas o para personal externo.

Actualmente, la interpretación y auditoria de la lógica de control en los sistemas de control DCS de Collahuasi presenta una complejidad elevada. Los ingenieros y mantenedores se enfrentan a códigos extensos, principalmente en lenguajes gráficos como Function Block Diagram (FBD) o texto estructurado, cuya comprensión requiere un alto nivel de especialización y tiempo.

El problema central radica en la desactualización de la documentación, filosofías de control. Cuando un ingeniero necesita entender por qué un equipo se detuvo o cómo funciona un enclavamiento específico, no puede confiar plenamente en los manuales antiguos. Esto obliga a realizar una ingeniería inversa manual sobre el código fuente, un proceso que genera un consumo de una gran cantidad de horas-humanas (HH) de los especialistas, aumentando el tiempo de respuestas ante las fallas ocurridas durante operación e incrementa el riesgo de errores humanos en la interpretación de lógica de seguridad críticas. Esta dependencia de la ingeniería inversa manual no solo afecta a los costos operativos (OPEX), sino que impacta directamente en la disponibilidad de la planta. Cada hora adicional dedicada a entender el código durante una detención no planificada se traduce en pérdidas de producción significativas. Además, la falta de una herramienta de consulta rápida limita la capacidad de los operadores para realizar análisis preventivos, restringiendo la gestión del mantenimiento a un enfoque reactivo.

1.1. Estado del arte y Brecha Tecnológica

Tradicionalmente, la solución a la problemática de la documentación desactualizada ha sido la contratación de servicios de consultoría externa para la revisión manual de códigos, o la asignación de ingenieros internos para realizar ingeniería inversa.

Ambos procesos son costosos, lentos y propensos a quedar obsoletos ante cualquier cambio menor en la planta.

En el ámbito de la inteligencia artificial moderna, han surgido avances significativos que abordan parcialmente estos desafíos. A continuación, se analizan las investigaciones más relevantes y su desconexión con la realidad operativa de un sistema DCS.

1.1.1. Resumen del Estado del Arte de los Sistemas de Control Distribuido (DCS)

La evolución de los sistemas de control industrial ha transitado desde mecanismos puramente mecánicos hasta arquitecturas cognitivas basadas en datos. A continuación, se presenta una síntesis de la línea de tiempo tecnológica que define el estado del arte de los DCS:

- **Década de 1970 - El Nacimiento del Control Digital:** Antes de esta era, las plantas se controlaban de forma neumática o mediante paneles analógicos centralizados. En 1975, con la introducción del Honeywell TDC 2000 y el Yokogawa CENTUM, nace el concepto de DCS. Por primera vez, el control se distribuye en múltiples microprocesadores interconectados, permitiendo que el fallo de un controlador no detuviera la planta completa.
- **Décadas de 1980 y 1990 - Protocolos y Estándares:** Esta etapa estuvo marcada por la transición hacia interfaces gráficas (HMI) y la aparición de protocolos de comunicación. Se pasó de señales de corriente simple (4-20 mA) a buses de campo inteligentes como HART, Foundation Fieldbus y Profibus. Los DCS comenzaron a integrarse con computadoras personales y sistemas operativos comerciales, facilitando el almacenamiento de históricos de datos.
- **Años 2000 a 2015 - Convergencia y Gestión de Activos:** Los sistemas evolucionaron para monitorear la “salud” de los equipos. Aparecen plataformas como Emerson DeltaV o 800XA ABB, que integran diagnósticos predictivos y

ciberseguridad. La arquitectura se vuelve modular, permitiendo expansiones en caliente y una integración profunda con sistemas de gestión empresarial (ERP).

- **La Actualidad (Industria 4.0) - Sistemas Cognitivos y Big Data:** El estado del arte actual se define por la convergencia IT/OT. Los DCS modernos interactúan con entornos de computación en el borde y nubes privadas. El desafío actual —donde se inserta este proyecto— es la **interpretabilidad de la complejidad**. Con más de 500 controladores y miles de señales, el flujo de datos ha superado la capacidad de análisis manual, abriendo la puerta a la Inteligencia Artificial Generativa para actuar como una capa de interpretación lógica y soporte de decisiones.

1.1.2. Modelos de Lenguaje para la Explicación de Código

La capacidad de los LLMs para interpretar software ha sido ampliamente estudiada. Una de las investigaciones más relevantes es la propuesta de **LEDEX (Jiang et al., 2024)**[1], la cual aborda la generación de explicaciones de código y la depuración automática o self-debugging.

La metodología utilizada en LEDEX se basa en entrenar al modelo no solo para completar código, sino para identificar errores y generar una explicación en lenguaje natural del fallo antes de corregirlo. Los autores generaron un dataset masivo de pares código error-corregido, logrando que el modelo aprenda la relación causal entre sintaxis defectuosa y su corrección lógica. Sus resultados demostraron una mejora significativa en la tasa de éxito de la depuración en lenguajes como Python y Java, superando a modelos base sin este entrenamiento específico.

Sin embargo, la arquitectura LEDEX y estudios similares asumen que el código es una secuencia de texto lineal. En el caso de los sistemas DCS, la lógica se rige por el estándar **IEC 61131-3**[3], donde la ejecución no es secuencial, si no basada en el flujo

de datos entre bloques interconectados, compuertas lógicas, grafos y texto estructurado. Los LLMs entrenados con la metodología LEDEX fallan al intentar interpretar la topología de un archivo .fhx, ya que carecen de la capacidad nativa para entender la posición de un bloque en el texto no dicta su orden de ejecución, sino sus conexiones.

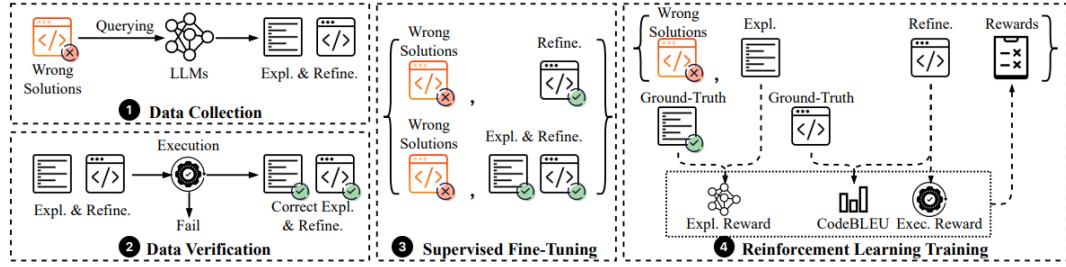


Fig. 1.1: Descripción LEDEX

1.1.3. Recuperación de información visual vía diagramas

Dado que los sistemas DCS como Emerson DeltaV se programan visualmente mediante interfaces gráficas, una aproximación lógica sería el uso de visión por computador. En esta línea, **Betancourt et al.** [2] exploraron el reconocimiento de diagramas de flujo mediante redes neuronales convolucionales (CNN).

Su investigación propone un pipeline que inicia con el preprocesamiento de imágenes de diagramas manuscritos o escaneados, aplicando filtros de binarización y esqueletización. Posteriormente, una red CNN clasifica las formas geométricas y reconstruye la lógica subyacente. Este trabajo es relevante pues valida teóricamente que la lógica de control debe estructurarse como una topología de nodos conectados.

Si bien este enfoque es válido conceptualmente, la técnica de visión artificial resulta ineficiente para la auditoría de sistemas DCS modernos donde la lógica ya existe en formato digital (.fhx). Sin embargo, cabe recalcar un aspecto crucial rescatado de esta metodología: **la representación basada en grafos y conexiones**. Aunque nuestra implementación descarta la lectura de imágenes, adopta esta estructura visual para abstraer la complejidad del código fuente. Esto permite generar diagramas de flujo interactivos que facilitan la comprensión de los procesos a profesionales de otras dis-

ciplinas, quienes pueden interpretar visualmente la lógica sin necesidad de dominar la sintaxis críptica de los controladores.

1.1.4. Técnicas de Optimización de entrenamiento: RAG y RAFT

Para dotar a los modelos de conocimiento específico sin entrenarlos desde cero, la industria ha adoptado técnicas de RAG. Recientemente, la propuesta de **RAFT (Retrieval Augmented Fine Tuning)** [4] ha ganado tracción al plantear un enfoque híbrido.

RAFT propone entrenar al modelo específicamente para comprender y filtrar información recuperada. A diferencia del RAG tradicional, donde el modelo recibe ciegamente los documentos entregados, RAFT entrena al LLM presentándole documentos que contienen la respuesta correcta (Oracle documents), mezclados con documentos irrelevantes (Distractor documents). El objetivo es que el modelo aprenda a descartar el ruido y citar solo la información verídica. Para realizar este entrenamiento sin la necesidad de modificar la totalidad de los pesos o crear una IA desde cero, se utilizan métodos eficientes como QLoRA (Quantized Low Rank Adapters), el cual reduce considerablemente el consumo de recursos al trabajar solo sobre matrices de bajo rango.

Aunque RAFT representa el estado del arte en precisión, su implementación en este proyecto es compleja debido a las restricciones de hardware. El proceso de Fine-Tuning necesario requiere una capacidad de cómputo (Clusters de GPUs tipo A100) que excede largamente los recursos de una estación de ingeniería estándar. Por lo tanto, la solución propuesta en esta memoria debe lograr una precisión similar, pero utilizando una arquitectura de inferencia pura y técnicas de inyección de contexto, evitando la etapa de re-entrenamiento masivo.

1.2. *Objetivos del Proyecto*

Ante la brecha tecnológica identificada y la necesidad operativa de CMDIC, el presente trabajo de titulación plantea los siguientes objetivos:

1.2.1. *Objetivo General*

Diseñar e implementar un asistente virtual inteligente (“Agente DCS”) para la interpretación de sistemas de control, basado en modelos de lenguaje locales y técnicas de extracción de información estructurada, que permita generar explicaciones técnicas y filosofías de control actualizadas en un entorno *offline*.

1.2.2. *Objetivo Específicos*

- Analizar la estructura de los archivos de exportación (FHX) de los controladores DCS Emerson para identificar patrones lógicos y generar una base de conocimientos estructurada.
- Desarrollar un motor de extracción híbrido que combine búsqueda basada en grafos con las capacidades generativas de un LLM local (Llama 3.1).
- Implementar una interfaz de consulta interactiva que permita a los operadores y mantenedores realizar consultas en lenguaje natural sobre la lógica de control.
- Validar la precisión y eficiencia del sistema mediante pruebas comparativas con documentación técnica real, asegurando su funcionamiento bajo las restricciones de hardware locales.

1.3. *Objetivos, Alcances y Limitaciones*

Ante la brecha tecnológica identificada anteriormente y la necesidad operativa de la minera Collahuasi (CMDIC), el objetivo general de este trabajo de titulación consiste en diseñar e implementar un asistente virtual inteligente para la interpretación de

sistemas de control, basado en modelos de lenguaje locales y técnicas de extracción de información estructurada. Este sistema busca generar explicaciones técnicas y filosofías de control actualizadas en un entorno fuera de línea, garantizando la continuidad operacional y la seguridad de los datos.

Para la consecución de este propósito principal, se establecen objetivos específicos escalonados que guían el desarrollo. En primera instancia, se procederá a analizar y caracterizar la estructura de los archivos de exportación .fhx de los controladores DCS Emerson DeltaV para identificar patrones lógicos y entender completamente su estructura. Posteriormente, se desarrollará un motor de extracción híbrido que combine la búsqueda determinista basada en grafos con las capacidades generativas de un LLM local, eligiendo un modelo liviano de 8 Mil Millones de datos (Llama 3.1 Instruct 8b). A nivel de usuario, se implementará una interfaz de consulta interactiva que permita realizar preguntas en lenguaje natural sobre la lógica de control, finalizando con la validación de la precisión y eficiencia del sistema mediante pruebas comparativas con documentación técnica.

En cuanto al alcance del desarrollo, el proyecto se encamina específicamente a la interpretación de archivos de configuración de la arquitectura **Emerson DeltaV**, abarcando desde la ingesta de archivos .fhx hasta su procesamiento algorítmico. El sistema garantiza una ejecución 100 % local , asegurando que ningún dato sensible sea transmitido a servidores externos, cumpliendo con los protocolos de ciberseguridad industrial de la compañía. El producto final entregará explicaciones textuales detalladas y representaciones visuales simplificadas de la topología de control, con el objetivo de que personas no expertas del área puedan comprender y aprender de los sistemas en lenguaje natural y visualizaciones generales.

No obstante, el diseño de la solución se encuentra con restricciones técnicas y operativas críticas. En primer lugar, existen limitaciones de hardware, ya que el sistema debe operar en estaciones de ingeniería estándar sin GPU nativa o CPU básicas, lo que impide el re-entrenamiento masivo de modelos y obliga al uso de técnicas de cuantización, RAG eficiente y llamadas a herramientas externas (Tool-Calling). En segundo

lugar, por seguridad operativa, el agente se concibe estrictamente como una herramienta de consulta pasiva, careciendo deliberadamente de privilegios para escribir o modificar configuraciones en los controladores reales, generando retrasos al momento de implementar ajustes promovidos por el agente creado. Finalmente, el análisis se limita a la lógica de configuración estática, sin acceso a variables de proceso en tiempo real provenientes del sistema SCADA o sistemas propios de los DCS, generando alucinaciones a la hora de asumir variables o estados inexistentes para poder generar el problema que pueda existir al momento de la consulta.

2. MARCO TEÓRICO

2.1. *Contexto Operacional y Normativo*

2.1.1. *Descripción del Proceso Minero en CMDIC*

La Compañía Minera Doña Inés de Collahuasi (CMDIC) constituye una de las operaciones de extracción de cobre más grandes a nivel mundial. Ubicada en la región de Tarapacá, a una altitud superior a los 4.400 m.s.n.m., su cadena de valor abarca desde la extracción a rajo abierto en el yacimiento Rosario hasta la producción de concentrado de cobre en la planta Ujina y la exportación de molibdeno y cátodos desde el puerto Collahuasi.

La operación se caracteriza por un alto nivel de automatización, estructurándose en macro-etapas secuenciales donde la intervención de los sistemas de control es crítica. El ciclo inicia con la Fragmentación Mecánica del Mineral (Chancado y Molienda), etapa destinada a la reducción granulométrica del mineral. Aquí, el material extraído es procesado en chancadores primarios y secundarios para luego ser transportado hacia el acopio del material. Posteriormente, ingresa al circuito de molienda (Molinos SAG y de Bolas), considerada la fase de mayor consumo energético y complejidad de control, donde se regulan variables como potencia, presión y alimentación para obtener la granulometría objetivo y la cual tiene los mayores estándares de seguridad con sensores en las correas, sonido, velocidad, pilar de enclavamiento, entre otros.

Una vez acondicionado el material, este avanza hacia la etapa de concentración o Flotación. En esta fase, el mineral molido pasa a celdas especializadas donde, mediante la adición de reactivos químicos y burbujeo de aire, se separa el sulfuro de cobre de la ganga. La lógica de control en esta área debe gestionar niveles de pulpa, flujos de

aire y dosificación de reactivos, coordinando cientos de actuadores simultáneamente para maximizar la recuperación metalúrgica, es un proceso crítico el cual tiene que funcionar 24/7 sin fallas. Finalmente, el proceso culmina con el Transporte de Concentrado, donde el producto espesado es enviado gravitacionalmente hacia el puerto a través de un mineroducto de más de 200 km, sistema que opera bajo estrictos lazos de seguridad y detección de fugas.

La continuidad operativa de estas áreas depende directamente de la robustez de sus Sistemas de Control Distribuido (DCS). Actualmente, la planta gestiona más de **10.000 señales de entrada y salida (I/O)**, distribuidas en un parque instalado de más de **500 controladores** de diversos fabricantes, predominando arquitecturas híbridas entre ABB, Siemens y Emerson DeltaV. Esta heterogeneidad tecnológica genera una complejidad exponencial en la administración de la ingeniería, pues cada fabricante implementa su propia filosofía de configuración. Esto, sumado al volumen masivo de señales, dificulta la trazabilidad de fallas, donde cualquier error de interpretación en la lógica de seguridad puede derivar en detenciones no programadas que impactan severamente la productividad y seguridad de la compañía.

2.2. Gestión de Riesgos

La gestión de riesgos en la industria minera se define como un proceso sistemático que permite identificar, evaluar y controlar los peligros asociados a las operaciones, con el fin de reducir la probabilidad de incidentes y sus consecuencias en la seguridad, salud, medio ambiente y continuidad operacional. En el contexto de la Compañía Minera Doña Inés de Collahuasi (CMDIC), este proceso implica la aplicación de metodologías estructuradas para evaluar los peligros en cada etapa del proceso productivo, considerando factores operacionales, técnicos y humanos.

2.2.1. Ciclo de Gestión de Riesgo (CGR)

CMDIC cuenta con un Modelo de Gestión estructurado que sustenta la organización bajo tres pilares fundamentales: Gestión de Procesos, Gestión de Activos y Gestión de Riesgos. El principal objetivo de este modelo es agregar valor mediante la reducción de la incertidumbre y variabilidad, asegurando que las operaciones se desarrollen bajo ambientes de control predecibles y seguros.

El Ciclo de Gestión de Riesgo (CGR) se compone de 4 etapas y 6 cajas operativas. A diferencia de un ciclo lineal convencional, este proyecto de titulación tiene su **génesis en la Caja 6 (Aprendizaje)**, detonando un nuevo ciclo de planificación para resolver una brecha técnica identificada.



Fig. 2.1: Ciclo de Gestión de Riesgo CMDIC.

A continuación, se describen las etapas del ciclo y su integración específica con el desarrollo del Agente DCS:

Etapas 4: Conocimiento, Aprendizaje y Comunicación (Origen)

El proceso de gestión de riesgos en Collahuasi no se detiene en la implementación de controles y su verificación, sino que incorpora un sistema de aprendizaje continuo

y comunicación efectiva, con el objetivo de mejorar la identificación y mitigación de riesgos.

Para ello, se establece un mecanismo de retroalimentación, basado en el análisis de eventos ocurridos, la evaluación de su impacto y la actualización de estrategias de control. Este proceso permite que la empresa aprenda de la experiencia operativa, optimizando sus protocolos y fortaleciendo su cultura de seguridad.

El aprendizaje se fomenta mediante el registro de incidentes y análisis de causas raíz, lo que permite identificar factores recurrentes y aplicar medidas correctivas. Además, se realizan capacitaciones y actividades de difusión, asegurando que los trabajadores conozcan los riesgos identificados, las mejores prácticas y las lecciones aprendidas en distintos niveles de la organización.

Asimismo, la información obtenida a partir de auditorías, inspecciones y reportes internos es utilizada para actualizar matrices de riesgo, mejorar los planes de mitigación y reforzar controles jerarquizados. De esta manera, se garantiza que la gestión de riesgos evolucione de forma dinámica, adaptándose a nuevos desafíos operacionales.

A través de la comunicación efectiva y la retroalimentación constante, Collahuasi promueve un entorno de trabajo más seguro y eficiente, donde la gestión de riesgos es un proceso proactivo y colaborativo, enfocado en la protección de las personas, el medio ambiente y los activos de la empresa.

- **Caja 6: Aprendizaje e Investigación:** Se fomenta mediante el registro de incidentes y análisis de causa raíz.

Génesis del Proyecto: La Caja 6 como Motor de Innovación

El desarrollo de este sistema de inteligencia artificial no surge como un requerimiento aislado, sino que tiene su génesis directamente en la etapa de **Conocimiento, Aprendizaje y Comunicación** del CGR. El proyecto se activa al detectar, mediante procesos de investigación interna y análisis de mantenimiento, una brecha crítica de conocimiento técnico que afecta la toma de decisiones en tiempo real.

Esta brecha se manifiesta específicamente en dos dimensiones técnicas:

- **Obsolescencia de la Documentación Estática:** En entornos industriales dinámicos como CMDIC, las estrategias de control evolucionan para optimizar procesos. Sin embargo, las “Filosofías de Control” (documentos narrativos que explican la lógica) suelen quedar rezagadas respecto a la configuración real cargada en los controladores. Esta desincronización genera una **incertidumbre operacional**, donde el personal técnico debe asumir el riesgo de interpretar una lógica que podría no coincidir con la documentación oficial.
- **Complejidad de la Ingeniería Inversa Manual:** Los archivos de exportación .fhx de Emerson DeltaV contienen la “verdad única” del sistema, pero su estructura jerárquica y declarativa no está diseñada para la lectura humana. La necesidad de interpretar manualmente miles de líneas de código para entender un simple enclavamiento de seguridad representa una tarea de alta carga cognitiva, propensa a errores de interpretación y que demanda una inversión excesiva de Horas-Hombre (HH).

Al identificar estos factores en la **Caja 6 (Aprendizaje e Investigación)**, se determina que la variabilidad en los tiempos de diagnóstico y la dependencia de expertos específicos para traducir el código DCS constituyen un riesgo para la continuidad operacional.

Por consiguiente, el Agente DCS se propone como una herramienta de **Transferencia de Conocimiento y Mitigación**. Su función es transformar esa “brecha de incertidumbre” en un activo digital accesible, permitiendo que el aprendizaje obtenido de los datos crudos del sistema se comunique de forma clara, estandarizada y automática a toda la organización. De esta manera, el proyecto cierra el ciclo de gestión de riesgos, convirtiendo una debilidad detectada en una fortaleza tecnológica que refuerza la cultura de seguridad y eficiencia de la compañía.



Fig. 2.2: Informe de aprendizaje.

Etapa 1: Planificación

En esta fase se establecen las bases del control mediante tres herramientas críticas:

- **Caja 1: Mapa del Proceso:** El mapa de proceso es una herramienta fundamental para conocer a fondo los procesos de una organización. Permite desglosar cada proceso de sus componentes esenciales, entendiendo cómo interactúan entre sí y cómo contribuyen al resultado final. Este mapa facilita clave, sino también de la membrana de los procesos, identificando no solo las actividades clave, sino también las causas subyacentes que afectan su rendimiento. A través de un

análisis detallado, el mapa de procesos ayuda a visualizar los flujos de trabajo y a comprender los porqués detrás de cada paso, lo que permite que los números y datos generados hablen por sí mismos, proporcionando información valiosa para la toma de decisiones y la mejora continua.

Mapa del Proyecto: Caja 1 como Planificación

Dentro del marco del CGR, el **Mapa de Procesos** permite desglosar la secuencia lógica de actividades para identificar los puntos críticos donde se genera o se pierde valor operacional. Para este proyecto, el mapeo se centró en el flujo de diagnóstico de fallas y auditoría de lógica en los controladores DeltaV de Emerson instalados en la Faena.

El análisis permitió identificar que el proceso tradicional de interpretación de lógica es lineal y altamente dependiente del factor humano, estructurándose en las siguientes etapas críticas:

1. **Extracción y Captura:** Obtención del archivo de configuración .fhx desde la ProfessionalPLUS.
 2. **Búsqueda y Localización:** Rastreo manual de etiquetas (*tags*) y parámetros dentro de miles de líneas de código ASCII.
 3. **Reconstrucción Mental de la Lógica:** El ingeniero debe interpretar las conexiones (*wires*) y bloques funcionales sin una referencia visual inmediata, lo que representa el mayor cuello de botella del proceso.
-
1. **Extracción y Captura:** Obtención del archivo de configuración .fhx desde la estación de ingeniería ProfessionalPLUS.
 2. **Búsqueda y Localización:** Rastreo manual de etiquetas (*tags*) y parámetros dentro de miles de líneas de código en formato ASCII.

3. **Reconstrucción Mental de la Lógica:** El ingeniero debe interpretar las conexiones (*wires*) y bloques funcionales sin una referencia visual inmediata, lo que representa el mayor cuello de botella del proceso.

Identificación de Puntos de Valor para la IA: Al mapear este flujo, se determinó que la integración del “Agente DCS” actúa como un catalizador en los puntos de mayor fricción técnica:

- **Reducción del Tiempo de Latencia en Diagnóstico:** El mapa de procesos reveló que un ingeniero senior puede tardar entre 4 a 24 horas en realizar una ingeniería inversa completa de un lazo de control complejo. La automatización mediante el Agente reduce este tiempo a minutos, eliminando la etapa de búsqueda manual.
- **Estandarización del Output:** El proceso actual presenta una alta variabilidad; dos ingenieros pueden interpretar una lógica de forma distinta dependiendo de su experiencia. El Agente DCS estandariza la salida de información, asegurando que el análisis técnico sea consistente y auditable.
- **Mitigación del Riesgo de Interpretación:** Al actuar en la fase de planificación, el mapa identifica que el error de lectura del archivo `.fhx` es un peligro latente. La IA, al procesar el archivo de forma determinista y estructural, elimina la omisión accidental de parámetros críticos.

De esta manera, la Caja 1 no solo cumple con la normativa de Collahuasi, sino que proporciona la justificación técnica para el diseño del software: el Agente no reemplaza al ingeniero, sino que optimiza el flujo de valor eliminando las tareas de bajo nivel y permitiendo que el personal experto se enfoque en la toma de decisiones críticas y la seguridad del proceso, además de obtener una herramienta de apoyo para poder acelerar el proceso de entendimiento.

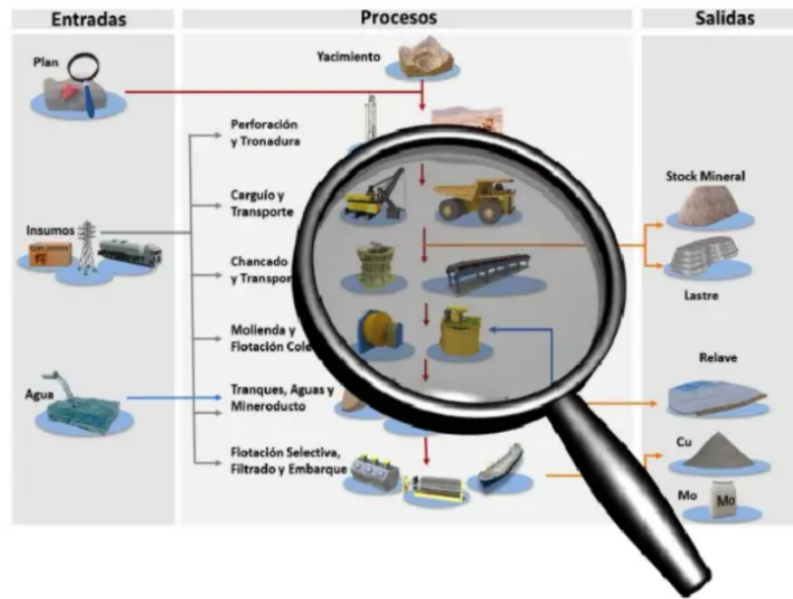


Fig. 2.3: Mapa de Procesos

- **Caja 2: Matriz de Riesgos:** La matriz de riesgo es una herramienta estratégica que permite identificar y evaluar los principales focos de dispersión que pueden amenazar el cumplimiento de la promesa de negocio. Esta matriz clasifica los riesgos según su probabilidad de ocurrencia y el impacto que tendrían en la organización permitiendo visualizar claramente los factores que pueden poner en peligro el rendimiento y la continuidad operativa. Al identificar estos riesgos de manera estructurada, la matriz facilita la implementación de medidas preventivas o correctivas, garantizando la protección de los objetivos estratégicos y el éxito del negocio

Matriz de Riesgos en el Proyecto: Caja 2 como Planificación

Una vez delimitado el alcance operativo, se procedió a la elaboración de la **Matriz de Riesgo**, herramienta estratégica diseñada para identificar y evaluar los factores de dispersión que podrían amenazar la integridad del proceso de diagnóstico o la seguridad de la información de CMDIC. En este análisis, se priorizaron los riesgos tecnológicos emergentes asociados al uso de modelos de lenguaje en entornos de control

crítico.

Los focos de riesgo identificados y sus respectivas estrategias de mitigación se detallan a continuación:

- **Riesgo de Integridad Técnica (Alucinación de Datos):** Dado que los LLMs operan bajo principios probabilísticos, existe el peligro de que el sistema genere respuestas sintácticamente coherentes pero técnicamente erróneas por ejemplo invertir una lógica de enclavamiento o inventar un parámetro de bloque.
- **Riesgo de Ciberseguridad y Fuga de Información Sensible:** Los archivos de configuración DCS representan la propiedad intelectual y el núcleo estratégico de la operación de CMDIC. El uso de modelos en la nube (como GPT-4) implicaría la salida de datos críticos fuera del control de la compañía.
- **Riesgo de Factibilidad Técnica (Restricción de Hardware):** El despliegue de modelos de lenguaje de alto rendimiento suele requerir infraestructura de centro de datos. El riesgo evaluado fue la imposibilidad de ejecutar la solución en estaciones de ingeniería estándar de 6GB o 12GB de VRAM.

La evaluación de estos riesgos permitió clasificar el proyecto dentro de un ambiente de control preventivo. Al abordar proactivamente la alucinación y la ciberseguridad, la matriz de riesgo validó que el Agente DCS no solo es una herramienta de productividad, sino una solución robusta alineada con los estándares de seguridad informática y operacional de la compañía.

INVENTARIO DE MATRIZ DE IDENTIFICACIÓN DE PELIGROS Y EVALUACIÓN DE RIESGOS															
ACTIVIDAD	FASE	SUBFASE	NÚMERO DE ACTIVIDAD	I. IDENTIFICACIÓN DE PELIGROS Y EVALUACIÓN DE RIESGOS				II. EVALUACIÓN DE RIESGOS				III. PLAN DE MITIGACIÓN Y CONTROL			
				A. IDENTIFICACIÓN DE PELIGROS				B. EVALUACIÓN DE RIESGOS				C. PLAN DE MITIGACIÓN Y CONTROL			
				FASE	FASE	FASE	FASE	FASE	FASE	FASE	FASE	FASE	FASE	FASE	FASE
ACTIVIDAD	FASE	SUBFASE	NÚMERO DE ACTIVIDAD	A. IDENTIFICACIÓN DE PELIGROS				B. EVALUACIÓN DE RIESGOS				C. PLAN DE MITIGACIÓN Y CONTROL			
				FASE	FASE	FASE	FASE	FASE	FASE	FASE	FASE	FASE	FASE	FASE	FASE
ACTIVIDAD 1: PLAN DE MITIGACIÓN Y CONTROL	FASE 1	SUBFASE 1	NÚMERO DE ACTIVIDAD 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1
				FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1
				FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1
				FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1
				FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1
				FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1
				FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1
				FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1
				FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1
				FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1
				FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1
				FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1
				FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1	FASE 1

Fig. 2.4: Ejemplo de matriz de riesgo

- Caja 3: Planes de Mitigación y Control:** Los estándares para la prevención de fatalidades representan los mejores procedimientos actuales en el control de riesgos fatales, estos derivan de evaluaciones de riesgos detalladas sobre los incidentes ocurridos en el pasado. Los estándares para la prevención de fatalidades son apenas una parte de los requerimientos para prevenir accidentes fatales. Los procesos regulares de gestión de riesgos ayudarán a identificar los controles adicionales y las actividades gerenciales asociadas, requeridas para facilitar una gestión efectiva de riesgos fatales. Estos controles deberán ser implementados adicionalmente a los estándares para asegurarse de que sean adecuados y efectivos.
- Jerarquía de Controles:** La jerarquía de controles establece niveles cuantitativos y cualitativos para las medidas implementadas en la mitigación y control de un riesgo. Esta clasificación permite priorizar las acciones según su efectividad asegurando un enfoque estructurado de la gestión del riesgo. A continuación, se presenta cada nivel de control en la siguiente figura 2.5

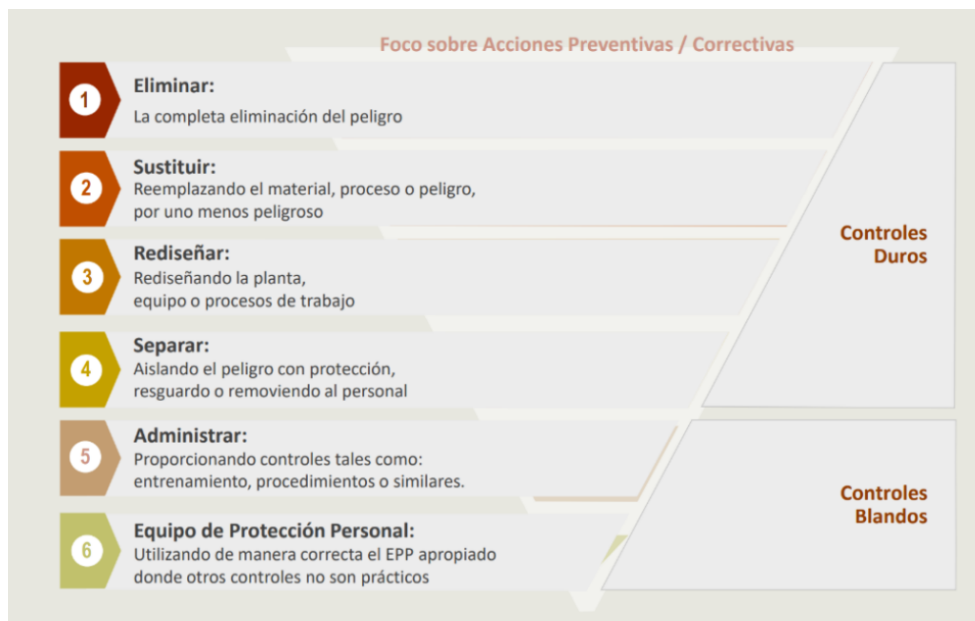


Fig. 2.5: Jerarquía de Controles.

Planes de Mitigación y Control: Caja 3 como Planificación

Una vez evaluados los riesgos en la fase anterior, la Caja 3 se enfoca en el diseño de medidas concretas para eliminar o minimizar dichos peligros. Esta etapa se rige por la **Jerarquía de Controles**, un sistema que prioriza las soluciones según su efectividad, moviéndose desde los controles blandos (administrativos) hacia los controles duros (ingeniería y eliminación).

En el marco de este proyecto, el Agente DCS no se diseñó únicamente como una herramienta de consulta, sino como un plan de mitigación técnica estructurado de la siguiente manera:

- **Clasificación como Control de Ingeniería:** El software se posiciona en el tercer nivel de la jerarquía. Al automatizar la extracción y el análisis del archivo .fhx, se crea una barrera tecnológica que protege al ingeniero de la fatiga cognitiva y de la interpretación errónea. No se limita a dar instrucciones (control administrativo), sino que procesa el dato crudo para entregar una salida verificada.

- **Mitigación de la Dependencia del Experto :** El plan de mitigación busca sustituir el método tradicional de ingeniería inversa manual —que, como se mencionó, puede tomar entre **4 a 24 horas**— por un proceso asistido por IA. Esto reduce la exposición del proceso a la variabilidad del criterio individual, estandarizando la respuesta técnica ante una falla.
- **Control de la Incertidumbre Operacional:** La medida de control implementada en el código (la arquitectura de grafos) garantiza que el análisis de la lógica sea holístico. Al mapear todas las conexiones del controlador, el plan de mitigación asegura que ninguna condición de seguridad sea omitida por el ojo humano durante una revisión bajo presión operativa.

Efectividad del Control: El diseño de esta medida de control asegura que, ante una contingencia en la planta, el personal de mantenimiento cuente con una fuente de información veraz y de respuesta inmediata. Esto impacta directamente en la reducción de los tiempos de reparación y fortalece la continuidad operacional de CMDIC, cumpliendo con la promesa de valor de la gestión de riesgos: transformar un peligro identificado en un ambiente controlado mediante la ingeniería.

Etapas 2: Ejecución

Cada actividad en Collahuasi se rige por un procedimiento estandarizado específico, el cual detalla de manera estructurada cada paso a seguir. Este procedimiento incluye una descripción detallada de la tarea, las herramientas necesarias, los instructivos asociados, listas de verificación (Checklists) y un listado exhaustivo de los riesgos involucrados junto con las medidas de control establecidas para su mitigación.

		COMPAÑÍA MINERA GORRA INES DE		GERENCIA TRANQUE AGUA Y MINERODUCTO	
Housekipping de áreas					
IDENTIFICACIÓN	FECHA DE REVISIÓN	VERSIÓN	PRÓXIMA REVISIÓN	RESPONSABLES	PAGINA
GTAM-ITF-INS-4610	Febrero 2023	V-04	Febrero 2025	Generado: Julio Peralta Revisado: Felipe Munizaga Aprobado: Pablo Contreras	1 de 8

1. Antecedentes Generales

Personal requerido	Operador-Mantenedor SI ITF	Especialidad	SEMAGUA / SEODESA
Duración actividad	X	Std Job N°	X
Condición equipo	X	Frecuencia	WR

2. Condiciones de seguridad

2.1. Matriz de evaluación y gestión de Riesgos

Actividad está en Matriz

SI	NO
X	

Si es NO se debe agregar en Matriz previa a la ejecución de los trabajos.

2.2. EPF asociado a la actividad

EPF1	EPF2	EPF3	EPF4	EPF5	EPF6	EPF7	EPF8
X		X	X			X	X

2.3. Riesgo a personas.

Riesgo a personas						
	Caída mismo nivel	Caída a distinto nivel	Atrapamiento de manos	Bajas Temperaturas	Radiación UV	Tormenta Eléctrica

Riesgo a personas					
	Aplastamiento por caída de carga	Choque, colisión o volcamiento	Dermatitis	Contacto con partes móviles	Trabajo en Caliente

Fig. 2.6: Portada del Instructivo de procedimiento Estandarización de la ejecución.

Ejecución del Proyecto: Caja 4 como Procedimiento de Trabajo

La fase de Ejecución en el CGR se materializa a través del **Procedimiento de Trabajo**, el cual garantiza que la implementación de la solución se realice bajo estándares de calidad, seguridad y reproducibilidad. En el contexto de este proyecto, esta etapa corresponde al desarrollo técnico del software, donde la programación se rige por protocolos de ingeniería para asegurar que la herramienta sea confiable en un entorno industrial.

La ejecución del Agente DCS se desglosa en los siguientes lineamientos operativos:

- **Estandarización del Desarrollo en Python:** La escritura del código no se realiza de forma aislada, sino siguiendo estándares internacionales de programación (como PEP 8). Esto asegura que el software sea legible y mantenible por otros ingenieros de CMDIC en el futuro, eliminando la dependencia de un único desarrollador y cumpliendo con el objetivo de continuidad operativa.
- **Protocolo de Tratamiento de Datos (.flhx):** Se estableció un procedimiento de trabajo específico para la limpieza y *parsing* de los archivos de configuración. Este incluye la eliminación de metadatos irrelevantes y la extracción estructurada de parámetros, asegurando que la entrada de datos al modelo de lenguaje sea limpia y libre de ruido que pueda inducir a errores.
- **Integración de Librerías y Control de Versiones:** La ejecución incluyó la integración de herramientas robustas como **NetworkX** para la gestión de grafos y modelos cuantizados de **Meta AI**. El uso de entornos virtuales y documentación técnica detallada garantiza la reproducibilidad del sistema: cualquier estación de ingeniería con los requisitos mínimos puede desplegar el agente siguiendo el procedimiento establecido.

Aseguramiento de la Calidad en la Ejecución: A diferencia de un desarrollo de software convencional, la Caja 4 exige que cada módulo del Agente DCS sea auditable. El procedimiento de trabajo contempla la generación de *logs* y registros de actividad, permitiendo que el equipo de soporte técnico pueda verificar la lógica interna del programa ante cualquier discrepancia. De esta manera, la ejecución se transforma en un proceso controlado y documentado, alineado con la excelencia operativa de Collahuasi.

Etapas 3: Verificación

En CMDIC, la etapa de verificación y autorización de actividades es un proceso fundamental dentro de la gestión de riesgos y la seguridad operacional asegurando que cada tarea se realice bajo condiciones controladas y en cumplimiento con los estándares establecidos. Previo a la ejecución de cualquier actividad, se lleva a cabo una verificación exhaustiva garantizando que los equipos, herramientas y condiciones del entorno sean adecuados para su desarrollo. Además, se realiza una evaluación detallada de los riesgos asociados a la tarea, junto con la implementación de las medidas de control necesarias para mitigar cualquier potencial incidente. Para dejar constancia de este proceso, se utiliza el documento VATS (Verificación y Autorización para Trabajo Seguro), el cual certifica que todas las condiciones operacionales y de seguridad han sido revisadas y aprobadas antes el inicio de la actividad. Este documento es clave para asegurar la trazabilidad de la verificación y el compromiso con una ejecución segura de las tareas.

Verificación y Autorización: Caja 5 como Verificación

La etapa de Verificación constituye el último control de calidad antes de la entrega formal de la solución. En CMDIC, este proceso asegura que las medidas de control (el Agente DCS) funcionen según lo diseñado en la etapa de planificación y cumplan con los estándares de seguridad operacional. Para este proyecto, la Caja 5 representa la transición del entorno de desarrollo a la validación en un entorno de ingeniería real.

El protocolo de verificación y autorización del Agente DCS se estructuró bajo los siguientes criterios:

- **Validación contra el “Ground Truth” (Verdad de Campo):** Las respuestas generadas por el sistema no se aceptan de forma automática. Se estableció un procedimiento de validación cruzada donde las interpretaciones de la IA se contrastan con la documentación histórica oficial y, fundamentalmente, con el criterio técnico de **ingenieros expertos (Seniors)** de la planta. Esta comparación

permite medir la precisión del modelo en escenarios de control reales.

- **Criterios de Aceptación y Tasa de Alucinación:** Para autorizar el uso del sistema como herramienta de consulta técnica, este debe superar un umbral de precisión predefinido. La verificación se enfoca en minimizar la tasa de alucinación; el sistema solo recibe la autorización si demuestra que es capaz de reconocer sus limitaciones (absteniéndose de responder si el dato no está en el archivo .fhx) en lugar de generar información falsa.
- **Certificación de Operación Segura (VATS Digital):** De manera análoga al documento VATS utilizado en terreno, el software se somete a una verificación de sus condiciones de seguridad informática y lógica. La autorización final depende de asegurar que la herramienta no interfiera con la red de control y que el flujo de información sea unidireccional, protegiendo los activos de la compañía.

Formalización de la Entrega: Una vez que el Agente DCS ha superado las pruebas de consistencia y precisión, se procede a la autorización de su uso. Esta etapa cierra el ciclo iniciado en la Caja 6, transformando una brecha de conocimiento identificada en una herramienta tecnológica validada, certificada y lista para aportar valor en la toma de decisiones críticas de mantenimiento.

COLLAHUASI		COMPañIA MINERA DOñA INÉS DE COLLAHUASI			
Revisión N° 5		Página 1 - 4		Herramienta 5- Ciclo de gestión de Riesgos	
Verificación y Autorización para Trabajo Seguro					
OT:					
Fecha:		Hora:		Fecha:	
Hora:		Hora:			
Descripción de la actividad por parte del Ejecutor					
CGR	Área y/o Equipo Solicitada:				
	Área				
	Equipo				
	Gcia/Si/E/SED				
Revisión Sup. Ejecutor (CMDIC/ESED)		Marcar (x)		Revisión Sup. Que solicita la Actividad (CMDIC/ESED)	
Cuenta con:		SI NO		Cuenta con:	
SI		NO		SI NO	
C1,C2,C3	Con la O.T (Contrato del Servicio, orden de servicio, libro de obra) y evaluación de riesgos (Matriz).		Evaluación de riesgos (Matriz), Procedimientos e Instructivos ESPECÍFICOS. (Es decir están todos los pasos y controles para desarrollar el trabajo).		
	Instructivos ESPECÍFICOS. (Es decir están todos los pasos y controles para desarrollar el trabajo).		Personal Entrenado y Capacitado para realizar la actividad. (Instructivos y competencias especiales).		
	Personal Entrenado y Capacitado para realizar la actividad. (Instructivos y competencias especiales).		Se cuentan con todos los RECURSOS necesarios para realizar la Actividad (Personal, Herramientas, Equipos, EPP, otros).		
	Se cuentan con los RECURSOS necesarios para realizar la Actividad (Personal, Herramientas, Equipos, EPP, otros).				
Si es; NO la Actividad no debe realizarse. Hasta que se cuenten con todos los Controles					
ENERGÍAS DE ORIGEN IDENTIFICADAS POR EJECUTOR		1. Eléctrica ☐ 2. Mecánica ☐ 3. Hidráulica ☐ 4. Gravitacional ☐ 5. Rotatoria ☐ 6. Neumática ☐ 7. Nuclear ☐ 8. Eólica ☐			
9. Química ☐ 10. Biológica ☐ 11. Biomecánica ☐ 12. Compromiso Medioambiental ☐ 13. Otras Especificar: _____					
ENERGÍAS DE ORIGEN IDENTIFICADAS POR SOLICITANTE		1. Eléctrica ☐ 2. Mecánica ☐ 3. Hidráulica ☐ 4. Gravitacional ☐ 5. Rotatoria ☐ 6. Neumática ☐ 7. Nuclear ☐ 8. Eólica ☐			
9. Química ☐ 10. Biológica ☐ 11. Biomecánica ☐ 12. Compromiso Medioambiental ☐ 13. Otras Especificar: _____					
Condiciones Revisadas por el Sup. Que EJECUTA la Actividad					
C1,C2,C3	Puntos de Bloqueo (Indicar si cuenta con libro de bloqueo, Plan de Bloqueo y verificación de potencial cero)		Requerimientos para espacios confinados (Gases Tóxicos, O2, etc.)		NOTA: Indique con SI, NO o NA los requerimientos. NA: No Aplica
	Indicar Tags:		Plan de Izaje		
	Protección contra incendio		A.R.T.A.		
	Cierre de Fuentes Radiactivas: (Indicar)		EPPs Aplicables (Especificar)		
	E.P.P. Especial (Indicar):		Permisos Especiales. (Espacio Confinado, Excavación, Alta y Media Tensión, Trabajos en Caliente, Intervención de Equipos Radiactivos).		
	Sectorización del Área: (Indicar Sector)		Indicar cuales:		
Si es; NO la Actividad no debe realizarse. Hasta que se cuenten con todos los Controles					
Entrega de Área y/o Equipos					
Observaciones Especiales de parte del Dueño de Área (Puntos de bloqueo, segregación o cierre de áreas, traslado de personal, transferencias, traslado de equipos, intervenciones de otras Empresas, etc.)					
C3,C5	Responsables		Nombre		Empresa
	Supervisor Ejecutor CMDIC/ ESED				
	Sup. Que Solicita la Actividad CMDIC/ESED				
	Dueño del Área y/o Equipo CMDIC / ESED				
	Responsables		Nombre		Empresa
	Supervisor Ejecutor CMDIC/ ESED				
Devolución de Área y/o Equipos					
Responsables		Nombre		Empresa	Firma
Supervisor Ejecutor CMDIC/ ESED					
Dueño del Área y/o Equipo					

Fig. 2.7: Verificación y Autorización para el uso del sistema VATS.

2.3. Sistemas de Control Distribuido (DCS)

Un Sistema de Control Distribuido es una plataforma de control automatizada diseñada para gestionar procesos industriales continuos y complejos, como los presentes en la minería, petróleo y gas. A diferencia de los PLC tradicionales, que centralizan la lógica en unidades discretas, un DCS distribuye las funciones de control en múltiples controladores autónomos repartidos por la planta, pero supervisados desde una sala

de control centralizada. Esta arquitectura garantiza alta disponibilidad, redundancia y escalabilidad. Estos dispositivos son considerados de alta disponibilidad, lo que significa que estos tienen que funcionar 24/7 sin detenciones con una tasa de fiabilidad del 96 %, a pesar de ser un gran número ese porcentaje restante puede llegar a ocurrir, debido a esto los sistemas funcionan de manera redundante existiendo 2 controladores DCS en paralelo llegando a una fiabilidad del 99,6 %. Esto último genera constantes cambios dentro de la lógica o forzados, esto debido a que los cambios realizados en uno tienen que ser homologados en el otro debido a que si llegase a fallar el primero, el segundo entraría a operación de manera automática, y si estos tienen diferencias muy grandes pueden derivar a problemas mayores.

2.3.1. Arquitectura Emerson DeltaV

El sistema DCS utilizado como estándar principal en las áreas críticas de CMDIC es el Emerson DeltaV, el cual tienen como plan futuro estandarizar esta marca como el uso general dentro de la empresa. Esta plataforma se caracteriza por su arquitectura orientada a objetos y su integración nativa de dispositivos de campo inteligentes.

La arquitectura de DeltaV se compone de los siguientes elementos fundamentales que interactúan en la red de control:

- **Estación de Ingeniería (ProfessionalPLUS):** Es el nodo central donde reside la base de datos de configuración global. Desde aquí se crean, modifican y descargan las estrategias de control hacia los controladores. Es en esta estación donde se generan los archivos de exportación (.fhx) que alimentan al sistema propuesto en esta memoria.
- **Controladores (Controller):** Son las unidades de procesamiento (CPU) encargadas de ejecutar la lógica de control en tiempo real. En la industria minera es común encontrar modelos de la Serie M y Serie S, diseñados para soportar ambientes hostiles y ejecutar miles de bloques de función por segundo.

- **Subsistema de I/O (Entradas/Salidas):** Interfaz física que conecta los sensores y actuadores de la planta con el controlador.
- **Red de Control (Area Control Network - ACN):** Red de comunicación redundante propietaria que permite el intercambio de datos entre controladores y estaciones de operación, asegurando determinismo y baja latencia.

2.3.2. Estándar IEC 61131-3 y Lenguajes de Programación

La lógica que gobierna los controladores industriales no es arbitraria; se rige bajo el estándar internacional IEC 61131-3[3], regulado por la Comisión Electrotécnica Internacional. Esta norma define la sintaxis y semántica de los lenguajes de programación para controladores programables, permitiendo la interoperatividad y estandarización del software industrial, todos los controladores se rigen bajo esta norma, sin embargo las plataformas para su configuración y programación varía según el proveedor, debido a esto cambian muchos atajos, compuertas y librerías internas, además cabe destacar que dentro de la compañía minera existe una gran cantidad de librerías personalizadas creadas solamente para esta empresa generada por las distintas marcas, de esta manera creando una gran segmentación con la competencia y una segmentación de uso dentro de esta misma, dado que estas librerías no son de libre acceso.

Entre los lenguajes definidos por la norma, el sistema DeltaV y este proyecto se centran principalmente en dos:

- **Diagrama de Bloques de Funciones (FBD):** Es el lenguaje predominante en la minería debido a su naturaleza gráfica. Representa la lógica mediante “bloques” (que encapsulan funciones como PID, sumas, lógicas booleanas) conectados por “líneas” que representan el flujo de datos. Para la IA, el desafío radica en que FBD no es texto lineal, sino una topología de red.
- **Texto Estructurado (ST):** Lenguaje de alto nivel similar a Pascal o C. Se utiliza dentro de bloques de cálculo complejos donde la lógica gráfica sería demasia-

do extensa. El análisis de este lenguaje requiere técnicas de procesamiento de lenguaje natural (NLP) más tradicionales.

- **Grafos de Función Secuencial (SFC):** Utilizado para definir secuencias de estados, como los procedimientos de arranque y parada de un molino. Estructura la lógica en etapas (pasos) y transiciones.

Los otros dos lenguajes definidos como LD (Ladder) e IL (Instruction List) no se consideran para el uso general del proyecto dado que, LD es mas utilizado dentro de los PLC y es muy poca la utilización dentro de los controladores DCS, en cambio IL se dejo de utilizar hace mas de 15 años dado esto a día de hoy ninguna marca o empresa tiene integrada esta opción como lenguaje dentro de sus plataformas.

2.3.3. Estructura de Archivos de Exportación (FHX)

Para realizar ingeniería externa, auditorías o migraciones, el sistema DeltaV permite exportar su base de datos de configuración a un formato legible denominado FHX. Una de las razones fundamentales para trabajar con este tipo de archivo y con esta marca en general es la facilidad de exportación de estos archivos, dado que cada proveedor tiene su plataforma y estas distintas de maneras de exportar la lógica programada, ademas de distintos niveles de ciberseguridad que en casos como ABB es imposible encontrar exportaciones en formato codigo dado políticas internas y contextos globales de seguridad. Por otro lado, el sistema DeltaV tiene esta opción para la creación masiva de items e integración de bloques y visualización de la lógica programada en formato estructurado.

Un archivo FHX es, en esencia, un archivo de texto plano basado en estándares ASCII o Unicode que describe la configuración completa de los objetos del sistema. Sin embargo, no sigue una estructura de programación secuencial. Su sintaxis es jerárquica y declarativa, organizada mediante definiciones de clases e instancias.

La estructura típica de un archivo FHX incluye:

- **Encabezado:** Metadatos sobre la versión del sistema y la fecha de exportación.

- **Definición de Módulos:** Cada lazo de control se define como un “Módulo”. Dentro de este, se declaran los bloques de función (ej. AI1, PID1, AO1).
- **Parámetros y Atributos:** Dentro de cada bloque, el archivo lista exhaustivamente sus parámetros internos (Gain, Reset, Setpoint, Alarm Limits).
- **Wires (Conexiones):** Esta es la sección más crítica para la reconstrucción lógica. El FHX no *"dibuja"* el diagrama, sino que define las conexiones mediante instrucciones de tipo `WIRE SOURCE -> DESTINATION`.
- **Bloques personalizados:** Dentro de estos archivos se puede encontrar librerías y bloques FBD personalizados, estos vienen con sus nombres configurados, además de la lógica interna la cual es utilizada en distintos controladores conectados, cabe destacar que esto es encontrado en el controlador de origen, si otro de los controladores hace uso de este bloque aparece su nombre con la dirección o nombre del controlador master asociado, sin embargo, la lógica no esta considerada por seguridad interna entre controladores.

El desafío técnico de esta memoria reside en que el archivo FHX es extremadamente verboso y difícil de interpretar por un humano sin la herramienta gráfica propietaria. Un solo lazo de control puede generar cientos de líneas de texto, haciendo inviable su lectura manual, generando mayor demora dentro de lo necesitado, sin embargo junto a la Inteligencia artificial esto puede ser de gran ayuda y mas eficiente a la hora de interpretar y explicar el funcionamiento.

2.4. Inteligencia Artificial Generativa y LLMs

La Inteligencia Artificial Generativa (GenAI) representa un cambio de paradigma en el campo del aprendizaje profundo (*Deep Learning*). Tradicionalmente, la IA se centraba en modelos discriminatorios, cuyo objetivo es modelar la probabilidad condicional $P(Y|X)$ para clasificar una entrada X en una etiqueta Y . En contraste,

la IA Generativa busca modelar la distribución de probabilidad conjunta $P(X, Y)$ o la distribución de los datos mismos $P(X)$, permitiendo generar nuevas muestras que comparten las características estadísticas del conjunto de entrenamiento original.

En el contexto de la generación de texto y código, estos sistemas se materializan en los denominados Modelos de Lenguaje Grande (LLMs, por sus siglas en inglés). Un LLM es, fundamentalmente, un sistema probabilístico entrenado para predecir el siguiente *token*, unidad mínima de texto, en una secuencia, basándose en el contexto previo. Matemáticamente, dada una secuencia de tokens $x_1, x_2, \dots, x_t$, el modelo maximiza la probabilidad:

$$P(x) = \prod_{i=1}^T P(x_i | x_1, \dots, x_{i-1}) \quad (2.1)$$

Esta capacidad de autocompletado, llevada a una escala masiva de parámetros y datos, da lugar a propiedades emergentes como el razonamiento lógico, la traducción y, crucial para este proyecto, la interpretación de sintaxis de programación compleja.

2.4.1. Modelos de Lenguaje Grande (LLMs) y Transformadores

Los Modelos de Lenguaje Grande (LLMs) son redes neuronales entrenadas con volúmenes masivos de texto para comprender y generar lenguaje humano y código de programación. La arquitectura fundamental que permite este avance es el **Transformer**, introducida por Vaswani et al. en 2017 [5].

[Image of Transformer architecture]

El componente revolucionario del Transformer es el mecanismo de **Auto-atención (Self-Attention)**. En términos simples, este mecanismo permite al modelo ponderar la importancia de cada *token* en relación con los demás dentro de una secuencia, independientemente de su distancia posicional. Para conceptualizar el término *token*, este debe entenderse como la unidad mínima de procesamiento semántico. Si bien en inglés una palabra simple suele equivaler a un token, en la práctica técnica se considera que las palabras complejas o compuestas (ej. “activa-mente”, “normal-mente”)

se fragmentan en 1.5 o 2 tokens. Esta tokenización permite al modelo gestionar la morfología del lenguaje de manera eficiente.

Para la interpretación de código DCS, la auto-atención es vital. Esta capacidad permite al modelo entender que una variable declarada al inicio de un archivo está vinculada funcionalmente a un bloque lógico ubicado cientos o miles de líneas más abajo, manteniendo la coherencia semántica del análisis y permitiendo recrear la lógica completa del controlador sin necesidad de una representación visual previa.

El proyecto presentado utiliza modelos de la familia **Meta AI**, específicamente la versión **Llama 3.1 Instruct 8B**. La nomenclatura “8B” hace referencia a la complejidad del modelo, indicando que posee **8 mil millones de parámetros** (pesos entrenables), lo cual le otorga una capacidad de razonamiento compacta pero potente. Por otro lado, el término “Instruct” señala que esta variante ha sido sometida a un proceso de ajuste fino (*Instruction Tuning*), optimizándola para seguir instrucciones precisas y responder preguntas, metodología que se alinea directamente con los requerimientos del modelo necesitado. A diferencia de modelos cerrados como GPT-4 o Gemini, Llama permite la descarga de sus pesos para una ejecución local, un requisito indispensable para garantizar la privacidad de los datos estratégicos de la compañía, además de ofrecer la flexibilidad para modificar o re-entrenar estos pesos según sea necesario.

2.4.2. Desafíos: Alucinación y Ventana de Contexto

La implementación de Grandes Modelos de Lenguaje (LLMs) en ingeniería crítica enfrenta dos desafíos técnicos mayores que determinan la arquitectura del software, áreas que actualmente continúan en activo desarrollo e investigación en la industria tecnológica:

1. Alucinación: Los LLMs son sistemas probabilísticos, no deterministas. Su funcionamiento se basa en predecir el “siguiente token más probable”, lo que puede derivar en el fenómeno conocido como **alucinación**. Esto ocurre cuando el modelo genera información sintácticamente correcta y coherente, pero factualmente falsa. En el con-

texto de este proyecto, una alucinación podría manifestarse como la invención de un parámetro inexistente en un bloque PID o la creación de conexiones (*wires*) ficticias dentro del código.

En un entorno minero, este comportamiento es inaceptable y peligroso. Una interpretación errónea sobre un enclavamiento de seguridad no solo puede generar detenciones injustificadas que afectan la producción, sino que, en el peor de los casos, podría comprometer la integridad física del personal al omitir condiciones de riesgo. Por ello, es imperativo restringir la “creatividad” del modelo mediante mecanismos de anclaje a la realidad, tales como el uso de herramientas externas (*Tool Calling*) y bases de datos vectoriales para la Recuperación Aumentada (RAG), entregando información verídica solo cuando sea requerida.

2. Ventana de Contexto Limitada: La “ventana de contexto” se define como la cantidad máxima de información (tokens) que el modelo puede retener y procesar en una sola interacción. Esta restricción técnica impone límites severos en la capacidad del modelo para leer y comprender grandes volúmenes de datos simultáneamente.

En el desarrollo del proyecto, esta limitación se hace crítica durante el procesamiento de los archivos de exportación .fhx. Dependiendo de la envergadura de la planta, estos archivos pueden contener más de un millón de líneas de código y miles de interconexiones. Intentar procesar esta cantidad de información en un solo paso es computacionalmente inviable; puede resultar en tiempos de procesamiento excesivos (superando las 24 horas en hardware estándar), desbordamiento de memoria y, finalmente, en respuestas incoherentes o repetitivas por parte del modelo al perder la referencia de los datos iniciales.

En consecuencia, la estrategia de “copiar y pegar” el código completo es inviable. La solución de ingeniería requiere fragmentar la información, etiquetarla según su relevancia lógica y entregarla al modelo de manera segmentada, asegurando que el LLM reciba solo el contexto preciso necesario para resolver una consulta específica sin saturar su memoria.

2.4.3. Optimización y Cuantización para Hardware Limitado

Uno de los pilares fundamentales de este proyecto es la viabilidad de ejecución en hardware de consumo y recursos limitados. La infraestructura disponible consta de una estación de trabajo con GPU NVIDIA RTX 3060 (6GB de VRAM) y una instancia de servidor proporcionada por la empresa con 4 vCPUs y 36GB de memoria RAM.

Sin embargo, un modelo de lenguaje moderno como Llama 3 de 8 mil millones de parámetros (8B), en su precisión original de coma flotante de 16 bits (FP16), requiere aproximadamente 16 GB de VRAM solo para cargarse y realizar inferencia. Si se considerara el entrenamiento convencional, este requisito ascendería a más de 36 GB de VRAM, excediendo largamente los recursos físicos disponibles.

Para resolver esta brecha tecnológica, se utiliza la técnica de **Cuantización**. Esta metodología reduce la precisión numérica de los pesos del modelo, transformando la representación de datos de coma flotante de 16 bits (FP16) a enteros de 4 bits (INT4) o incluso formatos menores si fuese necesario, comprimiendo la información sin alterar la arquitectura de la red.

Para la implementación, se utilizan formatos de archivo optimizados como **GGUF** (GPT-Generated Unified Format). Este formato permite mapear el modelo en la memoria RAM y VRAM de manera eficiente, gestionando inteligentemente la carga de capas entre la CPU y la GPU (Offloading) para maximizar el uso de los recursos. Al aplicar una cuantización de 4 bits (Q4_K_M), el modelo de 8B parámetros reduce su peso en memoria de ~ 16 GB a tan solo 4.5 GB. Esta reducción drástica permite su ejecución fluida y completa dentro de la tarjeta gráfica de 6GB disponible, con una pérdida de precisión despreciable para las tareas de razonamiento lógico requeridas en este proyecto.

2.5. Recuperación de Información y Grafos

2.5.1. RAG (Retrieval-Augmented Generation)

Los Modelos de Lenguaje Grande (LLMs) poseen una limitación fundamental denominada “conocimiento paramétrico congelado”, lo que significa que su entendimiento del mundo está limitado a la fecha de corte de su entrenamiento. Un modelo como Llama 3 carece de acceso a la configuración específica del controlador PID del Molino SAG de Collahuasi, dado que se trata de información privada, dinámica y propietaria. Adicionalmente, puesto que los datos exactos del entrenamiento inicial son conocidos únicamente por los desarrolladores del modelo, este actúa como una “caja negra” cuyo comportamiento debe ser acotado mediante herramientas externas para garantizar la veracidad de sus respuestas.

Para solventar esta limitación sin incurrir en los altos costos computacionales de un re-entrenamiento (*Fine-tuning*), se implementa la arquitectura **RAG (Generación Aumentada por Recuperación)**.

Esta arquitectura combina la capacidad de razonamiento lingüístico del LLM con la precisión de una base de datos externa, configurable y actualizable en tiempo real. El proceso se ejecuta en tres fases secuenciales:

1. **Recuperación (Retrieval):** Ante una consulta del usuario (ej. “¿Cuál es el Setpoint del bloque 101?”), el sistema no transfiere la pregunta directamente al modelo. En su lugar, un algoritmo de búsqueda consulta una base de conocimientos donde reside la información segmentada de los archivos .fhx, extrayendo los fragmentos técnicos relevantes para dicha consulta.
2. **Aumentación (Augmentation):** La información recuperada se inyecta en el *prompt* del sistema de manera transparente para el usuario (*backend*). El *prompt* original se transforma en una instrucción enriquecida: “Utilizando el siguiente contexto técnico extraído del archivo FHX: [Datos del bloque 101], responda a la siguiente pregunta: ¿Cuál es el Setpoint?”.

3. **Generación (Generation):** El LLM procesa la pregunta junto al contexto verídico. En esta etapa, el modelo actúa como un motor de síntesis y redacción, eliminando la necesidad de alucinar o inventar información, entregando una respuesta coherente en lenguaje natural basada estrictamente en los datos proporcionados.

Esta técnica es vital para la minería de gran escala, ya que garantiza la **trazabilidad**. El sistema no solo responde, sino que es capaz de citar la fuente exacta dentro del archivo FHX, permitiendo auditorías rápidas y actualizaciones constantes de la base de conocimientos sin necesidad de modificar los pesos del modelo de IA.

2.5.2. Teoría de Grafos aplicada a Lógica de Control

A diferencia del código de programación tradicional, que posee una estructura lineal, los sistemas de control industriales son inherentemente topológicos. Un diagrama de bloques de funciones (FBD) en DeltaV se define, matemáticamente, como un **Grafo Dirigido**. Esta abstracción es el pilar fundamental para que el Agente DCS y los usuarios sin conocimientos profundos de control comprendan el flujo de las señales.

Definición Matemática: Un grafo G se define como un par ordenado $G = (V, E)$, donde:

- V (Vértices o Nodos): En el contexto de un sistema DCS, cada nodo representa un Bloque de Función (ej. PID, Analog Input, Arithmetic). Cada nodo almacena atributos críticos como su Tag, Ganancia, Tiempo de Reset y otros parámetros de configuración.
- E (Aristas o Enlaces): Representan las conexiones físicas y lógicas (*wires*). Una arista dirigida (u, v) indica un flujo de datos unidireccional desde la salida del bloque u hacia la entrada del bloque v .

Mediante esta representación, la Inteligencia Artificial obtiene un mapeo conceptual de las dependencias lógicas que, en el código .fhx original, resultan extremada-

mente complejas de procesar. Para el usuario final, estos grafos se traducen en representaciones visuales intuitivas de nodos y aristas, facilitando la interpretación de la lógica de los controladores DeltaV.

Aplicación en la Ingeniería Inversa: Al transformar el archivo de texto plano .fhx en una estructura de datos de grafos utilizando librerías especializadas como **NetworkX**, el desafío de interpretar código se convierte en un problema de “recorrido de caminos”:

- **Trazabilidad de Señales:** Ante la activación de una alarma, se aplican algoritmos de búsqueda como *BFS (Breadth-First Search)* o *DFS (Depth-First Search)* para realizar un recorrido inverso desde el nodo afectado, identificando automáticamente todos los bloques y señales precedentes que influyeron en dicho estado.
- **Detección de Comunidades:** Permite aislar y agrupar automáticamente bloques que forman “Lazos de Control” cerrados (Sensor → Controlador → Actuador), separándolos del ruido visual de la planta completa.

Finalmente, el uso de grafos resuelve el problema de la **Ventana de Contexto**. En lugar de saturar la memoria del LLM con un archivo de un millón de líneas, el sistema extrae únicamente el “vecindario” del nodo consultado (nodos adyacentes y sus conexiones), optimizando el uso de tokens y garantizando una precisión técnica superior en la respuesta.

3. DISEÑO DE LA SOLUCIÓN

En este capítulo se describe el diseño arquitectónico y lógico del sistema “Agente DCS” (en adelante, la solución propuesta). Se detalla la integración de los componentes de procesamiento de datos, la estructura de almacenamiento basada en grafos y la lógica de orquestación del modelo de lenguaje para la interpretación de sistemas de control en el contexto de CMDIC.

3.1. *Arquitectura General del Sistema*

La arquitectura del sistema se ha diseñado bajo un enfoque modular y desacoplado, permitiendo que cada componente cumpla una función específica dentro del pipeline de procesamiento de datos. El objetivo es transformar archivos de configuración industrial de baja legibilidad humana (.fhx) en una base de conocimientos estructurada y accionable, garantizando la trazabilidad y seguridad en cada etapa.

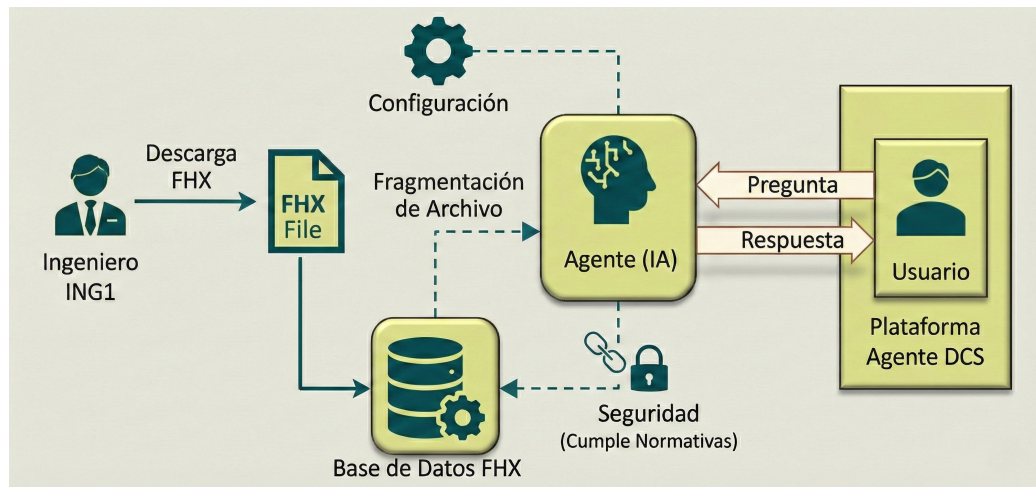


Fig. 3.1: Arquitectura General y Flujo de Datos del Agente DCS.

3.1.1. Capas del Sistema y Procesos de Integración

A continuación, se describen detalladamente las cuatro capas fundamentales que componen la solución, integrando los roles de ingeniería y usuario en el flujo de datos:

1. **Capa de Ingesta y Parsing:** El proceso inicia con la carga del archivo `.fhx`, donde el sistema realiza un análisis léxico exhaustivo para identificar bloques de función e interconexiones. Esta capa se desglosa en dos etapas críticas de integración:

- **Gestión de Ingeniería:** El ingeniero a cargo es el responsable de la descarga inicial de los archivos desde la plataforma DeltaV. Estos deben mantenerse en el formato original (`.fhx`) y ser subidos a la base de datos centralizada con sus nombres respectivos, siguiendo estrictamente la norma **ISA 5.1** [6]. Esta base de datos actúa como el **cerebro de toda la información**, donde cada archivo queda conectado mediante código para que el Agente DCS pueda extraer datos precisos ante cualquier consulta.
- **Intervención del Usuario y Segmentación:** El usuario final cumple un rol activo antes de iniciar el diálogo con la IA. Debe seleccionar el archivo específico de una lista desplegable que muestra todos los elementos cargados previamente por ingeniería. Una vez seleccionado, el sistema ejecuta una fragmentación en bloques de datos específicos: **TAG, WIRES y FUNCTION BLOCK**. Esta segmentación no es volátil; se guarda en una carpeta específica dentro de la plataforma, la cual es **creada y nombrada por el propio usuario**. Este diseño permite que el archivo quede procesado y listo para futuras consultas, evitando tener que repetir el ciclo de fragmentación y mejorando drásticamente el tiempo de respuesta efectivo.

2. **Capa de Modelado Estructural (Grafo de Control):** Los datos limpios se inyectan en un motor de grafos que preserva la topología lógica. Dentro de la partición del archivo, el segmento de **WIRES** es el eje principal, ya que contie-

ne todas las conexiones y nombres de bloques. Un aspecto diferenciador de este diseño es que, aunque algunos bloques funcionales no informan su funcionamiento interno de forma explícita, el sistema identifica e integra las **compuertas lógicas (AND, OR, XOR, etc.)**. Estas compuertas son contempladas como nodos fundamentales para generar la representación de la lógica completa, permitiendo que el grafo entienda no solo el camino de la señal, sino la condición que la activa.

3. **Capa de Orquestación e Inferencia:** Aquí el orquestador prepara el *prompt* para el modelo **Llama 3.1 8B** en un entorno local. Para blindar el sistema contra alucinaciones y saturación de memoria (*tokens*), se diseñó una capa de seguridad extra:

- **Procesamiento Fragmentado:** El sistema impone un límite de caracteres por consulta. Si la información es extensa, el orquestador realiza **varias consultas internas de forma autónoma**, recolecta todas las respuestas parciales y finalmente genera una respuesta general que satisface la consulta inicial del usuario sin perder contexto.
- **Desacoplamiento Visual:** Como medida de seguridad y eficiencia, el sistema genera imágenes de la lógica **previo a la respuesta textual**. Estas representaciones visuales nacen de librerías de código y no de la IA generativa; esto garantiza que el usuario tenga un primer acercamiento visual 100 % fiel al archivo, ahorrando recursos y evitando que el Agente DCS cometa errores en la interpretación gráfica.

4. **Capa de Interfaz y Entrega:** Es la interfaz reactiva que conecta al mantenedor con la lógica del DCS. El diseño considera críticamente que el usuario puede ser nuevo en la planta o tener poca antigüedad operando el sistema. Por ello, la entrega de información se complementa con:

- **Descripciones Exactas y Manuales:** Se integran guías de usuario y una

”militancia.ºperativa dentro de la plataforma para orientar al operario.

- **Estandarización de Consultas:** El sistema guía al usuario para que las preguntas sean genéricas para cada archivo, asegurando que, independientemente de la experiencia del mantenedor, el Agente DCS entregue un diagnóstico de alta calidad y utilidad técnica.

3.1.2. Infraestructura y Entorno de Ejecución

El cumplimiento de las normativas de ciberseguridad de CMDIC es el pilar del entorno técnico:

- **Ejecución Local:** Todos los componentes (Base de datos, Grafos y LLM) residen en un servidor físico dentro de la infraestructura privada de la empresa. Esto garantiza que la información sensible jamás salga del perímetro de la compañía.
- **Sin Dependencias Externas:** No se realizan llamadas a APIs de terceros ni se depende de nubes públicas, protegiendo tanto los datos del proceso como las librerías exclusivas desarrolladas para la minera.
- **Optimización de Recursos:** Mediante la **cuantización de 4 bits**, el pipeline completo se ajusta a los límites de una GPU *Nvidia RTX 3060* (hardware de nivel personal/estación de trabajo), permitiendo que la solución sea escalable sin requerir servidores de cómputo masivo.

3.2. Interfaz de Adquisición de Datos Crudos y Preprocesamiento Estructural

El archivo `.fhx` es un formato propietario de Emerson DeltaV que contiene la configuración integral de un nodo de control en formato de texto plano (ASCII). Debido a su naturaleza redundante, este archivo incluye metadatos de autoría, versiones históricas y parámetros de visualización que, si bien son útiles para el entorno gráfico

del DCS, carecen de valor para la interpretación lógica algorítmica. El diseño de este módulo realiza una “limpieza quirúrgica” para transformar este archivo masivo en una estructura de datos ligera y jerárquica, optimizando la ventana de contexto y evitando la pérdida innecesaria de *tokens* durante el procesamiento del modelo de lenguaje.

3.2.1. Metodología de Extracción y Segmentación de Datos

Para procesar el archivo de manera eficiente sin saturar los recursos de memoria del servidor local, se diseñó un motor de adquisición basado en **Expresiones Regulares (Regex)** y un algoritmo de búsqueda por palabras clave. Esta lógica de extracción garantiza que el sistema identifique la semántica industrial oculta en el texto crudo.

La segmentación se rige por la identificación de etiquetas críticas que definen la arquitectura de control:

1. **Identificación de Bloques y Estructuras:** El algoritmo detecta los delimitadores de inicio y fin de cada bloque, priorizando las siguientes categorías:

- **FUNCTION_BLOCK:** Representa el “ADN” de la lógica. Incluye bloques creados en librerías internas exclusivas de Collahuasi y bloques compuestos. Estos últimos son fundamentales, ya que encapsulan lógicas predefinidas, uniones entre compuertas lógicas y reglas de funcionamiento complejas que definen estados operativos del proceso minero.
- **WIRES:** Constituye el sistema nervioso del controlador. Aquí se definen las uniones entre bloques mediante un formato de origen (*source*), destino (*destination*) y vectores de señal. Esta sección permite reconstruir la topología completa para que el Agente DCS pueda graficarla y analizarla, independientemente del estado dinámico de la señal.
- **SOURCE:** Gestiona la comunicación inter-controladores. En una planta de gran escala como CMDIC, las señales suelen viajar entre distintos controladores. El sistema captura estas fuentes externas como variables de

entrada que afectan la lógica interna, permitiendo al Agente comprender que el comportamiento del lazo local depende de una condición de otra área de la planta.

2. **Recuperación de Datos de Auditoría y Contexto:** Más allá de la lógica pura, el módulo rescata metadatos críticos para la trazabilidad. Se extrae la fecha de creación, la última modificación, el autor de la lógica y la identificación del controlador bajo la norma **ISA 5.1**. Esto permite que el Agente DCS asocie la lógica con un área geográfica y operacional específica (*Molienda, Correas, Flotación, Chancado, etc.*), otorgando conciencia situacional a la respuesta generada.
3. **Filtrado Selectivo y Normalización:** Se aplica un filtro de exclusión para eliminar datos de “ruido” (coordenadas X, Y del HMI, colores, fuentes). Posteriormente, el parser valida los nombres de los bloques. Si se detectan **Tags** con errores de tipeo o que no cumplen con la nomenclatura de CMDIC, el sistema genera una alerta automática. Esta funcionalidad permite detectar errores en la descarga del archivo o inconsistencias en la base de datos, incentivando un bucle de mejora continua entre el usuario y el ingeniero a cargo.

3.2.2. Transformación a Objeto Jerárquico Estructurado

Una vez filtrado, el archivo deja de ser texto plano para convertirse en un objeto estructurado (JSON/Diccionario) organizado en niveles de profundidad:

- **Nivel 1 (Modulo):** Identificación del módulo y área funcional (ej. Molienda SAG 1).
- **Nivel 2 (Nodos):** Inventario de bloques de función, compuertas lógicas y fuentes externas con sus respectivos parámetros operacionales (*Setpoints, Gains, Limits*).
- **Nivel 3 (Vínculos):** Mapeo de conexiones directas que definen la causalidad del sistema de control.

3.2.3. Manejo Crítico de Inconsistencias y Lógica Probabilística

Una innovación fundamental en el diseño es su capacidad para gestionar lógicas incompletas o errores de indentado en el archivo original. En lugar de detener el procesamiento, el sistema actúa de la siguiente manera:

- **Diagnóstico de Desconexión:** Si una unión lógica está rota (falta de *wire*), el sistema informa explícitamente que la señal de salida (OUT) no guarda dependencia con la fuente de entrada (IN), basándose en la configuración real del archivo.
- **Simulación de Escenarios Ideales:** Como valor agregado, el sistema es capaz de plantear una situación hipotética: informa al usuario cómo se comportaría la lógica “si la conexión estuviese unida”. Esto permite al usuario descubrir fallas internas y encontrar soluciones a problemas de configuración que podrían haber pasado desapercibidos durante años, generando una lógica interna de pesos que asume variables críticas para el soporte de decisiones.

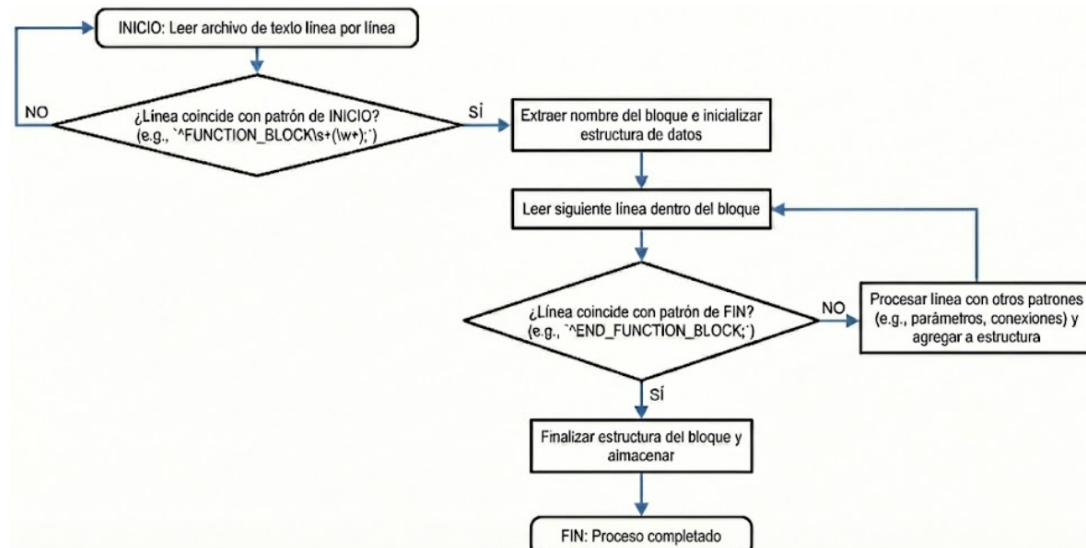


Fig. 3.2: Flujo de identificación

3.3. Modelado Estructural: Conversión a Grafo

La lógica de control de un Sistema de Control Distribuido no debe interpretarse como una secuencia lineal de instrucciones de programación, sino como una red compleja de dependencias funcionales dinámicas. En los sistemas Emerson DeltaV, el comportamiento de un proceso está definido por la interconexión de bloques que intercambian señales en tiempo real para mantener la estabilidad operacional del proceso de la planta.

Para que el Agente DCS pueda realizar razonamientos topológicos y diagnósticos completos, se ha diseñado un modelo basado en la **Teoría de Grafos**. Esta abstracción permite transformar el archivo plano en un **Grafo Dirigido** $G = (V, E)$, donde la topología del control se convierte en una estructura de datos navegable algorítmicamente.

En esta arquitectura, un enlace (u, v) existe únicamente si existe un *Wire* explícito que conecte la salida (*OUT*) del bloque u con una entrada o fuente del bloque v . Esta formalización es crítica, ya que permite aplicar algoritmos de búsqueda sobre la lógica para identificar flujos de datos que, en el código fuente, estarían separados por miles de líneas de texto.

3.3.1. Caracterización Profunda de Nodos

Cada bloque funcional identificado es instanciado como un objeto enriquecido dentro del grafo. A diferencia de un sistema de búsqueda textual, el nodo almacena el “estado estructural” del bloque.

Atributos y Dimensiones del Nodo

Cada vértice v posee un diccionario de atributos que el Agente DCS utiliza para generar el contexto. Estos se clasifican en tres categorías:

1. **Identificación y Localización:**

- **Tag (ID):** Identificador único bajo norma ISA 5.1.
- **Jerarquía Operacional:** Información del área (Molienda, Flotación) extraída del encabezado del módulo.
- **Descripción de Ingeniería:** Texto aclaratorio que define la función del bloque en lenguaje natural.

2. Parámetros de Comportamiento:

- **Tipo de Bloque:** Define la clase de algoritmo (ej. *PID*, *AI*, *AO*, *COMP*, *SPLT*).
- **Sintonización:** Valores de ganancia proporcional (K_p), tiempo integral (T_i), y derivada (T_d).
- **Rango y Escalamiento:** Límites de operación (0-100 %) y unidades de ingeniería (PSI, GPM, %).

3.3.2. Aristas

Las aristas constituyen el elemento de vinculación de la arquitectura, representando las conexiones lógicas y físicas entre los bloques o nodos previamente generados. La representación mediante grafos resulta ideal para el análisis de sistemas DCS, dado que permite modelar de forma natural las señales de lazo cerrado, las cuales son tratadas algorítmicamente como flujos de retorno o bloques de estado dentro de la lógica del controlador.

Esta configuración de aristas es lo que permite que el motor de grafos transforme un archivo de texto plano en un mapa topológico funcional, donde el Agente DCS puede "navegar" para explicar las condiciones de operación al usuario final.

Las aristas generan los subgrafos no conectados, es decir, dentro de los controladores existen distintas lógicas en paralelo las cuales no tienen relación entre sí dado esto para analizar la lógica en específico, se separa en pequeños diagramas para así

hacer consultas más específicas sin necesidad de analizar los grafos en paralelo que no afectan.

Identificación de Puertos de Destino

Dada la naturaleza de los bloques de función, un nodo puede recibir múltiples señales. El modelo de grafo almacena en la arista el atributo *Destination*. Por ejemplo, una arista hacia un bloque de suma (*SUM*) llevará la etiqueta *IN1* o *IN2*. Esto permite al Agente DCS responder preguntas específicas como: “¿Qué señal está entrando por el puerto de ”SUM” del bloque de molienda?”.

3.3.3. Modelado de Compuertas Lógicas como Puentes de Decisión

Un desafío técnico resuelto en este diseño es la representación de las compuertas booleanas (*AND*, *OR*, *NOT*, *XOR*). En lugar de tratarlas como simples parámetros, se modelan como **Nodos de Decisión Estructural**.

Esto permite al sistema identificar decisiones lógicas. Si un motor no arranca, el Agente DCS puede recorrer el grafo hacia atrás, realizando un backtracking, desde el nodo del motor, pasando por la compuerta *AND* de permisivos, e identificar exactamente qué entrada interlock no está en el estado lógico correcto para permitir la salida.

3.3.4. Implementación Algorítmica con NetworkX

La construcción del grafo se realiza en memoria volátil utilizando la librería **NetworkX** en Python, lo que otorga acceso a un conjunto de herramientas matemáticas de análisis de redes.

Filtrado Topológico para Optimización de Contexto

Como se menciona anteriormente, para no saturar la ventana de contexto del LLM, el diseño utiliza el concepto de Vecindad de Nodo. Es decir, Si el usuario pregunta por

el bloque LC-100, el sistema genera un sub-grafo 3.3 inducido:

$$G_{ego}(v, d) = \{u \in V : dist(v, u) \leq d\} \quad (3.1)$$

Donde d es el radio de búsqueda considerando una variable grande hasta no recibir más uniones, de esta manera el algoritmo llega al final de ese nodo generando un grafo completo en donde se aloja la búsqueda necesitada. Esto garantiza que el Agente DCS reciba solo la información estructural relevante, maximizando la precisión de la respuesta y minimizando la latencia de inferencia en el servidor local.

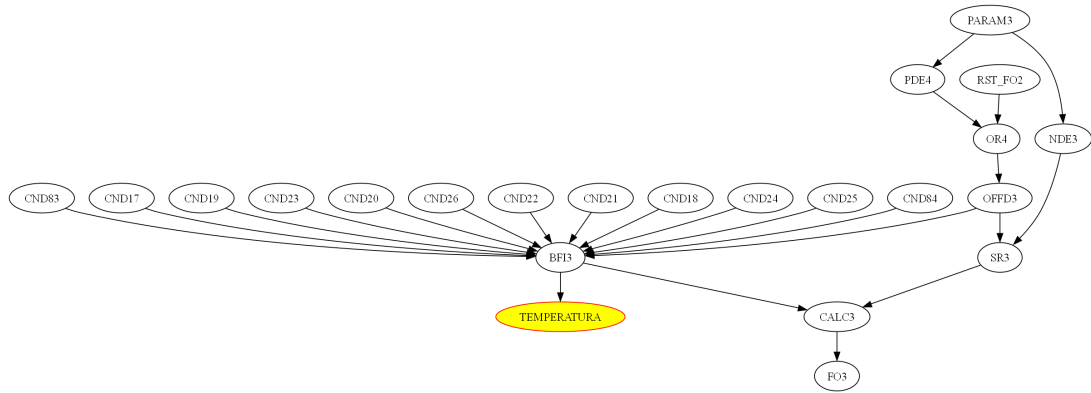


Fig. 3.3: Grafo generado de conexiones que regulan enclavamiento de Temperatura

3.4. Diseño de la Lógica del Agente y Orquestación RAG

El corazón del sistema “Agente DCS” reside en su motor de orquestación, el cual implementa una arquitectura de **Generación Aumentada por Recuperación (RAG)** adaptada a entornos de control industrial.

3.4.1. Estrategia de Búsqueda y Recuperación Basada en Grafos

La eficiencia del agente depende de su capacidad para seleccionar la información técnica relevante dentro de archivos .fhx que pueden contener millones de caracteres. El proceso de búsqueda se divide en tres capas lógicas:

Identificación de Entidades y Resolución de Consultas

Cuando el usuario realiza una consulta por ejemplo “¿Por qué no abre la válvula XV-101?”, el sistema activa un display para seleccionar todos los tag asociados dentro del grafo asociado, de esta manera evita que el usuario haga una consulta sobre una señal inexistente dentro del controlador y le da un contexto real al LLM para realizar el análisis. A partir de esta extracción se genera lo siguiente:

- **Extracción de Nodo Raíz:** El sistema localiza el nodo correspondiente en el grafo generado por *NetworkX*.
- **Análisis de Vecindad:** Para evitar la saturación de *tokens*, el algoritmo no envía toda la lógica al modelo. En su lugar, extrae un sub-grafo que comprende el nodo consultado y sus conexiones directas e indirectas. Esto asegura que el LLM vea los permisivos, enclavamientos y señales de entrada que afectan directamente al componente en cuestión.

Tratamiento de Señales Externas

Dada la complejidad de CMDIC, muchas señales provienen de otros controladores. La lógica de búsqueda detecta las etiquetas *SOURCE* o fuentes, en el fragmento segmentado. Si una señal crítica proviene de un controlador externo, el orquestador marca esta dependencia, informando al modelo que la causa raíz podría residir fuera del archivo actual, mitigando así conclusiones erróneas basadas en información parcial.

3.4.2. Diseño del Prompt Engineering

El Prompt Engineering es una herramienta crítica a día de hoy al momento de trabajar con la IA, esta toma la labor de dar las ordenes exactas a los modelos para que generen lo que realmente uno requiere. Dado esto, se ha diseñado para transformar un modelo de propósito general (Llama 3.1 8B) en un especialista en sistemas de control.

El diseño del prompt parte indicándole el contexto minero, la necesidad del análisis y las limitancias al momento de generar una respuesta, estas instrucciones vienen diseñadas en el lenguaje exacto que son entrenados los modelos, de esta manera se obtiene mejores resultados a las instrucciones deseadas perdiendo un pequeño margen de Tokens para diseñar una explicación detallada de instrucciones, esto debido a que los Tokens son considerados desde el prompt hasta las respuestas generadas.

3.4.3. Orquestación y Razonamiento Iterativo (Sub-consultas)

Para manejar la limitación de la ventana de contexto y garantizar la precisión en lógicas extensas, se implementó un flujo de trabajo llamado **Razonamiento Fragmentado**:

Ciclo de Sub-consultas Internas

Ante preguntas que requieren analizar múltiples lazos de control, el orquestador descompone la tarea en pasos secuenciales:

1. **Análisis de Componentes:** El modelo identifica primero qué bloques individuales están involucrados.
2. **Síntesis Global:** Se recopilan las respuestas de las etapas anteriores para generar el diagnóstico final.

Este método evita que el modelo ignore detalles técnicos situados al final de un contexto largo.

3.4.4. Eficiencia en el Uso de Tokens y Cuantización

Para operar en la infraestructura **Air-Gapped** de la compañía con hardware limitado (RTX 3060), se implementó una estrategia de gestión de recursos:

- **Prompt Truncation:** El orquestador monitorea constantemente la cantidad de caracteres. Si el contexto del sub-grafo excede el límite crítico, se activan filtros

de prioridad que descartan metadatos secundarios para priorizar los parámetros de control, un ejemplo de los prompt utilizados esta representado en la siguiente figura ??.

- **Inferencia Local Cuantizada:** El uso de modelos en formato GGUF permite que el motor de orquestación mantenga el modelo en la memoria VRAM de forma permanente, reduciendo la latencia de respuesta de minutos a segundos, transformando así la experiencia de usuario en un diálogo fluido, el consumo utilizado por el hardware se puede ver representado en la siguiente figura 3.4.

```
1 def dividir_texto(texto, tamaño=4000):
2     """Divide el texto en fragmentos de máximo 4000
3     ↪ caracteres."""
4     return [texto[i:i+tamaño] for i in range(0, len(texto),
5     ↪ tamaño)]
6
7 def analizar_fragmento(fragmento, numero):
8     """Envía cada fragmento al modelo y lo analiza
9     ↪ individualmente."""
10    prompt = (
11        "Eres un ingeniero de control experto en
12        ↪ automatización minera. "
13        "Analiza el siguiente fragmento de un código FHX de un
14        ↪ Sistema de Control "
15        "Distribuido (DCS) utilizado en minería.\n"
16        "No asumas que el sistema es una correa
17        ↪ transportadora.\n\n"
18        "Identifica qué sistema controla este código (molino,
19        ↪ correa transportadora, "
20        "bomba, flotación, espesador, ventilador, agua o
21        ↪ válvula de control).\n"
22        "Explica cómo interactúan los bloques de función (OND,
23        ↪ AND, OR, CND, RS, CMP, SEL).\n"
24        "Identifica variables clave (presión, caudal,
25        ↪ temperatura, velocidad, torque, "
26        "apertura de válvulas, estado de motores, nivel,
27        ↪ alarmas).\n"
28        "Determina si contiene lógica de seguridad (alarmas,
29        ↪ bloqueos o trip: TRIP1, TRIP2, ALARMA_ROC).\n"
30        "Explica tiempos de espera/retardos y su impacto.\n"
31        "Si el fragmento es insuficiente, indica
32        ↪ qué información falta.\n\n"
33        f"Fragmento #{numero}:\n{fragmento}"
34    )
35    respuesta = llm.invoke(prompt)
```

```

24     return respuesta
25
26 def guardar_salida(salida,
27     ↪ nombre_archivo="E_136-ST_Llama.txt"):
28     """Guarda la salida en un archivo de texto."""
29     with open(nombre_archivo, "w", encoding="utf-8") as
30         ↪ archivo:
31         archivo.write(salida)
32
33 if __name__ == "__main__":
34     ruta = "136-ST-2305_P_UTF8.txt" # Archivo de texto con la
35     ↪ lógica de control
36     texto = cargar_txt(ruta) # Cargar el archivo como texto
37
38     fragmentos = dividir_texto(texto)
39
40     print(f"[PIN] Se encontraron {len(fragmentos)} fragmentos.
41     ↪ Procesando...")
42
43     resultados = []
44     for i, fragmento in enumerate(fragmentos):
45         print(f"[SEARCH] Analizando fragmento
46         ↪ {i+1}/{len(fragmentos)}...")
47         resultado_parcial = analizar_fragmento(fragmento, i+1)
48         resultados.append(f"--- Fragmento {i+1}
49         ↪ ---\n{resultado_parcial}\n")
50
51     texto_completo = "\n".join(resultados)
52
53     prompt_resumen = (
54         "Basado en los siguientes análisis parciales de código
55         ↪ de un sistema de control minero, "
56         "proporciona una conclusión general sobre su
57         ↪ funcionamiento:\n\n"
58         "- ¿Qué equipo o sistema controla este código?
59         ↪ (correa, válvula, PID, bomba, flotación,
60         ↪ espesador, etc.)\n"
61         "- ¿Cómo interactúan las variables principales dentro
62         ↪ del sistema?\n"
63         "- Si es un sistema complejo, describe la lógica
64         ↪ general del control.\n\n"
65         "Importante: no hagas solo un resumen; analiza el
66         ↪ propósito y funcionamiento completo.\n\n"
67         f"Análisis previos:\n{texto_completo}"
68     )
69
70     resumen_final = llm.invoke(prompt_resumen)
71
72     guardar_salida(
73         f"{texto_completo}\n\n--- Resumen Final
74         ↪ ---\n{resumen_final}",
75         "E_136-ST_Llama.txt"
76     )

```

```

62     )
63
64     print("\n[OK] Análisis completado y guardado en
        ↪ 'E_136-ST_Llama.txt'")

```

Código 1: Ejemplo de prompt 1

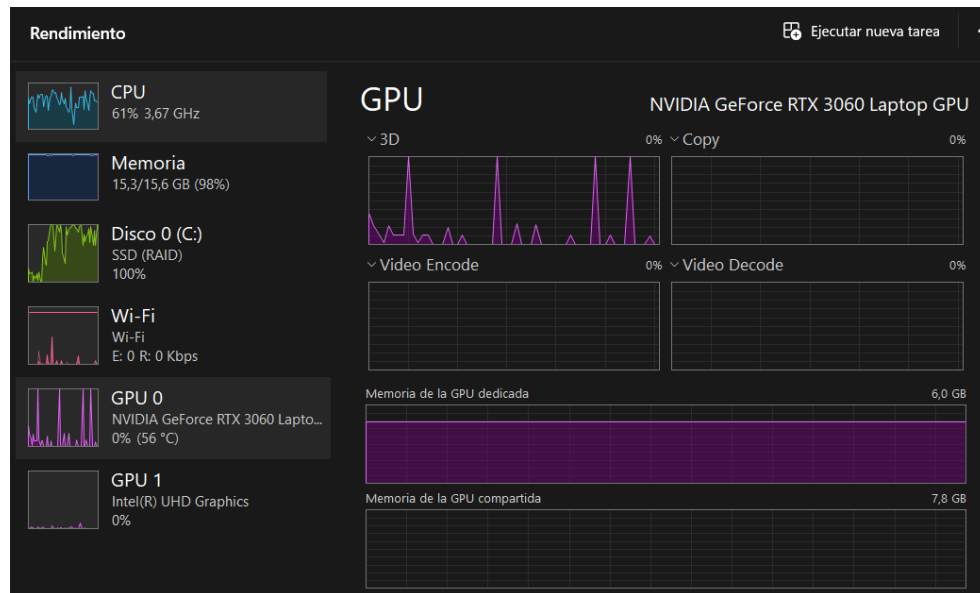


Fig. 3.4: Rendimiento de la GPU AL generar respuesta

3.5. *Arquitectura de Visualización y Experiencia de Usuario (UI/UX)*

La interfaz de usuario del Agente DCS se ha diseñado bajo el paradigma de que el usuario no tiene los conocimientos base para el entendimiento de los controladores, dado esto la interfaz y guía de esta es detallada para poder hacer las consultas sin necesidad de entendimiento previo, priorizando la entrega de información técnica crítica sobre la estética superficial. El objetivo es proporcionar una capa de interacción intuitiva que oculte la complejidad del procesamiento de grafos y la inferencia de lenguaje natural, permitiendo que un ingeniero de planta interactúe con el sistema mediante lenguaje técnico especializado y el usuario común pueda aprender y a diferenciar funciones internas desconocidas.

3.5.1. Selección del Framework: Streamlit como Motor Visual

Se seleccionó **Streamlit** como el núcleo del desarrollo de la interfaz debido a su capacidad para generar aplicaciones web didácticas y simples directamente en Python, lo que facilita la integración nativa con el motor de grafos y la lógica de inferencia, esto debido a que todo lo anterior es nativo de python.

Las ventajas técnicas de esta elección incluyen:

- **Estado de Sesión Persistente:** Permite mantener el historial de la conversación y el contexto del archivo seleccionado sin necesidad de re-procesar los datos en cada interacción.
- **Visualización de Datos Dinámica:** Integración fluida con librerías para la renderización de los lazos de control en tiempo real.
- **Arquitectura de Micro-servicios Locales:** Al ejecutarse en un puerto local dentro de la red privada de CMDIC, garantiza el cumplimiento del entorno.

3.5.2. Diseño Funcional y Componentes de la Interfaz

La interfaz de usuario se organiza en distintas áreas estratégicas diseñadas para minimizar el error humano y maximizar la eficiencia en la recuperación de información. Los componentes principales que integran el diseño funcional se describen a continuación:

- **Gestión de Archivos y Base de Datos:** Incluye las funciones para la selección y el guardado de los archivos fragmentados. Este módulo establece la conexión directa con la base de datos, asegurando la disponibilidad de la información procesada para el sistema.
- **Módulo de Consultas Generales:** Corresponde al apartado destinado a la interacción con el modelo de inteligencia artificial. Este componente permite realizar consultas técnicas de carácter general sin requerir dependencia de una conexión a internet, garantizando la autonomía del sistema.

- **Selector de Preguntas Asistido:** Herramienta diseñada para facilitar la elección de consultas basadas específicamente en las conexiones y los *tags* del archivo. Al utilizar un selector predefinido en lugar de entrada de texto libre, se eliminan los errores humanos de tipeo y se crea un entorno de ejecución estandarizado que protege la integridad y el desempeño del *prompt* de fondo.
- **Creador de grafos:** Dentro del ítem de preguntas técnicas, existe la opción de generar los grafos asociados al archivo seleccionado desplegando en el display todas las imágenes generadas a partir de las conexiones, además de presentar resaltado el tag seleccionado en la consulta a realizar para entender a que conjunto de conexiones se integra esta señal.

Consola de Diálogo Técnico

Espacio reactivo donde se despliega la orquestación entre el usuario y la IA, diseñada para soportar lenguaje natural técnico.

3.5.3. Galería de Implementación y Capturas de Diseño

En las siguientes imágenes se presenta la interfaz de interacción del usuario con sus distintas opciones de uso:

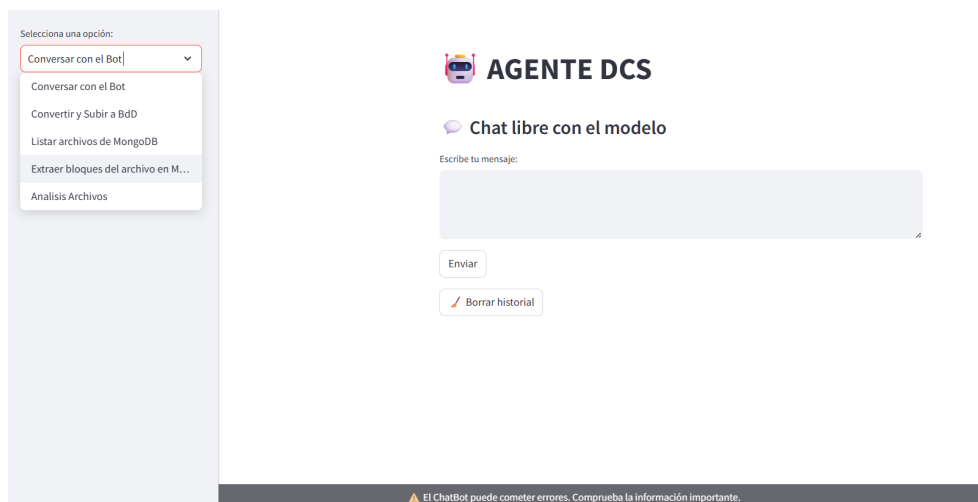


Fig. 3.5: Interfaz Inicial del Agente DCS con sus opciones

AGENTE DCS

Archivos almacenados en MongoDB

- 136-CV-3001-A_LG_UTF8
- 136-ST-2305_P_UTF8
- 141-ML-1012-CTRL_UTF8
- 150-BL-8007_SEQ_UTF8
- 153-COU-MUES_UTF8
- 153-ISIO-VALID-D_UTF8
- 105-TEST

Fig. 3.6: Archivos almacenados en la base de datos

AGENTE DCS

Extraer y guardar bloques del documento de MongoDB

 Nombre del documento (sin .txt):

 Nombre de la carpeta donde se guardarán los bloques:

 La carpeta 'bloques_extraidos' ya existe.

¿Deseas sobrescribirla?

☒ No

☐ Sí

 Ingresa un nuevo nombre para la carpeta:

 El nuevo nombre de carpeta también ya existe. Por favor, elige otro.

Fig. 3.7: Display de extracción de archivo y segmentación

AGENTE DCS

ChatBot DCS

 Selecciona la carpeta con los bloques extraídos:

 Selecciona archivo WIRE:

 Selecciona archivo ATTRIBUTE_INSTANCE:

 Selecciona un TAG a analizar:

 Selecciona el tipo de consulta:

 El ChatBot puede cometer errores. Comprueba la información importante.

Fig. 3.8: Interfaz de consultas sobre tags y conexiones

3.6. Síntesis Metodológica del Diseño

La metodología aplicada en el desarrollo de la solución se fundamenta en un enfoque ingenieril de carácter investigativo y experimental, orientado a resolver la brecha tecnológica identificada en CMDIC respecto a la interpretación de la lógica de control DCS. Este enfoque, integra técnicas de ingeniería de control clásico, modelamiento estructural mediante grafos y procesamiento de lenguaje natural asistido por modelos de lenguaje de gran escala.

El proceso metodológico se estructuró en las siguientes fases críticas:

1. **Análisis de Requerimientos y Restricciones:** Se definieron los requerimientos funcionales priorizando la seguridad de la información y la operación 100 % *offline*, condición para cumplir con los protocolos de ciberseguridad industrial de la compañía.
2. **Modelamiento Estructural:** En esta etapa, los archivos crudos .*fhx* fueron sometidos a un proceso de extracción y transformación. Se utilizó la teoría de grafos para representar las dependencias lógicas como una red navegable, permitiendo que el sistema supere la linealidad del texto plano y comprenda la topología real del controlador.
3. **Implementación de la Arquitectura RAG:** Se diseñó la capa de orquestación utilizando un enfoque de *Retrieval Augmented Generation*. Esta técnica permite inyectar el "vecindario" del grafo consultado directamente en el *prompt* del modelo Llama 3.1, garantizando respuestas ancladas a la verdad técnica y minimizando el riesgo de alucinaciones lógicas logrando mejores respuestas técnicas y explicadas para todos los usuarios.
4. **Validación Operacional:** La metodología concluye con la evaluación del sistema mediante casos de estudio reales (Molienda, Correas y Flotación), midiendo el impacto en indicadores clave como el tiempo medio de reparación y la reducción de costos operativos (OPEX).

Este enfoque iterativo aseguró que la arquitectura propuesta no solo sea técnicamente robusta, sino que sea escalable a otras marcas de DCS presentes en la industria, consolidando un estándar de interpretación de lógica asistida y guiada por inteligencia artificial.

4. RESULTADOS Y CONCLUSIONES

En el presente capítulo se presentan los resultados obtenidos tras la implementación y validación del Agente DCS en entornos controlados y operativos. Se efectúa una evaluación comparativa del desempeño del sistema desarrollado frente a las metodologías convencionales de ingeniería inversa, analizando métricas de eficiencia temporal, precisión y capacidad de recuperación de información. Finalmente, se exponen las conclusiones fundamentales del proyecto y se identifican las oportunidades de mejora detectadas durante la fase experimental.

4.1. Casos de Estudio

Con el objetivo de validar la versatilidad del Agente DCS, se seleccionaron activos digitales críticos que representan la diversidad de las operaciones en CMDIC. A continuación, se detallan los archivos fundamentales empleados en las pruebas de diagnóstico:

- **Área de Molienda (Tag ML):** Se seleccionó el archivo 141-ML-1012-CTRL. Este documento representa la lógica de control de un molino, la cual se caracteriza por una elevada densidad de enclavamientos (*interlocks*) de seguridad y bucles PID en cascada.
- **Área de Correas (Tag CV/ST):** Se procesaron archivos tales como 136-CV-3001-A_LG (Válvula de salida correa) y 136-ST-2305_P (Correa Transportadora). Estas lógicas se consideran críticas para la continui-

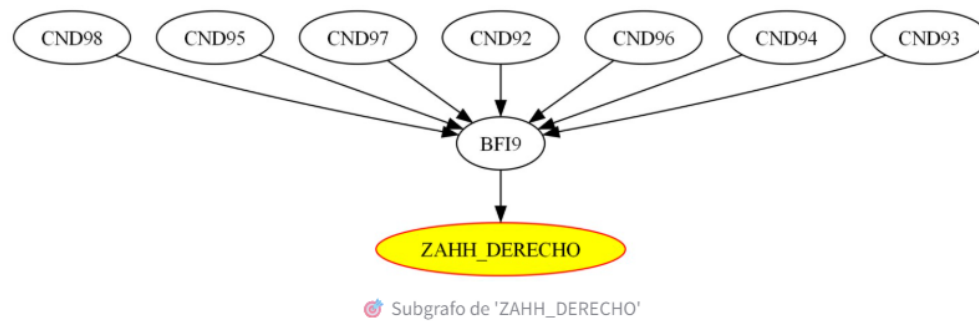
dad operacional, dado que involucran secuencias de arranque y parada sincronizadas.

- **Área de Flotación:** Los archivos 150-BL-8007_SEQ y 153-ISIO-VALID se emplearon para verificar la capacidad del agente en la interpretación de secuencias lógicas y la validación de señales de entrada/salida (I/O).
- **Casos de Prueba (Tag TEST):** El archivo 105-TEST se utilizó para ejecutar pruebas de estrés en el motor de *parsing* y validar la detección de errores de sintaxis en el archivo fuente .fhx.

La diversidad de los casos de estudio seleccionados garantiza que el Agente DCS posea la capacidad de generalizar su razonamiento técnico a través de distintas filosofías de control, abarcando desde la regulación de variables continuas en el proceso de molienda hasta las lógicas discretas de seguridad presentes en los sistemas de transporte por correas.

4.2. Evaluación de Desempeño y Métricas

La validación del Agente DCS se realizó mediante un análisis comparativo entre el proceso de ingeniería inversa tradicional de manera manual y el proceso asistido por Inteligencia Artificial. Para cuantificar el éxito, se definieron métricas de eficiencia temporal y precisión técnica.



Mapeos relevantes en el subgrafo: ⇔

- CND92 → //136-ZAHH-01261R1/DI1/DISC_ACT
- CND93 → //136-ZAHH-01261R2/DI1/DISC_ACT
- CND94 → //136-ZAHH-01261R3/DI1/DISC_ACT
- CND95 → //136-ZAHH-01261R4/DI1/DISC_ACT
- CND96 → //136-ZAHH-01261R5/DI1/DISC_ACT
- CND97 → //136-ZAHH-01261R6/DI1/DISC_ACT
- CND98 → //136-ZAHH-01261R7/DI1/DISC_ACT

 El ChatBot puede cometer errores. Comprueba la información importante.

Fig. 4.1: Generación grafo y lista de tags relevantes ZAHH (Alarma de posicionamiento Alto Alto)

4.2.1. *Análisis de Reducción de Tiempos (Eficiencia)*

Como se identificó en el Mapa de Procesos (CGR Caja 1), el diagnóstico manual de un lazo de control complejo en DeltaV requiere que un ingeniero senior dedique entre 4 y 24 horas a la interpretación de archivos. En contraste, el Agente DCS automatiza la extracción y el razonamiento, reduciendo este tiempo a un intervalo de segundos o minutos.

La Tabla 4.1 resume los tiempos promedio registrados durante las pruebas con los archivos de las Correas:

Tab. 4.1: Comparativa de Tiempos de Ejecución: Proceso Manual vs. Agente DCS.

Tarea de Diagnóstico	Tradicional (Manual)	Agente DCS (IA)	Mejora (%)
Lectura y Parsing de .fhx	120 min	~90 seg	~ 98 %
Mapeo de Wires e Interlocks	3 - 8 horas	~2 min	~ 98 %
Interpretación Lógica Final	20 - 30 horas	~ 2 horas	~ 88 %
Total Acumulado	~35 horas	~ 2 horas	90 %

4.2.2. Precisión y Fidelidad Técnica (Exactitud)

Para evaluar la precisión, se sometieron 50 consultas técnicas al Agente DCS sobre lógicas críticas, entre ellas entendimiento de los controladores o funciones críticas como los enclavamientos de seguridad. Los resultados fueron validados por personal experto de CMDIC (Supervisores del area/Ingenieros Senior), comparando la respuesta del Agente con la lógica real del controlador4.2.

- **Tasa de Acierto Operacional:** El sistema alcanzó un **88 % de precisión** en la identificación de causas raíz de detención, hubo fallos en las alucinaciones mientras mas tenia que analizar, sin embargo, en apoyo del Agente DCS hubo una reducción de los tiempos en la detención del **90 %**.
- **Detección de Interlock:** El motor de grafos permitió una fidelidad del **100 %** en el rastreo de conexiones (*wires*), superando la capacidad de rastreo manual que suele omitir señales en archivos con más de 10,000 líneas.
- **Control de Enclavamientos:** Aquí existe una mejora del **95 % de precisión**, la causa principal es un método de hardcode el cual los enclavamientos necesitan una gran fidelidad de ocurrencia, para ello se generan capas extras de detención las cuales al momento de detectarse las consultas sobre los enclavamientos, se hace el llamado de la tool-calling especifica entregando un nuevo prompt al Agente para poder explicar en detalle el funcionamiento de los enclavamientos, además de entregar una lista detallada de estos dentro el controlador junto a un análisis exhaustivo de estos.

 Pregunta personalizada:

Que salida sera si MOLINO_DET esta en alto o 1, y asumimos que el resto de posibles tags con valores e:

 Analizar TAG

Fig. 4.2: Consulta de lógica ante MolinoDet

Sin embargo, como mencioné anteriormente, el estado de RS2/SET no afecta la salida final. En su lugar, debemos seguir las conexiones desde OR1/OUT_D , que es la salida del OR gate.

La conexión más directa a la salida final es:

OR1/OUT_D → AND3/IN_D2

Sin embargo, esta conexión no lleva directamente a la salida final. Debemos seguir las conexiones desde AND3/OUT_D .

La conexión más directa a la salida final es:

AND3/OUT_D → OR1/IN_D4

Y luego:

OR1/IN_D4 → 136-ST-2305_P05

Dado que MOLINO_DET/OUT_D está en alto, la entrada OR1/IN_D3 también estará en alto. Como resultado, la salida del OR gate (OR1/OUT_D) también estará en alto.

Finalmente, dado que AND3/OUT_D está conectado a OR1/IN_D4 , y OR1/OUT_D está en alto, la entrada OR1/IN_D4 también estará en alto. Por lo tanto, la salida final 136-ST-2305_P05 también estará en alto.

La respuesta final es: ALTO

Fig. 4.3: Respuesta ante consulta de Lógica y señal de salida

Dada la anterior figura 4.2 la consulta realizada completa es, "Que salida sera si MOLINO_DET esta en alto o 1, y asumimos que el resto de posibles tags con valores estipulados en cero o bajo, considerar la lógica completa y analiza que valores va tomando paso a paso".

La respuesta de los ingenieros y técnicos del área fue positiva, dado que la lógica evaluada cumple con los estándares establecidos en un tiempo récord, lo que permitió generar una respuesta efectiva en un periodo reducido. No obstante, el requerimiento

ideal consiste en obtener distintas conexiones de lógica, considerando que los valores que pueden adoptar las variables dependen de otros controladores o sensores externos. Esto genera comportamientos basados en pulsos periódicos o estados tipo switch ON/OFF, los cuales, dentro del análisis actual, se encuentran limitados a la obtención de posiciones mediante supuestos y a la generación de valores al momento de la descarga de los archivos de los controladores.

4.2.3. Impacto en la Continuidad Operacional (MTTR)

El desempeño del Agente DCS tiene un impacto directo en la continuidad operacional. Al proporcionar un diagnóstico instantáneo, reduciendo drásticamente la fase de incertidumbre técnica durante una detención de planta.

- **Disponibilidad de Activos:** La reducción del tiempo de análisis permite que las decisiones de mantenimiento se tomen en minutos, lo que se traduce en una potencial recuperación de horas de producción en áreas críticas como la Molienda SAG, Correas o Flotación, las cuales tienen una continuidad operativa de las 24 horas.
- **Estandarización del Conocimiento:** Se verifica que ingenieros con menor antigüedad pudieron realizar diagnósticos de nivel experto utilizando el selector de preguntas asistido, cerrando la brecha de experiencia identificada al inicio del proyecto, además de creando una plataforma de auto enseñanza y entendimiento sobre controladores sin actualización dentro de las filosofías de control.

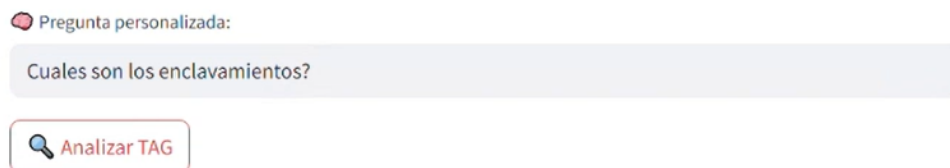


Fig. 4.4: Pregunta ante Enclavamiento

Enclavamientos detectados

Enclavamientos de proceso:

- Estado funcionando (RUN) → RUN
 - Este enclavamiento se activa cuando hay una desviación del modo normal del sistema. Puede ser un problema con el estado actual del sistema, lo que puede provocar una detención.

Enclavamientos de seguridad

- Pullcord → PULL
 - Se activa cuando se detecta un peligro para las personas o equipos.
- Desalineamiento → DESALINEAMIENTO
 - Este enclavamiento se activa cuando hay un problema con la alineación del sistema, lo que puede provocar una detención y garantizar la seguridad de los trabajadores.
- Vibración → VIBRACIÓN
 - Se activa cuando se detectan vibraciones anormales en el sistema, lo que puede ser un signo de problemas mecánicos o de desgaste excesivo.
- Temperatura → TEMPERATURA
 - Este enclavamiento se activa cuando la temperatura del sistema supera los límites establecidos, lo que puede provocar daños a las máquinas y equipos.

Fig. 4.5: Respuesta ante Enclavamiento

La respuesta anterior fue mejor recibida por el área de ingenieros senior, debido a que presenta información estándar necesaria y de rápida visualización, junto con un análisis detallado de los tags y una explicación acorde al nivel ingenieril requerido en los procesos de automatización y control de la planta.

4.2.4. Reducción de Tiempos de Diagnóstico y Análisis

El impacto más disruptivo del Agente DCS se observa en la contracción de los tiempos de latencia técnica. Mientras que el método tradicional depende de la capacidad de rastreo visual y memoria del ingeniero, el sistema basado en IA y grafos realiza búsquedas indexadas y correlacionales de forma instantánea.

La Tabla 4.1 presenta la comparativa final de desempeño, evidenciando una optimización que permite escalar de una respuesta reactiva a una proactiva.

Análisis de los Resultados:

- **Optimización del Ciclo de Ingeniería Inversa:** La interpretación de la lógica final, la tarea más compleja y que mayor tiempo consume (entre 20 a 30 horas manuales), experimentó una mejora del **88 %**. Pasar a un tiempo de ejecución de solo **2 horas** permite que el personal experto deje de ser un "buscador de datos" para convertirse en un "analista de soluciones", delegando la reconstrucción lógica al Agente y enfocándose en la toma de decisiones críticas para la planta.
- **Impacto Global en la Continuidad Operacional:** El tiempo total acumulado para un diagnóstico integral se redujo de **~35 horas a tan solo ~2 horas**. Este ahorro global del **90 %** impacta directamente en el MTTR (*Mean Time To Repair*) de CMDIC. En términos prácticos, una falla que antes mantenía un lazo de control en incertidumbre durante varios turnos de trabajo, ahora puede ser comprendida y direccionada en una fracción de un solo turno laboral.

4.3. Análisis de Valor Agregado para CMDIC: Impacto Económico y Operacional

El valor agregado del Agente DCS no se limita exclusivamente a la eficiencia técnica, sino que impacta directamente en la estructura de costos fijos y variables de la Gerencia de Mantenimiento de CMDIC. En este apartado se presenta un análisis comparativo de costos basado en la movilización de servicios de consultoría especializada frente a la operación asistida por la solución propuesta.

4.3.1. Determinación del Valor de la Hora-Humanas (HH) y Respaldo de Mercado

Para el cálculo de costos, se establece una tarifa de consultoría especializada para un **Ingeniero Senior de Control y Automatización** con experiencia en sistemas de control distribuido. Bajo estándares del mercado minero chileno proyectados al 2026, se define una tarifa de **2.7 UF por hora efectiva**, este valor nace de un promedio entre

2.2UF a 3.5 UF asociadas al cobro de un Ingeniero por la prestación de servicios en el ambiente minero.

Para garantizar la validez de este análisis, se tabularon las referencias de mercado utilizadas para el cálculo de la Hora-Humanas (HH), basadas en informes de remuneraciones y tarifas de consultoría técnica en Chile (Tabla 4.2).

Tab. 4.2: Benchmark de Referencia para Costos de Ingeniería Senior en Minería.

Fuente de Información	Perfil / Concepto	Valor de Referencia
Michael Page (Estudio Salarios)	Ingeniero Senior Proyectos/Control	\$5.5M - \$7.5M CLP (Bruto)
Mifuturo.cl (MINEDUC)	Ing. Civil Automatización (9no Decil)	Rango Superior Nacional
Tarifa de Consultoría Externa	Valor Hora Efectiva (Contratista)	2.5 - 3.2 UF / Hora
Valor Adoptado en Analisis	Consultor Senior DCS	2.7 UF / Hora

4.3.2. Comparativa de Costos por Análisis Integral (Caso Crítico)

Se evalúa el costo de realizar una auditoría de lógica completa para un controlador de Molienda SAG, tarea que demanda el entendimiento total de la filosofía de control y la generación de un documento de recomendaciones. Se comparan las 35 horas de esfuerzo manual promedio contra las 2 horas requeridas mediante el uso del Agente DCS (Tabla 4.3).

Tab. 4.3: Comparativa Económica de Diagnóstico: Manual vs. Agente DCS.

Ítem	Método Tradicional (Manual)	Agente DCS + Ingeniero
Tiempo Requerido (Promedio)	35 horas	2 horas
Tarifa Hora-Hombre (UF)	2.7 UF	2.7 UF
Costo Total en UF	94.5 UF	5.4 UF
Costo Total en USD	3,780.00 USD	216.00 USD
Ahorro Directo por Evento	—	3,564.00 USD

4.3.3. Proyección de Impacto en el OPEX Anual

Considerando que en una operación de la escala de CMDIC se realizan, en promedio, 4 intervenciones de alta complejidad al mes (un evento por semana), la proyección de ahorro en el gasto operacional, es decir el OPEX, se visualiza de la siguiente manera:

- **Ahorro Mensual:** $3,564 \text{ USD} \times 4 = 14,256 \text{ USD}$.

- **Ahorro Anual Proyectado:** $14,256 \text{ USD} \times 12 = 171,072 \text{ USD}$.

4.3.4. *Análisis de Costo de Oportunidad y MTTR*

Más allá del ahorro en facturación de contratistas, el valor más crítico reside en el **Costo de Oportunidad**. En minería, el costo de tener una planta detenida puede ascender a decenas de miles de dólares por hora.

1. **Reducción del Lucro Cesante:** Mediante la disminución del tiempo de interpretación de 35 horas a solo 2 horas, el Agente DCS facilita el restablecimiento de la operación óptima aproximadamente **30 horas antes aproximadamente** que el método tradicional. Es fundamental precisar que este ahorro de tiempo no implica necesariamente que la planta se encuentre en una detención total; sin embargo, la persistencia de fallas en un controlador degrada el rendimiento de sectores específicos, provocando discontinuidades operativas o funcionamientos parciales. Estas ineficiencias impactan severamente en la estructura de costos y reducen los márgenes de beneficio esperados. Bajo un enfoque conservador, esta reducción en los tiempos de respuesta representa una recuperación masiva en la continuidad de la producción ante cada evento crítico. Un ejemplo de los ahorros en costos corresponde a la evaluación ante una eventual detención provocada por un pilar de enclavamiento de seguridad de la correa transportadora, incluyendo el descubrimiento del TAG y el reset del sistema. Actualmente, esta situación genera una detención considerable, la cual puede extenderse entre una y dos horas completas.

Para efectos de evaluar el impacto económico, se considera el supuesto crítico, aunque poco probable, en el cual la detención de la correa implica la paralización total de la producción durante dos horas. Bajo este escenario, se estiman pérdidas aproximadas de 127 toneladas de cobre fino, calculadas en función de la producción anual de Collahuasi. Considerando un precio de 6 dólares por libra, ello representa un estimado de 1,6 millones de dólares en pérdidas asocia-

das a la detención. Este cálculo contempla el proceso completo, incluyendo la comprensión de la falla, el forzado para reiniciar la operación en condiciones de seguridad, el restablecimiento del enclavamiento inicial y el retorno al proceso productivo.

Por otra parte, al considerar el uso del Agente DCS, el entendimiento del evento, la asistencia en la detección de TAGs y su ubicación dentro del programa permiten reducir el tiempo de detención a aproximadamente 20 minutos. Bajo los mismos supuestos de paralización total durante el tiempo estipulado, las pérdidas se estiman en aproximadamente 270 mil dólares brutos.

En comparación con el caso anterior, ambos escenarios cumplen el supuesto de detención total dentro del periodo considerado; sin embargo, la mejora frente a esta eventualidad corresponde a una reducción del 83 % en las pérdidas brutas, lo que representa una mejora comercial significativa, considerando su bajo costo de mantenimiento y su alta capacidad de adaptabilidad al entendimiento de los procesos mineros.

2. **Documentación Instantánea:** A diferencia del proceso manual, el Agente DCS genera el resumen técnico de forma automática, eliminando el tiempo de redacción de informes y permitiendo la transferencia inmediata de información al equipo de ejecución.
3. **Eliminación de la Dependencia Externa:** El conocimiento de la lógica de control se transforma en un activo digital disponible 24/7 dentro de los servidores locales, eliminando la necesidad de movilizar consultores externos para diagnósticos de emergencia.

4.3.5. Justificación del Retorno de Inversión (ROI)

Dado que el desarrollo del sistema utiliza modelos de código abierto y hardware preexistente en la red industrial, el costo de implementación inicial es marginal en com-

paración con los beneficios. Con un ahorro anual proyectado superior a los **170,000 USD**, el proyecto alcanza su punto de equilibrio en menos de dos meses de operación, validando su rentabilidad económica absoluta para la compañía.

4.4. Oportunidades de Mejora y Trabajo Futuro

A partir de los resultados obtenidos y la validación en entorno real, se identifican ejes estratégicos para el escalamiento del Agente DCS. Estas oportunidades buscan transformar el prototipo funcional actual en una plataforma de inteligencia operativa integral a un nivel corporativo para CMDIC.

4.4.1. Escalamiento de Recursos Computacionales y Capacidad de Cómputo

Uno de los pilares fundamentales para la evolución del proyecto es la implementación de una infraestructura de hardware dedicada (Granja de GPUs) dentro de los servidores privados de la compañía, este objetivo es logable a corto plazo.

- **Aumento de la Ventana de Contexto:** La disponibilidad de mayor VRAM permitiría procesar una mayor cantidad de tokens de forma simultánea, facilitando consultas sobre lógicas extremadamente complejas que involucren múltiples controladores interconectados sin fragmentar la información.
- **Reducción de Latencia:** El uso de hardware de alto rendimiento permitiría alcanzar tiempos de respuesta cercanos a la inmediatez, asimilando la experiencia de usuario a los estándares de modelos comerciales, pero manteniendo la soberanía de los datos.
- **Soporte Multiusuario:** Una infraestructura robusta permitiría la integración de múltiples consultas concurrentes desde distintas consolas de ingeniería en Faena Cordillera y Puerto, convirtiendo al agente en una herramienta dinámica para toda la organización.

4.4.2. Evolución hacia Modelos de Lenguaje de Gran Escala

Actualmente, el sistema opera de forma eficiente con modelos de 8 mil millones de parámetros (Llama 3.1 8B). Sin embargo, el escalamiento de hardware abre la puerta a la implementación de modelos de mayor envergadura como serían los modelos de 70B o 400B de parámetros.

- **Mejora en el Razonamiento Lógico:** Modelos más densos poseen una capacidad de abstracción superior, lo que permitiría interpretaciones más profundas de filosofías de control implícitas y una detección de errores de configuración más sutiles.
- **Personalización Mediante Fine-Tuning:** Con recursos suficientes, se propone transitar del método *QLoRA* a un *Fine-tuning* completo. Esto permitiría entrenar al modelo con el histórico de reportes de fallas y bitácoras de mantenimiento de Collahuasi, creando un agente con "memoria institucional" única en la industria, finalmente logrando el RAFT deseado en el estado del arte.

4.4.3. Optimización de los Métodos de Entrenamiento y Recuperación (RAFT)

Se identifica como mejora crítica la transición hacia metodologías híbridas como *RAFT*. Este enfoque permitiría que el agente no solo recupere datos de forma determinista, sino que sea entrenado específicamente para ignorar ruido.^{en} los archivos .fhx y citar con precisión absoluta la base de conocimientos, optimizando el uso de recursos computacionales durante la inferencia.

4.4.4. Integración Multi-Marca y Estandarización de Datos

Si bien el proyecto se validó bajo la arquitectura Emerson DeltaV, el objetivo estratégico es la integración de las demás marcas presentes en CMDIC (ABB 800xA, Siemens, etc.).

- **Investigación de Formatos Propietarios:** Se requiere un estudio exhaustivo para realizar la ingeniería inversa de los archivos de exportación de otros fabricantes, superando barreras de encriptación y lenguajes de bajo nivel.
- **Base de Datos Universal de Control:** La meta final es consolidar una base de datos centralizada que albergue la lógica de todos los sistemas de control de la compañía (desde Cordillera hasta Puerto), permitiendo una trazabilidad completa del proceso productivo en una sola interfaz.

4.4.5. *Integración interna a Controladores DCS*

Dentro del alcance final del proyecto, el objetivo con mayor ambición es lograr integrar al Agente DCS en los controladores, para generar un asistente en tiempo real dando apoyo en distintas áreas como la creación de la lógica, detección de errores en lógica que pueden afectar la operación, detención de unión entre controladores, apoyo en tareas automatizadas de ajustes PID, integración en HMI para consultas directas del Usuario, entre muchas mas opciones.

Para poder llegar a este objetivo es necesario contar con el apoyo de los desarrolladores de los controladores, un equipo de ingeniería dedicado al entrenamiento y un servidor cerrado que pueda comunicar internamente lo solicitado o un servidor online con una gran capacidad de ciberseguridad.

4.4.6. *Derechos de Autor, Propiedad Intelectual y Restricciones de Uso*

El presente trabajo de memoria, así como la arquitectura metodológica, los algoritmos desarrollados, el código fuente implementado y la documentación técnica asociada, constituyen una obra original de autoría de Daniel Sebastián Lazo Larraín.

En consecuencia:

- El código fuente desarrollado no se encuentra disponible para distribución pública.

- La arquitectura detallada del sistema, en su implementación operativa, posee carácter confidencial.
- Los archivos de configuración, bases de datos, ejemplos reales y cualquier información derivada de sistemas industriales pertenecen a CMDIC y están protegidos por acuerdos de confidencialidad y políticas internas de ciberseguridad.

Se autoriza únicamente la reproducción parcial del contenido de esta memoria con fines académicos o de citación, siempre que se realice la referencia correspondiente al autor. No se permite la reproducción total del software, su redistribución, ingeniería inversa, ni su utilización con fines comerciales sin autorización expresa y por escrito del autor y de la entidad involucrada.

La presente memoria tiene carácter estrictamente académico y no constituye una liberación pública del software ni de la tecnología implementada

Sin perjuicio de lo anterior, el desarrollo del software descrito en esta memoria se realizó en el contexto de colaboración con la Compañía Minera Doña Inés de Collahuasi (CMDIC), utilizando información técnica interna, configuraciones reales de sistemas de control industrial y archivos propietarios correspondientes a plataformas DCS de uso exclusivo de la empresa.

4.5. Conclusiones

El desarrollo e implementación del Agente DCS representa un hito importante en la integración de Inteligencia Artificial Generativa aplicada a sistemas de control industrial en la gran minería chilena, esto es debido a que no existe precedente de una implementación igual en el mercado. Tras el análisis de resultados, se concluyen los siguientes puntos fundamentales:

1. **Validación de la Hipótesis Técnica:** Se demostró que es factible procesar lenguajes de control no lineales mediante una arquitectura basada en grafos y modelos de lenguaje locales. El sistema logró reducir el tiempo de interpretación

lógica de un promedio de **35 horas a solo 2 horas**, alcanzando una mejora en la eficiencia técnica superior al **90 %**.

2. **Impacto Económico y Operacional:** La solución no solo optimiza el uso de Horas-Hombre, sino que proyecta un ahorro directo en costos de consultoría superior a los **170,000 USD anuales**. Más importante aún es su impacto en el *MTTR* y la reducción del lucro cesante, permitiendo que la planta retome su operación óptima de forma optima y temprana ante fallas críticas.
3. **Soberanía y Seguridad de la Información:** Mediante el uso de modelos *Open-Source* y ejecución 100 % local, se garantizó la protección de la propiedad intelectual de CMDIC, cumpliendo con los más estrictos estándares de ciberseguridad industrial, logrando así no tener filtración de información sensible con otras empresas o directamente con la competencia.
4. **Democratización del Conocimiento:** El Agente DCS actúa como un puente entre la complejidad técnica del código fuente y el personal de mantenimiento. Esto permite reducir la brecha de experiencia entre ingenieros senior y personal en formación, estandarizando la calidad de los diagnósticos en toda la gerencia.
5. **Sinergia Humano-IA:** Es importante destacar que el agente no sustituye el criterio del experto. Su valor reside en actuar como una herramienta de soporte que elimina tareas mecánicas de búsqueda, permitiendo que el personal calificado se enfoque en la toma de decisiones críticas. Como toda herramienta basada en modelos probabilísticos, la verificación por parte de especialistas sigue siendo un pilar de la seguridad operativa.

Finalmente, este proyecto sienta las bases para una nueva era de mantenimiento inteligente en Collahuasi, donde el conocimiento de la lógica de control deja de ser un activo estático y oculto para convertirse en una capacidad dinámica, accesible y de alto valor estratégico para la continuidad operacional de la compañía.

Bibliografía

- [1] N. Jiang, X. Li, S. Wang, et al., “LEDEX: Training LLMs to Better Self-Debug and Explain Code,” en *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS)*, 2024.
- [2] C. D. Betancourt Montellano, C. O. F. Campos Garcia, y R. O. Cruz Leija, “Recognition of handwritten flowcharts using convolutional neural networks,” *International Journal of Computer Applications*, vol. 184, no. 1, pp. 37–41, 2018.
- [3] International Electrotechnical Commission, “IEC 61131-3:2013 Programmable controllers - Part 3: Programming languages,” IEC, Estándar Internacional, 2013.
- [4] T. Zhang, et al., “RAFT: Adapting Language Model to Domain Specific RAG,” *arXiv preprint arXiv:2403.10131*, 2024.
- [5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, y I. Polosukhin, “Attention is all you need,” en *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.
- [6] International Society of Automation (ISA), “Instrumentation Symbols and Identification,” *Standard ANSI/ISA-5.1-2009*, Research Triangle Park, NC, USA, 2009.