



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA**

**FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO
TÍTULO DEL TRABAJO DE TITULACIÓN**

DESARROLLO E IMPLEMENTACIÓN DE UNA RED INDUSTRIAL MULTI
PROTOCOLO PARA EL MONITOREO Y REGISTRO REMOTO DE VARIABLES
PARA LA DETECCIÓN DE FALLAS DE UN MOTOR A COMBUSTIÓN DE UN
REMOLCADOR MARÍTIMO

AUTOR

Salazar Espinoza, Christian Fernando

TRABAJO DE TITULACIÓN

Previo a la obtención del grado académico en
MAGISTER EN ELECTRÓNICA Y AUTOMATIZACIÓN

TUTOR

. Chuquimarca Jiménez, Luis

Santa Elena, Ecuador

Año 2024



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO**

TRIBUNAL DE SUSTENTACIÓN

**Ing. Alicia Andrade Vera, Mgtr.
COORDINADORA DEL
PROGRAMA**

**Ing. Luis Chuquimarca Jiménez, Mgtr.
TUTOR**

**Ing. Samuel Bustos Gaibor, Mgtr.
DOCENTE
ESPECIALISTA**

**Ing. José Sánchez Aquino, Mgtr.
DOCENTE
ESPECIALISTA**

**Abg. María Rivera, MSc.
SECRETARIO GENERAL
UPSE**



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO**

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por SALAZAR ESPINOZA CHRISTIAN FERNANDO, como requerimiento para la obtención del título de Magister en Electrónica y Automatización.

TUTOR



firmado electrónicamente por:
**LUIS ENRIQUE
CHUQUIMARCA JIMENEZ**

Ing. Luis Chuquimarca Jiménez Msc.

15 de junio de 2024



UPSE

**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO**

DECLARACIÓN DE RESPONSABILIDAD

Yo, **SALAZAR ESPINOZA CHRISTIAN FERNANDO**

DECLARO QUE:

El trabajo de Titulación, Desarrollo e implementación de una red industrial multi protocolo para el monitoreo y registro remoto de variables para la detección de fallas de un motor a combustión de un remolcador marítimo; previo a la obtención del título en Magister en Electrónica y Automatización, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Santa Elena, 15 de junio de 2024

EL AUTOR



Plinado electrónicamente por:
**CHRISTIAN FERNANDO
SALAZAR ESPINOZA**

Ing. Christian Fernando Salazar Espinoza



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE CIENCIAS DE LA INGENIERÍA
INSTITUTO DE POSTGRADO**

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado (Desarrollo e implementación de una red industrial multi protocolo para el monitoreo y registro remoto de variables para la detección de fallas de un motor a combustión de un remolcador marítimo), presentado por el estudiante, SALAZAR ESPINOZA CHRISTIAN FERNANDO fue enviado al Sistema Anti plagio COMPILATIO, presentando un porcentaje de similitud correspondiente al 02%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.

 CERTIFICADO DE ANÁLISIS magister		
CS_TesisVersion03		
2% Textos sospechosos		< 1% Similitudes 0% similitudes entre comillas < 1% entre las fuentes mencionadas 1% Idiomas no reconocidos
Nombre del documento: CS_TesisVersion03.docx ID del documento: 4204c0b3fcd455dbdab679bacce9a4cda903f266 Tamaño del documento original: 6.06 MB	Depositante: LUIS ENRIQUE CHUQUIMARCA JIMENEZ Fecha de depósito: 31/5/2024 Tipo de carga: interface fecha de fin de análisis: 31/5/2024	Número de palabras: 15.991 Número de caracteres: 106.186

TUTOR



Firmado electrónicamente por:
**LUIS ENRIQUE
CHUQUIMARCA JIMENEZ**

Ing. Luis Chuquimarca Jiménez Msc.



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO**

AUTORIZACIÓN

Yo, **CHRISTIAN FERNANDO SALAZAR ESPINOZA**

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales de artículo profesional de alto nivel con fines de difusión pública, además apruebo la reproducción de este artículo académico dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, 15 de junio 2024

EL AUTOR



CHRISTIAN FERNANDO
SALAZAR ESPINOZA

Ing. Christian Fernando Salazar Espinoza

AGRADECIMIENTO

Quisiera expresar mi más profundo agradecimiento a todas las personas que han sido parte fundamental de este importante viaje académico. En primer lugar, mi gratitud a los excepcionales profesores que me acompañaron durante mi formación; su dedicación, paciencia y conocimientos no solo me formaron profesionalmente, sino que también inspiraron mi pasión por aprender y crecer. A mis compañeros, gracias por compartir tantos momentos de desafíos y descubrimientos, por la amistad y el apoyo constante que hicieron de este camino una experiencia inolvidable. No puedo dejar de agradecer a mi familia, cuyo amor incondicional y aliento constante han sido mi roca y refugio en los momentos más desafiantes. Finalmente, mi reconocimiento a la Universidad, por brindarme las herramientas y oportunidades para construir un futuro prometedor y por ser un espacio donde pude desarrollar mis habilidades y construir sueños. A todos, gracias de corazón por ser parte esencial de mi formación y por contribuir a este significativo logro en mi vida.

Christian Fernando, Salazar Espinoza

DEDICATORIA

Este logro está dedicado con todo mi amor a mi madre Lourdes Espinoza, quien ha sido mi fuente inagotable de inspiración y apoyo incondicional tanto en los momentos buenos como en los difíciles. Mamá, tu fortaleza, sabiduría y cariño han iluminado cada paso que he dado y cada desafío que he superado. Eres mi ejemplo a seguir y mi constante recordatorio de que con amor y perseverancia todo es posible.

También quiero dedicar este triunfo a mis hermanos, quienes no solo han sido mis compañeros de vida, sino también mi soporte y alegría diaria. Su presencia constante y su apoyo incondicional han sido esenciales para mí. Juntos hemos compartido más que una vida, hemos construido un hogar de apoyo mutuo y comprensión.

A mi madre y a mis hermanos, mi corazón les pertenece. Gracias por ser parte de cada momento, cada desafío y cada éxito. Este logro es tan mío como suyo.

Christian Fernando, Salazar Espinoza

ÍNDICE GENERAL

TITULO DEL TRABAJO DE TITULACIÓN	I
TRIBUNAL DE SUSTENTACIÓN	II
CERTIFICACIÓN.....	III
DECLARACIÓN DE RESPONSABILIDAD	IV
DECLARO QUE:	IV
CERTIFICACIÓN DE ANTIPLAGIO	V
AUTORIZACIÓN	VI
AGRADECIMIENTO	VII
DEDICATORIA.....	VIII
ÍNDICE GENERAL	IX
ÍNDICE DE TABLAS	XIII
ÍNDICE DE FIGURAS	XIV
RESUMEN	XVI
ABSTRACT.....	XVII
INTRODUCCIÓN	2
Formulación del problema de investigación	3
Objetivo General:	3
Objetivos Específicos:.....	3
Planteamiento hipotético	4
CAPÍTULO 1. MARCO TEÓRICO REFERENCIAL.....	5
1.1. Revisión de literatura	5
1.2. Desarrollo teórico y conceptual	7
1.2.1. Norma IEC 61162.....	7
1.2.2. Norma IEC 61158.....	8

1.2.3. Estándar ISA101	9
1.2.4. Red Industrial.....	9
1.2.5. Multiprotocolo	9
1.2.6. Gateway	10
1.2.7. Modbus	10
1.2.8. CAN.....	11
1.2.9. MQTT	11
1.2.10. Node Red	11
1.2.11. IIOT	12
1.2.12. Motor de combustión interna a Diesel	12
1.2.13. Sistema Embebido	13
CAPÍTULO 2. METODOLOGÍA.....	14
2.1. Contexto de la investigación	14
2.2. Diseño y alcance de la investigación propuesta.....	15
2.2.1. Tipo de embarcación.....	15
2.2.2. Diagrama general	16
2.2.3. Motor CAT 3508C.....	16
2.2.4. Diagrama de Comunicación del Sistema	17
2.3. Componentes de la propuesta.....	18
2.3.1. Hardware.....	18
2.3.2. Software	27
2.4. Características de la red	34
2.4.1. Topología de la Red.....	34
2.4.2. Protocolos y seguridades	36
2.4.3. Velocidades.....	38

2.4.4. Cantidad de datos.....	39
2.5. Métodos y desarrollo de la investigación.....	40
2.5.1. Enlace CANBUS J1939 Computadora del motor y DCU 410E.....	40
2.5.2. Enlace Modbus TCP/IP entre RP 410E y DCU 410E	41
2.5.3. Enlace de DCU 410E y Servidor de datos	42
2.5.4. Enlace de datos Mosquito Broker -InfluDB	43
2.5.5. Consulta y presentación de los datos en Grafana.....	46
2.5.6. Implementación de reglas en Grafana para Alarmas	47
CAPÍTULO 3. RESULTADOS Y DISCUSIÓN	50
3.1. Pruebas de respuestas del Gateway al servidor de datos.....	50
3.1.1. Media	50
3.1.2. Mediana	50
3.1.3. Desviación estándar	51
3.1.4. Mínimo.....	51
3.1.5. Máximo.....	51
3.2. Interface HMI remota (RP 410E) en el puente del capitán	52
3.3. Prueba de reglas implementadas en Grafana	52
3.4. Historial del registro de los parámetros del sistema.....	53
CONCLUSIONES	55
RECOMENDACIONES	57
REFERENCIAS.....	58
ANEXOS.....	62
ANEXO 1	62
ANEXO 2:.....	64
ANEXO 3	65
ANEXO 4.....	66

ANEXO 5.....	67
ANEXO 6.....	68
ANEXO 7.....	69

ÍNDICE DE TABLAS

Tabla 1 Características principales del Switch Ethernet EDS-205	19
Tabla 2 Características principales del Panel 410E	23
Tabla 3 Especificaciones del bus CAN.....	25
Tabla 4 Especificaciones de Ethernet	27
Tabla 5 Interpretación del trabajo de Docker	29
Tabla 6 Datos esenciales para la supervisión y monitoreo del motor.....	39
Tabla 7 Parámetros a considerar en las tramas de comunicación.....	43
Tabla 8 Datos de muestreo del tiempo de respuesta de la solicitud Modbus	65

ÍNDICE DE FIGURAS

Figura 1 Remolcador marítimo Camila III	15
Figura 2 Diagrama general	16
Figura 3 Motor CAT 3508C	17
Figura 4 Diagrama de comunicación del sistema	18
Figura 5 Switch Ethernet EDS - 205	19
Figura 6 Panel DCU 410E	20
Figura 7 Páginas de instrumentos, confirmación, alarmas y configuración	20
Figura 8 Módulo SDU 410, Safety Shutdown Unit.....	21
Figura 9 <i>Panel RP 410E HMI</i>	22
Figura 10 <i>Conexión CANBUS</i>	24
Figura 11 <i>Conexión Ethernet</i>	26
Figura 12 <i>Interfaz de presentación de Oracle VM Virtual Box ejecutando Linux Mint</i> 28	
Figura 13 <i>Conectividad del broker MQTT</i>	29
Figura 14 <i>Flujo de trabajo de NodeRed</i>	31
Figura 15 <i>Panel principal desarrollado en Grafana</i>	32
Figura 16 <i>Menú principal del servidor web</i>	33
Figura 17 <i>Topología de red</i>	35
Figura 18 <i>Comunicación de datos para el monitoreo</i>	36
Figura 19 <i>Puerto COM 4 en CAN J1939</i>	37
Figura 20 <i>Puerto COM 6 en Modbus TCP Ethernet</i>	38
Figura 21 <i>Configuración específica del nodo</i>	41
Figura 22 <i>Menú de administración</i>	42

Figura 23 <i>Configuración de MQTT en el nodo</i>	44
Figura 24 <i>Obtención de la carga útil de cada parámetro</i>	44
Figura 25 <i>Creación y configuración de la base de datos en InfluxDB</i>	45
Figura 26 <i>Configuración del nodo de InfluxDB</i>	45
Figura 27 <i>Filas creadas para el Dashboard</i>	46
Figura 28 <i>Gauge para la temperatura del refrigerante</i>	47
Figura 29 <i>Adición de la alerta de pérdida de datos</i>	47
Figura 30 <i>Regla de alerta “Sobrecarga”</i>	48
Figura 31 <i>Condición de alerta</i>	48
Figura 32 <i>Configuración de la condición de la regla de alerta</i>	49
Figura 33 <i>Frecuencia del muestreo.</i>	50
Figura 34 <i>Diagrama de caja de los tiempos de respuesta</i>	51
Figura 35 <i>Ubicación del RP 410E en consola en el puente del capitán</i>	52
Figura 36 <i>Historial de alertas</i>	53
Figura 37 <i>Evidencia del Dashboard web en Grafana</i>	53

RESUMEN

En este proyecto se desarrolló e implementó una red industrial multiprotocolo para optimizar la detección de fallas y el mantenimiento preventivo en motores de remolcadores marítimos. El objetivo central fue la implementación de un controlador Gateway que permitiera una eficiente transacción de datos desde la computadora del motor hacia la HMI y el servidor de datos. Este proceso se llevó a cabo mediante un enfoque práctico, en acceder y almacenar variables sin manipularlas, en el establecimiento de supervisión adicional para alertas y acumulación de datos para análisis futuros con técnicas avanzadas.

La infraestructura se ajustó a estándares industriales y marítimos, logrando tiempos de respuesta eficaces y sin errores de tiempo agotado significativos, lo que demuestra la robustez del sistema. Los resultados en términos de tiempos de respuesta fueron: media de 287.89 ms, mediana de 74.78 ms, y una desviación estándar de 611.49 ms, con valores mínimos y máximos de 4.75 ms y 4041.05 ms respectivamente.

La implementación se visualizó exitosamente mediante Grafana, mostrando alarmas e historiales de los datos almacenados, destacando su relevancia para el IIOT y la Industria 4.0.

Palabras claves: RED INDUSTRIAL, MULTIPROTOCOLO, GATEWAY, MQTT, IIOT

ABSTRACT

This project involved the development and implementation of a multiprotocol industrial network aimed at optimizing fault detection and preventive maintenance in maritime tugboat engines. The central objective was to implement a Gateway controller for efficient data transaction from the engine computer to the HMI and data server. The methodology adopted was practical, focusing on accessing and storing variables without manipulation, and establishing additional data supervision for alarms and data accumulation for advanced analysis techniques.

The infrastructure adhered to industrial and maritime standards, achieving effective response times without significant timeout errors, demonstrating the system's robustness. The response time results were as follows: mean of 287.89 ms, median of 74.78 ms, and a standard deviation of 611.49 ms, with minimum and maximum values of 4.75 ms and 4041.05 ms, respectively.

The implementation was successfully visualized through Grafana, displaying alarms and data history, highlighting its relevance to IIOT and Industry 4.0. The key terms defining this work are: Industrial Network, Multiprotocol, Gateway, MQTT, and IIOT.

Keywords: INDUSTRIAL NETWORK, MULTIPROTOCOL, GATEWAY, MQTT, IIOT.

INTRODUCCIÓN

En el ámbito marítimo, la eficiencia y confiabilidad de los motores a combustión diésel abordo en remolcadores son críticos para el exitoso desarrollo de operaciones portuarias y de navegación. Las fuentes de propulsión y generación casi siempre están instaladas en la sala de máquinas de la embarcación, por lo cual es importante modernizar el sistema de monitoreo de este tipo de equipos para reducir el riesgo de que la embarcación opere con variables fuera del rango de trabajo normal. La correcta supervisión de las variables en las embarcaciones ayuda a un desempeño eficaz en distintas áreas (Muñoz, 2022).

Se plantea el desarrollo e implementación de una red industrial multiprotocolo para el monitoreo y almacenamiento de variables tales como temperatura de aceite, presión de aceite, temperatura de refrigerante, revoluciones por minuto, presión de combustible, entre otros, de un motor a combustión de una embarcación marítima en Santa Elena, con el objetivo de que con esta información ayudar a reducir el tiempo de detección de fallas tempranas comunes en los motores en su entorno de trabajo.

En una red industrial es crucial el papel que desempeñan los Gateway, que desde un enfoque centralizado lo desempeña un dispositivo que se encarga de intercomunicar a dos o más dispositivos en una red, los cuales manejan protocolos de comunicación diferentes como en este caso leer los datos del motor por CANBUS y dejar disponibles datos asequibles por MODBUS desde cualquier maestro que tenga en cuenta las políticas de funcionamiento de la red.

En el enfoque distribuido de los Gateway se usa una filosofía de traductor de protocolos de comunicación de tipo software, esto es posible en casos donde algún tipo de dispositivo típico de una red industrial como PLC, HMI, servidor, etc tenga la capacidad no solo de realizar su trabajo sino además de ello incorporar rutinas o algoritmos para dar disponibilidad de datos de una subred a otra que usen distintos protocolos de comunicación (Amjad et al., 2021).

Tal es este caso el uso de un servidor de datos que tendrá entre otros, a NODERED ejecutando y brindando comunicación desde MODBUS del Gateway a MQTT en el servidor de almacenamiento de datos Grafana el cual nos ayudará a tener una interfaz amigable de manejo de datos importantes del motor.

La percepción cuantitativa de la información en muchas ocasiones es muy importante al momento de supervisar el comportamiento de un equipo en operación por lo cual la implementación de una HMI moderna en la red facilita esta labor.

La red de comunicación a implementar debe tener la capacidad de administrar toda esta información de una manera más intuitiva y además de tener la opción de obtener datos digitalizados para un posterior análisis en cuestiones de planificación de mantenimientos preventivos.

El IIOT (Industrial Internet of Things) ha llegado para quedarse en este concepto de Industria 4.0 en donde se ha evidenciado que las empresas mejoran su desempeño e impulsan soluciones más sostenibles debido al avance tecnológico en la gestión de recursos. En este punto, la posibilidad de introducir los datos recopilados a la nube donde se puede aprovechar estas herramientas TI que se ha visto grandes evidencias de apoyo al desarrollo y evolución de sectores completos en la industria y no podría ser la excepción en el sector marítimo (Peter et al., 2023).

Este documento consta por tres capítulos: En el capítulo 1 se hace una revisión referencial y conceptual respecto a normativas y trabajos anteriores relacionados al tema desarrollado. En el capítulo 2 se detalla más a detalle el contexto, diseño y alcance de la investigación, así como el tipo y método de la investigación. Finalmente, en el capítulo 3 se aborda los temas de resultados obtenidos de la implementación.

Formulación del problema de investigación

¿Se podrá monitorear y almacenar las variables principales de un motor de propulsión del remolcador Camila 3 en un servidor de datos para poder detectar fallas?

Objetivo General:

Desarrollar e implementar una red industrial multi protocolo para el monitoreo y registro remoto de variables para la detección de fallas de un motor a combustión de un remolcador marítimo.

Objetivos Específicos:

- Implementar un controlador Gateway para la transacción de datos desde la computadora del motor hacia la HMI y servidor de datos.

- Diseñar e implementar un programa de HMI para monitoreo en el puente del capitán.
- Establecer y programar reglas para enviar advertencias a la HMI.
- Implementar un Servidor local para el almacenamiento de las variables relevantes para la operatividad del sistema.

Planteamiento hipotético

Idea a Defender:

"La integración de tecnologías de comunicación avanzadas mediante una red multiprotocolo es esencial para optimizar la detección de fallas y el mantenimiento preventivo en los motores de los remolcadores marítimos."

CAPÍTULO 1. MARCO TEÓRICO REFERENCIAL

1.1. Revisión de literatura

La evolución de las comunicaciones industriales ha visto en un proceso de evolución complejo y dinámico, debido a la variedad de protocolos y tecnologías que han surgido para satisfacer las necesidades específicas de la industria y especialmente por la pugna del poder del dominio en el ámbito comercial de equipos de automatización industrial.

Ethernet es fundamental hoy en día en la comunicación industrial, aunque su uso no siempre es la mejor opción al momento de implementar una solución, especialmente cuando se busca simplicidad y velocidad de desarrollo de soluciones aplicables. Según Wang et al. (2022), en este punto, protocolos como CANBUS, MODBUS y LINBUS han ganado terreno porque se destacan por su relativa facilidad de implementación y capacidad para cubrir una amplia gama de aplicaciones sobre todo a nivel de actuadores e instrumentación. Por otro lado, Ethernet ha evolucionado para ofrecer alternativas más avanzadas y elaboradas como EtherCAT, PROFINET y Modbus TCP/IP. Aunque aprovechan la infraestructura de Ethernet, se debe tomar en cuenta aspectos de las redes comerciales en base a seguridades, su enfoque en la comunicación sea al tiempo real, la determinación de tiempo y la alta velocidad los hacen adecuados para aplicaciones industriales exigentes. Sin embargo, su diversidad y falta de estandarización pueden plantear desafíos en términos de integración entre dispositivos de diferentes fabricantes.

Conforme al estudio de Wang et al. (2022), OPC UA se ha posicionado como un estándar clave para la comunicación entre dispositivos en entornos industriales, pero no el único, ofreciendo una arquitectura robusta, elaborada, segura y escalable de acuerdo a los requerimientos que se van presentando como la empresa crece. Su capacidad para intercomunicar con varios sistemas y su soporte para modelos de datos complejos lo convierten en una opción atractiva para empresas que buscan integrar equipos de varios fabricantes ya que tiene un enfoque a la comunicación independientemente de la arquitectura de donde se ejecuta. Por su parte, MQTT ha ganado mucha popularidad dando cara a las nuevas exigencias de la industria con su enfoque en la comunicación máquina a máquina (M2M) y su capacidad para gestionar grandes volúmenes de datos en entornos distribuidos. Tiene diseño ligero y alta eficiencia en la transmisión de mensajes

que lo hacen ideal para aplicaciones donde se requiere una comunicación ágil y confiable entre dispositivos conectados.

La adopción de protocolos como OPC UA y MQTT no solo ayuda a eliminar los problemas de coexistencia de múltiples protocolos en un mismo sistema, sino que también facilita la integración de subsistemas y la creación de entornos de producción más flexibles orientados a aumentar su eficiencia. Al estandarizar las comunicaciones industriales en este nivel, las empresas pueden aprovechar al máximo las ventajas del enfoque de la Industria 4.0, como la automatización inteligente, la monitorización en tiempo real y la toma de decisiones basada en datos precisos y oportunos.

Según Galloway & Hancke (2013), las redes industriales son esenciales en diversos ámbitos industriales como la fabricación, generación de electricidad, procesamiento de alimentos, transporte y refinación química, incluido el marítimo. Estas redes se distinguen por su arquitectura compleja, donde la tecnología Ethernet presenta mayor participación junto con sus estándares WEB, donde la conexión de instrumentos, controladores, HMIs y DACs se organizan de manera meticulosa. Además, la gravedad de las fallas en estas redes es relativamente mayor que en las redes comerciales debido a su conexión con equipos físicos, pudiendo ocasionar daños graves, pérdida de producción o llegar incluso a presentar riesgos para la vida humana. En cuanto a los requisitos de tiempo real, las redes industriales deben transmitir y procesar datos con tiempos de respuesta muy bajos y un enfoque a la velocidad e integridad de los datos, a diferencia de las redes comerciales que tienen una flexibilidad mayor en este aspecto que tienen un enfoque a la experiencia del usuario. Además, las redes industriales están diseñadas para ser robustas y resistir condiciones ambientales duras como humedad, polvo y vibraciones, aspecto crucial en entornos industriales de varios sectores como el minero o marítimo.

Los Gateways tienen un papel fundamental en las redes industriales al facilitar la comunicación entre diferentes protocolos industriales ya sea la implementación de su filosofía a solo a nivel de servicio o con hardware incluido. El rápido crecimiento de la conectividad de redes aisladas a través de un Gateway capaz de operar como una interfaz entre ellas, se ha vuelto una solución fundamental. Entre las características más importantes que debe cumplir un dispositivo de esta índole es que deben ser compatibles con al menos dos protocolos de comunicación, se han desarrollado de varios tipos con el

fin de integrar sensores, nodos industriales, robots, máquinas CNC, entre otros dispositivos, a Internet o una red privada donde trabaja algún tipo de servidor (Hemmatpour et al., 2017).

La evolución del IoT (Internet de las Cosas) al IIOT (Internet Industrial de las Cosas) ha marcado un gran cambio en la forma en que interactuamos con la tecnología. El IoT ha proporcionado una plataforma amplia para diversas tecnologías al conectar dispositivos y automatizarlos mediante el uso de sensores y actuadores conectados a la red de una u otra forma, lo que ha dado lugar a una red de dispositivos inteligentes capaces de comunicarse entre sí sin la intervención directa humana. Esta evolución ha encontrado aplicación en el sector de la industria, cambiando la manera en que trabajan y optimizando procesos de producción y logística. Junto con esta revolución tecnológica se han presentado desafíos importantes, especialmente en la seguridad y privacidad de datos, así como en la intercomunicación de sistemas. Mientras avanzamos hacia un futuro cada vez más conectado, es importante abordar estos desafíos y desarrollar soluciones efectivas para garantizar un entorno seguro y confiable en el ecosistema del IoT y el IIoT. En un estudio realizado en el 2020 proporciona una visión integral de esta transformación y nos ofrece ideas valiosas sobre las aplicaciones y desafíos que enfrentamos en esta era digital en constante evolución (Himanshu et al., 2020).

1.2. Desarrollo teórico y conceptual

1.2.1. Norma IEC 61162

IEC 61162 es un estándar utilizado en la industria marítima y naval cuyo objetivo principal es garantizar la comunicación de datos entre equipos y sistemas a bordo de las embarcaciones en consecuencia aumentando la compatibilidad de dichos equipos. En él se especifican las características técnicas y funcionales de las interfaces digitales utilizadas en sistemas de navegación, comunicación y control en el ámbito marítimo.

La norma IEC 61162 establece requisitos claros para la transmisión de datos, incluyendo el formato de los mensajes, los protocolos de comunicación y las velocidades de transmisión.

Algunos de los aspectos cubiertos por la norma IEC 61162 incluyen:

- **Definición de protocolos de comunicación:** La norma aclara los protocolos estándar para la transmisión de datos entre equipos, como NMEA 0183 y NMEA 2000, que son ampliamente utilizados en la industria marítima para la comunicación de datos de navegación, posición, velocidad, rumbo, entre otros, así también llega a cubrir especificaciones básicas para la comunicación serial en un nodo de comunicación RTU.
- **Interfaces físicas y eléctricas:** Define las características físicas y eléctricas de las interfaces de comunicación, como los tipos de conectores, niveles de voltaje, velocidades de transmisión y protocolos de señalización, para garantizar una conexión adecuada y fiable entre los equipos.
- **Formato de mensajes:** Establece la estructura de los mensajes de datos transmitidos entre equipos, asegurando la coherencia y la interpretación correcta de la información recibida por los sistemas de navegación, control y comunicación a bordo.
- **Gestión de errores y redundancia:** Incluye mecanismos para la detección y corrección de errores en la transmisión de datos, así como la implementación de medidas de redundancia para garantizar la integridad y fiabilidad de la información en entornos marítimos exigentes.

1.2.2. Norma IEC 61158

Esta norma internacional establece protocolos para interconectar diferentes componentes o dispositivos en una red de automatización, control o monitorización, es decir para la implementación de redes de bus de campo, la norma abarca desde la interconexión en los ambientes más duros por altas temperaturas, humedad o tensiones mecánicas hasta las conexiones estándares para los niveles altos de la pirámide de la automatización con servidores SCADAS, entre las bases de un cable de uso industrial comprende básicamente el material de envoltura que soporte condiciones poco favorables, apantallamiento contra campos electromagnéticos hasta el porcentaje de cobre mínimo que debe tener el conductor interno (Eland Cables, 2024).

1.2.3. Estándar ISA101

Este estándar aborda convenciones y normas a considerarse en el diseño de una HMI para uso industrial, en términos de eficacia y eficiencia que garanticen la operatividad en un entorno de control automático.

Por otro lado, tomar en cuenta las recomendaciones del estándar en detalles como jerarquías de menús, detalles de navegación en la pantalla, convenciones de gráficos y colores, objetos dinámicos, filosofía del sistema de alarmas, seguridades, históricos, pantallas de ayuda, etc, ayudan al usuario u operario a ser más eficaces en temas de rendimiento, calidad, producción y fiabilidad (García, 2015).

1.2.4. Red Industrial

Una red industrial hace referencia a la infraestructura de comunicación diseñada para facilitar la interconexión y la transferencia de datos entre dispositivos, máquinas y sistemas en un entorno industrial. Estas están diseñadas para soportar aplicaciones críticas en tiempo real, como control de procesos, monitoreo de datos, gestión de la producción y la automatización. La arquitectura suele estar optimizada para tener alta disponibilidad y seguridad, usando protocolos y tecnologías que garantizan un rendimiento óptimo en entornos industriales. Estas redes desempeñan un papel fundamental para mejorar la eficiencia de una industria, para tomar de decisiones basada en datos e históricos y en especial para la integración de sistemas en el contexto de la Industria 4.0 (Zurawski, 2014).

1.2.5. Multiprotocolo

Multiprotocolo esencialmente conceptualiza al uso de dos o más protocolos en un sistema comunicado, una red industrial puede estar subdivida en varias secciones de acuerdo a sus funcionalidades o capas en la pirámide del control industrial, por ejemplo en la capa más baja, la de sensores y actuadores pueden usar un protocolo a nivel de campo como CANBUS hacia un PLC, mientras que para entregar información a la siguiente capa puede hacer uso de un protocolo como Profinet (PLC – HMI), el PLC desempeñaría la función un dispositivo de enlace entre 2 subredes (CANBUS y Profinet).

1.2.6. Gateway

Un Gateway, en el contexto de las redes y la integración de sistemas, es un dispositivo o servicio que actúa como un traductor entre dos o más subredes que usan distinto protocolo, facilitando la comunicación entre ellos, esencial en la capa de aplicación en la industria.

Esta filosofía contextualiza la conversión del protocolo del dispositivo emisor al protocolo del dispositivo receptor, permitiendo así que estos dispositivos puedan intercambiar datos de manera efectiva a pesar de las diferencias en sus protocolos de comunicación nativos. El Gateway puede ser implementado de tipo hardware y/o funcionalidad integrada en algunos dispositivos como controladores lógicos programables (PLC) o interfaces hombre-máquina (HMI) que tienen la capacidad de acceder a sus datos con 2 o más protocolos de comunicación (Amjad et al., 2021), actuando como un intermediario adicional para garantizar una integración práctica y eficiente entre los dispositivos que no pueden comunicarse directamente debido a sus limitaciones en los protocolos de comunicación que son capaces de ejecutar.

1.2.7. Modbus

El protocolo Modbus es una especificación de capa de aplicación diseñado por MODICOM para la comunicación cliente/servidor entre dispositivos conectados en diferentes tipos de buses o redes. Desde su estandarización en 1979, Modbus se ha convertido en el estándar en la industria, permitiendo la comunicación de millones de dispositivos de automatización. Su estructura simple sigue siendo ampliamente respaldada, con acceso a través de un puerto reservado en el stack TCP/IP. Modbus opera mediante un enfoque de solicitud/respuesta, ofreciendo servicios definidos por códigos de función que se utilizan en transacciones Modbus. Estos códigos de función forman parte de los PDU (Protocol Data Units) de solicitud/respuesta de Modbus.

El protocolo Modbus se implementa actualmente utilizando TCP/IP sobre Ethernet, transmisión serie asíncrona sobre diversos medios como cableado, fibra o radio. Este enfoque permite una comunicación fácil y efectiva entre diversos dispositivos como PLCs, HMIs, controladores, y dispositivos de E/S en entornos industriales (Modbus-IDA, 2016).

1.2.8. CAN

Controller Area Network (CAN) es un protocolo de comunicación en serie diseñado para aplicaciones en tiempo real en entornos industriales y automotrices. Su estructura de comunicación entre pares permite una buena comunicación de baja latencia, lo que la hace ideal para sistemas que requieren respuestas rápidas y sincronizadas. Además, CAN se utiliza a menudo en aplicaciones críticas en tiempo real porque proporciona funciones como detección de errores, priorización de mensajes y la capacidad de gestionar múltiples nodos en una red. CAN proporciona una red robusta y confiable para transmitir datos entre varios dispositivos, como sensores, actuadores, dispositivos de control y otros componentes electrónicos. La versatilidad de CAN se refleja en su uso en una variedad de industrias, incluidas la automotriz, aeroespacial y sistemas de control y automatización industrial, donde su rendimiento y fiabilidad son fundamentales en entornos industriales.

1.2.9. MQTT

Es un protocolo de mensajería de filosofía publicación/suscripción, es ligero y basado en intermediarios llamado MQ Telemetry Transport (MQTT), está diseñado para ser abierto y fácil de implementar. Ideal para entornos donde la red es costosa, tiene limitaciones de ancho de banda o no es confiable, así como para chips integrados con recursos limitados. El patrón de mensajes de publicación/suscripción utilizado por el protocolo permite la distribución de uno a muchos y el desacoplamiento de aplicaciones, lo que facilita la comunicación efectiva entre dispositivos. Además, es independiente del contenido de la carga útil o payload, por ende, adaptable a una variedad de tipos de datos. Proporciona una conectividad de red básica utilizando TCP/IP y ofrece tres características de servicio para la entrega de mensajes, que se adaptan a las necesidades específicas de cada aplicación. La eficiencia mejorada y la reducción del tráfico de la red son el resultado de su bajo costo de transporte y la minimización de intercambios de protocolos (Eurotech & International Business Machines Corporation [IBM], 2017).

1.2.10. Node Red

Node Red es una herramienta de software que facilita la interconexión de dispositivos de hardware, API y servicios en línea. Su IDE basado en navegador facilita la creación de flujos al ofrecer una amplia gama de nodos en una paleta fácil de entender que se pueden conectar y desplegar en tiempo real con un simple clic y arrastre de objetos. Además,

utiliza un editor de texto para la creación de funciones en el lenguaje JavaScript, lo que permite personalización. Una de las principales ventajas de Node Red es su integración con Node.js, lo que le permite una ejecución rápida y eficiente. Esto lo hace ideal para su implementación en dispositivos de bajo costo como Raspberry Pi o en entornos de nube. En entornos con bajos recursos funciona perfectamente por su modelo basado en eventos. La biblioteca Node Red permite el almacenamiento y reutilización de funciones, plantillas y flujos útiles, lo que disminuye el tiempo de desarrollo y aumenta la productividad del usuario.

Node Red ofrece una amplia gama de nodos que le permiten acceder a más de 225.000 módulos de Node en su repositorio, lo que le permite agregar nuevas capacidades y personalizar aún más sus proyectos de conectividad (Advantech Czech s.r.o., 2022).

1.2.11. IIOT

El Internet Industrial de las Cosas satisface esas necesidades que el IOT convencional no lo hace y está siendo aplicado ampliamente en la industria minera para mejorar la productividad y la seguridad. Investigadores del Instituto Nacional de Seguridad y Salud Ocupacional de EE.UU. se encuentran explorando las aplicaciones de IIoT para identificar desafíos y soluciones en esta industria de alto riesgo. NIOSH ha revisado los sistemas de sensores y comunicaciones existentes en minas subterráneas de carbón en EE.UU. para determinar si son capaces de soportar sistemas de IIoT. Los resultados muestran que alrededor del 40 por ciento de los sistemas de comunicación post accidente instalados hasta 2014 no requieren modificaciones importantes para admitir aplicaciones de IIoT. Además, se ha desarrollado un prototipo de sistema de monitoreo y control de IIoT utilizando placas de microcontrolador WiFi de bajo costo para procesos de control industrial como detectar la apertura de una puerta en un refugio alternativo, activar ventiladores ubicados dentro de la mina y accionar una alarma en la superficie (Baudoin et al., 2021).

1.2.12. Motor de combustión interna a Diesel

El motor de combustión interna, como su nombre lo indica es una máquina rotacional que utiliza la energía proporcionada por el proceso físico de combustión de diésel y alta temperatura por la compresión del aire del pistón en el cilindro para generar movimientos lineales a lo largo del cilindro que a su vez comunica ese movimiento a una biela que

transporta fuerzas tangenciales al eje del motor a través del cigüeñal para hacerlo girar, este proceso de combustión se lo realiza de forma sincronizada para generar la rotación del motor, existen motores de 4 tiempos como la serie 3600 de Caterpillar que realiza el proceso anterior 4 veces para completar una vuelta completa, el diseño de estos motores le permiten funcionar con diésel o fuel oil, son sobrealimentados con turbos y cuentan con un post enfriamiento en el aire de admisión a los cilindros, cuentan con un sistema de control central y alarmas que advierten de anomalías en su funcionamiento (Hernández, 2016, p.23).

1.2.13. Sistema Embebido

Un sistema embebido es una combinación de hardware y software diseñado para realizar una determinada función, normalmente forma parte de un sistema mucho mayor, el hardware normalmente usado en los sistemas embebido suele ser FPGAs, SoC, IPs o ICs y el software un sistema operativo en tiempo real que permite satisfacer necesidades duras o estrictas en el tiempo. Los sistemas embebidos generalmente tienen un alto rendimiento en comparación con los dispositivos de uso general.

CAPÍTULO 2. METODOLOGÍA

2.1. Contexto de la investigación

La red industrial como propósito principal tiene comunicar el puente del capitán y la sala de máquinas en el Remolcador Camila 3, el recorrido que realiza un dato de temperatura por ejemplo, se lee la temperatura con la ayuda de una sonda en la tubería de paso del motor en el frente del motor, la sonda está conectada a la computadora del motor del motor CAT 3508C, este a su vez envía el valor a través del bus CAN hasta el panel de control y monitoreo local, propio de Caterpillar, en este mismo nodo se conecta un HMI - Gateway DCU 410E que pregunta por el valor de la temperatura y otros valores más de interés por medio del uso del bus CAN para almacenarlos en una tabla de memoria a la cual se los puede acceder ya que también está configurado en modo Modbus TCP/IP esclavo o conocido también como modo Servidor el cual está conectado a una red LAN a través de un Switch Ethernet EDS-205, el cual nos va a proveer la capacidad de conectar en red Gateway, una HMI remota en el puente del capitán en modo Maestro Modbus TCP IP o en también conocido como en Modo Cliente y un servidor el cual va a poder acceder a los mismos datos del módulo Gateway mediante la ejecución de modo Modbus TCP/IP master desde NodeRed y enviarlos al servidor de datos Grafana, usando el protocolo de comunicación MQTT.

La HMI en el puente va a tener la capacidad de leer y supervisar los parámetros actuales y relevantes de la máquina de propulsión en cuestión, pero esos datos son solamente instantáneos y de carácter volátil, como es importante hoy en día en el mantenimiento preventivo acceder a un historial del comportamiento de algún parámetro se va a almacenarlos en una pequeña base de datos local, en el servidor antes mencionado vía red LAN.

En la HMI se tiene la capacidad de establecer reglas de advertencia y seguridad para variables del motor que estén fuera del rango normal de operación, así mismo, una vez los datos en el servidor.

2.2. Diseño y alcance de la investigación propuesta

2.2.1. Tipo de embarcación

La implementación de la red multiprotocolo fue concebida en el Remolcador marítimo Camila 3 que se muestra en la Figura 1 que consta de una sala de máquinas donde se encuentran 2 motores de propulsión (Izquierda y Derecha), dos generadores eléctricos a combustión y una bomba a combustión para el sistema contra incendios, como primera instancia se realiza la integración del motor de propulsión de Babor (Izquierdo) a la red para su supervisión en el puente del capitán (Acomodación en la parte superior de la embarcación).

Figura 1

Remolcador marítimo Camila III



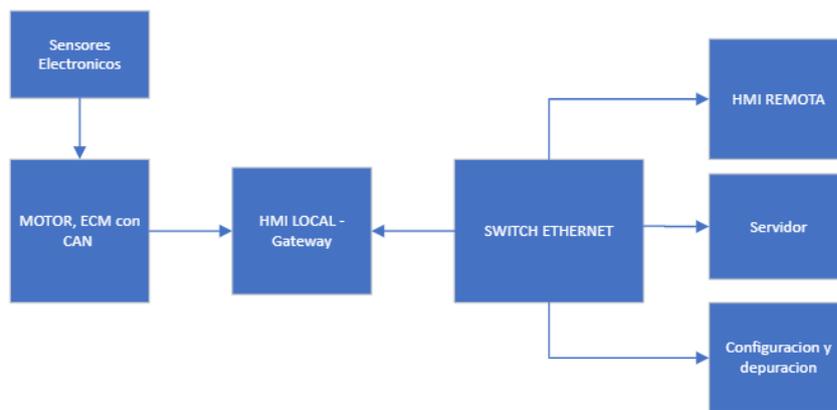
La implementación de las HMI permite la comunicación entre puente y sala de máquinas (en la parte inferior de la embarcación) ya que las maniobras que realiza comúnmente demandan poner a los motores a grandes exigencias y por tal razón, la cuantificación de los parámetros del estado del motor es gran importancia ya que una maniobra, por ejemplo, de remolque puede verse comprometida si alguno de los motores de propulsión presenta una falla inminente, conllevando a problemas portuarios y de la empresa que presta el servicio.

2.2.2. Diagrama general

El diagrama general del proyecto se presenta en la Figura 2, donde los sensores o transductores de temperatura, de presión hidráulica, de pulsos, etc., están dispuestos estratégicamente en el motor por el fabricante, la lectura de estos lo realiza el controlador de la computadora del motor para monitoreo y control, este módulo del motor tiene la capacidad de comunicación con CAN J1939 con la cual se proporciona la lectura de estos sensores por parte del HMI local que a su vez se conecta a una red LAN en modo servidor Modbus TCP IP con la ayuda de un switch Ethernet, a esta red LAN se conecta al puente del capitán donde se implementara la HMI Remota y el servidor, finalmente se puede usar un puerto ethernet para configurar dispositivos que tienen esta opción vía HTML.

Figura 2

Diagrama general

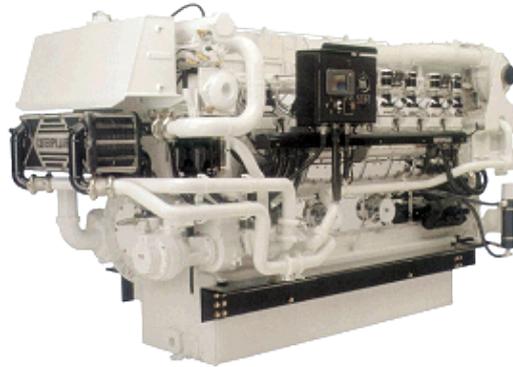


2.2.3. Motor CAT 3508C

El motor CAT 3508C se puede visualizar en la Figura 3, el cual es un motor a diésel de la marca Caterpillar de la serie 3500, entre sus principales características destaca la robustez para aplicaciones en grupos electrógenos, industriales y marinos, su diseño y mejora ofrecen un alto rendimiento en términos de potencia y torque, las potencias que pueden manejar está comprendida en el rango de 1000 a 2000 KW, 8, 12 y 16 cilindros en línea o en configuración V.

Figura 3

Motor CAT 3508C



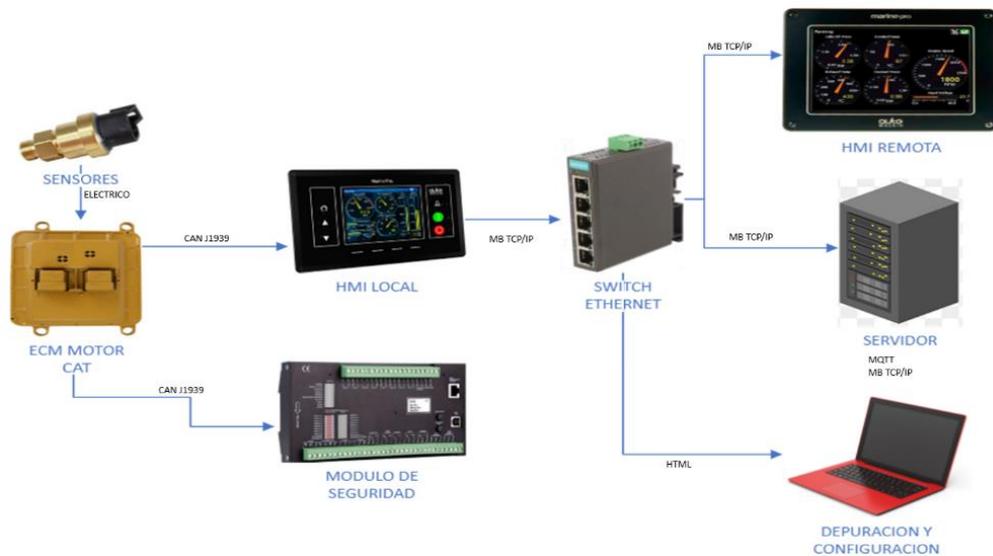
Nota. Tomado de *Motores Marinos 3500C*, por Caterpillar, 2005, <https://boatdiesel.com/News/Caterpillar/N-501/New-3500C-Marine-Engine-Ratings-Give-Caterpillar-the-Edge-in-Displacement-Semi-displacement-and-Planing-Hull-Yacht-Power-Stakes.cfm?NC=1>.

2.2.4. Diagrama de Comunicación del Sistema

En la Figura 4 se puede apreciar la interfaz A, donde se tiene la conexión desde los sensores, las señales eléctricas de tipo industrial (4-20 mA, 0-10V, 24 VDC), hacia la computadora del motor Caterpillar vía cableado directo, la interfaz B se lo hace vía un par de cables trenzados para el nodo CAN (J1939) hacia el HMI local, la cual presenta parámetros y permite control del motor a través de la computadora del motor, la interfaz C obtiene datos redundantes y de seguridad del estado del motor, es decir, tiene acceso a los datos de la computadora del motor y además a través de sus propios sensores instalados adicionalmente al motor, la interfaz D es la conexión Ethernet al switch de corta distancia para conectar el motor a la red LAN, la interfaz E de igual manera es una conexión Ethernet desde sala de máquinas al puente del capitán para leer variables vía Modbus TCP IP, la interfaz E y F conectan un PC para uso como servidor y a un PC para configurar los dispositivos en la red LAN.

Figura 4

Diagrama de comunicación del sistema



2.3. Componentes de la propuesta

Toda red industrial y la mayoría de los sistemas electrónicos están compuestos, como toda computadora, por un hardware y un software que se acoplan el uno con el otro para cumplir una determinada función y responden a demandas cada vez más complejas.

2.3.1. Hardware

El hardware en un sistema es el corazón, aquí comprenden todos los componentes electrónicos, básicamente hacemos referencia a la parte palpable físicamente que compone en este caso la red industrial multiprotocolo.

2.3.1.1. Switch Ethernet EDS-205

El EDS-205, que se muestra en la Figura 5 y cuyas características principales se visualizan en la Tabla 1, es un switch ethernet de 5 puertos, con soporte 10/100M, funcionalidades para full/half-duplex, es de uso industrial, con auto detección de nuevo puerto conectado, negociación automática de velocidad, almacenamiento y envío de paquetes con un buffer de 512 Kbits, trabaja en temperaturas entre -10 y 60 °C, grado de protección IP30, para instalación en riel, alimentación de 24 VDC a 0.11 A y medidas de 24.9 x 100 x 86.5 mm.

Figura 5

Switch Ethernet EDS - 205



Nota. Tomado de *EDS-205 Series*, por Moxa, s. f.,

<https://www.moxa.com/getmedia/e4068be8-1168-4cf8-9437-0d08ca4ac4be/moxa-eds-205-series-datasheet-v1.2.pdf>

Tabla 1

Características principales del Switch Ethernet EDS-205

Característica	Valor
Tamaño de la tabla MAC	1K
Tamaño del Paquete Buffer	384 Kbits
Tipo de Procesamiento	Almacenamiento y envío
Conexión	4 terminales RJ45
Voltaje	24Vdc
Corriente	0,1A
Protección	IP30

2.3.1.2. Panel DCU 410E (HMI local)

El DCU 410E es un dispositivo de control digital versátil, diseñado para la gestión y monitorización de maquinaria pesada, probablemente motores y sistemas relacionados. Este dispositivo admite múltiples fuentes de alimentación y tiene varias entradas y salidas para el control y la monitorización de diferentes aspectos del sistema al que está conectado. En la Figura 6 se puede observar la presentación del dispositivo.

Figura 6

Panel DCU 410E



Nota. Tomado de DCU 410E, por Auto-Maskin AS., 2017b, https://www.auto-maskin.com/wp-content/uploads/2015/04/400E_Series_Installation_Manual.pdf

En la Figura 7, se presenta las diferentes pantallas de la interfaz de usuario donde se dispone de un panel de monitoreo o instrumentación, una ventana emergente de confirmación por seguridad, una pantalla de alarmas y una pantalla de configuraciones generales del DCU 410E.

Figura 7

Páginas de instrumentos, confirmación, alarmas y configuración



Entre las recomendaciones del fabricante son asegurarse de usar cables de calibre adecuado para minimizar la resistencia y la caída de voltaje, configurar adecuadamente las alarmas y los parámetros de seguridad para evitar operaciones incorrectas o daños al equipo y monitorear regularmente el estado de las conexiones y la funcionalidad del DCU 410E para mantener un rendimiento óptimo.

2.3.1.3. Módulo de parada de seguridad SDU 410

El SDU 410, que se muestra en la Figura 8, es un componente esencial en el sistema de seguridad y control de motores. Su función principal es asegurar el funcionamiento seguro de los motores, especialmente cuando se utiliza en conjunto con la Serie Marine Pro 400. Sin embargo, también tiene la capacidad de operar de manera independiente, lo que lo hace versátil en diferentes configuraciones de sistemas de control.

Figura 8

Módulo SDU 410, Safety Shutdown Unit.



Nota. Tomado de *SDU 410*, por Auto-Maskin AS., 2020, <https://www.auto-maskin.com/wp-content/uploads/2021/01/SDU-410-Installation-Manual-Nov-2020.pdf>

Una de las características clave del SDU 410 es su capacidad para monitorear y controlar los interruptores de apagado del motor. Estos interruptores se conectan a los canales de entrada de interruptores en el SDU 410, lo que permite que el sistema detecte y responda rápidamente a cualquier situación de apagado necesario del motor.

El SDU 410 también está equipado con una interfaz de LED que proporciona información detallada sobre el estado del motor. Estos LEDs se dividen en tres grupos: Estado (LEDs verdes), Apagado (LEDs rojos) y Falla (LEDs ámbar). Cada grupo de LEDs indica diferentes condiciones operativas, como la disponibilidad de alimentación, el estado de

funcionamiento del motor, la presencia de fallas o alarmas, entre otros aspectos importantes para la supervisión y el diagnóstico del motor.

Además de sus capacidades de monitoreo y control, el SDU 410 también se destaca por su capacidad de comunicación. Puede establecer conexiones a través de diferentes interfaces, como COM 1 RS-232, COM 2 RS-232, COM 3 Ethernet Modbus/TCP, lo que le permite integrarse fácilmente en redes de control más amplias y sistemas de monitoreo remoto.

En cuanto a las conexiones eléctricas, el SDU 410 requiere una alimentación de 24VDC que se conecta a terminales específicos, junto con la conexión adecuada de los canales de interruptores, bobinas de apagado y otros componentes según las directrices del fabricante. Estas conexiones están diseñadas para garantizar un suministro eléctrico confiable y seguro, fundamental para el correcto funcionamiento del sistema de control del motor en diferentes entornos y aplicaciones marinas.

2.3.1.4. Panel 410E (HMI remoto)

El RP 410E (Unidad de Panel Remoto), que se visualiza en la Figura 9, sirve como interfaz de control y monitoreo para la Unidad de Control de Motor (DCU 410E) en sistemas de control de motores y equipos relacionados.

Figura 9

Panel RP 410E HMI



Nota. Tomado de RP 410E, por Auto-Maskin AS., 2017b, https://www.auto-maskin.com/wp-content/uploads/2015/04/400E_Series_Installation_Manual.pdf

Las características técnicas principales del equipo se evidencian en la Tabla 2, sus funciones principales incluyen monitorear el estado y controlar el funcionamiento de hasta ocho unidades de motor DCU 410E desde una ubicación remota. Proporciona una interfaz a través de la cual los operadores pueden visualizar datos importantes del motor, como la temperatura, la presión, el estado de los interruptores, entre otros, y realizar acciones de control, como encendido/apagado, ajuste de parámetros, etc. Establece comunicación con la DCU 410E a través de Ethernet, lo que facilita la transmisión de datos y comandos entre el RP 410E y la DCU 410E. Tiene entradas para detectar el estado de interruptores y salidas configurables para controlar dispositivos externos, como relés y puede ser integrado en redes de monitoreo más amplias, permitiendo la supervisión centralizada de múltiples motores o equipos desde una ubicación central.

Tabla 2

Características principales del Panel 410E

Sección	Valor	Unidad
Voltaje	24	VDC
Corriente	2	A
Entradas	24	VDC
Entradas Corriente	4-20	mA
PT100	800	°C
Salidas	VCC-0,5	VDC
Salida Crank	1,8	A
Tacómetro Freq	5	KHz
tacómetro Voltaje	32	Vpp
CanBus 1	250	Kbs
CanBus 2	250	Kbs
Ethernet MBs	10/100	Mbs
Modbus RTU	19200- 115200	Baudios

2.3.1.5. Servidor PC

La laptop Dell Inspiron 15 7000 equipada con Windows 11 y configurada con un procesador Intel Core i7-8550U, 24 GB de memoria RAM y discos SSD de 240 GB y 480 GB, ofrece un potente entorno para simular una máquina virtual para usar como de servidor de aplicaciones de Internet de las Cosas Industrial (IIoT). Esta configuración

resulta suficiente para manejar las exigencias de aplicaciones como Mosquitto, Node-RED, InfluxDB, Grafana.

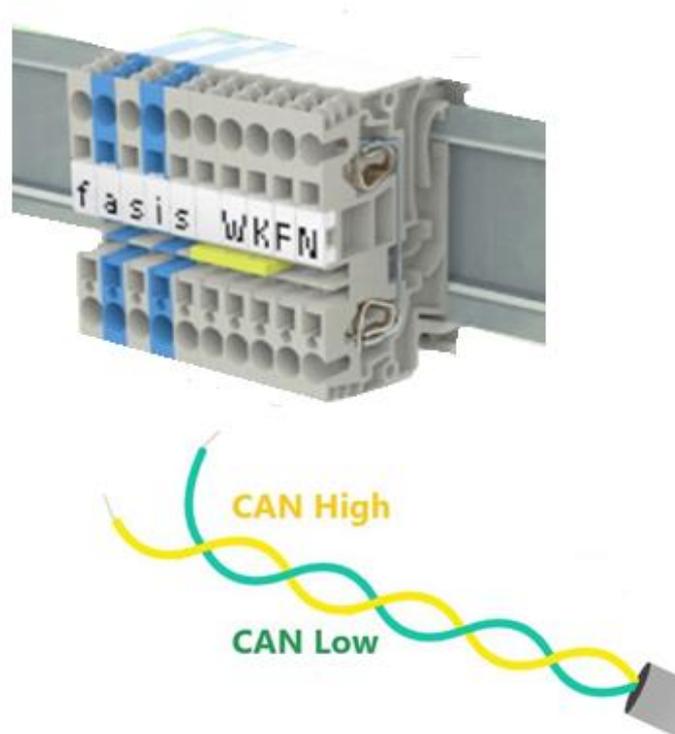
Los discos SSD garantizan tiempos de respuesta rápidos y eficiencia en el acceso a los datos, lo cual es crucial cuando se manejan grandes volúmenes de datos de sensores en tiempo real. La suficiente RAM y la capacidad de procesamiento gráfico no solo permiten un manejo eficiente de múltiples flujos de datos sino también la visualización efectiva de los mismos a través de amigables dashboards en Grafana,

2.3.1.6. Conexiones CANBUS J1939

Las velocidades de transmisión de CANBUS J1939 normalmente son de 250 y 500 Kbps para lo cual la calidad y tipo del cable son un factor importante para la implementación de un nodo de buena calidad en cuanto al calibre y pureza del metal conductor (cobre). El uso de un par trenzado, como se aprecia en la Figura 10, está diseñado para condiciones ambientales adversas, como en este caso, en maquinaria pesada de propulsión marítima.

Figura 10

Conexión CANBUS



El bus CAN usa una topología lineal con resistencias de terminación de 120 Ohms en ambos externos del nodo para evitar reflexiones de datos.

En este caso, se conecta la DCU 410E a través de cable apantallado hacia borneras de tipo riel a donde llegan el par trenzado de la computadora o controlador local del motor dispuesto por el fabricante y disponible para conectar equipos de terceros.

En la Tabla 3 se presenta las especificaciones relevantes que se deben tomar en cuenta para la implementación del bus CAN.

Tabla 3

Especificaciones del bus CAN

Especificación	Detalles
Protocolo	J1939
Capa de red	Protocolo de comunicación de red de área de control (CAN)
Uso	Comunicación en vehículos pesados y maquinaria agrícola
Velocidad de transmisión	250 kbps (predominante), 500 kbps, 1 Mbps (varía según aplicación)
Arquitectura de mensajes	Basada en mensajes PGN (Número de grupo de parámetros)
Tipo de mensajes	PDU (Protocol Data Unit)
ID de mensaje	Identificación mediante J1939 PGN
Conectores típicos	Conector de 9 pines (Deutsch J1939), conector OBD-II (vehículos)
Cableado	Apantallado, generalmente par trenzado
Longitud máxima de cable	40 metros (aproximadamente, dependiendo de la velocidad de transmisión)
Aplicaciones típicas	Motores diésel, sistemas de control y monitoreo en vehículos pesados

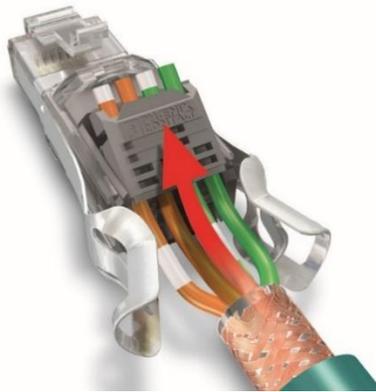
2.3.1.7. Conexiones Ethernet

En las redes industriales Ethernet, la robustez y seguridad de las conexiones juegan un papel muy importante. Ethernet en la automatización ha permitido una comunicación vertical desde el nivel más bajo, al nivel más alto de la pirámide del control automático.

En el entorno marítimo se presenta adversidades como la humedad, variaciones extremas de temperatura, vibraciones, etc. Es importante tener en cuenta algunos aspectos para el conector, el cual se ilustra en la Figura 11, como la robustez, la resistencia a tracciones y fuerzas laterales y protección contra aceites. También es vital conocer las especificaciones técnicas de Ethernet, las cuales se presentan en la Tabla 4.

Figura 11

Conexión Ethernet



Nota. Tomado de *Una conexión rápida in situ a través de la conexión rápida Quickon*, por Andueza, 2009, <https://www.conelectronica.com/ethernet-industrial/ethernet-la-conexion-optima>

El fabricante del módulo DCU 410E recomienda cable estándar CAT-5 para la red LAN, evitar doblar el cable durante la instalación, el cable instalado es un cable comúnmente usado para redes Profinet (CAT-6) de 4 polos y sección del cable 22 AWG diseñado para entornos duros.

Así mismo se consideró el uso de conectores RJ45 de uso industrial para 4 hilos y apantallamiento, el conector no requiere herramientas sofisticadas para la implementación de una conexión, lo que lo vuelve una opción robusta y versátil pal momento de considerarlo en un diseño.

Tabla 4*Especificaciones de Ethernet*

Especificación	Detalles
Protocolo	TCP IP, UDP
Estándares de red	IEEE 802.3 (Ethernet), IEC 61158 (Fieldbus), Profinet
Uso	Comunicación en entornos industriales y comerciales
Velocidad de transmisión	10 Mbps, 100 Mbps, 1 Gbps, 10 Gbps (varía según aplicación)
Medio de transmisión	Cable de cobre apantallado (STP), fibra óptica (POF, MMF, SMF)
Topología de red	Estrella, bus, anillo, malla
Conectores típicos	Conector RJ45, Conector M12, Conector de fibra óptica
Cableado	Par trenzado apantallado (STP), fibra óptica
Longitud máxima de cable	Hasta 100 metros (cobre), varía según tipo de fibra y velocidad
Aplicaciones típicas	Automatización industrial, sistemas de control y monitoreo

Nota: ISO/IEC 11801:2002-09 Tecnología de la Información – cableado genérico para instalaciones de clientes. Ginebra/Suiza: Oficina Central de la Comisión Internacional de Electrotécnica.

2.3.2. Software

En otro contexto tenemos al software en un sistema, que es el cerebro del mismo, este es el que detalla cómo se debe comportar el hardware, aquí comprenden todos los componentes virtuales, como programas, algoritmos, protocolos, etc, básicamente hacemos referencia a la parte no palpable físicamente que compone un entorno computacional.

2.3.2.1. Linux Mint 21.3 Virginia

Mediante el software Oracle VM Virtual Box, cuya interfaz de presentación se indica en la Figura 12, se crea una máquina virtual con Linux Mint 21.3 Virginia.

Figura 12

Interfaz de presentación de Oracle VM Virtual Box ejecutando Linux Mint



Linux Mint es un sistema operativo basado en Ubuntu que destaca por su estabilidad, seguridad y facilidad de uso, cuenta con potentes características heredadas de su ascendencia Debian. Aunque es ampliamente reconocido y preferido para aplicaciones de escritorio por su interfaz intuitiva y la inclusión de software completo, Linux Mint también puede ser configurado como un servidor efectivo para pequeñas o medianas aplicaciones como en este caso. Su compatibilidad con el software de servidor habitual y la disponibilidad de paquetes actualizados lo convierten en una opción viable para ambientes de producción ligeros.

En la implementación de servicios como servidores web, aplicaciones de IoT o ambientes de pruebas, Linux Mint proporciona una base sólida y segura con acceso a un amplio repositorio de software a través de APT y la posibilidad de utilizar contenedores Docker para una gestión eficiente de aplicaciones. Esto lo hace adecuado para un entorno de servidor donde se requiere estabilidad sin sacrificar la facilidad de mantenimiento y actualización.

2.3.2.2. Docker versión 26.1.0, build 9714adc

Docker es una plataforma que permite a los desarrolladores empaquetar aplicaciones junto con todas sus dependencias en contenedores independientes. Esto asegura que el software funcione de manera uniforme y eficiente en cualquier sistema que soporte Docker, muy independientemente de las configuraciones del sistema operativo subyacente o las variaciones en el entorno de desarrollo y producción. Los contenedores Docker son ligeros porque comparten el kernel del sistema operativo anfitrión como

muestra la Tabla 5, pero mantienen procesos aislados en espacios de usuario, lo que permite un despliegue rápido y escalable de aplicaciones. Además, Docker facilita la gestión del ciclo de vida del software mediante el uso de imágenes de contenedores, las cuales se pueden usar para crear contenedores en cualquier momento, garantizando así la reproducibilidad y la minimización de inconsistencias durante el desarrollo y la implementación.

Tabla 5

Interpretación del trabajo de Docker

Contenedor 1	Contenedor 2	Contenedor 3
Java	PHP	
Debian	MySQL	
Tomcat	Ubuntu	Alpine
KERNEL		

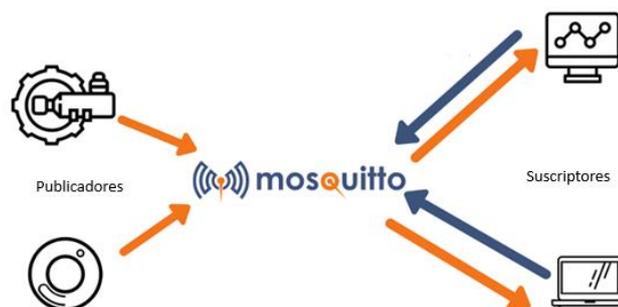
En este caso se usa Docker para la instalación de los softwares de servicio Mosquito como Broker de MQTT, InfluxDB como base de datos, Node Red como software con filosofía Gateway y Grafana para la interfaz y acceso a los datos, con la ayuda de Docker podremos trasladar fácilmente a cualquier computador de uso industrial.

2.3.2.3. Mosquito Broker

Eclipse Mosquitto es un broker MQTT de código abierto desarrollado bajo las licencias EPL/EDL, diseñado para facilitar la comunicación entre dispositivos en redes IoT. Como broker MQTT, Mosquitto actúa como intermediario entre los dispositivos que publican mensajes y aquellos que los reciben, siguiendo el modelo de publicación/suscripción del protocolo MQTT, tal como se presenta en la Figura 13.

Figura 13

Conectividad del broker MQTT



El funcionamiento técnico según Eclipse Mosquitto (2018), del protocolo MQTT implementa las versiones 5.0, 3.1.1 y 3.1, proporcionando una comunicación eficiente y ligera para aplicaciones IoT. Utiliza el modelo de publicación/suscripción, donde los dispositivos publican mensajes en tópicos específicos y otros dispositivos se suscriben a esos tópicos para recibir los mensajes pertinentes.

Además del broker, Eclipse Mosquitto ofrece una biblioteca en C que permite a los desarrolladores implementar clientes MQTT personalizados para sus aplicaciones y es altamente portable, está disponible para una amplia gama de plataformas, lo que permite su despliegue en dispositivos de baja potencia hasta servidores completos. También ofrece opciones de seguridad como MQTT sobre TLS (Transport Layer Security) para encriptar las comunicaciones y garantizar la integridad de los datos en entornos sensibles.

2.3.2.4. InfluxDB

Para la descripción del software InfluxDB, se toma como referencia a Martínez (2022), donde se detalla la definición, las características, los conceptos claves, las ventajas, InfluxDB vs MySQL y los primeros pasos con el software.

InfluxDB es una base de datos de series temporales con la capacidad de almacenar y consultar datos temporales eficientemente, orientada a aplicaciones de monitoreo y análisis de datos en tiempo real, como en el Internet de las Cosas (IoT), diseñado desde cero para manejar datos temporales de manera óptima. InfluxDB utiliza un motor de compresión de datos que permite una alta velocidad de ingesta y compresión, ofrece una API HTTP de escritura y consulta de alto rendimiento. Compatible con varios protocolos de ingestión de datos como Graphite, collectd, y OpenTSDB.

InfluxDB posee un lenguaje de consulta similar a SQL adaptado para consultar datos temporales, permite etiquetar series para consultas rápidas y define políticas de retención para manejar datos obsoletos de manera automática.

InfluxDB tiene integraciones con una amplia gama de herramientas y servicios, incluyendo herramientas de visualización como Grafana, plataformas de monitoreo como Telegraf y Prometheus, servicios en la nube como AWS, Azure y GCP, así como diversos lenguajes de programación como Python, Java, JavaScript, y Go.

En Martínez (2022), también se menciona que InfluxDB admite varios lenguajes de consulta, incluyendo InfluxQL, similar a SQL pero optimizado para datos temporales, además de Flux, que es un lenguaje de scripting diseñado específicamente para consultas, análisis y acciones en datos de series temporales.

2.3.2.5. Programación en Node Red

Node Red es una plataforma de desarrollo visual, de arrastrar y conectar bloques (nodos) y que se centra en simplificar la creación de aplicaciones para el Internet de las Cosas (IoT). Cada nodo representa una función específica, como para procesar datos, tomar decisiones y activar acciones.

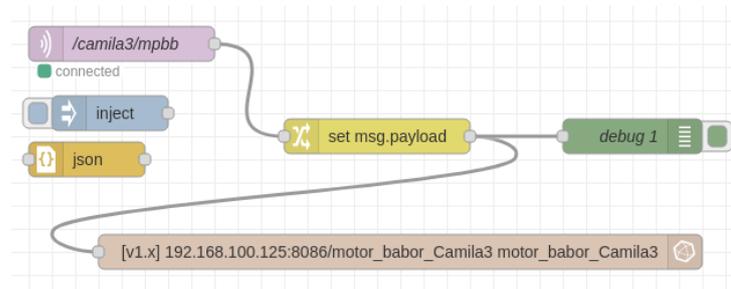
Node Red ofrece una amplia gama de nodos predefinidos, desde trabajo con servicios web y bases de datos hasta el control de dispositivos físicos como sensores y actuadores.

Su modelo de programación se basa en flujos de trabajo donde los bloques representan una función que puede tener gráficamente disponible el ingreso de argumentos, retorno de resultado o ambos, como se puede ver en la

Figura 14, el nodo de color lila tiene salida de resultado, el nodo debug y nodo café admiten gráficamente entrada de datos y el nodo amarillo tiene entrada y salida de datos.

Figura 14

Flujo de trabajo de NodeRed



Node Red se integra fácilmente con APIs de terceros y servicios web, lo que lo hace ideal para la creación de aplicaciones conectadas en IoT y automatización, además permite integrar fácilmente servicios como Twitter, Fitbit, bases de datos, sistemas de mensajería, entre otros, para crear flujos de trabajo complejos.

2.3.2.6. Grafana

Grafana es una plataforma de análisis, tratamiento y visualización de datos de código abierto, convertida en una herramienta muy popular para monitorear métricas en tiempo real y visualizar datos a gran escala. Permite a los usuarios crear tableros dinámicos que ayudan a asimilar grandes volúmenes de datos de diversas fuentes. Grafana proporciona opciones de visualización, como gráficos de líneas, barras, mapas de calor, pasteles, histogramas, indicadores radiales, diagramas geométricos, etc. En la Figura 15, se puede apreciar el panel principal desarrollado en Grafana.

Figura 15

Panel principal desarrollado en Grafana



Las gráficas de configuran y ajustan para mostrar la información que los usuarios necesitan. Puede integrarse con una o varias fuentes de datos a la vez como Prometheus, Elasticsearch, MySQL, PostgreSQL, y en nuestro caso InfluxDB, esto nos permite recopilar y visualizar datos directamente.

Existe la posibilidad de configurar alertas basadas en comparaciones numéricas que nos pueden notificar a través de varios canales como correo electrónico cuando ciertos umbrales son cruzados.

Los tableros de Grafana tienen la posibilidad de ser compartidos entre equipos y organizaciones con el medio adecuado, lo que facilita la colaboración ya que pueden ser vistos y editados por múltiples usuarios.

La seguridad como la autenticación, el control de acceso basado en roles, el uso del protocolo HTTPS para asegurar que los datos y dashboards sean accedidos solo por usuarios autorizados es importante para su empleo en una aplicación.

2.3.2.7. Firmware 1.9 de la serie 400 de Auto Maskin

Se debe habilitar la interface web previo a cualquier procedimiento por este método de configuración, lo cual se puede hacer una vez se inicie sesión y se muestre el menú principal que se puede visualizar en la Figura 16.

Figura 16

Menú principal del servidor web



Nota. Tomado de *Web Server Configuration* (p. 10), por (Auto-Maskin AS., 2023), https://www.auto-maskin.com/wp-content/uploads/2023/11/400E_200E-Configuration-Manual-3.10P5.pdf

Para acceder a la configuración del software con el hardware instalado lo hacemos con la ayuda de la interfaz de comunicación Ethernet en este caso, usando el modo servidor de página web incrustado que tiene el módulo DCU 410E, se lo hacer conectando el DCU 410E a la red con un cable de red directo al Switch dispuesto y energizamos el sistema parcialmente, el DCU 410E se pondrá en red con la IP de fabrica 192.168.0.101 como indica el fabricante.

Conectamos una PC a la misma red y colocamos la IP para que pueda estar en red la DCU 410E y la PC, podemos comprobar haciendo ping a la IP antes mencionada desde la PC para verificar conexión exitosa.

Con la ayuda de un buscador de internet como Mozilla o Chrome introducimos la URL <http://192.168.0.101> e inmediatamente obtendremos como respuesta el panel de seguridad para identificarse, usamos las credenciales de fabrica: DCU y 1234 como nombre de usuario y contraseña, una vez aceptada tendremos acceso a las configuraciones del módulo DCU 410E como alarmas, tiempos, contadores, etc. En la misma interfaz

configuramos el hardware conectado como el tipo de sensores y en que bornera se localiza.

2.3.2.8. Python

De acuerdo a Python TM (2024), Python es un lenguaje de programación interpretado, es decir, su código fuente se convierte en lenguaje máquina a medida que se va ejecutando línea por línea. Además, es un lenguaje orientado a objetos, de alto nivel y con semántica dinámica, por lo que es muy conveniente para el desarrollo rápido de aplicaciones. Python se caracteriza por su sintaxis simple y fácil de aprender, junto a esto también acepta módulos y paquetes externos, fomentando el modularidad del programa y la reutilización del código, lo que contribuye a una mayor productividad para los programadores. Cabe resaltar que el intérprete de Python y su extensa biblioteca estándar son gratuitas

2.3.2.9. JSON (JavaScript Object Notation)

Según JSON (2024), JavaScript Object Notation es un formato ligero para el intercambio de datos. Para el programador es relativamente sencillo leerlo y escribirlo, de igual manera, para las máquinas es simple interpretarlo y generarlo. JSON es totalmente independiente del lenguaje, pero emplea convenciones de los lenguajes de programación populares, como C, C++, Java, Python, entre otros, siendo ideal para el intercambio de datos, esto es posible debido a que las estructuras que utiliza JSON son universales para cualquier lenguaje de programación. Sus formas de presentación son como objeto, arreglo, valor, cadena de caracteres, y número, con la posibilidad de inserción de espacios en blanco entre cualquier par de símbolos.

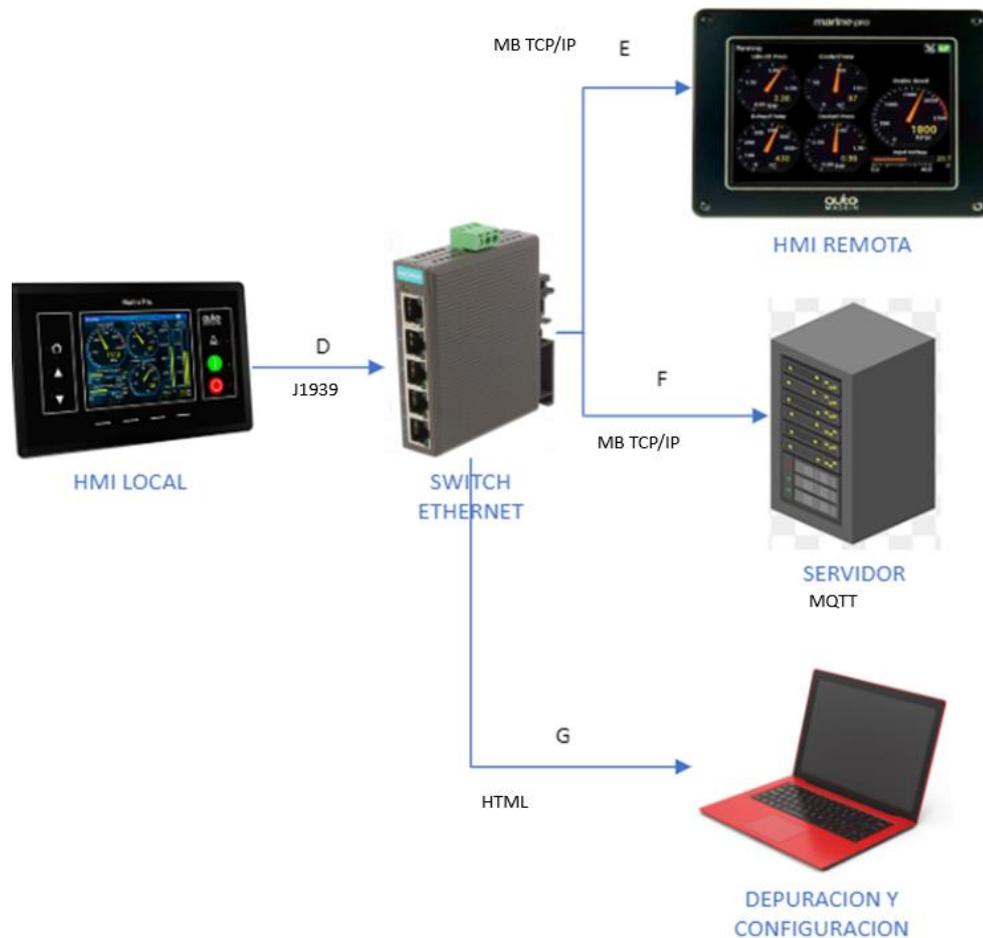
2.4. Características de la red

2.4.1. Topología de la Red

En la Figura 17, podemos observar la disposición de la red LAN, la comunicación entre la DCU 410E (HMI local) y RP 410E (HMI remota) es la más saturada debido a que esta última está destinada a la supervisión del estado del motor, por otro lado tenemos al servidor que accede a datos con baja tasa de actualización de datos y finalmente podemos ver una PC que se usa para configuración y depuración vía Ethernet de los host presentes en la red LAN.

Figura 17

Topología de red

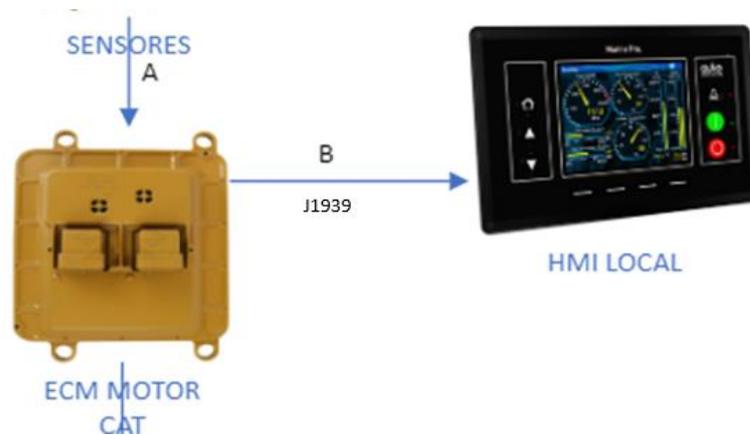


Se tiene una red LAN de tipo estrella a través de un switch, este está situado con el DCU 410E en sala de máquinas y un panel RP 410E en puente del capitán junto a un servidor de almacenamiento y monitoreo de datos a una distancia aproximada de 40 m, contextualmente se tiene la coexistencia de 2 Gateway, el uno se ejecuta en el DCU 410E que comunica los datos obtenidos desde CAN J1939 y los dispone vía Modbus TCP IP hacia el RP 410E y al Servidor de datos, el segundo Gateway se ejecuta a través del servidor con un Script de Python, Mosquitto y Node Red ejecutándose para recibir los datos desde la DCU 410E a través de Modbus TCP IP y enviarlos en una trama tipo JSON a Node Red vía MQTT.

Entre la computadora del motor y el HMI local tenemos una subred CAN J1939 para comunicar los datos necesarios para monitoreo, tal como se representa en la Figura 18.

Figura 18

Comunicación de datos para el monitoreo



2.4.2. Protocolos y seguridades

Los paneles usados en la implementación son certificados y revisados la Asociación Internacional de Sociedades de Clasificación que se encarga de establecer en general estándares para seguridad marítima, entre otros aspectos, quienes han evaluado rigurosamente los procesos de fabricación de los equipos, además de que su diseño y concepción cumple armoniosamente con el estándar IEC 60092-504 que abarca la especificación de equipos eléctricos, electrónicos y programables usados en el área de la automatización, monitoreo y control en embarcaciones.

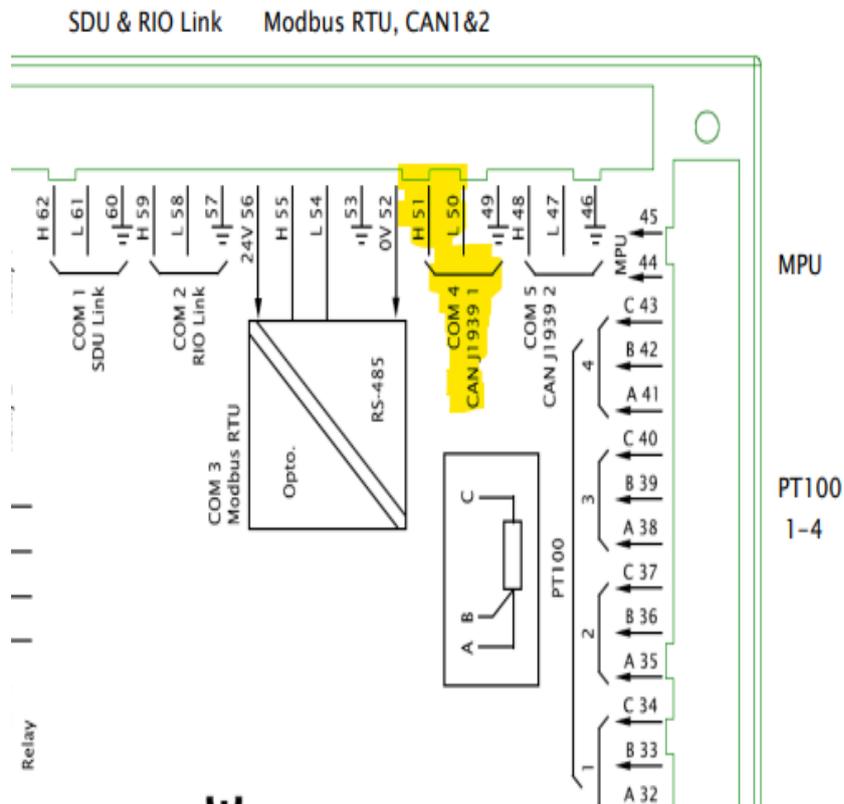
De igual manera el switch de uso industrial implementado está regido por su propio estándar internacional IEC 60950-1 que rigen a los equipos de tecnologías de la información y entre sus características destacables esta la respuesta a los requerimientos de seguridad de los datos.

Por otro lado, tenemos contextos eléctricos y electrónicos a tomar en cuenta para asegurar el correcto funcionamiento del sistema, teniendo así desde la DCU 410E (HMI Local) conexión a 3 hilos del bus CAN para J1939, tal como se observa en la Figura 19, hacia la ECU del motor usando un par de cables trenzado desde el tablero implementado junto al motor al tablero de conexiones dispuesto por el fabricante del motor más una bornera para conectar una conexión a tierra el apantallamiento del cable si se lo instala, esto ayuda a reducir ruido eléctrico inducido en las líneas de comunicación, eléctricamente está diseñado para ambientes con alto ruido electromagnético ya que usa un diferencial de voltaje entre los cables H y L respectivamente lo que nos permite tener integridad en los

datos ya que de existir alguna perturbación en esas líneas, la magnitud de la perturbación va a afectar a todo el par trenzado lo que va a desfazar la señal en amplitud de voltaje mas no va a alterar frecuencia de transmisión por ruido eléctrico lo que nos brinda alta confianza en el uso del protocolo, además de la disponibilidad de blindaje conectado a tierra ratifica más la estabilidad del nodo de datos.

Figura 19

Puerto COM 4 en CAN J1939

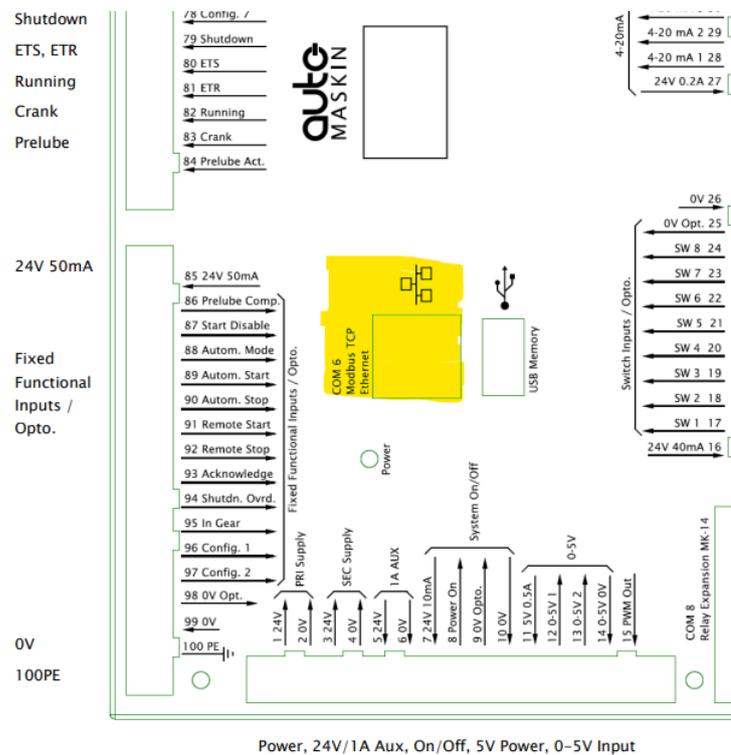


Por el puerto COM 6 de la DCU 410E, que se representa en la

Figura 20, se tiene disponible Modbus TCP/IP Server o Client, resguardadas por las seguridades y características del switch industrial, el protocolo TCP/IP es un protocolo orientado a la conexión y con su filosofía de ejecución garantiza la integridad de los datos, eléctricamente usa mismos principios de pares trenzados para conectar dos puntos a la red.

Figura 20

Puerto COM 6 en Modbus TCP Ethernet



El módulo de software de Modbus ejecutándose en el DCU 410E están bajo la supervisión y control de uno o varios Watchdog, los cuales nos van a permitir restablecer el enlace y la comunicación Modbus del módulo en caso de que este se viera afectado por alguna causa.

Desde el servidor Modbus TCP/IP hacia el servidor de datos usa Ethernet convencional y de cara a la ciberseguridad, un tema de mucho cuidado últimamente, podemos deducir que está garantizada debido a que ninguno de los dispositivos presentes en la red LAN van a tener acceso al internet.

2.4.3. Velocidades

La velocidad de un sistema de comunicación juega un papel fundamental debido que al usar grandes velocidades aumenta potencialmente el riesgo a tener errores en la comunicación.

La velocidad de CANBUS J1939 para el puerto de comunicación usado es de 250 Kbs con su respectiva resistencia de 120 Ohms para configurar terminación de nodo y evitar reflejos de datos en el nodo.

La velocidad disponible para el enlace Ethernet usado en el COM6 es de 10 Mbs, es una velocidad suficiente para la transmisión de datos de tipo cuantitativo mediante el protocolo Modbus, mientras que cuando se utiliza los servicios de configuración con el protocolo de comunicación HTTP es de carga moderada y de muy baja frecuencia, es decir se lo hace estrictamente para puesta en marcha del sistema o para alguna configuración específica del comportamiento del mismo.

2.4.4. Cantidad de datos

Los 12 datos básicos más relevantes para la supervisión y monitoreo del estado del motor, tal como se aprecia en la Tabla 6, son la temperatura del refrigerante, presión del combustible, temperatura de combustible, consumo de combustible, presión de la transmisión, temperatura de la transmisión, velocidad del eje del motor, presión de aceite, temperatura del escape, carga del motor y diferencial de presión de presión del filtro de aceite, son datos almacenados en 2 palabras de 16 bits en la tabla de memoria Modbus en la DCU 410E con sus respectivas unidades o magnitudes físicas.

Tabla 6

Datos esenciales para la supervisión y monitoreo del motor.

	A	B	C	D	E	F	G	H	I	J	K	L	M
		PGN	SPN	Read/ Write	Bit/ Word	Warning Ala. Shdn.	W/A/S/FMI ¹	Bit/ Word	FMI -1 = no DM1 alarm	Value MSW	Value LSW	Unit	Unit Address
1	DCU 210/410 Engine Panel / LT-ONE Main Panel												
3	J1939 Accelerator Pedal Position	0xF003	91	R	B	101900	105000	W	451600	451400	451401	m%	401001
4	J1939 Air Filter Diff Pressure	0xFFE6	107	R	B	101901	105001	W	451601	451402	451403	Pa	401101
5	J1939 Air Inlet Pressure	0xFFE6	106	R	B	101902	105002	W	451602	451404	451405	Pa	401201
6	J1939 Barometric Pressure	0xFFE5	108	R	B	101903	105003	W	451603	451406	451407	Pa	401301
7	J1939 Boost Pressure	0xFFE6	102	R	B	101904	105004	W	451604	451408	451409	Pa	401401
8	J1939 Coolant Level	0xFFEF	111	R	B	101905	105005	W	451605	451410	451411	m%	401501
9	J1939 Coolant Pressure	0xFFEF	109	R	B	101906	105006	W	451606	451412	451413	Pa	401601
10	J1939 Crankcase Temperature	0xFFEE	110	R	B	101907	105007	W	451607	451414	451415	mC	401701
11	J1939 Crankcase Pressure	0xFFEF	101	R	B	101908	105008	W	451608	451416	451417	Pa	401801
12	J1939 Engine Aux Coolant Pressure	0xFFE9	1203	R	B	101909	105009	W	451609	451418	451419	Pa	401901
13	J1939 Engine Intercool Temperature	0xFFEE	52	R	B	101910	105010	W	451610	451420	451421	mC	402001
14	J1939 Engine Load	0xF003	92	R	B	101911	105011	W	451611	451422	451423	m%	402101
15	J1939 Engine Oil Level	0xFFEF	98	R	B	101912	105012	W	451612	451424	451425	m%	402201
16	J1939 Engine Oil Pressure	0xFFEF	100	R	B	101913	105013	W	451613	451426	451427	Pa	402301
17	J1939 Engine Speed	0xF004	190	R	B	101914	105014	W	451614	451428	451429	mRPM	402401
18	J1939 Exhaust Temperature	0xFFE6	173	R	B	101915	105015	W	451615	451430	451431	mC	402501
19	J1939 Fuel Delivery Pressure	0xFFEF	94	R	B	101916	105016	W	451616	451432	451433	Pa	402601
20	J1939 Fuel Filter Diff Pressure	0xFFE6	95	R	B	101917	105017	W	451617	451434	451435	Pa	402701
21	J1939 Fuel Rate	0xFFE2	183	R	B	101918	105018	W	451618	451436	451437	ml/h	402801
22	J1939 Fuel Temperature	0xFFEE	174	R	B	101919	105019	W	451619	451438	451439	mC	402901
23	J1939 Intake Manifold Temperature	0xFFE6	105	R	B	101920	105020	W	451620	451440	451441	mC	403001
24	J1939 Left Manifold Exhaust Temperature	0xFE07	2434	R	B	101921	105021	W	451621	451442	451443	mC	403101
25	J1939 Oil Filter Diff Pressure	0xFFE6	99	R	B	101922	105022	W	451622	451444	451445	Pa	403201
26	J1939 Prefilter Fuel Pressure	0xFFE6	1382	R	B	101923	105023	W	451623	451446	451447	Pa	403301
27	J1939 Prefilter Oil Pressure	0xFFE2	1208	R	B	101924	105024	W	451624	451448	451449	Pa	403401
28	J1939 Right Manifold Exhaust Temperature	0xFE07	2433	R	B	101925	105025	W	451625	451450	451451	mC	403501
29	J1939 Transmission Oil Pressure	0xFFE8	127	R	B	101926	105026	W	451626	451452	451453	Pa	403601
30	J1939 Transmission Oil Temperature	0xFFE8	177	R	B	101927	105027	W	451627	451454	451455	mC	403701
31	J1939 Turboch 1 Comp Inlet Pressure	0xFFE9	1176	R	B	101928	105028	W	451628	451456	451457	Pa	403801
32	J1939 Turboch 2 Comp Inlet Pressure	0xFFE9	1177	R	B	101929	105029	W	451629	451458	451459	Pa	403901
33	J1939 Engine Oil Temperature	0xFFEE	175	R	B	101930	105030	W	451630	451460	451461	mC	404001
34	J1939 Exhaust Port Temperature 1	0xFE03	1137	R	B	101931	105031	W	451631	451462	451463	mC	404101
35	J1939 Exhaust Port Temperature 2	0xFE03	1138	R	B	101932	105032	W	451632	451464	451465	mC	404201
36	J1939 Exhaust Port Temperature 3	0xFE03	1139	R	B	101933	105033	W	451633	451466	451467	mC	404301

Nota. Adaptado de *Auto-Maskin 200/400 Series Communication List*, por Auto-Maskin AS., 2017a, <https://docs.google.com/spreadsheets/d/1yE8VU858j559ICNR-3cQeg9R78ncSFocq0p0T2rU6o/edit#gid=1531835075>

La consulta de esta información mediante Modbus se realiza de tal manera que se pueda optimizar el bus de comunicación y no sobrecargar a la DCU 410E en el motor, esto se logra usando la información de la tabla de Modbus proporcionada por el fabricante en el documento de excel Auto-Maskin 200/400 Series Communication List rescatado de la red en abril del 2024, en la página “J1939 1 Status” donde se deduce que se puede realizar una sola consulta de datos desde la celda 10 a la 33 (24 Datos) con las direcciones Modbus de los valores de interés e implican 48 direcciones, desde la 451414 a la 451461, para su posterior tratamiento en el servidor.

La carga del bus de comunicación entre las 2 HMI presenta una carga moderada ya que debe realizar al menos 2 consultas Modbus, una para leer datos a presentar y otra para leer estados o alarmas o ambos a la vez.

2.5. Métodos y desarrollo de la investigación

2.5.1. Enlace CANBUS J1939 Computadora del motor y DCU 410E

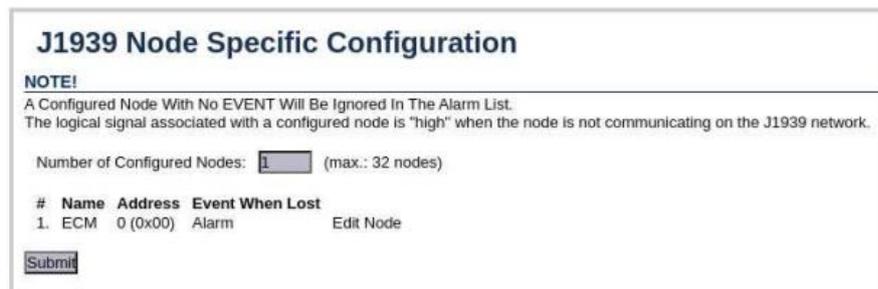
Una vez identificado el enlace físico anteriormente a usar en el DCU 410E, se realiza la conexión de un par de cables trenzados de preferencia para conectar CAN_H y CAN_L desde el tablero de alojamiento de la DCU 410E hacia el tablero del motor que tiene por parte del fabricante del motor expuesto en bornera tipo riel este enlace. Posteriormente accedemos a la configuración del DCU 410E por el explorador de internet para su configuración y accedemos a la dirección /home/dcu/communication/j1939 configuration para configurar la interfaz que en este caso seleccionamos COM4.

Luego en la dirección /home/dcu/communication/j1939 configuration/j1939:1 tendremos los parámetros principales del nodo y dejamos por defecto la dirección de la DCU 410E el valor hexadecimal 0xF2, en la dirección de respuesta dejamos colocado 0x00 que indica que se espera que un dispositivo en la dirección más baja responda a los datos y como solo tendremos a la computadora, entonces es la que responde.

Las funciones configuraciones especiales quedan deshabilitadas ya que el control del motor no se lo realiza por con esos criterios y finalmente en el apartado de configuración especial del nodo habilitamos el evento al nodo 1 para que se tenga una alarma en el panel cuando se pierda la comunicación con la computadora como indica la Figura 21.

Figura 21

Configuración específica del nodo.



The screenshot shows a web interface titled "J1939 Node Specific Configuration". It includes a "NOTE!" section with the text: "A Configured Node With No EVENT Will Be Ignored In The Alarm List. The logical signal associated with a configured node is 'high' when the node is not communicating on the J1939 network." Below the note, there is a field for "Number of Configured Nodes" with a value of "1" and a maximum of "32 nodes". A table lists the configured nodes with columns for "#", "Name", "Address", "Event When Lost", and "Edit Node". The table contains one entry: "1. ECM 0 (0x00) Alarm Edit Node". A "Submit" button is located at the bottom left of the form.

#	Name	Address	Event When Lost	Edit Node
1.	ECM	0 (0x00)	Alarm	Edit Node

Nota. Tomado de *J1939 Node Specific Configuration* (p. 74), por (Auto-Maskin AS., 2023), https://www.auto-maskin.com/wp-content/uploads/2023/11/400E_200E-Configuration-Manual-3.10P5.pdf

2.5.2. Enlace Modbus TCP/IP entre RP 410E y DCU 410E

Como primer paso en la parte física se conectó el panel remoto RP410E HMI 2 en el puente del capitán a 24 VDC desde la fuente de energía del tablero de la DCU 410E local y se lo puso en red usando un cable Ethernet de uso industrial de 4 hilos de aproximadamente 40 m en modo directo al switch.

En el RP 410E tiene la dirección IP 192.168.0.201 de fábrica, y contraseña es 1234, con la IP comprobada comunicación podemos acceder de la misma manera a la configuración vía <http://192.168.0.201>.

En el manual de comunicación para Modbus TCP/IP tenemos dos configuraciones importantes como son el máximo tiempo de espera a una respuesta para posterior a ese tiempo silenciar la réplica e intentar una nueva solicitud y finalmente la opción de bloquear funciones de escritura vía Modbus y en la selección podemos habilitar o deshabilitar la escritura por las FC 05, 06, 15 y 16.

Como muestra la Figura 22 en el panel RP 410E en el apartado /Home/Settings/Administration/DCU Connections de la interfaz gráfica se puede presionar el botón buscar DCU, entonces el RP410E enlistara los DCU disponibles en la red, seleccionar la DCU de la sala de máquinas y presionar el botón OK para que automáticamente se autoconfigure para emparejarse con la DCU seleccionada, este

modelo puede conectarse a 8 DCU, de ser necesario aquí se puede configurar la IP de la DCU objetivo manualmente.

Figura 22

Menú de administración



Nota. Tomado de *Administration Menu* (p. 97), por (Auto-Maskin AS., 2023), https://www.auto-maskin.com/wp-content/uploads/2023/11/400E_200E-Configuration-Manual-3.10P5.pdf

2.5.3. Enlace de DCU 410E y Servidor de datos

La PC servidor se le debe configurar con la dirección IP estáticamente 192.168.0.301/24 y puerta de enlace 192.168.0.1 como sugiere el fabricante, posterior a ello podemos hacer prueba de conectividad a la DCU 410E y RP 410E usando servicio de ping. Para obtener los datos por Modbus TCP/IP desde el DCU 410E a través del switch en el PC servidor se lo realiza usando un script de Python, se utiliza la librería pymodbus para leer los registros 20 registros de J1939 que implican 40 direcciones de la tabla Modbus, desde la dirección 451414 según el documento de apoyo para la comunicación del fabricante, a pesar de que son solo 12 los registros que se interesa leer de la tabla, debido a que no están distribuidos de forma continua como muestra en la Tabla 7, es más optimo realizar una sola lectura de 20 parámetros y tomarlos con asignación directa y concatenación cada 5 segundos los 12 de interés, optimizando así en canal de comunicación evitando peticiones separadas. Para leer los parámetros es necesario usar la dirección IP

192.168.0.101 que es la dirección de la DCU 410E de fábrica, usando el puerto por defecto para servicios de Modbus 502.

Tabla 7

Parámetros a considerar en las tramas de comunicación

9	J1939 Coolant Pressure
10	J1939 Coolant Temperature
11	J1939 Crankcase Pressure
12	J1939 Engine Aux Coolant Pressure
13	J1939 Engine Intercool Temperature
14	J1939 Engine Load
15	J1939 Engine Oil Level
16	J1939 Engine Oil Pressure
17	J1939 Engine Speed
18	J1939 Exhaust Temperature
19	J1939 Fuel Delivery Pressure
20	J1939 Fuel Filter Diff Pressure
21	J1939 Fuel Rate
22	J1939 Fuel Temperature
23	J1939 Intake Manifold Temperature
24	J1939 Left Manifold Exhaust Temperature
25	J1939 Oil Filter Diff Pressure
26	J1939 Prefilter Fuel Pressure
27	J1939 Prefilter Oil Pressure
28	J1939 Right Manifold Exhaust Temperature
29	J1939 Transmission Oil Pressure
30	J1939 Transmission Oil Temperature
31	J1939 Turboch 1 Comp Inlet Pressure
32	J1939 Turboch 2 Comp Inlet Pressure
33	J1939 Engine Oil Temperature
34	J1939 Exhaust Port Temperature 1
35	J1939 Exhaust Port Temperature 2
36	J1939 Exhaust Port Temperature 3
--	--

Nota. Adaptado de *Auto-Maskin 200/400 Series Communication List*, por Auto-Maskin AS., 2017a, <https://docs.google.com/spreadsheets/d/1yE8VU858j5591CNR-3cQeg9R78ncSFoczq0p0T2rU6o/edit#gid=1531835075>

Una vez obtenidos los parámetros podemos elaborar un objeto JSON para el envío a Node Red a través de Broker Mosquitto, se debe asignar el payload correspondiente a cada variable que vamos a transmitir obtenidas en Modbus, por otro lado, ya que el script de Python se va a ejecutar en el mismo PC servidor donde se va a ejecutar Mosquitto la IP la podemos usar la 192.168.0.301 antes configurada o hacer referencia a localhost por el puerto 1883 para MQTT.

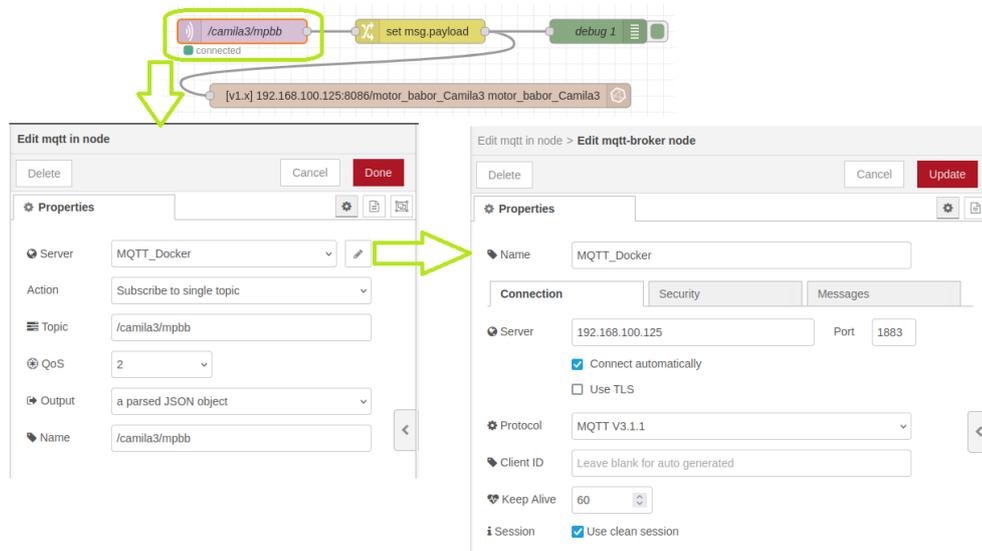
Para obtener más detalles del código consultar el ANEXO 1.

2.5.4. Enlace de datos Mosquito Broker -InfluDB

En Node Red se usa el bloque o nodo In MQTT para leer el stream de texto tipo JSON, se lo configura como indica la Figura 23.

Figura 23

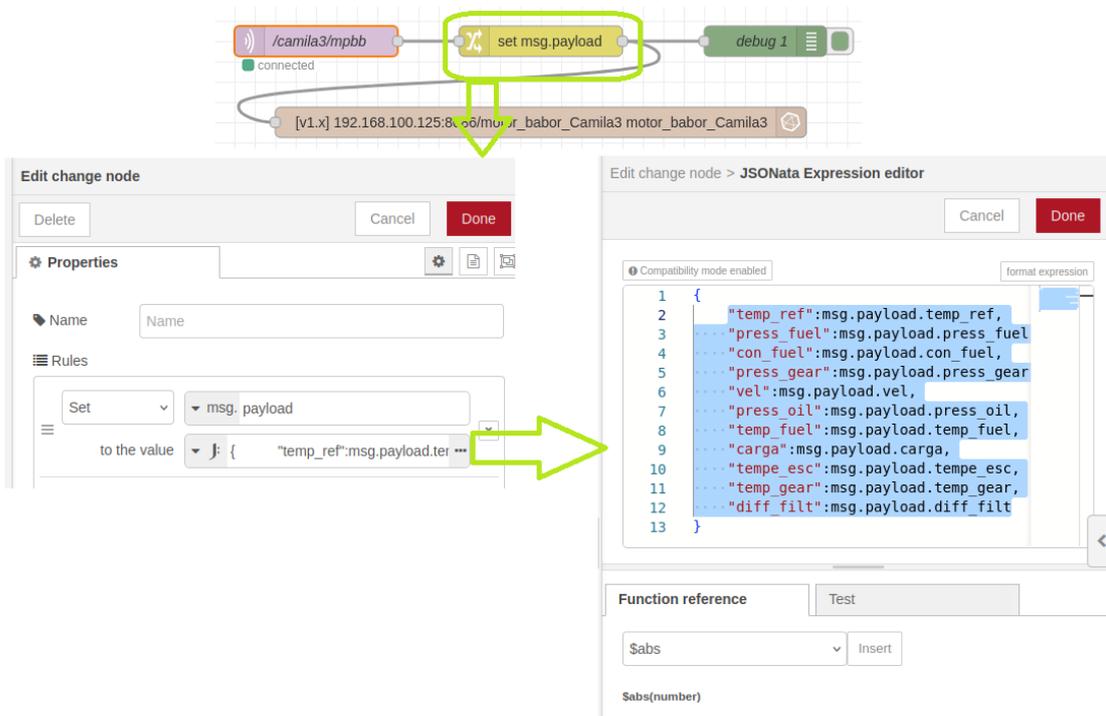
Configuración de MQTT en el nodo



Del JSON procedemos a tomar los payload o carga útil de cada parámetro que se envió en el script antes mencionado y esto se lo realiza usando el bloque o nodo de tipo Función y debe tener el código como detalla la Figura 24.

Figura 24

Obtención de la carga útil de cada parámetro



Antes de usar los bloques en Node Red para inyectar datos en la base de datos se debe crear y configurar el uso de esa base de datos en InfluxDB con los siguientes comandos en el menú IOTStack como se muestra en la Figura 25.

Figura 25

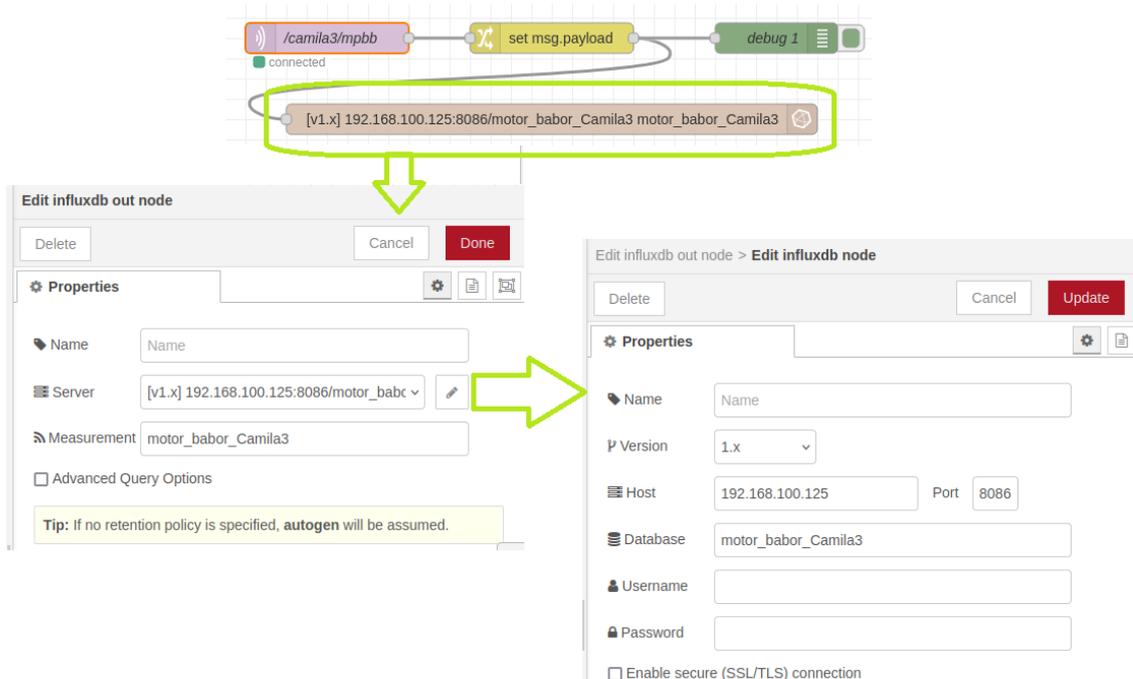
Creación y configuración de la base de datos en InfluxDB

```
cris@cris-VirtualBox:~/IOTstack$ docker exec -it influxdb influx
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
> show databases
name: databases
name
----
motor_babor_Camila3
> create database motor_estribor_Camila3
> show databases
name: databases
name
----
motor_babor_Camila3
motor_estribor_Camila3
> use motor_estribor_Camila3
Using database motor_estribor_Camila3
>
```

Finalmente, la salida del nodo de tipo Función conectamos a un nodo de tipo entrada de InfluxDB con las configuraciones del servidor como se visualiza en la Figura 26.

Figura 26

Configuración del nodo de InfluxDB



2.5.5. Consulta y presentación de los datos en Grafana

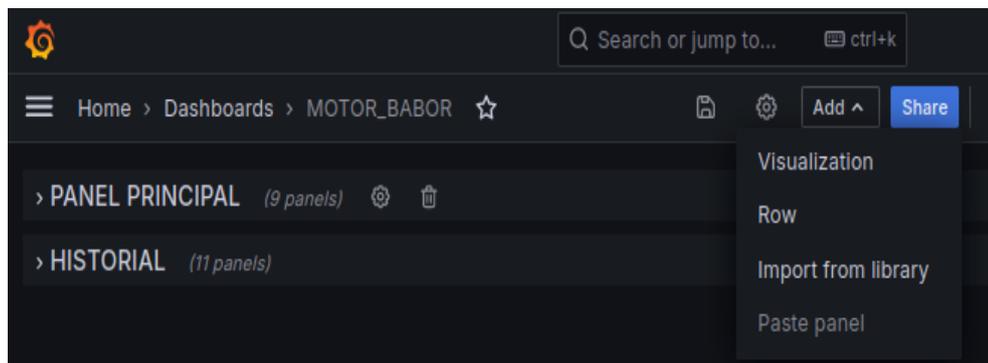
Para acceder al servidor de Grafana se utiliza un explorador de internet y se ingresa a la URL <http://192.168.100.125:3000>, donde se pide configurar usuario y contraseña.

Para usar Grafana un paso indispensable es configurar la base de datos en la sección /Connections/Data sources, tal como se indica en ANEXO 2, donde se selecciona el Lenguaje Query como InfluxQL que es una variante de MySQL, se agrega la URL del servidor seguido del puerto 8086, en la opción Database se coloca el nombre de la base de datos que se creó previamente, en este caso, se llama motor_babor_Camila3 y POST para el método usado en HTTP.

Se guarda y comprueba con el botón inferior, y se procede a crear dos Filas (Rows), una para el Dashboard visual para monitoreo y el segundo para presentar los datos en curvas históricas como se muestra en la Figura 27.

Figura 27

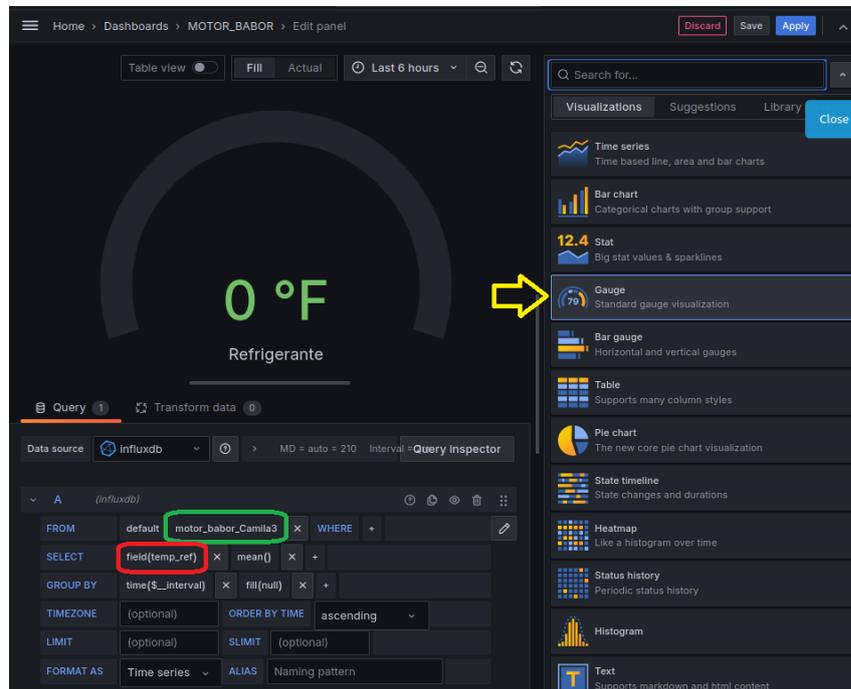
Filas creadas para el Dashboard



En cada fila se puede agregar objetos de visualización y se los configura cada uno con la base de datos de entrada y el dato de esa base de datos, así mismo características del visualizador que pueden ser Indicadores radiales, indicadores tipo barra e históricos. Como ejemplo, se presenta en la Figura 28, un gauge para indicar la temperatura del refrigerante.

Figura 28

Gauge para la temperatura del refrigerante

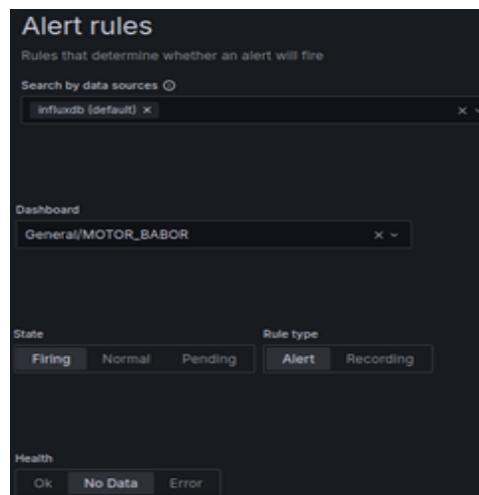


2.5.6. Implementación de reglas en Grafana para Alarmas

Grafana tiene la interfaz correspondiente para la implementación de reglas para alertar en paralelo a las alarmas implementadas en la DCU 410E y RP 410E. En la Figura 29 se presenta la adición de la alerta principal de pérdida de Datos en la sección /Alerting/Alert rules.

Figura 29

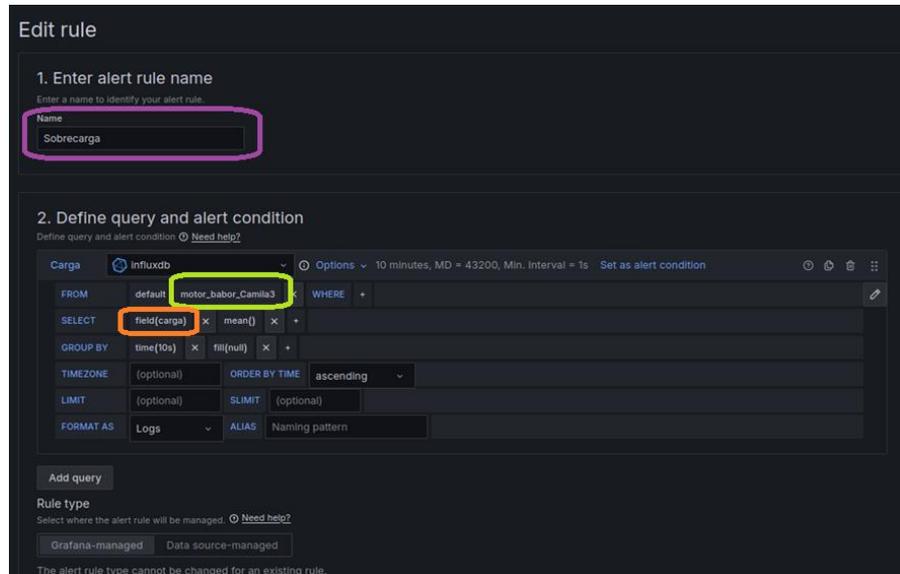
Adición de la alerta de pérdida de datos



Posteriormente, se selecciona agregar nueva regla y se asigna el nombre “Sobrecarga”, apuntando al dato en la correspondiente base de datos como se aprecia en la Figura 30.

Figura 30

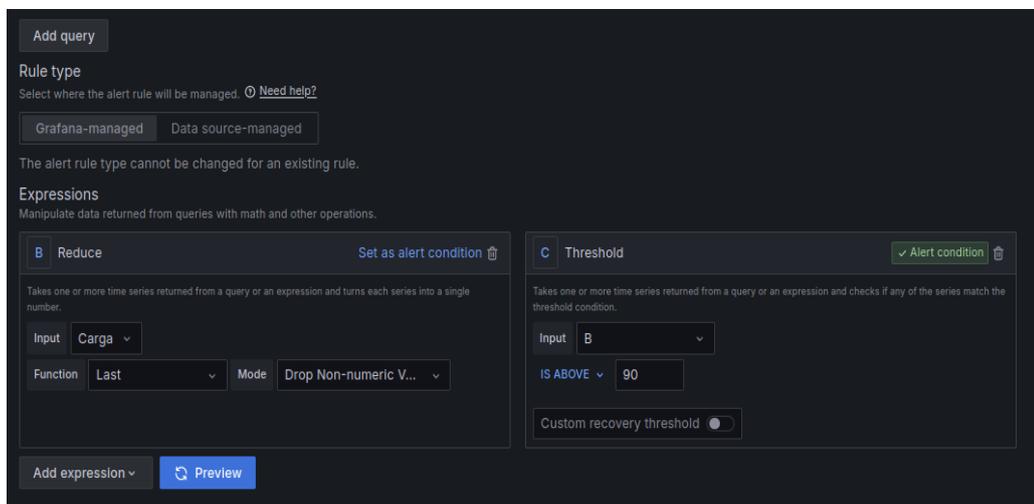
Regla de alerta “Sobrecarga”



Se selecciona el manejo por Grafana el tipo de regla, se deja solo la expresión Threshold C donde se pone la condición de alerta cuando sobrepase 90%, tal como se evidencia en la Figura 31, debido a que llegar a usar el motor a ese porcentaje de carga por un tiempo muy prolongado aumenta las posibilidades de mayor desgaste y reducción de la vida útil del motor.

Figura 31

Condición de alerta



Se procede a configurar el comportamiento de la regla como se ilustra en la Figura 32, donde guardar la regla, asignar a un grupo de evaluación todas las reglas ya que se requiere una supervisión lo más completa con los 12 valores disponibles.

Figura 32

Configuración de la condición de la regla de alerta

The screenshot displays the configuration interface for an alert rule, divided into two main sections:

- 3. Set evaluation behavior**
 - Define how the alert rule is evaluated. [Need help?](#)
 - Folder**
Select a folder to store your rule.
AlertaSobrecarga or + New folder
 - Evaluation group**
Rules within the same group are evaluated concurrently over the same time interval.
Babor2 or + New evaluation group
 - All rules in the selected group are evaluated every 1m.
 - Pending period**
Period in which an alert rule can be in breach of the condition until the alert rule fires.
1m
 - Pause evaluation [?](#)
 - [Configure no data and error handling](#)
- 4. Configure labels and notifications**
 - Labels**
Add labels to your rule to annotate your rules, ease searching, or route to a notification policy. [Need help?](#)
 - tipo = grave
 - Choose key = Choose value
 - [Add label](#)
 - Alert instance routing preview**
When you have your folder selected and your query and labels are configured, click "Preview routing" to see the results here

Una vez finalizado el procedimiento, ya tenemos disponibles las reglas y ejecutandose en grafana, existe la posibilidad de implementar notificaciones a raíz de estas reglas ya sea por correo electrónico, Telegram, etc.

CAPÍTULO 3. RESULTADOS Y DISCUSIÓN

3.1. Pruebas de respuestas del Gateway al servidor de datos

De acuerdo a los datos del ANEXO 3, se obtuvo la Tabla 6:

Tabla 6

Datos estadísticos

Dato	Valor (mS)
Media	287.89
Mediana	74.78
Desviación Estandar	611.49
Mínimo	4.75
Máximo	4041.05

3.1.1. Media

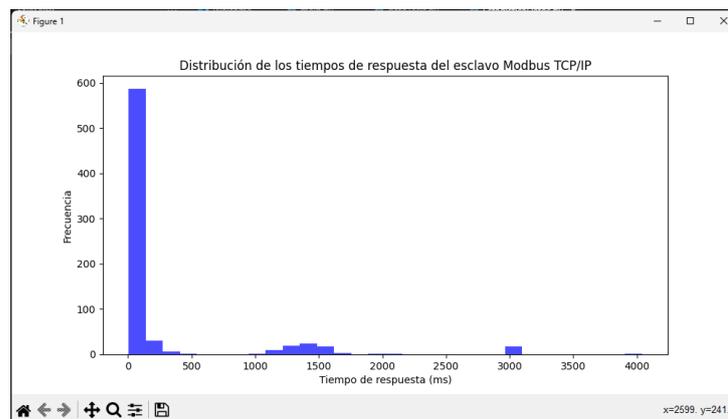
Que quiere decir que, en promedio, la duración o tiempo que tarda el esclavo en responder es aproximadamente 288 ms para solicitar 40 registros Modbus de 16 bits.

3.1.2. Mediana

Lo que nos presenta que el 50% de los valores está por debajo de 74.78 ms y como es menor que 288 ms (Promedio) nos indica que tenemos una distribución asimétrica como indica la Figura 33 con una cola hacia los valores más altos.

Figura 33

Frecuencia del muestreo.

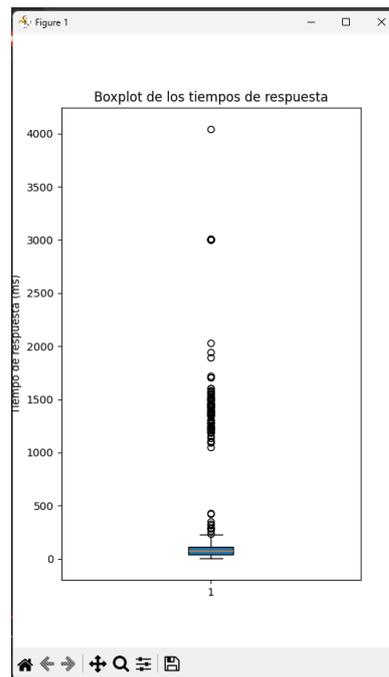


3.1.3. Desviación estándar

Vemos que 611.49 ms es mucho mayor que la media e indica que los datos son muy dispersos y hay valores extremos que influyen significativamente en el análisis, llegando a alcanzar 4041.05 ms, como podemos observar en la Figura 34 son pocos los datos, pero en una proporción de 13 a 1 aproximadamente.

Figura 34

Diagrama de caja de los tiempos de respuesta



3.1.4. Mínimo

Siendo 4.75 ms el tiempo el más corto o pequeño transcurrido desde que el maestro envía una solicitud de datos al esclavo y obtener una correcta respuesta.

3.1.5. Máximo

Aquí observamos que el máximo tiempo es 4041.05 ms, siendo menor que el periodo de solicitud es 5000 ms, nos indica que no hubo errores por tiempo de espera para el almacenamiento de datos, pero al ser un valor muy alto, la posibilidad de obtener errores es latente, aunque esto nos daría una tasa de error relativamente aceptable.

3.2. Interface HMI remota (RP 410E) en el puente del capitán

Como se puede observar en la Figura 35, se tiene la adaptación e instalación física en el lado izquierdo de la consola de instrumentos de la embarcación, tiene una inclinación estratégica para poder observar desde varios ángulos desde el lado derecho del puente que es desde donde el capitán normalmente realiza las maniobras, este ángulo resulta adecuado cuando existe demasiada luz durante el día y para poder monitorear el motor.

El empotramiento se logra con la ayuda de una modificación en la consola.

Figura 35

Ubicación del RP 410E en consola en el puente del capitán



3.3. Prueba de reglas implementadas en Grafana

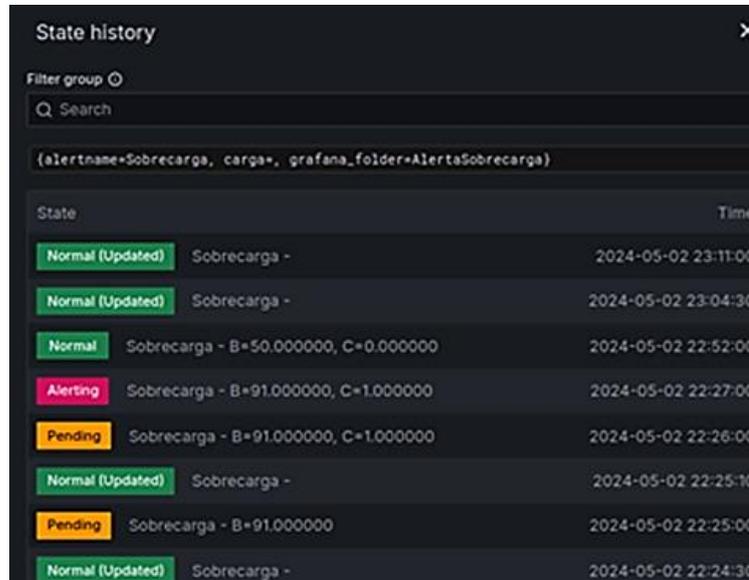
Como se puede observar en los ANEXO 4, 5 y 6, las reglas una vez guardadas, pasana al estado de normal hasta el siguiente diagnostico, una vez que cumpla la condicion de de la regla del valor establecido, se puede ver al sistema alarmado en el dashboard web.

El estado del ANEXO 4 muestra una alarma normal, el ANEXO 5 indica el estado que la conficion se sigue cumpliendo y el estado del ANEXO 6 significa que la condicion se cumplio almenos una vez y regreso a la normalidad.

Ademas tal como se ilustra en la Figura 36, se puede acceder al historial de alertas, con el objetivo de tomar decisiones en función del análisis de las nuevas variables cualitativas.

Figura 36

Historial de alertas



3.4. Historial del registro de los parámetros del sistema

En la Figura 37 podemos observar un dashboard web en Grafana con 9 de los 12 parámetros disponibles en el almacenamiento, estando localizado en la parte central la velocidad del motor, siendo una de las variables de control más importantes, las otras variables son de monitoreo dependientes en cierta medida de la velocidad. Se implemento este dashboard con la disposición de los objetos de tal manera que se parezca lo más posible a la interfaz gráfica del panel remoto.

Figura 37

Evidencia del Dashboard web en Grafana



En la ANEXO 7 tenemos la tabla de graficas del comportamiento de las 12 variables en el tiempo, recordar que nuestro tiempo de muestreo es 5 segundos, están dispuestos para observar el historial de la ultima hora. Como se puede observar que en algunos casos tienen rasgos similares, pero en diferentes rangos, esto se debe por la relación que tienen y en diferentes rangos o similares se observa las gráficas ya que los gráficos en el tiempo tienen la opción de auto ajustar el grafico dinámicamente al valor máximo encontrado en el historial o poderlo dejar fijo como esta en algunos casos para mejorar la experiencia de percepción.

CONCLUSIONES

- La revisión de las normativas y estándares esenciales para el sistema de comunicación industrial ha garantizado su funcionamiento efectivo, incluso bajo las condiciones potencialmente disruptivas del entorno marítimo. Este análisis destaca la crucial importancia de elegir componentes de alta calidad y adecuados para la infraestructura de comunicación en dichas aplicaciones, donde cualquier fallo puede acarrear consecuencias significativas.
- La configuración del Gateway DCU 410E para la comunicación CAN J1939 a Modbus TCP/IP y el Gateway en el servidor para la comunicación Modbus TCP/IP a MQTT representa una integración efectiva de tecnologías para el monitoreo y control en entornos marítimos. La consideración de factores como la velocidad de comunicación, la latencia y la cantidad de datos es fundamental para garantizar la fiabilidad de los enlaces de datos, especialmente en condiciones marítimas desafiantes. La baja tasa de errores de tiempo vencido obtenida como resultado de esta configuración demuestra la robustez y eficiencia del sistema implementado.
- La implementación de una HMI en el puente del capitán para el monitoreo del motor ha permitido presentar un espectro más amplio del estado del motor antes, durante y después de las maniobras lo que conlleva a una mejor operación de la embarcación evitando poner al motor a trabajar en condiciones fuera de lo normal.
- La implementación de tecnologías del IIoT en la red marítima representa un avance significativo hacia la integración del sector marítimo en la Industria 4.0. Este enfoque innovador para la captura de datos ha logrado romper barreras tradicionales, si se enfoca la implementación en términos de seguridad informática va a ser común el tener motores de combustión interna conectados a la nube.
- Al tener configuraciones de alerta tanto en Grafana como en el HMI, se garantiza que las advertencias se generen desde múltiples fuentes, además de tener una supervisión abstracta de la embarcación, lo que reduce la dependencia de una única plataforma y minimiza el riesgo de no detectar a tiempo una situación crítica. Este enfoque aumenta la integridad del sistema de supervisión y control,

asegurando que las alertas puedan ser evaluadas y gestionadas prontamente, incluso si uno de los sistemas falla o se ve comprometido.

- La recopilación de datos robusta y sistemática es fundamental para la creación de bases de datos sólidas, las cuales son imprescindibles para la implementación efectiva de técnicas computacionales inteligentes. Estas tecnologías dependen críticamente de la calidad, cantidad y diversidad de los datos recolectados para desarrollar modelos predictivos, precisos y confiables.
- El método adoptado para realizar solicitudes extensas de 40 registros con baja frecuencia, en lugar de solicitudes cortas más frecuentes, mientras se descartan activamente los datos no esenciales, ha demostrado ser particularmente beneficioso. Este enfoque minimiza la saturación de la red y reduce la carga en el Gateway DCU 410E, permitiendo una comunicación eficiente sin errores entre el esclavo y el servidor de datos.

RECOMENDACIONES

- Cuando se requiera modificar la topología o los datos a comunicar en la red explorar la posibilidad de ajustar la frecuencia y el tamaño de las solicitudes según las condiciones de la carga de trabajo del Gateway para mantener una óptima eficiencia.
- Realizar estudios comparativos con otros sistemas de monitorización para validar y, si es necesario, mejorar la efectividad de la HMI implementada, esto puede implicar en aumentar el número de variables relevantes en el HMI.
- Realizar inspecciones y mantenimientos periódicos del estado físico de la red debido a que se encuentra expuesta a perturbaciones tales como altas temperaturas, humedad, salinidad, vibración y ruido electromagnético, para asegurar su durabilidad y funcionalidad a largo plazo.
- Priorizar la seguridad cibernética cuando se vea la necesidad de conectar dispositivos y subsistemas al Internet para proteger contra vulnerabilidades y amenazas externas.
- Desarrollar protocolos de respuesta rápida ante las alarmas o advertencias generadas para asegurar las medidas adecuadas y oportunas para prevenir inconvenientes.

REFERENCIAS

- Advantech Czech s.r.o. (2022). *User Module Node-RED*.
- Amjad, A., Azam, F., Anwar, M. W., & Butt, W. H. (2021). A Systematic Review on the Data Interoperability of Application Layer Protocols in Industrial IoT. *IEEE Access*, 9, 96528-96545. <https://doi.org/10.1109/ACCESS.2021.3094763>
- Andueza, E. (2009, noviembre 10). *Ethernet—La conexión óptima*. Conectronica. <https://www.conectronica.com/ethernet-industrial/ethernet-la-conexion-optima>
- Auto-Maskin AS. (2017a). *Auto-Maskin 200/400 Series Communication List*. Google Docs. https://docs.google.com/spreadsheets/d/1yE8VU858j5591CNR-3cQeg9R78ncSFoczq0p0T2rU6o/edit?usp=embed_facebook
- Auto-Maskin AS. (2017b). *Installation Manual for Marine Pro 400E Series*. https://www.auto-maskin.com/wp-content/uploads/2015/04/400E_Series_Installation_Manual.pdf
- Auto-Maskin AS. (2020). *Installation Manual for Marine Pro 400 Series*. <https://www.auto-maskin.com/wp-content/uploads/2021/01/SDU-410-Installation-Manual-Nov-2020.pdf>
- Auto-Maskin AS. (2023). *Configuration Manual 400 Series / 200 Series*. https://www.auto-maskin.com/wp-content/uploads/2023/11/400E_200E-Configuration-Manual-3.10P5.pdf
- Baudoin, C., Bournival, E., & Clauer, E. (2021). *Global Industry Standards for Industrial IoT*.

Caterpillar. (2005). *New 3500C Marine Engine Ratings Give Caterpillar the Edge in Displacement, Semi-displacement and Planing Hull Yacht Power Stakes* by Caterpillar. boatdiesel.com. <https://boatdiesel.com/News/Caterpillar/N-501/New-3500C-Marine-Engine-Ratings-Give-Caterpillar-the-Edge-in-Displacement-Semi-displacement-and-Planing-Hull-Yacht-Power-Stakes.cfm?NC=1>

Eclipse Mosquito. (2018, enero 8). Eclipse Mosquito. <https://mosquito.org/>

Eland Cables. (2024). *Cable IEC 61158 / Eland Cables*.

<https://www.elandcables.com/es/electrical-cable-and-accessories/cables-by-standard/iec-61158-cable>

Eurotech & International Business Machines Corporation [IBM]. (2017). *MQTT V3.1 Protocol Specification*. <https://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html>

Galloway, B., & Hancke, G. P. (2013). Introduction to Industrial Control Networks.

IEEE Communications Surveys & Tutorials, 15(2), 860-880.

<https://doi.org/10.1109/SURV.2012.071812.00124>

García, D. (2015, noviembre 18). *ISA101 Norma para el diseño HMI*. infoPLC.

<https://www.infoplcn.net/actualidad-industrial/item/102902-isa101-hmi>

Hemmatpour, M., Ghazivakili, M., Montrucchio, B., & Rebaudengo, M. (2017). DIIG:

A Distributed Industrial IoT Gateway. 2017 IEEE 41st Annual Computer

Software and Applications Conference (COMPSAC), 755-759.

<https://doi.org/10.1109/COMPSAC.2017.110>

- Hernández, J. A. (2016). *DESCRIPCION Y FUNCIONAMIENTO DEL MOTOR MARINO CATERPILLAR 3618.pdf* (p. 23). Universidad de La Laguna.
<https://riull.ull.es/xmlui/handle/915/3126>
- Himanshu, Sharma, N., & Singh, R. (2020). Evolution of IoT to IIoT: Applications & Challenges. *SSRN Electronic Journal*.
<https://doi.org/10.2139/ssrn.3603739>
- JSON. (2024). *Introducción a JSON*. json.org. <https://www.json.org/json-es.html>
- Martínez, J. (2022, febrero 25). *Qué es InfluxDB y primeros pasos*. OpenWebinars.net.
<https://openwebinars.net/blog/que-es-influxdb-y-primeros-pasos/>
- Modbus-IDA. (2016). *MODBUS APPLICATION PROTOCOL SPECIFICATION*.
https://modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf
- Moxa. (s. f.). *EDS-205 Series*. Recuperado 5 de mayo de 2024, de
<https://www.moxa.com/getmedia/e4068be8-1168-4cf8-9437-0d08ca4ac4be/moxa-eds-205-series-datasheet-v1.2.pdf>
- Muñoz, R. (2022). *Implementación de un sistema de control de combustible para un motor Caterpillar 3412 de la embarcación fluvial Miranda I* [Universidad Tecnológica de Perú].
https://repositorio.utp.edu.pe/bitstream/handle/20.500.12867/8235/R.Fasabi_Programa_Especial_Titulacion_Titulo_Profesional_2022.pdf?sequence=1&isAllowed=y
- Peter, O., Pradhan, A., & Mbohwa, C. (2023). Industrial internet of things (IIoT): Opportunities, challenges, and requirements in manufacturing businesses in

emerging economies. *Procedia Computer Science*, 217, 856-865.

<https://doi.org/10.1016/j.procs.2022.12.282>

Python TM. (2024). *What is Python?* Python.Org.

<https://www.python.org/doc/essays/blurb/>

Wang, Z., Han, D., Gong, Y., & Zhao, Y. (2022). Multi-protocol Integration and Intercommunication Technology Based on OPC UA and MQTT. *Journal of Physics: Conference Series*, 2173(1), 012070. <https://doi.org/10.1088/1742-6596/2173/1/012070>

Zurawski, R. (2014). *Industrial Communication Technology Handbook, Second Edition*. CRC Press.

ANEXOS

ANEXO 1

Código de Python para leer los parámetros vía Modbus TCP IP a la DCU 410E y enviarlos en forma de JSON a Node Red vía MQTT

```
import time
import json
import paho.mqtt.publish as publish
from pymodbus.client import ModbusTcpClient
from pymodbus.exceptions import ConnectionException

# Configuración de MQTT
BROKER_IP = "192.168.0.301"
BROKER_PORT = 1883
TOPIC = "/camila3/mpbb"

# Configuración Modbus TCP/IP
IP_ADDRESS = '192.168.0.101'
PORT = 502
SLAVE_ID = 5
PARAMETER_ADDRESS = 41610
COUNT = 22 # 11 parámetros * 2 registros por parámetro

# Crear cliente Modbus TCP/IP
client = ModbusTcpClient(IP_ADDRESS, PORT)

def read_modbus_values(client, address, count):
    try:
        result = client.read_holding_registers(address, count=count,
unit=SLAVE_ID)
        if result.isError():
            print(f"Error al leer los parámetros Modbus: {result}")
            return None
        if len(result.registers) < count:
            print("Número insuficiente de registros recibidos.")
            return None

        values = [(result.registers[i] << 16) + result.registers[i +
1] for i in range(0, count, 2)]
        print("Lectura exitosa de los valores Modbus.")
        print("Longitud de la lista de valores:", len(values))
        print("Datos leídos:", values)
        return values
    except IndexError as e:
        print(f"Error al procesar los registros: {e}")
        return None

def main():
    while True:
        try:
```

```

if not client.is_socket_open():
    client.connect()
    print("Conexión Modbus establecida.")

    modbus_values = read_modbus_values(client,
PARAMETER_ADDRESS, COUNT)
    if modbus_values:
        valores = {
            "temp_ref": modbus_values[0],
            "press_fuel": modbus_values[1],
            "con_fuel": modbus_values[2],
            "press_gear": modbus_values[3],
            "vel": modbus_values[4],
            "press_oil": modbus_values[5],
            "temp_fuel": modbus_values[6],
            "carga": modbus_values[7],
            "tempe_esc": modbus_values[8],
            "temp_gear": modbus_values[9],
            "diff_filt": modbus_values[10]
        }
        mensaje_json = json.dumps(valores)
        publish.single(TOPIC, mensaje_json,
hostname=BROKER_IP, port=BROKER_PORT)
        print("Datos enviados a", TOPIC, ":", mensaje_json)

        time.sleep(5)

except ConnectionException as e:
    print(f"Error de conexión Modbus, reintentando: {e}")
    client.close()
    time.sleep(5)

except Exception as e:
    print(f"Error inesperado: {e}")
    time.sleep(5)

if __name__ == "__main__":
    main()

```

ANEXO 2:

Configuración de la base de datos InfluxDB en Grafana

The screenshot shows the InfluxDB configuration page in Grafana. The interface is dark-themed and includes the following sections:

- influxdb** logo and "Type: InfluxDB" label.
- Settings** tab.
- Name:** "influxdb" (Default is checked).
- Query language:** "InfluxQL" (dropdown).
- HTTP** section:
 - URL:** "http://192.168.100.125:8086"
 - Allowed cookies:** "New tag (enter key to add)" (Add button)
 - Timeout:** "10"
- Auth** section:
 - Basic auth:** **With Credentials:**
 - TLS Client Auth:** **With CA Cert:**
 - Skip TLS Verify:**
 - Forward OAuth Identity:**
- Custom HTTP Headers:** "+ Add header" button.
- InfluxDB Details** section:
 - Database Access:** "Setting the database for this datasource does not deny access to other databases. The InfluxDB query syntax allows switching the database. To support data isolation and security, make sure appropriate permissions are configured in InfluxDB."
 - Database:** "motor_babor_Camila3"
 - User:** (empty)
 - Password:** "configured" (Reset button)
 - HTTP Method:** "POST" (dropdown)
 - Min time interval:** "1"
 - Max series:** "1000"
- Buttons:** "Delete" (red) and "Save & test" (blue).

ANEXO 3

Tiempo transcurrido desde que se envía una solicitud Modbus TCP/IP hasta obtener una respuesta correcta, recopilados en una hora cada 5 segundos

Tabla 8

Datos de muestreo del tiempo de respuesta de la solicitud Modbus

	1	49	97	145	193	241	289	337	385	433	481	529	577	625	673
1	100.52	26.30	3000.8	39.21	86.87	1349.9	6.19	102.00	142.23	105.57	51.73	70.00	120.07	56.75	126.76
2	50.23	43.57	108.19	58.46	91.47	175.91	5.79	98.43	9.81	1456.5	55.14	49.93	117.25	28.02	39.93
3	58.63	1220.0	93.48	134.69	107.76	160.22	21.20	72.69	28.79	68.08	103.46	105.43	41.53	48.46	77.56
4	101.10	105.34	1575.3	1104.7	1446.1	67.24	47.44	110.72	106.92	48.59	75.35	23.13	62.31	108.84	26.49
5	22.73	1379.3	25.12	18.97	12.25	40.77	174.16	107.36	82.81	1582.8	51.88	45.37	30.43	59.06	60.10
6	3004.3	80.14	14.92	1523.0	171.10	6.94	6.63	99.66	83.47	21.02	58.58	54.47	114.59	9.30	72.82
7	419.64	91.12	3002.4	17.54	3001.8	30.44	15.98	105.77	312.84	62.89	71.26	58.66	96.11	52.63	68.49
8	95.29	10.80	61.37	44.31	347.23	147.22	22.50	92.78	75.18	5.28	61.90	15.99	73.59	18.60	73.16
9	85.48	25.84	70.21	38.77	21.88	23.69	102.63	98.08	80.09	18.00	55.05	61.47	79.44	101.71	56.74
10	193.78	69.73	1243.6	64.14	42.61	21.23	26.87	110.13	96.11	1366.2	188.33	41.69	42.52	88.37	99.89
11	100.32	79.01	61.45	152.87	119.94	1247.3	103.32	39.12	293.04	23.67	54.72	25.41	39.46	12.31	29.57
12	91.03	63.25	27.59	5.74	77.98	1342.4	88.35	196.89	20.97	109.85	62.28	32.46	5.55	5.72	5.45
13	3001.7	214.72	1945.6	95.96	1702.2	32.05	64.09	19.37	145.35	115.85	47.48	44.72	29.04	77.68	17.40
14	134.25	78.37	3013.7	167.15	109.32	117.38	111.65	69.12	138.02	25.41	39.98	20.65	23.12	116.94	5.33
15	130.27	6.49	1605.3	66.99	134.44	21.32	25.61	110.30	32.42	92.80	60.05	51.47	100.17	112.49	57.23
16	86.71	33.23	74.45	3003.7	138.12	98.05	116.26	96.57	1308.0	74.92	7.50	109.60	87.04	264.45	78.94
17	32.15	7.40	70.36	23.90	106.23	18.31	110.57	85.39	1444.0	80.12	125.85	41.05	5.02	54.68	4.91
18	89.60	3004.7	75.14	25.52	58.84	37.88	94.78	96.98	5.21	84.77	1722.8	49.35	32.11	21.77	69.69
19	58.42	40.02	3003.8	4.75	97.63	32.75	7.47	9.00	8.62	47.08	118.84	1502.8	119.20	105.53	69.26
20	75.20	1451.1	65.38	64.66	46.33	23.91	109.64	77.58	100.77	1310.4	126.97	89.98	118.82	1556.3	56.79
21	138.92	1467.7	10.02	3003.5	47.70	34.73	1447.0	89.61	57.17	114.12	129.54	9.89	233.49	1210.2	95.00
22	1405.7	92.65	1296.2	2030.4	28.72	1155.0	78.10	70.56	87.85	70.20	19.37	30.24	11.70	52.31	39.00
23	117.75	71.77	1424.6	138.25	228.71	93.42	7.35	96.83	88.56	57.15	25.20	23.71	176.89	103.60	78.88
24	1523.9	66.68	8.74	7.76	1278.1	5.61	116.02	158.15	8.61	62.37	58.11	63.53	111.36	73.92	51.00
25	6.78	80.57	1552.5	1258.4	61.13	134.08	106.96	103.50	77.20	113.41	26.48	40.39	120.68	54.60	22.24
26	80.67	113.19	1485.9	1486.3	130.03	107.80	127.44	96.63	90.36	79.73	77.64	25.72	41.68	111.61	58.87
27	79.22	8.49	40.93	1607.7	1188.5	1094.3	83.43	129.27	6.27	76.31	80.30	5.87	55.75	132.76	45.34
28	3005.5	1505.5	121.84	100.37	103.98	426.66	8.07	92.80	28.61	174.24	9.19	90.01	80.68	6.37	20.68
29	47.85	40.80	1229.1	50.98	1389.4	117.33	83.50	71.11	127.15	7.15	143.17	33.05	7.33	47.01	81.85
30	282.77	78.10	29.13	1703.3	1374.6	18.93	5.63	100.03	7.62	103.61	52.11	38.81	85.22	64.32	1403.5
31	97.61	83.14	7.69	88.55	1538.4	25.53	152.27	65.68	1224.3	80.58	51.86	54.40	78.00	43.58	1197.3
32	1343.5	70.91	41.20	1278.4	1421.7	1504.6	35.36	107.42	6.00	5.21	65.50	55.66	71.03	83.28	5.04
33	1468.1	74.99	11.20	33.20	68.12	25.33	11.90	37.07	91.85	27.40	128.80	1512.4	23.05	4.77	26.22
34	8.38	70.25	89.27	10.35	1454.2	1892.5	99.21	110.20	81.71	68.98	56.67	10.59	73.36	56.13	77.57
35	75.88	63.39	1356.3	75.01	35.08	1349.4	12.43	1273.7	12.56	74.65	62.18	121.22	10.70	89.69	188.34
36	210.91	73.32	11.52	22.56	8.96	67.14	19.65	5.09	6.00	24.05	56.74	1051.1	116.32	56.49	46.96
37	45.62	67.56	54.25	3004.2	114.09	1232.7	48.80	1361.4	94.62	55.49	6.14	5.64	114.69	119.85	1135.8
38	77.62	12.03	9.60	22.35	16.27	1425.2	47.92	65.02	36.80	78.98	112.25	7.95	26.76	90.27	76.85
39	153.69	88.95	149.64	39.13	40.86	113.41	85.57	208.24	52.22	58.81	119.05	23.01	39.89	100.95	144.05
40	183.00	71.63	17.45	1380.6	13.92	102.47	103.22	46.10	87.39	57.08	56.61	21.29	32.23	1283.5	48.09
41	110.30	122.38	142.45	3002.5	120.01	1098.5	99.99	8.74	38.81	41.26	8.08	50.97	120.78	67.12	117.78
42	85.53	52.78	13.09	3002.1	26.28	87.78	104.40	79.40	68.53	115.27	81.30	46.14	25.95	327.66	78.09
43	85.82	217.18	10.36	3002.9	36.98	120.69	116.28	92.00	7.37	33.40	18.21	222.58	45.82	49.45	52.63
44	94.28	106.28	120.92	31.36	25.17	84.10	83.63	100.48	1494.8	72.90	48.67	41.49	29.63	52.61	18.60
45	1429.2	10.37	41.44	123.68	115.18	9.03	1432.4	93.24	37.22	70.07	66.30	11.85	1568.1	1245.4	101.71
46	53.16	119.26	6.46	4041.0	1188.2	44.20	5.43	48.22	5.88	36.89	41.97	54.84	52.13	3003.5	88.37
47	77.46	86.01	53.71	56.99	7.86	103.54	1400.7	285.96	1208.3	72.84	36.56	110.19	27.42	88.54	12.31
48	54.74	3001.4	59.98	71.11	19.64	55.54	20.46	89.82	1383.9	57.28	18.82	82.52	65.10	103.62	5.72

ANEXO 4

Alerta de sobrecarga en estado normal

The screenshot displays the Grafana alerting configuration page for a rule named "Sobrecarga". The rule is currently in a "Normal" state. The interface includes a top navigation bar with "1 rule 1 normal" and "Grafana" labels. The main content area is divided into several sections: "State" (Normal, Silence, Show state history), "Name" (Sobrecarga), "Health" (ok), "Summary", "Next evaluation" (in a few seconds), "Actions" (More), "Evaluate" (Every 1m), "Pending period" (1m), "Last evaluation" (a minute ago), "Evaluation time" (0s), "Labels" (tipo: grave), "Matching instances" (1 normal), and "Data source" (Influxdb). A "Labels" section at the bottom shows a tree structure: alertname: Sobrecarga, grafana_folder: AlertaSobrecarga, tipo: grave. The "Created" timestamp is 2024-05-02 23:14:00.

Section	Value
State	Normal
Name	Sobrecarga
Health	ok
Next evaluation	in a few seconds
Evaluate	Every 1m
Pending period	1m
Last evaluation	a minute ago
Evaluation time	0s
Labels	tipo: grave
Matching instances	1 normal
Created	2024-05-02 23:14:00

ANEXO 5

Alerta de sobrecarga en estado pendiente

The screenshot displays the Grafana alerting configuration page for a rule named 'Sobrecarga'. The rule is currently in a 'Pending' state. The configuration includes a 1-minute evaluation interval, a 1-minute pending period, and an 'Influxdb' data source. The alert is currently 'ok' in health. The labels for the alert are 'alertname: Sobrecarga', 'grafana_folder: AlertaSobrecarga', and 'tipo: grave'. The alert was created on 2024-05-02 at 23:19:00.

Alerting Rule Configuration:

- Name:** Sobrecarga
- State:** Pending
- Health:** ok
- Next evaluation:** within 1 minute
- Evaluate:** Every 1m
- Pending period:** 1m
- Last evaluation:** a minute ago
- Evaluation time:** 0s
- Labels:** tipo: grave
- Matching instances:** 1 pending
- Data source:** Influxdb
- Created:** 2024-05-02 23:19:00

Alert Labels:

- alertname: Sobrecarga
- grafana_folder: AlertaSobrecarga
- tipo: grave

ANEXO 6

Alerta de sobrecarga activada

The screenshot displays the Grafana Alerting configuration page for a rule named 'AlertaSobrecarga'. The interface is dark-themed and includes several sections:

- Header:** Shows '1 rule 1 firing' and 'Export rules'.
- Navigation:** Includes '1 firing', '1m', and 'More' buttons.
- Summary:** Displays 'Name: Sobrecarga', 'Health: ok', and 'Next evaluation: In a few seconds'.
- Configuration:**
 - State:** 'Firing for 39s' with a 'Show state history' link.
 - Evaluate:** 'Every 1m'.
 - Pending period:** '1m'.
 - Last evaluation:** 'a minute ago'.
 - Evaluation time:** '0s'.
 - Labels:** 'tipo: grave'.
 - Matching Instances:** '1 firing'.
- Data source:** 'Influxdb'.
- Labels:** A list of labels: 'Alerting', 'alertname: Sobrecarga', 'grafana_folder: AlertaSobrecarga', and 'tipo: grave'.
- Created:** '2024-05-02 23:20:00'.

ANEXO 7

Evidencia del comportamiento de cada variable en el tiempo

