



UNIVERSIDAD TÉCNICA  
FEDERICO SANTA MARÍA

# **MONITOREO REMOTO DE VARIABLES PARA EL MANTENIMIENTO BASADO EN LA CONDICIÓN DE COCEDORES EN LA INDUSTRIA PESQUERA**

Trabajo de Titulación para optar al  
título profesional de Ing. de (E) en control  
e Instrumentación industrial.

Alumnos:

Sr. Sebastián Romero Bustos.

Sr. César Fuentes Cid.

Profesor guía:

Sr. Luis Muñoz



## CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

### 1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

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

**Título del trabajo:** Monitoreo remoto de variables para el mantenimiento basado en la condición de cocedores en la industria pesquera

**Nombre del candidato(a):** César Ignacio Fuentes Cid

**Carrera / Grado:** Ingeniería de ejecución en control e instrumentación industrial

**Campus:** Sede Concepción **Departamento:** Departamento electrónica e informática

### 2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Luis Muñoz Quezada, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

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

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

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

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

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

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

---

---

---

### 4.- FIRMAS

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

**Fecha:** 13-03-2026 **Firma:** \_\_\_\_\_

**Estudiante o Candidato(a):**

**Fecha:** 04-03-2026

**Firma:** \_\_\_\_\_

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



## CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

### 1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

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

**Título del trabajo:** Monitoreo remoto de variables para el mantenimiento basado en la condición de cocedores en la industria pesquera

**Nombre del candidato(a):** Sebastián Romero Bustos

**Carrera / Grado:** Ingeniería de ejecución en control e instrumentación industrial

**Campus:** Sede Concepción **Departamento:** Departamento electrónica e informática

### 2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Luis Muñoz Quezada, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

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

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

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

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

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

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

---

---

---

### 4.- FIRMAS

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

**Fecha:** 13-03-2026 **Firma:** 

**Estudiante o Candidato(a):**

**Fecha:** 04-03-2026

**Firma:** 

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



INTRODUCCIÓN .....	9
1. CAPÍTULO : ANTECEDENTES .....	12
<b>1.1. Industria pesquera y sus desafíos .....</b>	<b>12</b>
1.1.1. Impacto económico .....	12
1.1.2. Retos principales.....	12
1.1.2.1. Eficiencia operativa.....	12
1.1.2.2. Sostenibilidad.....	12
1.1.2.3. Monitoreo en tiempo real de los procesos.....	13
<b>1.2. Tecnologías emergentes en la industria.....</b>	<b>13</b>
1.2.1. Computación en la nube y gestión de datos .....	14
1.2.2. IoT e IIoT .....	15
1.2.2.1. protocolos de comunicación en IoT e IIoT .....	15
1.2.2.2. Integración de servicios en la nube con IoT.....	16
1.2.2.3. Mejora de la visibilidad en los procesos de producción .....	16
1.2.2.4. Mejora de la eficiencia y la toma de decisiones .....	16
<b>1.3. Modelo de equipo industrial para la integración del proyecto.....</b>	<b>17</b>
1.3.1. COCEDOR C60-CSD1750C .....	17
1.3.2. Especificaciones técnicas: .....	17
1.3.3. Funcionamiento del Cocedor .....	18
<b>1.4. Identificación de variables de proceso.....</b>	<b>19</b>
1.4.1. Desglose y detalle del funcionamiento .....	19
1.4.2. Variables específicas que permiten realizar análisis del estado del proceso.....	20
<b>1.5. Mantenimiento de equipos .....</b>	<b>21</b>
1.5.1. Tipos de mantenimiento .....	21
1.5.1.1. Mantenimiento preventivo.....	21
1.5.1.2. Mantenimiento predeterminado.....	22
1.5.1.3. Mantenimiento basado en la condición .....	22
1.5.1.4. Mantenimiento continuo.....	22
1.5.1.5. Mantenimiento predictivo .....	22
1.5.1.6. Mantenimiento correctivo.....	22
1.5.1. Consecuencia del descuido del mantenimiento de los equipos .....	22
2. CAPÍTULO : MANTENIMIENTO BASADO EN LA CONDICIÓN .....	23
<b>2.1. Aplicación de mantenimiento basado en la condición a la cultura empresarial .....</b>	<b>23</b>
<b>2.2. KPI Y METRICAS DE MANTENIMIENTO.....</b>	<b>24</b>
2.2.1. IMPORTANCIA DE LOS KPI .....	25
2.2.2. TOMA DE DECISIONES BASADAS EN DATOS .....	25
2.2.3. MEJORA CONTINUA.....	26
2.2.4. IMPORTANCIA DE LOS KPI EN PROCESOS PRODUCTIVOS.....	28
2.2.5. INDICADORES ADECUADOS PARA EL MANTENIMIENTO .....	29
2.2.5.1. OEE (Overall Equipment Efectiveness) .....	29
2.2.5.2. MTTR (Mean Time To Repair) .....	30



2.2.5.3. MTTF (Mean Time to Failure) .....	30
2.2.5.4. MTBF (Mean Time Between Failures).....	30
2.2.5.5. Disponibilidad y confiabilidad .....	30
2.2.5.6. Backlog (Acomulacion de trabajos pendientes).....	31
3. CAPITULO: TECNOLOGIA DE INFORMACION Y OPERACIÓN (IT/OT).....	32
<b>3.1. Estructura metodologica para implementar el sistema de monitoreo .....</b>	<b>32</b>
3.1.1. Variables asociadas al equipo.....	33
3.1.1.1. Temperatura .....	33
3.1.1.2. Corriente en el tablero.....	33
3.1.1.3. Presion .....	34
3.1.1.4. Vibraciones .....	35
3.1.2. Requerimientos de internet (Wi-Fi o Ethernet).....	36
3.1.2.1. Factores.....	36
3.1.3. Determinacion de KPIs de mantenimiento .....	37
3.1.3.1. MTTR (Mean Time To Repair) .....	37
3.1.3.2. MTTF (Mean Time to Failure) .....	37
3.1.3.3. MTBF .....	37
<b>3.2. Selección de tecnologías .....</b>	<b>38</b>
3.2.1. Bróker HiveMQ .....	38
3.2.1.1. Características Clave de HiveMQ .....	38
3.2.1.2. EDICIONES DE HIVE MQ.....	39
<b>3.3. RASPBERRY PI4 .....</b>	<b>41</b>
3.3.1. Especificaciones tecnicas de Rasperry PI4 .....	41
3.3.2. Sistema operativo y software .....	43
3.3.2.1. Caracteristicas de ubuntu .....	43
<b>3.4. Modulos DAQ .....</b>	<b>44</b>
3.4.1. MODULO DAQ MCC 128 .....	45
3.4.2. MODULO DAQ MCC 134 PARA MEDICION DE TEMPERATURA.....	45
3.4.3. MODULO DAQ MCC 172 .....	46
<b>3.5. MONGO DB .....</b>	<b>49</b>
3.5.1. Estructura de almacenamiento en Mongo DB.....	51
3.5.1.1. Base de datos (Database) .....	51
3.5.1.2. Colección(collection).....	51
3.5.1.3. Documentos (Document).....	51
<b>3.6. Grafana.....</b>	<b>52</b>
<b>3.7. BIBLIOTECAS Y DEPENDENCIAS .....</b>	<b>52</b>
<b>3.8. Diseño de sistema.....</b>	<b>53</b>
3.8.1. Arquitectura de hardware .....	53
3.8.2. Arquitectura de software.....	54
4. CAPÍTULO : PROTOCOLO MQTT.....	55
<b>4.1. DEFINICIÓN DE MQTT .....</b>	<b>55</b>



<b>4.2. RAZONES DE POR QUE EMPRESAS UTILIZAN MQTT .....</b>	<b>56</b>
4.2.1. <i>Practicas de seguridad en MQTT .....</i>	56
<b>4.3. FUNCIONAMIENTO Y CONCEPTOS .....</b>	<b>57</b>
4.3.1. <i>Cientes MQTT.....</i>	57
4.3.2. <i>Broker MQTT.....</i>	58
4.3.2.1. <i>Pasos para comunicación broker MQTT con el cliente MQTT .....</i>	58
4.3.3. <i>Topicos, suscripciones, QoS y sesiones persistentes .....</i>	58
4.3.3.1. <i>Modelo de mensajería MQTT: temas y suscripciones .....</i>	58
4.3.4. <i>Descripcion de los niveles de calidad de servicio (QoS).....</i>	59
4.3.5. <i>Persistencia de mensajes para la comunicación IoT confiable .....</i>	60
4.3.6. <i>Arquitectura de publicación/suscripción de MQTT (pub/sub) .....</i>	61
4.3.6.1. <i>Funcion de filtrado de mensajes .....</i>	62
<b>4.4. ESTRUCTURA Y FORMATO DE PAQUETES MQTT .....</b>	<b>62</b>
4.4.1. <i>PAQUETES DE CONTROL MQTT .....</i>	65
4.4.1.1. <i>MQTT CONNECT y MQTT CONNACK.....</i>	65
4.4.1.2. <i>MQTT PUBLISH.....</i>	66
4.4.1.3. <i>MQTT SUBSCRIBE, UNSUBSCRIBE, SUBACK Y UNSUBACK .....</i>	66
4.4.1.4. <i>MQTT DISCONNECT .....</i>	66
<b>5. ADQUISICION REGISTRO Y SUPERVISION DE DATOS .....</b>	<b>67</b>
<b>5.1. CONSIDERACIONES PARA LA APLICACION .....</b>	<b>67</b>
5.1.1. <i>Teorema de Nyquist-Shannon.....</i>	67
<b>5.2. PUNTOS DE MEDICION .....</b>	<b>67</b>
5.2.1.1. <i>Termocuplas .....</i>	67
5.2.1.2. <i>Puntos clave para la medición .....</i>	68
5.2.1.3. <i>Determinación de tiempo de muestreo .....</i>	68
5.2.2. <i>Sensores de presión .....</i>	69
5.2.3. <i>Vibraciones .....</i>	70
5.2.3.1. <i>Principales Estándares y Guías de Vibración.....</i>	70
5.2.3.2. <i>Valores de vibración aceptables según ISO 10816: .....</i>	71
5.2.3.3. <i>Valores típicos según la frecuencia de vibración: .....</i>	72
5.2.3.4. <i>Tipos de defectos que las vibraciones pueden detectar:.....</i>	72
5.2.4. <i>CORRIENTE.....</i>	73
<b>5.3. ESPECIFICACIONES TECNICAS DE LOS SENSORES.....</b>	<b>73</b>
5.3.1. <i>Sensores de temperatura.....</i>	73
5.3.2. <i>sensor de presión WIKA A-10.....</i>	74
5.3.3. <i>Sensor de corriente .....</i>	75
5.3.4. <i>Sensor de vibración .....</i>	75
<b>5.4. FLUJO DE DATOS.....</b>	<b>76</b>
5.4.1. <i>Adquisición de Datos: .....</i>	76
5.4.1.1. <i>Habilitar puerto GPIO e interfaz I2C y SPI .....</i>	76
5.4.1.2. <i>Instalación de DAQ HAT, librerías y código de lectura de variables .....</i>	77
5.4.2. <i>Transmisión de datos.....</i>	78



5.4.3. <i>Procesamiento de datos</i> .....	80
5.4.3.1. <i>Procesamiento de datos Node-RED</i> .....	80
5.4.3.2. <i>KPIs de mantenimiento en Python</i> .....	81
5.4.4. <i>Almacenamiento de datos</i> .....	84
5.4.5. <i>Visualización y monitoreo</i> .....	87
5.4.5.1. <i>Instalación de fuente de datos</i> .....	87
5.4.5.2. <i>Arquitectura de Dashboard</i> .....	89
5.4.5.3. <i>Consultas y transformación de datos para MongoDB</i> .....	90
5.4.5.4. <i>Configuración de alarmas y notificaciones</i> .....	95
<b>5.5. Puertos y enlaces para acceso a plataformas.</b> .....	<b>99</b>
6. <i>Conclusión</i> .....	100
7. <i>Bibliografías</i> .....	101



## índice imágenes

<i>Figura 1 Cocedor C60-CSD1750C</i>	17
<i>Figura 2 . COCEDOR C60-CSD1750C PERFIL 1</i>	18
<i>Figura 3 COCEDOR C60-CSD1750C PERFIL 2</i>	18
<i>Figura 4 Tipos de mantenimiento industrial según la norma EN13306 2010</i>	21
<i>Figura 5 KPIs de mantenimiento</i>	28
<i>Figura 6 métricas de conexión, volumen de datos e insertos por segundo</i>	36
<i>Figura 7 Detalle de volumen de datos en la red</i>	37
<i>Figura 8:Ediciones de HIVEMQ</i>	39
<i>Figura 9 RED HIVEMQ</i>	40
<i>Figura 10 Servicios HIVEMQ</i>	40
<i>Figura 11 Raspberry PI 4 B</i>	42
<i>Figura 12 Diagrama de bloques MCC 128</i>	45
<i>Figura 13 Diagrama de bloques MCC 134</i>	46
<i>Figura 14 Diagrama de bloques MCC 172</i>	47
<i>Figura 15 Modulo DAQ HAT MCC 172</i>	48
<i>Figura 16 APILAMIENTO DE DAQ HATs en Raspberry PI</i>	48
<i>Figura 17 Modelo de datos en MongoDB Compass</i>	49
<i>Figura 18 Arquitectura de Hardware</i>	53
<i>Figura 19 Arquitectura software</i>	54
<i>Figura 20 Protocolo MQTT</i>	55
<i>Figura 21 Temas y suscripciones MQTT</i>	58
<i>Figura 22 Estructura y formato de paquete MQTT</i>	62
<i>Figura 23 Control header structure</i>	63
<i>Figura 24 Encabezado de variable</i>	63
<i>Figura 25 Ejemplo paquete MQTT</i>	64
<i>Figura 26 Paquetes de control</i>	65
<i>Figura 27 Habilitacion de puertos en ubuntu</i>	77
<i>Figura 28 Suscripción Node-red a HIVE MQ</i>	78
<i>Figura 29 suscripción Node-red a HIVE MQ 2</i>	79
<i>Figura 30 Enlace de datos Node-red</i>	79
<i>Figura 31 Procesamiento de datos 1</i>	80
<i>Figura 32 Procesamiento de datos 2</i>	81
<i>Figura 33 Código para KPIs</i>	82
<i>Figura 34 Código para KPIs 2</i>	83
<i>Figura 35 KPIs de mantenimiento</i>	83
<i>Figura 36 configuración de nodo MONGODB3</i>	84
<i>Figura 37 configuración de nodo MONGODB3 2</i>	85
<i>Figura 38 Ingreso de cadena de conexión desde mongodb compass</i>	85
<i>Figura 39 Documentos en base de datos "test" colección "corriente" en mongo db atlas</i>	86
<i>Figura 40 Documentos en la base de datos "test" colección "presión" en mongo db compass</i>	86
<i>Figura 41 configuración de base de datos para grafana</i>	88
<i>Figura 42 Ajustes adicionales base de datos grafana</i>	89
<i>Figura 43 Dashboard grafana</i>	89
<i>Figura 44 Ajuste de rango tiempo para visualizar los datos en el panel</i>	90
<i>Figura 45 Operación find para buscar documentos dentro de la base de datos</i>	90
<i>Figura 46 Funciones transformación</i>	91
<i>Figura 47 Funciones de transformación</i>	94
<i>Figura 48 configuración canal de notificación</i>	95
<i>Figura 49 Consulta y condición de la alerta</i>	96
<i>Figura 50 configuración de evaluación inmediata y mensaje de notificación</i>	97
<i>Figura 51 estado de alerta de métrica de corriente</i>	98



<i>Figura 52 Notificación al correo electrónico de alerta para métrica de corriente.</i>	98
<i>Figura 53 Separador raspberry</i>	102
<i>Figura 54 Receptáculo 2*20</i>	102
<i>Figura 55 Puentes DAQ HAT</i>	103
<i>Figura 56 Código lectura de datos 1</i>	106
<i>Figura 57 Código lectura de datos 2</i>	107
<i>Figura 58 Código lectura de datos 3</i>	107

## Índice de tablas

<i>Tabla 1 COMPARATIVA MODULOS DAQ HAT</i>	48
<i>Tabla 2 COMPARATIVA MONGODB ATLAS MONGODB COMPASS</i>	50
<i>Tabla 3 PUNTOS DE MEDICION DE TEMPERATURA</i>	69
<i>Tabla 4 CARACTERISTICAS DEL TRANSMISOR DE PRESION A-10</i>	69
<i>Tabla 5 PUNTOS DE MEDICION DE PRESION</i>	69
<i>Tabla 6 VALORES DE VIBRACION ACEPTABLES SEGÚN ISO 10816</i>	71
<i>Tabla 7 PUNTOS DE MEDICION DE VIBRACION</i>	72
<i>Tabla 8 PUNTOS DE MEDICION DE CORRIENTE</i>	73

## Siglas y simbología

KPI: Indicadores Clave de Desempeño

IoT: el internet de las cosas

IaaS: infraestructura como servicio

PaaS: plataforma como servicio

SaaS: software como servicio

DBaaS: base de datos como servicio

MQTT: Message Queuing Telemetry Transport

CoAP: Constrained Application Protocol

LoRaWAN: Long Range Wide Area Network

BLE: Bluetooth Low Energy

CBM: mantenimiento basado en la condición

RCM: mantenimiento centrado en la confiabilidad

KPIs: indicadores clave de rendimiento

Data-Driven Decision Making, DDDM: La toma de decisiones basada en datos

GMAO: Sistema de gestión del mantenimiento

OEE: Medición de la eficiencia general

MTTR: Reducción del tiempo de inactividad

MTTF: Tiempo medio hasta la falla

MTBF: Tiempo medio entre fallas



TPM: Mantenimiento de productividad total

IT: Tecnología de la Información

OT: Tecnología de Operaciones

SCADA: Supervisory Control and Data Acquisition

PLC: Programmable Logic Controllers

DAQ: adquisición de datos

HAT: hardware conectado en la parte superior

MCC: corporación de computación de información.

ADC: convertidor análogo digital.

I2C: Protocolo serie circuito Inter integrado

SPI: protocolo interfaz periférica serial

GNOME: Entorno de modelos de objetos de red GNU

UFW: Cortafuegos descomplicados.

LTS: Soporte a largo plazo.

EEPROM: Memoria de solo lectura programable y borrable electricamente

AWS: servicios web de amazon.

GPIO: Entradas y salidas de proposito general.

TC: Termopares diferenciales

CJC: compensación de unión fría.

OTD: Detección de termopar abierto

IEPE: electrónica integrada piezoeléctrica

TTL: Lógica de transistor-transistor

CMOS: semiconductor complementario de óxido metálico.

SQL: Lenguaje de consulta estructurado.

MQTT: Transporte de telemetría mediante cola de mensajes

TLS: Seguridad de la capa de transporte.

SSL: Capa de conexión segura.

QoS: Calidad de servicio

ISO: Organización internacional de normalización



## ***INTRODUCCIÓN***

En la industria, el monitoreo eficiente y en tiempo real de los equipos es esencial para mejorar su rendimiento, reducir costos operativos y garantizar la calidad del producto. Este proyecto tiene como objetivo desarrollar un sistema de monitoreo remoto de variables críticas para un equipo cocedor en la industria pesquera, utilizando tecnología de operación e información avanzadas que permitan la recolección, análisis y visualización de datos de manera continua y precisa. En este trabajo de título se propone un sistema que se basa en una arquitectura que combina dispositivos de bajo costo como la Raspberry PI, complementados con placas DAQ HAT MCC para la adquisición de datos. Estos datos serán procesados y gestionados mediante código en Python y enviados a través de HiveMQ para su comunicación en tiempo real. Además, se utilizará Node-RED para la creación de flujos de automatización, mientras que los datos recolectados serán almacenados en la nube utilizando MongoDB Atlas para una gestión eficiente y escalable de la información.

Para la visualización y análisis de los datos, se integrará grafana Cloud, que proporcionará dashboard interactivos para monitorear las variables críticas del equipo cocedor, con la capacidad de generar alertas en tiempo real y visualizar datos históricos. Además, se incorporarán indicadores KPI específicos de mantenimiento, que permitan evaluar la eficiencia de los equipos y detectar posibles fallas antes de que se conviertan en problemas mayores.

Este enfoque integrado no solo mejorara la visibilidad y el control sobre las operaciones de la planta pesquera, sino que también proporcionara información valiosa para la toma de decisiones, optimizando el mantenimiento predictivo y contribuyendo a la mejora continua de los procesos productivos.



## **OBJETIVOS**

### ***Objetivo general:***

Desarrollar un sistema de monitoreo para las variables clave del proceso productivo en los equipos cocedores de la industria pesquera. Este sistema permitirá anticipar eventos adversos causados por el desgaste o el mal funcionamiento de los componentes basándose en la condición del activo, mejorando la continuidad operacional, la eficiencia y la calidad de la producción. Además, facilitará la respuesta proactiva ante condiciones anormales, facilite el mantenimiento predictivo de los equipos para prevenir fallos y reducir los tiempos de inactividad.

### ***Objetivos específicos:***

- Identificar variables específicas que permitan realizar un análisis del estado del proceso mediante instrumentos de referencia, independientes del sistema de control establecido en el proceso, que permitan comparar señales, validar tendencias operacionales y habilitar el monitoreo basado en condición (MBC).
- Identificar y calcular Indicadores Clave de Desempeño (KPI) relacionados con el mantenimiento, a partir de los datos almacenados.
- Crear un sistema integrado para desarrollar un flujo de datos, que permita la captura eficiente de información mediante dispositivos de adquisición, protocolos de comunicación, servicios en la nube, almacenamiento y visualización.
- Proponer un sistema automatizado para la generación periódica de alertas y reportes de estado, optimizando la toma de decisiones del equipo de mantenimiento mediante información oportuna y precisa.





## **1. CAPÍTULO : ANTECEDENTES**

### **1.1. Industria pesquera y sus desafíos**

Se define industria de la pesca es de gran relevancia tanto para la economía global como para las comunidades locales. Contribuye significativamente a la seguridad alimentaria mundial, provee millones de empleos directos e indirectos y es una fuente importante de proteína para muchas poblaciones, especialmente en regiones costeras y de zonas en desarrollo.

#### **1.1.1. Impacto económico**

El sector pesquero nacional representa alrededor del 0,6% del PIB nacional, contribuyendo significativamente a la economía de países con costas ricas en recursos marinos, generando ingresos por la exportación de productos pesqueros. Además, impulsa industrias relacionadas como la acuicultura, la transformación de productos pesqueros, el transporte marítimo y el turismo. Se estima que alrededor de 200 millones de personas dependen de la pesca a nivel mundial, tanto en su aspecto directo como indirecto, lo que la convierte en una industria clave para millones de familias y regiones enteras.

#### **1.1.2. Retos principales**

La industria pesquera enfrenta varios retos que impactan su eficiencia operativa, sostenibilidad y rentabilidad. Algunos de los principales retos incluyen:

##### **1.1.2.1. Eficiencia operativa**

La eficiencia en las operaciones pesqueras es esencial para maximizar la producción mientras se minimizan los costos. Maximizar la rentabilidad y minimizar el impacto ambiental. Esto implica mejorar los procesos, optimizar el uso de la tecnología, la mano de obra y otros recursos disponibles, manteniendo o mejorando la calidad de los productos o servicios ofrecidos.

##### **1.1.2.2. Sostenibilidad**

La sostenibilidad es un tema de gran importancia ya que implica adoptar prácticas que permitan un equilibrio entre el crecimiento económico, la protección del medio ambiente y el bienestar social. Para asegurar la sostenibilidad, se requiere políticas responsables, la implementación de prácticas que minimicen el impacto ambiental y el uso de tecnologías para reducir la captura incidental de especies no deseadas.



### **1.1.2.3. Monitoreo en tiempo real de los procesos**

Con la llegada de nuevas tecnologías como la inteligencia artificial, el internet de las cosas (IoT) y los satélites, el monitoreo en tiempo real se está convirtiendo en una herramienta clave. Estos sistemas permiten mejorar la eficiencia y dar seguridad y confiabilidad a las operaciones. La industria pesquera enfrenta desafíos como la gestión de recursos, la calidad del pescado y la trazabilidad de los productos, por lo que la implementación de tecnologías para el monitoreo es un factor clave para mejorar estas áreas.

## **1.2. Tecnologías emergentes en la industria**

Los avances y herramientas que se están implementando en los procesos de producción, transformación o distribución tienen como objetivo mejorar la eficiencia, reducir costos, aumentar la calidad, y minimizar el impacto ambiental. El continuo desarrollo y actualización de estas tecnologías a dado lugar a nuevas tendencias, mejorando el flujo de información mediante el análisis y procesamiento en tiempo real. Estas tendencias tecnológicas son impulsadas por la constante innovación y la creciente demanda de mejorar la eficiencia, sostenibilidad y competitividad, reflejando un impacto transformador en el sector industrial, generando cambios profundos en los procesos y modelos de negocio.

Algunas de las principales tendencias son:

- Automatización y robótica
- Inteligencia artificial y machine learning
- Internet de las cosas
- Computación en la nube
- Blockchain y trazabilidad.



### 1.2.1. Computación en la nube y gestión de datos

Cloud computing es una tecnología que permite el acceso remoto a recursos informáticos (servidores, almacenamiento de datos, software, etc.) a través de internet, sin la necesidad de poseer y mantener infraestructura local. En lugar de almacenar datos y ejecutar aplicaciones en servidores locales o computadoras personales, los usuarios pueden hacerlo a través de servidores remotos proporcionados por un proveedor de servicios en la nube.

Principales características de Cloud computing:

- Acceso remoto: Los usuarios pueden acceder a la información y los servicios en la nube desde cualquier lugar y en cualquier dispositivo con conexión a Internet.
- Escalabilidad: permiten aumentar o reducir la capacidad de almacenamiento, procesamiento y recursos según las necesidades del usuario, lo que lo hace altamente flexible.
- Modelo de pago por uso: las empresas pagan solo por los recursos que utilizan, lo que reduce costos y mejora la eficiencia.
- Alta disponibilidad: alta disponibilidad de sistemas, con respaldo y recuperación ante desastres, minimizando el riesgo de pérdida de datos
- Seguridad: se implementa protección de cifrado de datos y autenticación de usuarios, para asegurar integridad y privacidad de la información

Tipos de servicio en la nube:

- IaaS (infraestructura como servicio): Ofrece recursos informáticos básicos como servidores virtuales, redes, almacenamiento, etc.
- PaaS (plataforma como servicio): Proporciona una plataforma completa para desarrollar, ejecutar y gestionar aplicaciones sin tener que preocuparse por la infraestructura subyacente.
- SaaS (software como servicio): Permite acceder a aplicaciones de software a través de la nube, sin necesidad de instalación o mantenimiento en los dispositivos del usuario.
- DBaaS (base de datos como servicio): permite a las empresas utilizar bases de datos sin tener que gestionar la infraestructura y el mantenimiento de estas. En este modelo, el proveedor de servicios en la nube se encarga de la administración, escalabilidad, seguridad y mantenimiento de la base de datos.



## 1.2.2. IoT e IIoT

IoT es el acrónimo de Internet of Things, que en español significa Internet de las Cosas. Se refiere a la red de objetos físicos que se conectan a internet para intercambiar dato

IIoT son las siglas de Internet Industrial de las Cosas. El IIoT es una subcategoría del Internet de las Cosas (IoT). Se utiliza en muchos sectores industriales.

Algunos ejemplos de dispositivos IIoT son: Sensores ambientales, Robots industriales, Dispositivos de registro electrónico.

### 1.2.2.1. protocolos de comunicación en IoT e IIoT

Los protocolos de comunicación en IoT son esenciales para asegurar la interoperabilidad y la eficiencia de los dispositivos conectados. Permiten una comunicación efectiva entre dispositivos de diferentes fabricantes, optimizan el uso del ancho de banda y recursos, y proporcionan seguridad mediante cifrado y autenticación. Además, garantizan la fiabilidad en la transmisión de datos y la flexibilidad para adaptarse a distintas topologías de red y necesidades específicas. Elegir el protocolo adecuado es fundamental para el éxito y escalabilidad de cualquier proyecto IoT, asegurando que los dispositivos puedan interactuar de manera segura y eficiente.

Los principales protocolos de comunicación en IoT e IIoT son :

- MQTT (Message Queuing Telemetry Transport): Protocolo ligero y eficiente, ideal para redes con ancho de banda limitado. Funciona en un modelo de publicación/suscripción y es ampliamente utilizado en aplicaciones de monitoreo remoto y telemetría industrial.
- HTTP/HTTPS: Protocolo basado en la web, fácil de implementar y ampliamente compatible con la mayoría de los dispositivos y plataformas. HTTPS proporciona una capa adicional de seguridad mediante cifrado.
- CoAP (Constrained Application Protocol): Diseñado para dispositivos IoT con limitaciones de recursos. Es similar en estructura a HTTP pero más ligero, y funciona sobre el protocolo UDP, lo que permite un menor consumo de energía.
- LoRaWAN (Long Range Wide Area Network): Permite la comunicación a largas distancias con un bajo consumo de recursos. Es ideal para aplicaciones en agricultura inteligente, seguimiento de activos y ciudades inteligentes.
- Bluetooth Low Energy (BLE): Protocolo de bajo consumo de energía, ideal para comunicación a corta distancia. Es compatible con la mayoría de los dispositivos móviles y gadgets portátiles.



### **1.2.2.2. Integración de servicios en la nube con IoT**

La integración de servicios en la nube con dispositivos IoT en el entorno de plantas de producción permite una captura de datos en tiempo real, los cuales son enviados de manera continua a una plataforma en la nube para su almacenamiento y procesamiento. Estos dispositivos IoT monitorean constantemente el estado de los equipos y la calidad de la producción, facilitando la identificación temprana de posibles fallos y permitiendo la implementación de acciones correctivas inmediatas al detectarse patrones y anomalías en los datos que indican desgaste o fallos inminentes. Esta capacidad de respuesta proactiva mejora la eficiencia operativa y reduce los costos asociados con las reparaciones de emergencia, asegurando un funcionamiento continuo y eficiente de la planta.

### **1.2.2.3. Mejora de la visibilidad en los procesos de producción**

La integración de IoT con la nube mejora significativamente la visibilidad de las variables de los procesos de producción, entregando los siguientes beneficios.

- **Monitoreo en tiempo real:** los supervisores y operadores pueden ver las condiciones de las máquinas, el rendimiento de la producción, y otros indicadores clave en tableros de control(dashboard) accesibles desde cualquier lugar, siempre que tengan acceso a internet. Esto les permite tomar decisiones informadas rápidamente.
- **Alerta y notificaciones:** si un sensor detecta alguna anomalía o fallo de alguna máquina, el sistema puede generar alertas automáticas a los responsables para que actúen de inmediato y eviten problemas mayores.
- **Visibilidad de flujo de trabajo:** Al integrar dispositivos IoT en un sistema centralizado en la nube, los responsables de la planta tienen visibilidad total de flujo de trabajo, desde la entrada de materias primas hasta la salida del producto terminado.

### **1.2.2.4. Mejora de la eficiencia y la toma de decisiones**

La mejora de la eficiencia y la toma de decisiones son dos aspectos clave para cualquier organización que busque optimizar sus operaciones y lograr un rendimiento superior. La tecnología juega un papel crucial en ambos aspectos.

- **Optimización en la producción:** Con el análisis en tiempo real y los modelos predictivos, se pueden hacer ajustes dinámicos en la producción, como reconfigurar máquinas para maximizar su eficiencia o ajustar los tiempos de ciclo de la línea de producción.
- **Mantenimiento predictivo:** La plataforma en la nube puede identificar signos de desgaste o fallos potenciales en las máquinas antes de que ocurran, permitiendo realizar mantenimiento preventivo y evitando costosos tiempos de inactividad no planificados.
- **Toma de decisiones:** Los datos recopilados y procesados permiten una toma de decisiones más objetiva. Los gerentes pueden hacer ajustes a la producción, asignar recursos de manera eficiente y reducir costos basados en información real y actualizada.



### 1.3. Modelo de equipo industrial para la integración del proyecto

Se presenta un modelo de un equipo perteneciente a una línea de producción en la industria pesquera, el cual tiene las características de cocimiento del pescado en donde se ven involucradas diversas variables como temperatura, presión, vibraciones, flujo, corriente, etc.

#### 1.3.1. COCEDOR C60-CSD1750C

Este equipo diseñado principalmente para la industria pesquera, es un equipo de operación continua con calefacción indirecta con una capacidad de proceso de hasta 60 t/h de materia prima, tanto el cilindro como el rotor son calefaccionados con vapor de caldera. El equipo está fabricado en forma de cilindro Horizontal, en donde la materia prima es conducida por un rotor conformado como un tornillo transportador. Las aletas del tornillo son estampadas y han sido especialmente diseñadas de modo de lograr la cocción en un menor tiempo de residencia y, por lo tanto, con una mejor mantención de las cualidades proteicas del producto, además de un diseño que permite una fácil limpieza química de este.

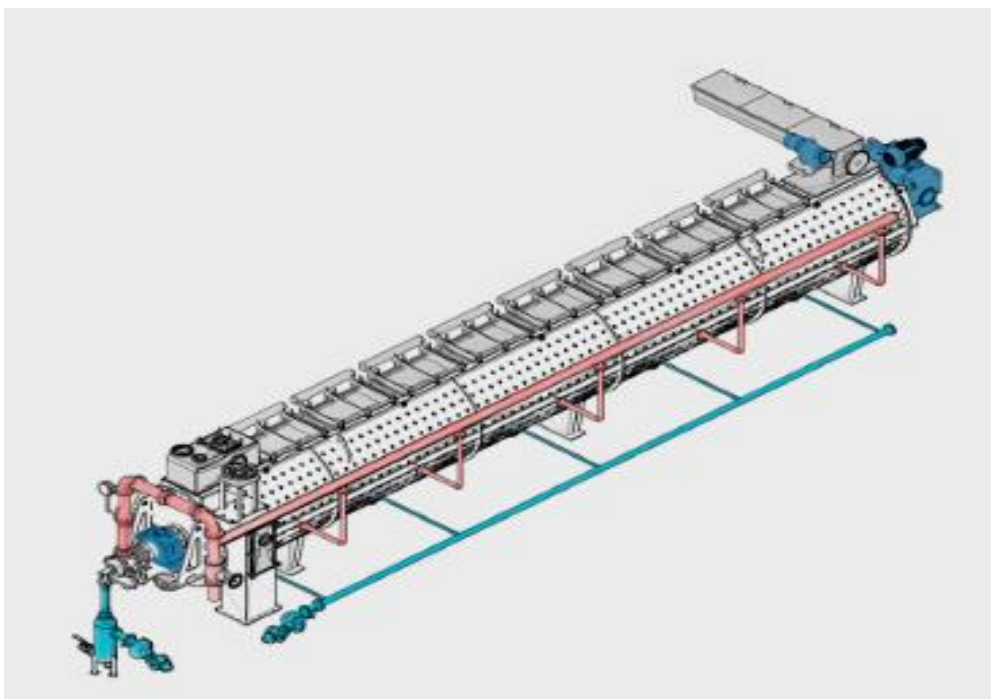


Figura 1 Cocedor C60-CSD1750C

Fuente: Conmetal. Año:2019

#### 1.3.2. Especificaciones técnicas:

- 220 m<sup>2</sup> de área de transferencia de calor (rotor 152,3 m<sup>2</sup>, estator 67,7 m<sup>2</sup>)
- 9200 kg/h consumo de vapor (presión de trabajo 6 bar, presión de prueba 12 bar)
- Capacidad de proceso de 60t/h
- Junta rotatoria de vapor y condensado de 8"
- Sistema motriz con reductor montado en eje con motor de 40 HP
- Largo total 17541 mm ancho total 2672 mm
- Peso total: 46827 kg



### 1.3.3. Funcionamiento del Cocedor

1. La materia prima ingresa por el lado motriz del cocedor y es calentada a lo largo del equipo hasta lograr una buena cocción.
2. Camisas calefaccionadas independientes con posibilidad de alimentación con revalorizado.
3. Evacuación de condensado sin sifón.
4. Junta rotatoria con baja pérdida de carga, se evita el flasheo de vapor al ingresar al equipo.
5. Equipo totalmente calefaccionado, se disminuye la corrosión bacteriana.
6. Permite ser alimentado con bombas lamella o tornillo.
7. Permite control por temperatura.



Figura 2 . COCEDOR C60-CSD1750C PERFIL 1

Fuente: conmetal. Año:2019

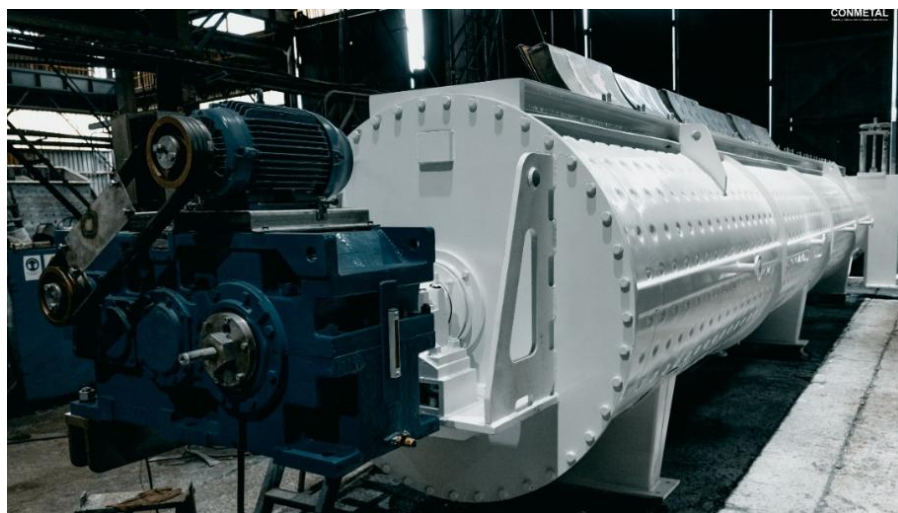


Figura 3 COCEDOR C60-CSD1750C PERFIL 2

Fuente: conmetal. Año: 2019



## 1.4. Identificación de variables de proceso

### 1.4.1. Desglose y detalle del funcionamiento

#### 1. Ingreso y calentamiento de la materia prima:

- La materia prima entra por el lado motriz del cocedor y es transportada a través del equipo.
- Durante su recorrido, se calienta de manera uniforme para lograr una cocción óptima, lo que asegura la desnaturalización de proteínas y la liberación de aceite.

#### 2. Camisas calefaccionadas independientes:

- Estas camisas permiten calentar de manera independiente cada sección del cocedor.
- Admiten alimentación con revalorizado, lo que optimiza el uso de vapor y mejora la eficiencia térmica del equipo.

#### 3. Evacuación de condensado sin sifón:

- Este diseño garantiza una evacuación más eficiente del condensado acumulado, evitando problemas como bloqueos o pérdidas de eficiencia térmica.

#### 4. Junta rotatoria de baja pérdida de carga:

- Esta junta permite la transferencia eficiente de vapor hacia el equipo, minimizando la resistencia al flujo.
- Previene el flasheo de vapor, lo que asegura una distribución térmica más estable y controlada.

#### 5. Calefacción total del equipo:

- Al mantener todo el equipo calefaccionado, se minimiza la corrosión bacteriana en el interior, preservando la higiene del proceso y la calidad del producto final.

#### 6. Flexibilidad de alimentación:

- El diseño permite utilizar sistemas de alimentación con bombas Lamella o transportadores de tornillo, adaptándose a las necesidades operativas.



### 1.4.2. Variables específicas que permiten realizar análisis del estado del proceso

Conocer cuáles son las principales variables de procesos, permitirá medir la eficiencia y las condiciones operativas de un equipo. Estas variables deben compararse con sus valores nominales estandarizados para lograr identificar desviaciones que puedan indicar fallas o pérdidas de rendimiento.

Para los equipos Cocedores, que forman parte de los dispositivos rotativos y cuyo principio de funcionamiento se basa en un eje que gira sobre un centro, transfiriendo este movimiento hacia un resultado final, es fundamental identificar las variables de proceso clave para asegurar su rendimiento óptimo. Entre estas variables se encuentran:

1. Variables de temperatura:
  - Temperatura del vapor en camisas calefaccionadas: Detectar sobrecalentamiento que pueda causar desgaste prematuro en componentes.
  - Temperatura de condensado evacuado: Indicador de eficiencia térmica y posibles problemas en la evacuación.
2. Variables mecánicas del sistema motriz
  - Velocidad del rotor: Identificar fluctuaciones que puedan indicar desalineaciones o problemas en el motor/reductor.
  - Torque aplicado en el reductor: Monitorear el esfuerzo mecánico para prevenir sobrecargas.
  - Vibraciones del sistema motriz: Detectar desbalances o fallos en rodamientos.
3. Variables de presión y flujo:
  - Presión del vapor en la junta rotatoria: Detectar fugas o bloqueos que afecten el suministro.
  - Flujo de condensado: Monitorear obstrucciones en el sistema de evacuación.



## 1.5. Mantenimiento de equipos

En la actualidad los equipos industriales son de suma importancia en las líneas de producción o en los procesos para mantener estos equipos funcionando de manera eficaz permitiendo aumentar los ciclos de trabajo.

El mantenimiento definido como la combinación de todas las acciones técnicas, administrativas y de gestión realizadas durante el ciclo de vida de un elemento destinadas a conservarlo o a devolverlo a un estado en el que pueda desempeñar la función requerida. En virtud de la estandarización por la norma europea EN13306. La norma define los tipos de mantenimiento en el escenario en donde no se presenten cambios en las características intrínsecas en las características de diseño relativas a confiabilidad, mantenibilidad y seguridad de los activos objeto de mantenimiento.

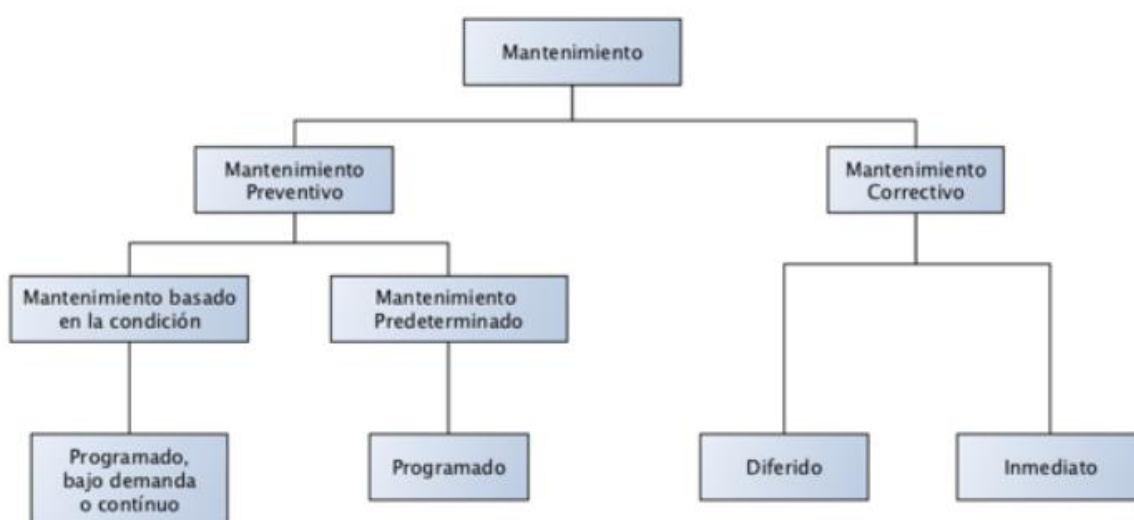


Figura 4 Tipos de mantenimiento industrial según la norma EN13306 2010

Fuente: <https://www.appvizer.es/revista/organizacion-planificacion/gestion-mantenimiento/tipos-de-mantenimiento>.

### 1.5.1. Tipos de mantenimiento

Si no hay cambios en las características intrínsecas del activo (aquellas determinadas por el diseño), el mantenimiento se divide en preventivo y correctivo.

#### 1.5.1.1. Mantenimiento preventivo

El mantenimiento preventivo es una estrategia de mantenimiento que tiene como objetivo prevenir fallos, averías o desgastes prematuros en los equipos o instalaciones realizando tareas programadas de inspección. Ajuste, limpieza, lubricación, reemplazo de piezas etc.

La idea central del mantenimiento preventivo es actuar de manera anticipada, antes de que ocurran los fallos.



### **1.5.1.2. Mantenimiento predeterminado**

El mantenimiento predeterminado es cíclico, independiente de la condición, tradicionalmente llamado mantenimiento preventivo planificado el cual sigue programaciones, procedimientos predeterminados o establecidos de antemano, es decir, tareas que se realizan en función de un plan o cronograma ya definido sin esperar a que se presente una falla.

### **1.5.1.3. Mantenimiento basado en la condición**

El mantenimiento basado en la condición (CBM) se centra en realizar las intervenciones únicamente cuando las condiciones del equipo indican que es necesario esto se puede lograr utilizando dispositivos IoT. En lugar de basarse en intervalos de tiempo o en un cronograma predeterminado, se basa en el monitoreo de las condiciones reales del equipo para determinar el momento óptimo para realizar el mantenimiento

### **1.5.1.4. Mantenimiento continuo**

Mantenimiento enfocado en la supervisión proactiva, diseñado para garantizar la eficiencia operativa, especialmente en sectores donde la continuidad del funcionamiento es crucial.

### **1.5.1.5. Mantenimiento predictivo**

El objetivo principal es predecir el fallo antes de que ocurra utilizando datos históricos y análisis avanzado. Se basa en el uso de tecnologías de monitoreo y análisis de datos para detectar patrones y tendencias que indiquen que una falla está próxima.

### **1.5.1.6. Mantenimiento correctivo**

Este se subdivide en aquel que se realiza inmediatamente después de la verificación de un fallo funcional (correctivo, inmediato) y el mantenimiento diferido que puede programarse.

## **1.5.1. Consecuencia del descuido del mantenimiento de los equipos**

Descuidar el mantenimiento de los equipos industriales puede provocar averías inesperadas, reparaciones costosas, tiempos de inactividad prolongados, disminución de la productividad y mayores riesgos de seguridad. Además, puede llevar a la anulación de las garantías y provocar fallas prematuras o averías completas, lo que requeriría reemplazos costosos. El mantenimiento regular es esencial para evitar estas consecuencias y garantizar un funcionamiento fiable y eficiente.

Los protocolos de comunicación son conjuntos de reglas y estándares que permiten que dispositivos y sistemas intercambien información de manera efectiva y coherente. Son esenciales en redes de computadoras e Internet, permitiendo que distintos sistemas, independientemente de su hardware, software o ubicación, puedan comunicarse entre sí sin problemas.



## 2. CAPÍTULO : MANTENIMIENTO BASADO EN LA CONDICIÓN

El mantenimiento basado en condición (CBM) es una estrategia de mantenimiento que consiste en monitorear el estado o rendimiento de un activo en tiempo real, mediante el uso de sensores, tecnologías, instrumentos o sistemas de información, para determinar cuándo y qué tipo de acción debe realizarse. Es el proceso para recolectar datos sistemáticamente y evaluarlos para identificar cambios en el desempeño o condición de un sistema o sus componentes, para que este pueda generar rentabilidad manteniendo su confiabilidad.

El objetivo de este mantenimiento es detectar los cambios y corregir algún tipo de desvío en las variables o predecir las fallas, optimizar la vida útil y el funcionamiento del equipo, reducir el tiempo y el coste del mantenimiento, incluyendo también la seguridad en los procesos.

Este mantenimiento es el más proactivo y óptimo, ya que permite actuar justo cuando es necesario al contrario del mantenimiento predictivo que busca predecir la falla en este tipo de mantenimiento la falla nunca sucede, aunque en ciertas condiciones también se relaciona con el mantenimiento centrado en la confiabilidad (RCM), por que ayuda a identificar los posibles problemas con los activos y terminar lo que se debe hacer para asegurar que sigan funcionando de manera óptima.

### Procedimiento para la implementación:

- Medir de manera selectiva y con una buena resolución los cambios en las condiciones de operación.
- Se comparan los cambios contra un estándar o con lo que se estima para parámetros ideales para identificar una posible desviación
- Determinar si existen indicios de una falla potencial
- Definir la trayectoria de la falla y como puede proyectarse en el resto de los equipos
- Se toman las acciones necesarias para regresar el activo a sus condiciones nominales.

### 2.1. Aplicación de mantenimiento basado en la condición a la cultura empresarial

Antes de su aplicación en la organización, es importante que el organigrama de mantenimiento deba ser modificado para adoptar esta metodología. Esto se refiere a que no basta con apoyarse solo en el área técnica meramente operativa: es mantenerse en un área de ingeniería, que trabajara con datos y modelos matemáticos junto con un área de diagnóstico. Un personal específico y profesional dentro de estos departamentos es el que debe decidir qué hacer con los activos.



## 2.2. KPI Y MÉTRICAS DE MANTENIMIENTO

Una eficiencia en la industria es una ventaja y necesidad en donde el mantenimiento tiene una gran importancia. Los indicadores clave de rendimiento (KPIs) de mantenimiento actúan como actúan para brindar bienestar a una empresa y a su dominio, en donde este puede marcar la diferencia entre un éxito rotundo y un fracaso catastrófico.

Los KPIs permiten evaluar factores que son de alta criticidad para que la organización se mantenga funcionando, asegurando que los equipos de mantenimiento tengan un mayor valor, integrando un ecosistema con una alta importancia. Este ecosistema involucra elementos que van desde profesionales capacitados, métodos de adquisición de datos y una estructuración en la planificación y reparación que implica números que cuentan el historial en la gestión de un activo.

KPI en mantenimiento significa “indicador clave de rendimiento” y sirve como un estándar utilizado para evaluar el cumplimiento de objetivos del personal y los grupos de mantenimiento.

Al establecer KPIs de mantenimiento los objetivos se vuelven cuantificables proporcionando una comprensión rápida sobre el estado de un proyecto a diferencia de una métrica de mantenimiento que es aquella que abarca cualquier punto de datos que una organización monitorea y funciona como parámetro cuantificable para evaluar logros a través de referencias KPI, para valorar el avance de tareas o estar alerta ante posibles averías.

Las métricas de mantenimiento examinan la fuerza laboral del equipo de mantenimiento, el desempeño de los activos o máquinas, así como los procesos. Esto se refiere a los elementos de datos en los cuales se fundamentan los objetivos o estándares (KPI de mantenimiento).



### **2.2.1. IMPORTANCIA DE LOS KPI**

Los son importantes porque permiten medir y evaluar el progreso de una organización, equipo o proyecto hacia el cumplimiento de sus objetivos estratégicos. Su importancia radica en los siguientes aspectos:

- **Medición del rendimiento:** Los KPI proporcionan métricas concretas para evaluar el desempeño en áreas clave, ayudando a identificar qué tan cerca estás de alcanzar tus metas
- **Toma de decisiones informadas:** Al contar con datos claros y cuantificables, los líderes y equipos pueden tomar decisiones basadas en evidencia en lugar de intuiciones, lo que mejora la efectividad de las acciones.
- **Monitoreo continuo:** Permiten un seguimiento constante del progreso, lo que facilita detectar problemas o desviaciones a tiempo para aplicar correcciones.
- **Motivación y responsabilidad:** Los KPI claros y bien definidos motivan a los equipos a alcanzar metas específicas y fomentan la responsabilidad, ya que todos saben cómo se mide su contribución.
- **Identificación de oportunidades de mejora:** Analizar el desempeño a través de KPI permite descubrir áreas que necesitan optimización o ajustes para maximizar resultados.
- **Comunicación clara:** Facilitan la comunicación del estado del proyecto o desempeño del negocio a todas las partes interesadas, como empleados, socios o inversores.

En resumen, los KPI son esenciales porque proporcionan un marco estructurado para medir el éxito, evaluar resultados y optimizar el desempeño en función de los objetivos establecidos.

### **2.2.2. TOMA DE DECISIONES BASADAS EN DATOS**

La toma de decisiones basada en datos (Data-Driven Decision Making, DDDM) es el proceso de recopilar datos en función de los indicadores clave de rendimiento (KPI) y transformarlos en información útil.

El concepto de “basarse en los datos” hace referencia a usar hechos concretos, o datos, para identificar patrones, hacer inferencias y realizar análisis a fin de guiar el proceso de toma de decisiones.

La toma de decisiones basada en datos permite a las empresas generar información y predicciones en tiempo real, optimizar el rendimiento y probar nuevas estrategias. Estas



decisiones informadas conducen a un crecimiento y una rentabilidad sostenibles, mientras que confiar en la intuición puede dar lugar a lo contrario.

#### Ventajas de la toma de decisiones basada en datos:

- Mayor precisión: Basarse en datos minimiza los errores derivados de sesgos o suposiciones y permite tomar decisiones fundamentadas en hechos comprobables.
- Mejor predicción de resultados: Al analizar tendencias y patrones históricos, las organizaciones pueden anticipar escenarios futuros y tomar medidas proactivas.
- Optimización de recursos: Ayuda a asignar recursos de manera más eficiente al identificar las áreas con mayor impacto potencial.
- Medición de resultados: Permite evaluar el éxito de las decisiones mediante métricas específicas, haciendo más fácil ajustar estrategias si es necesario.
- Prevención de fallas: El análisis de datos históricos y en tiempo real permite predecir problemas antes de que ocurran, minimizando paradas no planificadas.
- Reducción de costos: Al optimizar el mantenimiento, se evitan reparaciones mayores y el reemplazo innecesario de componentes.
- Optimización de recursos: Permite planificar mejor los recursos humanos, las herramientas y los materiales necesarios para las intervenciones.

### **2.2.3. MEJORA CONTINUA**

La mejora continua es el proceso continuo de analizar el rendimiento, identificar oportunidades y realizar cambios graduales en los procesos, los productos y el personal. Al analizar y ajustar los procesos, la empresa puede ahorrar tiempo y ofrecer un producto mejor en general.

La mejora continua en el mantenimiento se centra en las medidas destinadas para optimizar el proceso de mantenimiento y para conseguir la reducción o eliminación de los defectos del mantenimiento con la finalidad de prolongar la vida útil de los activos y evitar las paradas no planificadas de la producción.

#### Objetivos de la mejora continua en el mantenimiento industrial

El mantenimiento debe ser una actividad de mejora continua, que mejore la calidad del servicio, agilice los procesos y optimice los costes de explotación.

Un entorno de mejora continua incluye los elementos necesarios para evaluar las situaciones y determinar las oportunidades de mejora mediante la aplicación de varios factores para alcanzar los objetivos.

A continuación, detallamos cuáles son los más importantes:



- Definir los procesos de mantenimiento y la forma en que se mantendrán los equipos críticos, centrándose en los modos de fallo de los componentes críticos.
- Cada tarea de mantenimiento debe estar directamente orientada a crear el mejor valor de los activos.
- Evaluar los recursos disponibles y las herramientas necesarias para los trabajos de mantenimiento proactivo.
- El trabajo debe ejecutarse de forma eficiente y eficaz, lo que requiere un proceso meticuloso de gestión del trabajo para evitar el despilfarro de trabajos.
- Establecer objetivos globales con KPI para los puntos de control del progreso.
- Integrar el proceso en un sistema de gestión del mantenimiento GMAO para capturar todos los datos, hacer un seguimiento de la actividad y proporcionar un análisis de los informes.
- Todos los datos relacionados con los equipos, los activos críticos, las órdenes de trabajo, los retrasos, las horas de trabajo/técnico, los costes, los calendarios de mantenimiento, las piezas de repuesto y otros aspectos deben ser supervisados y medidos para identificar dónde se han realizado mejoras y dónde se necesitan.

#### ¿Cómo mejorar la estrategia de mantenimiento?

Para obtener el máximo rendimiento, las empresas deben estar preparadas para los cambios. Existen tres áreas que deben estar interconectadas para el concepto de cambio:

- Procesos y flujos de trabajo para conseguir las mejoras (por ejemplo, hacer más trabajo preventivo en lugar de correctivo, etc.)
- Tecnologías para facilitar o habilitar las tareas de mantenimiento.
- La empresa y personal de mantenimiento deben validar cualquier cambio, por lo que se necesitan herramientas que faciliten los cambios.

En la actualidad, el mantenimiento basado en la condición y las estrategias predictivas basadas en tecnologías de vanguardia están llegando al mercado y su continua reducción de costes abre amplias oportunidades, ayudando al personal de operaciones y mantenimiento a realizar las tareas con mayor eficacia.



## 2.2.4. IMPORTANCIA DE LOS KPI EN PROCESOS PRODUCTIVOS

Los KPI tienen gran importancia para el mejoramiento continuo de los procesos productivos que se llevan a cabo en las industrias, al servir como métrica para describir diferentes patrones de comportamientos dentro de las mismas.

La importancia de los KPI en los procesos productivos es mayor al incluir indicadores avanzados como OEE (Overall Equipment Effectiveness), MTTR (Mean Time to Repair), MTBF (Mean Time Between Failures) y MTTF (Mean Time to Failure), ya que estos ofrecen una visión integral del desempeño de los equipos, la eficiencia del sistema y la calidad del mantenimiento. Estos indicadores son esenciales para optimizar el rendimiento, minimizar los costos y garantizar la confiabilidad y disponibilidad de los activos productivos.

Importancia de los KPI específicos (OEE, MTTR, MTBF Y MTTF)

- Medición de la eficiencia general (OEE): evalúa la efectividad de los equipos en las tres dimensiones críticas: disponibilidad, rendimiento y calidad. Y permite identificar áreas problemáticas, como tiempo de inactividad, baja velocidad de operación o productos defectuosos.
- Reducción del tiempo de inactividad (MTTR): Permite evaluar la eficiencia de los equipos de mantenimiento y la rapidez con la que se recuperan las operaciones tras una falla.
- Mejora de la confiabilidad y la planificación (MTBF y MTTF):
  - MTBF mide la confiabilidad de los equipos y permite planificar el mantenimiento preventivo antes de que ocurran fallas.
  - MTTF es especialmente útil para prever la vida útil de componentes que no son reparables y garantizar la reposición oportuna.

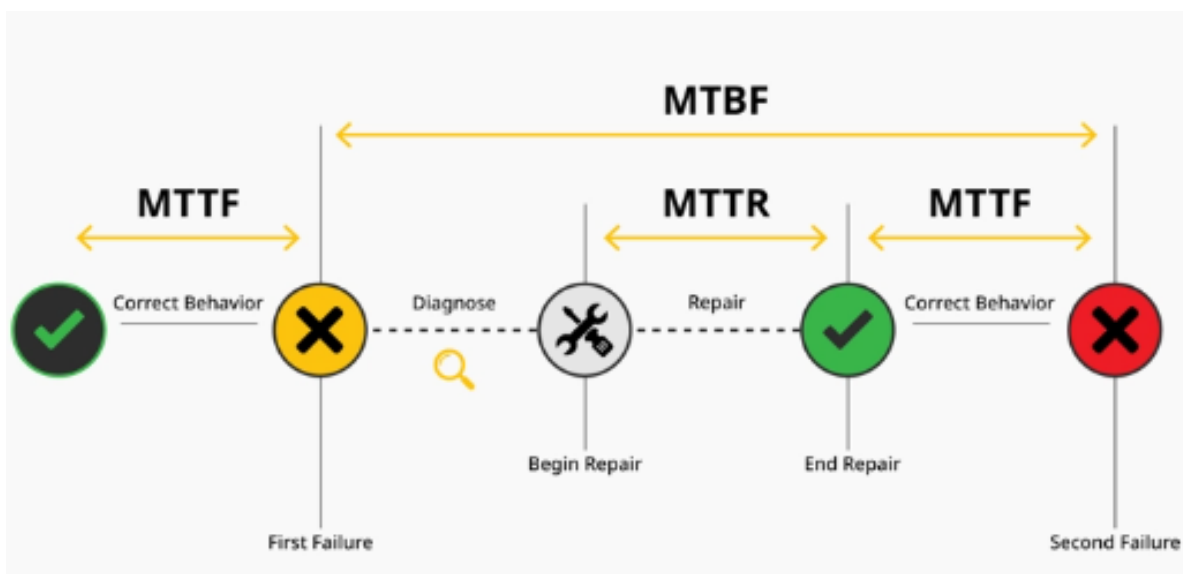


Figura 5 KPIs de mantenimiento

Fuente: <https://www.emaint.com/es/blog-what-is-mean-time-to-failure-mttf/> Año: 2024



## 2.2.5. INDICADORES ADECUADOS PARA EL MANTENIMIENTO

### 2.2.5.1. OEE (Overall Equipment Effectiveness)

Es un indicador de eficiencia general de los equipos que es responsable de medir la eficiencia de un equipo y proporcionar indicaciones valiosas acerca de donde pueden mejorar los procesos. De esta manera, a partir del mantenimiento, es posible corregir diferencias y garantizar la buena productividad de la industria. El OEE presenta de manera clara y objetiva si un equipo fue utilizado con efectividad. Esto significa cuantos elementos con la calidad deseada produjo el equipo en comparación con su capacidad total.

El OEE surge con la idea de suplir las necesidades de las metodologías del TPM (Mantenimiento de productividad total) de entender de donde provenían esas pérdidas y cuál era el nivel de eficiencia de los equipos.

Con este indicador es posible saber cuanto tiempo disponible se aplicó efectivamente en la fabricación de las piezas. Dentro de ese tiempo, el indicador también informa cuán rápido fue el equipo en la producción, esto permite tener análisis completo del proceso productivo y saber exactamente donde están las deficiencias

Esta métrica evidencia factores como:

- La producción máxima de las máquinas
- Que máquina produce más y cuál produce menos
- Si la calidad de los productos es satisfactoria
- Si alguna máquina está generando pérdidas

#### Cálculo de OEE

- $OEE = Disponibilidad \times Rendimiento \times Calidad$ .
- $Tiempo\ de\ producción\ planificado = Duración\ del\ turno - Pausas$
- $Tiempo\ de\ funcionamiento = Tiempo\ de\ producción\ Planificado - tiempo\ de\ Paradas$
- $Disponibilidad = Tiempo\ de\ Funcionamiento / Tiempo\ de\ Producción\ Planificado$
- $Desempeño = (Tiempo\ de\ Ciclo\ Ideal \times Total\ de\ piezas) / Tiempo\ de\ Funcionamiento$ .
- $Calidad = Piezas\ con\ buena\ calidad / Total\ de\ piezas$



### **2.2.5.2. MTTR (Mean Time To Repair)**

El MTTR (MEAN TIME TO REPAIR) es un indicador fundamental que mide el tiempo promedio necesario para reparar un activo después de una falla. Su importancia radica en la reducción del tiempo de inactividad (downtime) y la eficiencia del mantenimiento, ya que muestra el tiempo que se requiere para poner en funcionamiento nuevamente un equipo tras una parada no planificada. Se calcula mediante el tiempo de reparación (downtime) dividido por el número total de reparaciones

Un MTTR bajo indica que se están realizando reparaciones de manera rápida y efectiva, lo que garantiza que los activos vuelvan a estar operativos dentro de un menor tiempo posible.

### **2.2.5.3. MTTF (Mean Time to Failure)**

MTTF es un indicador que proporciona información valiosa sobre la confiabilidad de los activos y el tiempo promedio en que ocurren las fallas. El objetivo es maximizar el MTTF, lo que implica reducir la frecuencia de fallas y prolongar la vida útil del equipo. Un MTTF alto indica que los equipos son confiables y tienen una menor probabilidad de experimentar fallas.

Para mejorar el MTTF, es esencial implementar programas de mantenimiento preventivo y predictivo, realizar inspecciones regulares, mantener registros adecuados de las actividades de mantenimiento y utilizar técnicas de monitoreo avanzadas, como el análisis de vibraciones y termografía.

### **2.2.5.4. MTBF (Mean Time Between Failures)**

El MTBF es un indicador clave que complementa el MTTF y se utiliza para medir el tiempo promedio entre fallas en un equipo. Un MTBF alto indica una mayor confiabilidad y menor frecuencia.

Para mejorar el MTBF es crucial identificar las causas raíz de las fallas. Implementar acciones correctivas y preventivas y mejorar la calidad de los repuestos utilizados. Además, es fundamental realizar análisis de fallas adecuado para comprender los patrones y las tendencias que pueden afectar el sistema o el equipo.

### **2.2.5.5. Disponibilidad y confiabilidad**

La disponibilidad y confiabilidad son indicadores fundamentales para garantizar el buen funcionamiento de los activos y evaluar su rendimiento. La disponibilidad se refiere al tiempo en que los activos están disponibles y listos para operar. Una alta disponibilidad implica que los activos están disponibles cuando se les necesita, lo que reduce el tiempo de inactividad no planificado y aumenta la eficiencia operativa.

La confiabilidad, por otra parte, se refiere a la probabilidad de que un activo funcione sin fallas durante un periodo futuro específico de tiempo. Una alta confiabilidad implica que los activos son más propensos a funcionar correctamente sin experimentar fallas.



Para medir la disponibilidad y la confiabilidad, se utiliza diferentes métricas e indicadores, el tiempo medio entre fallas (MTBF), el tiempo medio para reparación (MTTR), entre otros. Estos indicadores ayudan a las organizaciones a evaluar el rendimiento de sus activos, identificar áreas de mejora y tomar decisiones informadas para optimizar la disponibilidad y la confiabilidad

#### **2.2.5.6. Backlog (Acumulación de trabajos pendientes)**

El backlog se refiere a la acumulación de trabajo pendiente en el departamento de mantenimiento. Puede incluir tareas de mantenimiento preventivo, correctivo y predictivo que aún no se han completado. Un backlog alto puede tener un impacto negativo en la eficiencia del departamento, ya que puede provocar demoras en la ejecución de tareas importantes y un aumento del tiempo de respuesta en la ejecución y un aumento del tiempo de respuesta ante fallas o requerimientos de mantenimiento urgente.

Reducir el backlog es muy importante para mejorar la eficiencia del mantenimiento y garantizar la disponibilidad de recursos para tareas críticas. Para gestionar el backlog de manera adecuada, es necesario establecer prioridades claras, asignar recursos adecuados, utilizar herramientas de programación de mantenimiento y seguimiento de tareas, así como mejorar los procesos de comunicación y coordinación dentro del departamento de mantenimiento.



### 3. CAPÍTULO: TECNOLOGÍA DE INFORMACIÓN Y OPERACIÓN (IT/OT)

La convergencia de la Tecnología de la Información (IT) y la Tecnología de Operaciones (OT) está transformando rápidamente el paisaje industrial y empresarial.

La IT se refiere a la utilización de sistemas, computadoras y redes para procesar y gestionar información. Incluye hardware, software, bases de datos y redes que permiten a las empresas almacenar, recuperar y enviar datos de manera eficiente. Algunas áreas clave de IT incluyen:

- Ciberseguridad: Protección de datos y sistemas contra ataques.
- Infraestructura de Red: Redes y servidores que soportan la conectividad y el intercambio de información.
- Gestión de Datos: Bases de datos y almacenamiento de datos.
- Desarrollo de Software: Creación de aplicaciones y sistemas que permiten a las empresas operar de manera efectiva.

La OT se enfoca en el monitoreo y control de dispositivos y procesos físicos en entornos industriales y de manufactura. Incluye sistemas como SCADA (Supervisory Control and Data Acquisition), PLCs (Programmable Logic Controllers) y otros dispositivos que gestionan maquinaria y procesos industriales. Áreas clave de OT incluyen:

- Automatización Industrial: Uso de tecnologías para automatizar procesos de producción.
- Control de Procesos: Monitoreo y control de operaciones industriales en tiempo real.
- Mantenimiento Predictivo: Uso de datos y análisis para predecir y prevenir fallas en equipos.

Al combinar los datos y capacidades de IT con los sistemas y procesos de OT, las empresas pueden:

- Optimizar Operaciones: Mejorar la eficiencia operativa mediante el análisis de datos en tiempo real y la automatización de procesos.
- Mejorar la Toma de Decisiones: Utilizar datos integrados para tomar decisiones más informadas y precisas.
- Facilitar la Innovación: Desarrollar nuevas tecnologías y aplicaciones que aprovechen la sinergia entre IT y OT.

#### 3.1. Estructura metodológica para implementar el sistema de monitoreo

La estructura metodológica para implementar un sistema de monitoreo remoto, basado en dispositivos como Raspberry Pi y el DAQ HAT MCC, empleando protocolos de comunicación, plataformas de desarrollo y servicios en la nube, así como la obtención de indicadores clave de rendimiento (KPIs) de mantenimiento, requiere una planificación detallada compuesta por diversas etapas. Estas etapas permiten guiar el flujo de datos de manera eficiente y ordenada.



### 3.1.1. Variables asociadas al equipo

El objetivo principal de sistema de monitoreo remoto es mantener la continuidad operacional y notificar de manera temprana cualquier anomalía en el equipo, para lograr la efectividad de este sistema se requieren de las siguientes necesidades:

#### 3.1.1.1. Temperatura

La temperatura es una variable clave que se utiliza para supervisar y controlar el comportamiento térmico de componentes específicos de la máquina. Esta variable puede determinar la eficiencia, la calidad del producto, y la seguridad operativa en sistemas industriales. A continuación, se detalla su importancia:

- Evitar sobrecalentamiento: Componentes críticos, como motores, engranajes, o rodamientos, pueden dañarse si operan fuera de sus rangos de temperatura.
- Optimización del proceso: En muchas máquinas productivas, ciertas temperaturas deben mantenerse para garantizar la calidad del producto final (e.g., en extrusoras, hornos o prensas térmicas).
- Aumentar la vida útil del equipo: Mantener la temperatura en rangos óptimos reduce el desgaste y las fallas.
- Seguridad: Protege al personal y al entorno de situaciones peligrosas, como incendios o explosiones.

##### Origen de aumento de temperatura:

- Fricción: El contacto entre superficies móviles genera calor debido a la resistencia al movimiento.
- Sobrecarga de trabajo: Un equipo sometido a cargas excesivas puede generar más calor.
- Fallos en sistemas de enfriamiento: Si los sistemas de ventilación, refrigeración o lubricación fallan, la temperatura se eleva.
- Energía eléctrica mal gestionada: Cortocircuitos o sobrecargas en sistemas eléctricos producen calor adicional.

#### 3.1.1.2. Corriente en el tablero

La corriente en un tablero eléctrico se refiere al flujo de electricidad que circula por el sistema, distribuyendo energía a los distintos circuitos conectados. En un entorno industrial, un tablero eléctrico es el punto principal de distribución de energía eléctrica, y la corriente en él debe ser monitoreada y controlada cuidadosamente para evitar problemas como sobrecargas, cortocircuitos o fallas eléctricas.

##### Importancia de monitorear la corriente en el tablero:

- Evitar sobrecargas y fallas: Garantiza que el sistema opere dentro de su capacidad nominal.



- Prolongar la vida útil del equipo: Reduce el estrés térmico en los componentes del tablero.
- Garantizar la seguridad: Previene riesgos de incendio y descargas eléctricas.
- Cumplir normativas eléctricas: Los sistemas deben operar de acuerdo con normas locales.

#### Origen de aumento de corriente:

- Sobrecarga del sistema: Demasiados dispositivos conectados a un solo circuito.
- Mal dimensionamiento de cables o breakers: Componentes insuficientes para manejar la corriente.
- Conexiones sueltas: Pueden generar arcos eléctricos y sobrecalentamiento.

### **3.1.1.3. Presión**

La presión se define como la fuerza por unidad de área que un fluido ejerce sobre las paredes de un recipiente o sobre un objeto sumergido en él. Se mide generalmente en pascales (Pa).

#### Factores que afectan la presión en un sistema:

- Temperatura: En gases y líquidos, la presión aumenta con el incremento de temperatura debido a la mayor energía cinética de las moléculas.
- Volumen del recipiente: Según la ley de Boyle (para gases ideales), la presión aumenta cuando el volumen de un recipiente se reduce, si la temperatura permanece constante.
- Densidad del fluido: A mayor densidad del fluido (líquido o gas), mayor será la presión.

#### Importancia de medir la presión:

1. Seguridad: La presión debe mantenerse dentro de los límites operativos para evitar explosiones, fugas o fallos catastróficos en equipos como calderas, tanques de gas o sistemas de aire comprimido.
2. Eficiencia del proceso: Mantener una presión constante en sistemas de fluidos puede optimizar el rendimiento de maquinaria y equipos industriales, como compresores, bombas y sistemas hidráulicos.
3. Control de calidad: En muchos procesos, como la inyección de plásticos o la fabricación de productos farmacéuticos, controlar la presión es crucial para la consistencia del producto final.

#### Problemas Comunes Relacionados con la Presión:



- **Sobrepresión:** Si la presión excede el límite de seguridad de un sistema, puede causar fallas en las válvulas, explosiones o fuga de fluidos peligrosos.
- **Baja presión:** Puede indicar una fuga o mal funcionamiento de una bomba o compresor, lo que afecta el rendimiento del sistema.
- **Pulsación de presión:** En sistemas de bombeo o de compresión, las fluctuaciones en la presión pueden generar daños o interferir en el proceso.

#### 3.1.1.4. Vibraciones

La vibración es el movimiento oscilante alrededor de una posición de equilibrio. Puede ocurrir en máquinas, estructuras o componentes mecánicos, y generalmente se mide en términos de amplitud, frecuencia y aceleración. Se expresa en unidades como milímetros (mm) para desplazamiento,  $m/s^2$  para aceleración, o Hz para frecuencia.

##### Importancia de medir la vibración

- **Detección temprana de problemas:** Las vibraciones anómalas son un indicador temprano de fallas en máquinas o equipos, como desbalance, desalineación, desgaste en rodamientos o problemas estructurales.
- **Menor desgaste:** Controlar las vibraciones evita daños en componentes críticos como ejes, rodamientos, engranajes y motores.
- **Eficiencia energética:** Equipos que operan sin vibraciones anómalas consumen menos energía, lo que reduce los costos operativos.
- **Prevención de riesgos:** Las vibraciones excesivas pueden debilitar estructuras, causar desprendimientos o aumentar la probabilidad de accidentes laborales.

##### Normas asociadas a las vibraciones

- **ISO 10816:**
  - Establece directrices para la evaluación de las vibraciones en máquinas rotativas en función de mediciones de vibración en puntos específicos.
  - Se utiliza para determinar los niveles de vibración aceptables y para identificar problemas potenciales.
  - Dividida en sub partes para diferentes tipos de máquinas (e.g., motores eléctricos, bombas, ventiladores).
- **ISO 20816:**
  - Reemplaza progresivamente la ISO 10816 con lineamientos actualizados para la medición y evaluación de vibraciones en máquinas.
  - Incluye análisis detallado para condiciones específicas, como turbinas y equipos grandes.



- ISO 7919:
  - Cubre la medición de vibraciones en máquinas rotativas utilizando mediciones directas en los ejes.
- ANSI/ASA S2.11:
  - Especifica métodos para evaluar vibraciones en maquinaria con enfoque en estándares norteamericanos.

### 3.1.2. Requerimientos de internet (Wi-Fi o Ethernet)

Para implementar un flujo de datos de monitoreo remoto, el cual integra MQTT, Node RED, MongoDB Atlas y Grafana Cloud, se debe tener en cuenta varios factores relacionados con el ancho de banda, la latencia y la estabilidad de la conexión, el intervalo de tiempo de los mensajes  $\geq 30$  segundo. A continuación, se detallan los requisitos de internet necesarios para soportar un flujo de datos eficiente y estable en este escenario:

#### 3.1.2.1. Factores

- Frecuencia de envío de datos: la raspberry PI enviará datos mediante diferentes tópicos utilizando una frecuencia de muestreo en función de la variable que se está monitoreando, los datos adquiridos serán enviados en un tiempo  $\geq 30$  segundos, por lo que la cantidad de ancho de banda requerido no debe ser muy alta. Para este flujo de datos, una conexión mínima de 10-20 Mbps es adecuada para manejar el tráfico de datos.
- Latencia: para un flujo de datos en tiempo real, y la integración de consultas KPI en Grafana, la latencia ideal debe ser al menos de 200 ms para una actualización rápida sin retrasos
- Estabilidad y redundancia: Es importante que la conexión a internet sea estable, ya que cualquier desconexión o fluctuación puede afectar el flujo de datos entre la etapa de adquisición y visualización.

En los siguientes gráficos de estado se visualizan las métricas de volumen de datos de entrada (309b/s) y de salida (5,2kb/s), los mensajes insertados (promedio de 0,4 mensajes /seg) y las conexiones establecidas a diferentes colecciones de la base de datos(5).

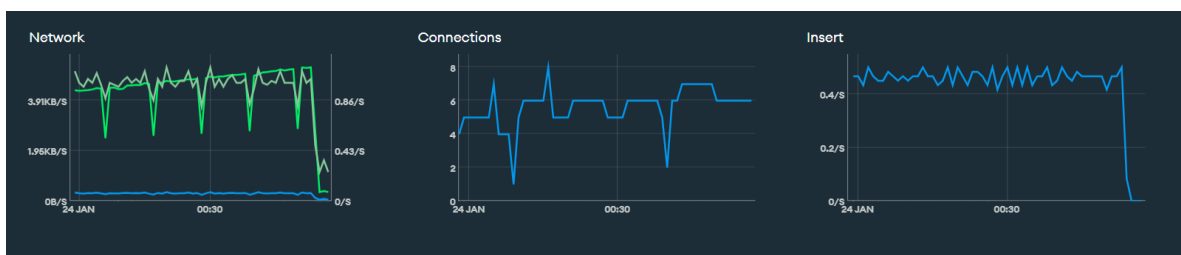


Figura 6 métricas de conexión, volumen de datos e insertos por segundo

Fuente: MongoDB Atlas. Año:2025



## Network



Figura 7 Detalle de volumen de datos en la red

Fuente: MongoDB Atlas. Año: 2025

### 3.1.3. Determinación de KPIs de mantenimiento

los KPIs propuestos para el mantenimiento son fundamentales para medir el rendimiento de las operaciones del equipo. Estos KPIs permiten evaluar la efectividad, eficiencia y proactividad del mantenimiento y con ello tomar decisiones informadas para mejorar la disponibilidad de los equipos, reducir costos y aumentar la vida útil de los activos. Los KPIs serán analizados mediante los registros de detención evaluados en la colección de “corriente” cuando esta sea =0, lo que nos dirá que el motor está detenido.

#### 3.1.3.1. MTTR (Mean Time To Repair)

El MTTR (MEAN TIME TO REPAIR) es un indicador fundamental que mide el tiempo promedio necesario para reparar un activo después de una falla.

Un MTTR bajo indica que se están realizando reparaciones de manera rápida y efectiva, lo que garantiza que los activos vuelvan a estar operativos dentro de un menor tiempo posible.

#### 3.1.3.2. MTTF (Mean Time to Failure)

MTTF es un indicador que proporciona información valiosa sobre la confiabilidad de los activos y el tiempo promedio en que ocurren las fallas

Para mejorar el MTTF, es esencial implementar programas de mantenimiento preventivo y predictivo, realizar inspecciones regulares, mantener registros adecuados de las actividades de mantenimiento y utilizar técnicas de monitoreo avanzadas, como el análisis de vibraciones y termografía.

#### 3.1.3.3. MTBF

Un MTBF alto indica una mayor confiabilidad y menor frecuencia.

Para mejorar el MTBF es crucial identificar las causas raíz de las fallas. Implementar acciones correctivas y preventivas y mejorar la calidad de los repuestos utilizados.



## 3.2. Selección de tecnologías

### 3.2.1. Bróker HiveMQ

HiveMQ es un broker de mensajería MQTT, lo que permite la publicación y suscripción de mensajes entre dispositivos y sistemas de manera eficiente y escalable.

#### 3.2.1.1. Características Clave de HiveMQ

- Escalabilidad y Rendimiento:
  - Alta Concurrencia: Capaz de manejar millones de conexiones simultáneas, HiveMQ está diseñado para escalar horizontalmente, permitiendo la adición de más nodos sin comprometer el rendimiento.
  - Bajo Retardo: Optimizado para minimizar la latencia, HiveMQ asegura que los mensajes se entreguen de manera rápida y eficiente, incluso en redes congestionadas.
- Seguridad:
  - Autenticación y Autorización: Soporta múltiples métodos de autenticación (como OAuth 2.0, LDAP, TLS/SSL) y permite definir políticas de autorización granulares para controlar el acceso a los datos.
  - Cifrado: Utiliza cifrado TLS/SSL para proteger los datos en tránsito, asegurando que las comunicaciones sean seguras.
- Fiabilidad y persistencia:
  - Persistencia de Mensajes: HiveMQ garantiza la entrega de mensajes mediante la persistencia de estos en bases de datos configurables, asegurando que los datos no se pierdan incluso si hay interrupciones en el servicio.
  - Alta Disponibilidad: Diseñado para entornos críticos, HiveMQ puede configurarse en clústeres para proporcionar redundancia y asegurar la continuidad del servicio.
- Integración y Extensibilidad:
  - Plugins y Extensiones: HiveMQ soporta una arquitectura de plugins que permite extender su funcionalidad mediante la adición de módulos personalizados. Esto facilita la integración con sistemas existentes y aplicaciones específicas.
  - Integración Empresarial: Compatible con múltiples servicios de nube, bases de datos y plataformas de análisis de datos, lo que facilita la integración en infraestructuras IT/OT existentes.
- Gestión y Monitoreo:
  - Panel de Control: HiveMQ ofrece una interfaz de usuario intuitiva para la administración y monitoreo en tiempo real del estado del broker, las conexiones de clientes, y el tráfico de mensajes.
  - Métricas y Alertas: Proporciona métricas detalladas y alertas configurables que permiten a los administradores supervisar el rendimiento y la salud del sistema.



### 3.2.1.2. EDICIONES DE HIVE MQ

Es una plataforma de mensajería basada en MQTT, es totalmente compatible con todas las características estándar de MQTT y ofrece funcionalidad ampliada, como clústeres de alta disponibilidad, seguridad de nivel empresarial e integración profunda en sus sistemas existentes.

Hive MQ está disponible en ediciones comerciales y de código abierto:

Hive MQ enterprise MQTT broker: es la mejor edición comercial de Hive MQ, ofrece un conjunto avanzado de características, herramientas y servicios diseñados para empresas y aplicaciones críticas que requieren escalabilidad, seguridad y fiabilidad adecuada para manejar grandes volúmenes de dispositivos y mensajes.

<p><b>Suite de IoT de propósito general</b></p> <p>Diseñado para casos de uso generales de IoT o dispositivos conectados en una variedad de industrias</p> <p><b>Precio basado en:</b></p> <ul style="list-style-type: none"><li>✓ Número de dispositivos</li><li>✓ Complejidad de los dispositivos</li><li>✓ Necesidades de HiveMQ Edge (opcional)</li></ul>	<p><b>Suite Industria 4.0</b></p> <p>Diseñado para casos de uso de IoT industrial con diversos requisitos en todo el sector industrial</p> <p><b>Precio basado en:</b></p> <ul style="list-style-type: none"><li>✓ Número de sitios</li><li>✓ Complejidad de cada sitio</li><li>✓ Requisitos de conectividad centralizada</li></ul>	<p><b>Suite de vehículos conectados</b></p> <p>Diseñado para casos de uso modernos de vehículos conectados, incluidos automóviles, camiones, tractores, aviones o autobuses</p> <p><b>Precio basado en:</b></p> <ul style="list-style-type: none"><li>✓ Número de vehículos</li><li>✓ Conexiones por vehículo</li></ul>
---	---	---

Figura 8:Ediciones de HIVE MQ

Fuente: <https://www.hivemq.com/>. Año:2024



Hive MQ Edge: es una puerta de enlace MQTT perimetral basada en software y un convertidor de protocolos que ayuda a las organizaciones a cerrar la brecha entre OT y TI. Ayuda a organizar los datos en un espacio de nombres unificado, lo que facilita la administración y la transmisión de datos en toda la infraestructura.

Características de Hive MQ Edge:

- Corredor (broker)
- Puente MQTT
- Adaptador de protocolos: admite diversos protocolos de comunicación y automatización industrial como: OPC-UA, Modbus, Siemens S7, Beckhoff ADS, etc.
- Adaptador personalizado
- Centro de datos

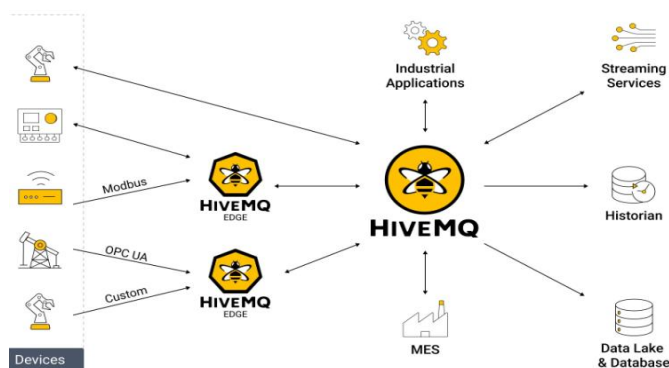


Figura 9 RED HIVEMQ

Fuente: <https://www.hivemq.com/>. Año: 2024

Nube de Hive MQ: ofrece una plataforma de servicio autogestionada para desarrollar, probar, implementar y escalar casos de IoT de producción sin una gran inversión. Gracias a la infraestructura de AWS, garantiza alta disponibilidad, seguridad y escalabilidad, facilitando la comunicación eficiente entre dispositivos IoT.

- Serverless (sin servidor): broker MQTT básico para aprender y experimentar con MQTT
- Iniciador: plataforma MQTT completa para pruebas y producción a pequeña escala.

Comenzar	Empieza GRATIS
No se requiere tarjeta de crédito	Prueba de 15 días: no se requiere tarjeta de crédito
Un broker MQTT básico para aprender y experimentar con MQTT.	Plataforma MQTT completa para pruebas y producción a pequeña escala.
<b>Un gran lugar para jugar</b>	<b>Todo en Serverless, más</b>
<ul style="list-style-type: none"><li>✓ Plataforma MQTT compartida</li><li>✓ 100 conexiones gratis</li><li>✓ 10 GB de tráfico de datos al mes</li><li>✓ MQTT 3.1, 3.1.1, 5.0</li><li>✓ MQTT a través de TLS/SSL</li><li>✓ Reglas básicas de autorización</li><li>✓ Compatibilidad con WebSocket</li></ul>	<ul style="list-style-type: none"><li>✓ Plataforma MQTT dedicada</li><li>✓ Conexiones ilimitadas</li><li>✓ Rendimiento de datos de hasta 1 MB/s</li><li>✓ Autenticación de certificado de cliente</li><li>✓ Centro de control de HiveMQ</li><li>✓ REST API</li><li>✓ Integraciones de transmisión de datos</li></ul>

Figura 10 Servicios HIVEMQ

Fuente: <https://www.hivemq.com/>. Año: 2024



### 3.3. RASPBERRY PI4

La Raspberry Pi4 es una computadora de placa reducida, desarrollada por la Raspberry Pi Foundation, que ofrece una gran potencia de procesamiento en un formato compacto y accesible. Esta última generación de Raspberry Pi fue lanzada en 2019 y se diferencia de sus predecesoras por su aumento significativo en rendimiento y capacidades, acercándose al nivel de una computadora de escritorio básica.

#### 3.3.1. Especificaciones técnicas de Raspberry PI4

- Procesador:
  - Utiliza un Broadcom BCM2711 de cuatro núcleos Cortex-A72 ARM V8 de 64 bits a 1.5 GHz, que proporciona un rendimiento notablemente superior a versiones anteriores.
  - Este procesador ARM de arquitectura de 64 bits permite ejecutar sistemas operativos más avanzados y manejar aplicaciones complejas.
- Memoria RAM
  - Disponible en variantes de 2 GB, 4 GB y 8 GB de LPDDR4. Esta variedad de opciones permite elegir la versión adecuada según las necesidades del proyecto.
  - La memoria LPDDR4 proporciona un ancho de banda rápido y es más eficiente en el consumo energético.
- Conectividad de video y audio
  - Doble salida de video micro-HDMI, cada una capaz de soportar hasta 4K a 60 FPS (en el puerto HDMI principal y 4K a 30 FPS en el secundario), permitiendo conectar dos monitores simultáneamente para configuraciones de doble pantalla.
  - Aceleración por hardware para decodificación de video H.265 y H.264, lo que permite reproducir contenido de alta calidad.
  - Puerto de 3.5 mm que combina salida de audio estéreo y video compuesto, para conexiones tradicionales.
- Almacenamiento
  - Ranura para tarjeta microSD para el sistema operativo y el almacenamiento de datos.
  - Soporte para almacenamiento mediante USB 3.0 y USB 2.0, lo que permite conectar discos externos de alta velocidad y otros dispositivos.
- Puertos y conectividad
  - Ethernet Gigabit: Proporciona una conexión de red rápida y estable.
  - Wifi 802.11ac de doble banda (2.4 GHz y 5 GHz), lo que permite conexión inalámbrica.
  - Bluetooth 5.0 para conectividad con dispositivos de baja energía como teclados, ratones y dispositivos IoT.
  - Cuatro puertos USB (dos USB 3.0 y dos USB 2.0), útiles para conectar múltiples periféricos.
- Interfaz GPIO



- 40 pines GPIO (General Purpose Input/Output) que ofrecen acceso a una variedad de interfaces, incluyendo UART, SPI e I2C.
- Esta interfaz permite a los usuarios interactuar directamente con hardware externo, sensores, motores y otros módulos electrónicos, haciéndola ideal para proyectos de electrónica y robótica.
- Alimentación y consumo de energía
  - Alimentación mediante un conector USB-C, con una entrada de 5 V/3A, que proporciona energía suficiente para manejar dispositivos externos conectados a los puertos USB.
  - Mayor eficiencia energética en comparación con modelos anteriores, aunque se recomienda un disipador de calor o ventilador en proyectos exigentes debido a que la Raspberry Pi 4 tiende a calentarse en uso intensivo.



Figura 11 Raspberry PI 4 B

Fuente: <https://www.mouser.cl/>. Año: 2024



### 3.3.2. Sistema operativo y software

Ubuntu es un sistema operativo de código abierto basado en Linux, desarrollado y mantenido por la empresa británica Canonical Ltd. Es una de las distribuciones de Linux más populares en el mundo, conocida por su facilidad de uso, estabilidad y fuerte comunidad de soporte. Ubuntu está diseñado para ser intuitivo y accesible para todos, desde usuarios novatos hasta expertos en tecnología, y se utiliza ampliamente en computadoras personales, servidores y en el ámbito de la computación en la nube.

#### 3.3.2.1. Características de Ubuntu

- Interfaz
  - Facilidad de Uso: Ubuntu está diseñado para ser intuitivo y accesible, incluso para aquellos que no tienen experiencia previa con Linux. Su interfaz gráfica, basada en GNOME, es amigable y fácil de navegar.
- Estabilidad y seguridad
  - Ubuntu recibe actualizaciones de seguridad regulares y es conocido por su estabilidad, lo cual lo convierte en una elección común para servidores y entornos críticos.
  - Utiliza un sistema de permisos estrictos y aislamiento de usuarios para proteger contra ataques y acceso no autorizado.
  - Incluye herramientas de encriptación y autenticación, además de soporte para software como el cortafuegos UFW (Uncomplicated Firewall), que facilita la configuración de reglas de red y seguridad
  - Ubuntu se lanza en versiones regulares cada seis meses y versiones de soporte extendido (LTS) cada dos años. Las versiones LTS tienen cinco años de soporte y actualizaciones de seguridad, lo que las hace ideales para entornos que necesitan estabilidad a largo plazo.
- Facilidad para Programación y Desarrollo
  - Es una de las plataformas favoritas para programadores y desarrolladores, ofreciendo soporte para múltiples lenguajes de programación (Python, Java, Ruby, C/C++ y otros).
  - Incluye herramientas de desarrollo como Git, Docker, y VS Code, y es ampliamente compatible con entornos como Node.js, Ansible y Kubernetes.
- Compatibilidad en la Nube y Servidores
  - Ubuntu Server es popular en la computación en la nube, utilizado por proveedores como AWS, Microsoft Azure y Google Cloud.
  - Ofrece soporte para contenedores, virtualización y herramientas como OpenStack, Kubernetes y Ansible, facilitando la administración y despliegue de aplicaciones en la nube.



### 3.4. Módulos DAQ

La adquisición de datos (DAQ) es un hardware que proporciona una amplia gama de aplicaciones e interfaces. Permite capturar información del mundo real mediante sensores de corriente, voltaje, temperatura, deformación o señales digitales.

Las DAQ HAT MCC permiten transferir los datos adquiridos a la Raspberry PI para integrarlos al flujo de datos designado para la aplicación deseada. Una DAQ HAT MCC es capaz de:

- Leer datos de sensores
- Conversión de señales utilizando ADC (convertidor análogo digital)
- Transferencia de datos mediante protocolos como I2C o SPI
- Aplicaciones de monitoreo.



### 3.4.1. MÓDULO DAQ MCC 128

La MCC 118 es una placa HAT (hardware attached on top) de voltaje diseñado para su uso con raspberry PI. Un HAT es una placa adicional con conector GPIO(entrada salida de propósito general) de 40 W que se ajusta la especificación de HAT de Raspberry Pi. Este módulo proporciona 8 entradas analógicas separadas y 4 entradas diferenciales para mediciones de voltaje, se pueden apilar hasta 8 MCC HAT en una Raspberry Pi.

- Interface de Raspberry Pi: el cabezal MCC128 se conecta al conector de E/S de propósito general(GPIO)de 40 pines de una Raspberry PI y se encuentra probado para todos los modelos con el conector GPIO de 40 pines.
- Configuración de HAT: Los parámetros de configuración de HAT se almacenan en una EEPROM integrada que permite a la Raspberry Pi configurar automáticamente los pines GPIO cuando se conecta al HAT.
- Entradas analógicas de MCC 128:Proporciona 8 entradas analógicas de un solo extremo o 4 entradas diferenciales. El rango de entrada se puede ser de  $\pm 10$  V,  $\pm 5$  V,  $\pm 2$  V o  $\pm 1$  V.
- Frecuencia de muestreo: en una única placa el rendimiento máximo es de 100KS/s (100000 muestras por segundo) y en placas apiladas el rendimiento máximo es de 320Ks/s agregados.
- Reloj de escaneo externo: Un pin de E/S de reloj bidireccional permite a los usuarios controlar las operaciones con una señal de reloj externa de escaneo interno de la placa.
- Disparo digital: la entrada de disparador digital externa es configurable por software para sensible al borde o al nivel. Al borde asciende o desciende y a nivel alto o bajo.

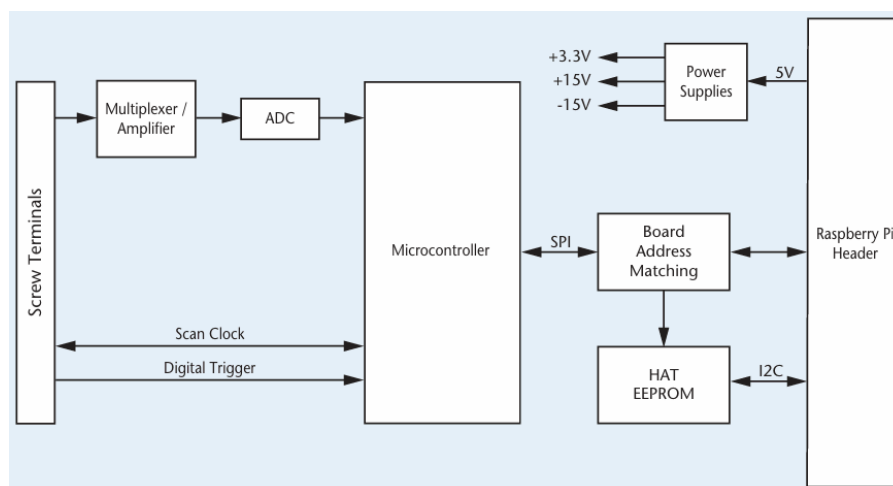


Figura 12 Diagrama de bloques MCC 128

Fuente: <https://digilent.com/>. Año: 2024

### 3.4.2. MÓDULO DAQ MCC 134 PARA MEDICIÓN DE TEMPERATURA

- Interface de Raspberry Pi: el cabezal MCC134 se conecta al conector de E/S de propósito general (GPIO) de 40 pines de una Raspberry PI y se encuentra probado para todos los modelos con el conector GPIO de 40 pines.



- Configuración de HAT: Los parámetros de configuración de HAT se almacenan en una EEPROM integrada que permite a la Raspberry Pi configurar automáticamente los pines GPIO cuando se conecta al HAT.
- Entrada de termocuplas: los usuarios pueden conectar hasta cuatro termopares diferenciales (TC) a los canales de entrada MCC134. Los tipos de TC son seleccionables por software por canal. Los valores de TC se pueden actualizar cada 1 segundo como mínimo. Las entradas de termopar está aisladas eléctricamente desde la raspberry pi para minimizar el ruido y proporcionar protección contra entornos eléctricos hostiles.
- Compensación de unión fría: el MCC134 tiene tres sensores de compensación de unión fría (CJC) de alta resolución.
- Detección de termopar abierto: EL MCC134 está equipado con detección de termopar abierto (OTD) para todos los canales de entrada TC, por lo que los usuarios pueden monitorear la placa en busca de termopares rotos o desconectados.

#### Características:

- Cuatro entradas de termopar aisladas
- Convertidor A/D de 24 Bits
- Intervalo de actualización de 1 segundo, mínimo
- Compatible con termopar J, K, R, S, T, N y B
- Compensación de unión fría
- Linealización
- Conexiones de terminal de tornillo

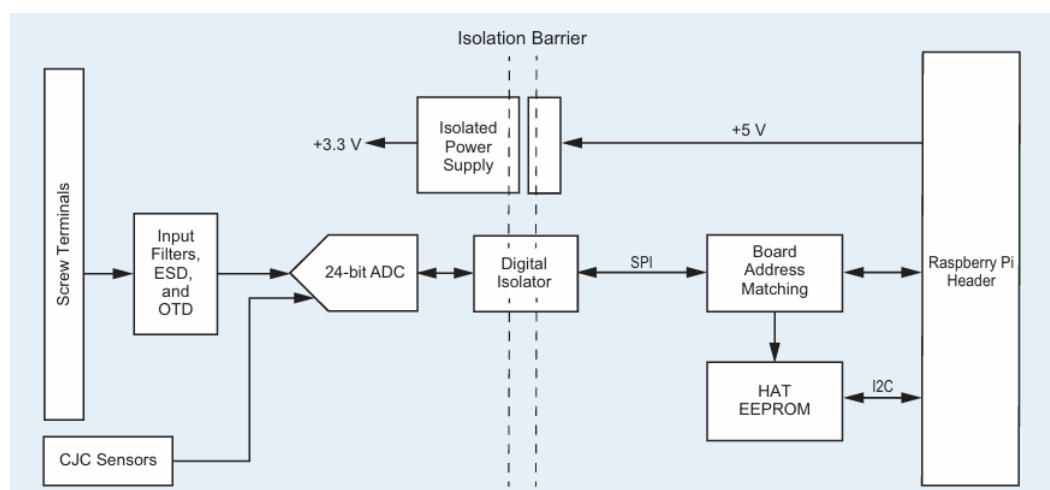


Figura 13 Diagrama de bloques MCC 134

Fuente: <https://digilent.com/>. Año: 2024

### 3.4.3. MÓDULO DAQ MCC 172

Este dispositivo es un DAC HAT de 24 bits para realizar mediciones de sonido y vibraciones a partir de sensores IEPE.

- Interface de Raspberry Pi: el cabezal MCC172 se conecta al conector de E/S de propósito general (GPIO) de 40 pines de una Raspberry Pi y se encuentra probado para todos los modelos con el conector GPIO de 40 pines.



- Configuración de HAT: Los parámetros de configuración de HAT se almacenan en una EEPROM integrada que permite a la Raspberry Pi configurar automáticamente los pines GPIO cuando se conecta al HAT.
- Entrada analógica: Los dos canales de entrada analógica diferencial de 24 bits adquieren datos simultáneamente a velocidades de hasta 51,2 kS/s (cada 19.5  $\mu$ s). Los usuarios pueden activar o desactivar la corriente de excitación IEPE. Cada canal tiene un convertidor A/D dedicado. Ambos ADC comparten el mismo reloj y están sincronizados para indicar conversiones al mismo tiempo para datos sincrónicos.
- Sincronización de múltiples HAT Se pueden sincronizar varios HAT MCC 172 con un solo reloj de muestreo. El reloj es programable para frecuencias de muestreo entre 51,2 kS/s y 200 S/s.
- Frecuencias de muestreo • Placa única: el rendimiento máximo es de 102,4 kS/s (51,2 kS  $\times$  2 canales) • Placas apiladas: el rendimiento máximo es de 307,2 kS/s agregados.
- Disparador digital La entrada de disparo (terminal TRIG) se utiliza para retrasar un escaneo de entrada hasta que se cumpla una condición especificada en la entrada del disparador. La señal de entrada de disparo puede ser una señal lógica TTL o CMOS de 3,3 V o 5 V. La condición de entrada puede ser sensible al borde o al nivel, al borde ascendente o descendente, o al nivel alto o bajo. Este terminal se puede utilizar para desencadenar el inicio de una adquisición en múltiples HAT MCC 172 sincronizados.
- Los usuarios pueden conectar señales de entrada analógicas a las entradas coaxiales 10-32 o a los terminales de tornillo. Solo se puede conectar una fuente a un canal a la vez.

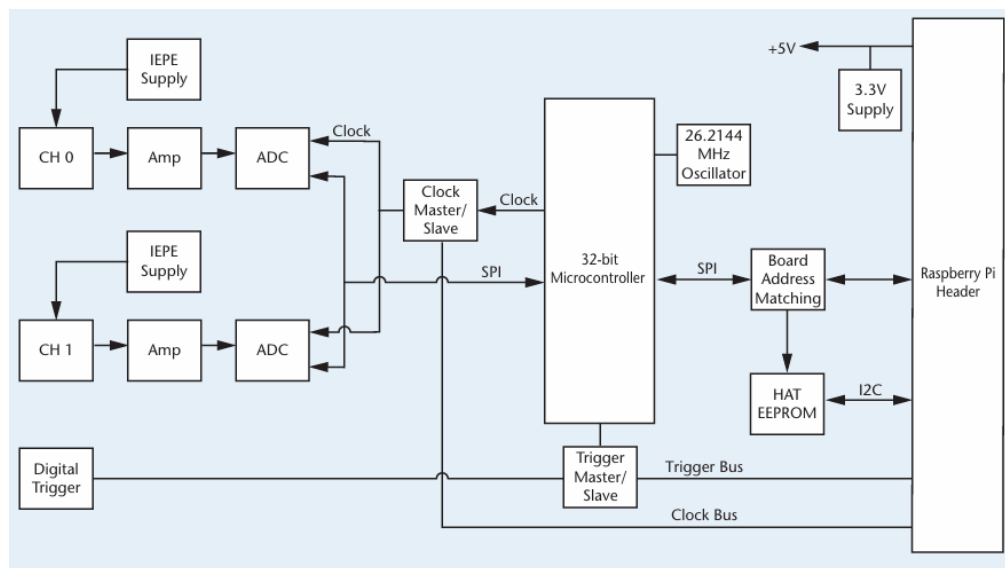


Figura 14 Diagrama de bloques MCC 172

Fuente: <https://digilent.com/>. Año: 2024

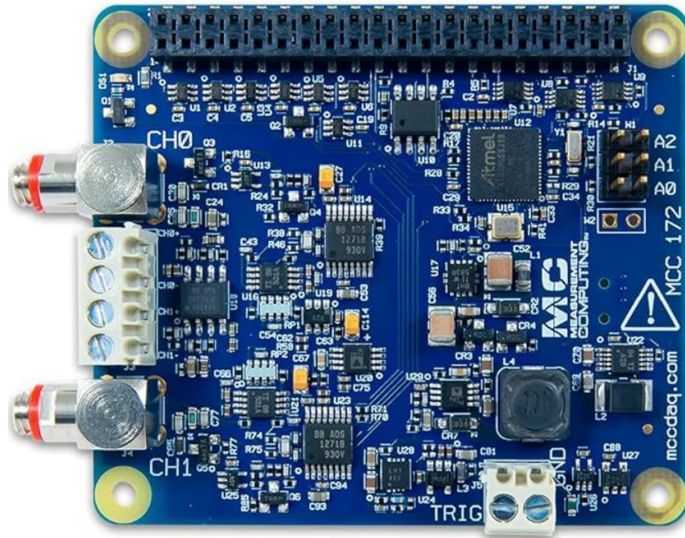


Figura 15 Módulo DAQ HAT MCC 172

Fuente: <https://digilent.com/>. Año: 2024



Figura 16 APILAMIENTO DE DAQ HATs en Raspberry PI

Fuente: <https://idm-instrumentos.es/>. Año: 2024

	MCC118	MCC134	MCC172
Convertidor A/D	12bits y 100KS/s	24bits	2x 24bits y 51,2KS/s
Entradas	8xAnalógicas de un solo extremo 10V	4xTermopar aisladas tipo J, K, R, S, T, N, E, B	2xanalógicas diferenciales acopladas de +-5V CA
características	Calibración de entrada de 20,8mV	Detección de termopar abierto	Compatibilidad con sensores IEPE
características	Entrada de disparo digital	Compensación de unión fría	Entrada de disparo digital

Tabla 1 Comparativa módulos DAQ HAT

Fuente: elaboración propia. Año: 2024



### 3.5. MONGO DB

MongoDB es una base de datos NoSQL orientada a documentos es útil para el monitoreo de datos, especialmente cuando los datos son de tipo no estructurado o semiestructurado, como logs, métricas, eventos o información de sensores. A continuación, se explica cómo MongoDB puede ser utilizado en el contexto de monitoreo de datos y su implementación.

- Modelo de datos para monitoreo: MongoDB utiliza un modelo basado en documentos JSON (BSON), lo que facilita la flexibilidad en el almacenamiento de datos sin necesidad de definir una estructura rígida de esquema. Esto es ideal para almacenar datos de monitoreo, que pueden variar en formato o tener información dinámica. Por ejemplo, un documento para el monitoreo de una aplicación podría tener un formato como el siguiente:

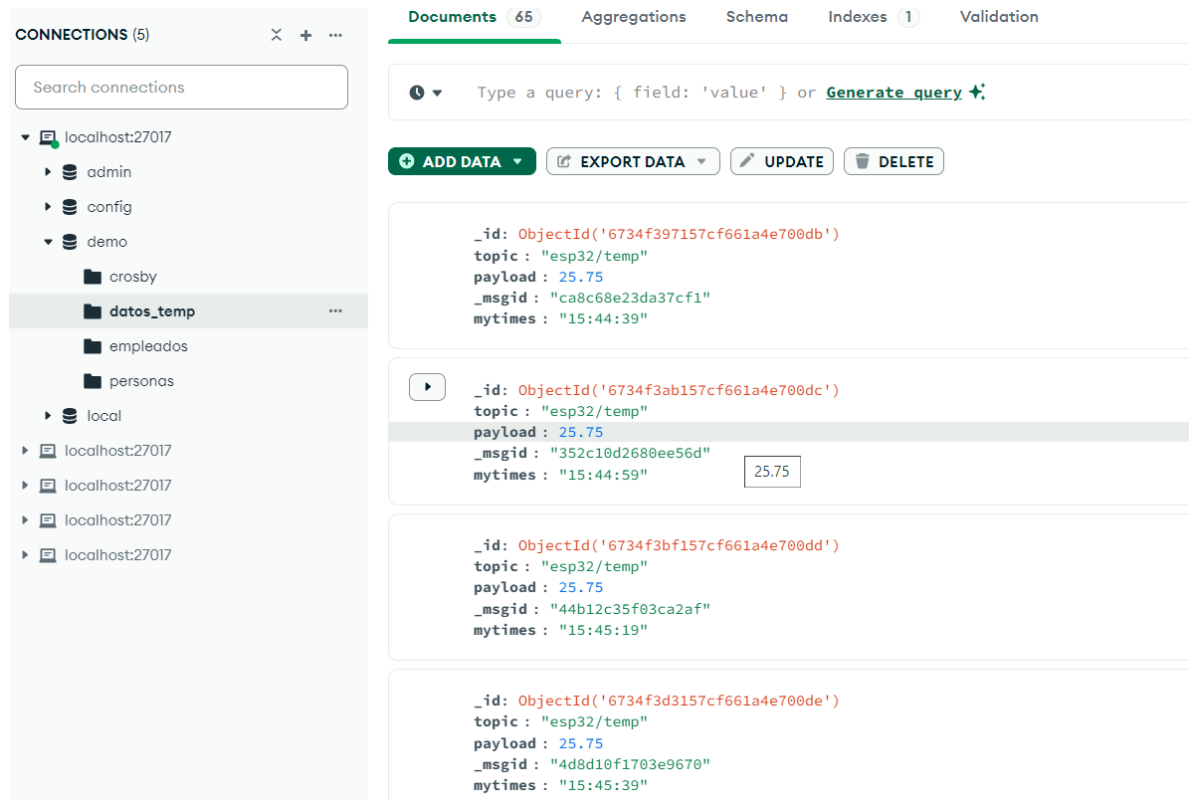


Figura 17 Modelo de datos en MongoDB Compass

Fuente <https://www.mongodb.com/>. Año: 2024



### Ventajas de MongoDB para el monitoreo de datos

- Escalabilidad horizontal: MongoDB es altamente escalable, lo que te permite distribuir grandes volúmenes de datos de monitoreo entre varios servidores o clústeres.
- Consultas complejas: Con MongoDB, puedes realizar consultas complejas sobre los datos para obtener información relevante, como promedios, máximos, mínimos, o incluso realizar análisis en tiempo real utilizando agregaciones.
- Almacenamiento flexible: MongoDB puede almacenar datos de diferentes tipos y estructuras, lo cual es ideal para datos de monitoreo que a menudo varían en formato.
- Alta disponibilidad: MongoDB soporta replicación y tolerancia a fallos, lo que asegura que los datos de monitoreo estén disponibles incluso en caso de fallos de hardware.
- Visualización de datos con Grafana: se pueden integrar herramientas de visualización como Grafana creando dashboard personalizado con indicadores de producción y condiciones de equipo

Tabla 2 Comparativa MongoDB atlas MongoDB compass

Característica	MongoDB Atlas	MongoDB Compass
Propósito	Base de datos en la nube como servicio (DBaaS)	Herramienta de GUI para administrar y visualizar bases de datos MongoDB
Ubicación	Nube (AWS, Google Cloud, Azure)	Local (puede conectarse a MongoDB remoto o localmente)
Gestión de infraestructura	Gestión automática de la infraestructura y escalabilidad	No gestiona infraestructura, solo visualiza y manipula datos
Escalabilidad	Escalabilidad automática en la nube	No tiene función de escalabilidad automática
Seguridad	Cifrado, autenticación y control de acceso avanzados	Dependiente de la configuración de MongoDB
Interfaz de usuario	Interfaz web para administración remota	Interfaz gráfica local para interactuar con bases de datos
Casos de uso	Creación, gestión y escalado de bases de datos en producción	Exploración y consulta de datos, creación de consultas, y administración de bases de datos en desarrollo

Fuente: elaboración propia. Año: 2024



### **3.5.1. Estructura de almacenamiento en MongoDB**

En MongoDB, la estructura de almacenamiento de datos se organiza de manera jerárquica en varios niveles que permiten almacenar datos de una manera eficiente y flexible, a continuación, se detalla los elementos clave que conforman esta estructura:

#### **3.5.1.1. Base de datos (Database)**

Una base de datos en MongoDB es un contenedor de colecciones. MongoDB puede manejar múltiples bases de datos en un solo servidor. Cada base de datos tiene un nombre único dentro del sistema

#### **3.5.1.2. Colección(collection)**

Una colección es un conjunto de documentos dentro de una base de datos. a diferencia de las tablas en base de datos relacionales, las colecciones no requieren un esquema fijo, esto significa que los documentos dentro de una misma colección pueden tener diferentes estructuras, lo que otorga flexibilidad en el diseño de los datos.

#### **3.5.1.3. Documentos (Document)**

Un documento es la unidad básica de almacenamiento en MongoDB. Está representado como un conjunto de pares clave-valor (similar a un objeto JSON). Los documentos pueden tener diferentes estructuras dentro de la misma colección y son almacenados en formato BSON (Binary JSON). Cada documento tiene un campo especial llamado “*id*”, que es un identificador único de ese documento.



### 3.6. Grafana

Grafana es una plataforma de código abierto diseñada para la visualización, monitorización y análisis de datos en tiempo real. Su principal función es permitir a los usuarios crear paneles interactivos y personalizados que representen datos provenientes de diversas fuentes, lo que facilita la toma de decisiones basada en información precisa.

#### Características principales de Grafana:

- Soporte para múltiples fuentes de datos: Grafana es compatible con una amplia variedad de bases de datos y fuentes de datos como:
  - Bases de datos de series temporales (como Prometheus, InfluxDB y Graphite).
  - Fuentes de datos SQL (como MySQL y PostgreSQL).
  - Soluciones en la nube (como AWS, CloudWatch, Google Cloud Monitoring).
- Paneles interactivos: Los usuarios pueden diseñar dashboards visuales que incluyan gráficos, tablas, medidores, mapas de calor y otras visualizaciones. Estas herramientas son altamente configurables y permiten hacer análisis detallados y personalizados.
- Alertas en tiempo real: Grafana tiene un potente sistema de alertas que permite configurar notificaciones basadas en ciertos umbrales o condiciones. Estas alertas pueden enviarse por correo electrónico, Slack, Microsoft Teams, entre otros canales.
- Flexibilidad y personalización: Cada componente dentro de un panel (llamado "widget") puede ajustarse según las necesidades específicas del usuario. Esto incluye opciones como rangos de tiempo, tipos de gráficos y combinaciones de datos.
- Código abierto y gratuito: Aunque hay una versión Enterprise con características avanzadas, la versión básica de Grafana es completamente gratuita, lo que la hace accesible para startups, pequeñas empresas e incluso proyectos personales.

### 3.7. BIBLIOTECAS Y DEPENDENCIAS

Las bibliotecas y dependencias son instancias clave en el desarrollo de tecnologías ayudando a reducir el tiempo y esfuerzo necesario para construir aplicaciones al reutilizar código. Las bibliotecas son colecciones de funciones y recursos predefinidos que los desarrolladores pueden integrar en sus aplicaciones para realizar tareas comunes o específicas, mientras que las dependencias son otros paquetes o bibliotecas de software que una aplicación necesita para funcionar correctamente.



### 3.8. Diseño de sistema

Crear la arquitectura del sistema de monitoreo se debe considerar todos los componentes principales y como interactúan entre sí. Un sistema de monitoreo generalmente se utiliza para supervisar parámetros en tiempo real. Para esta propuesta estamos desarrollando un sistema de monitoreo basado en sensores que captura las variables involucradas en el funcionamiento de un equipo en donde estos son llevados a través de DAQ hacia una unidad central de procesamiento y comunicación junto con plataformas de desarrollo y servicio en la nube.

#### 3.8.1. Arquitectura de hardware

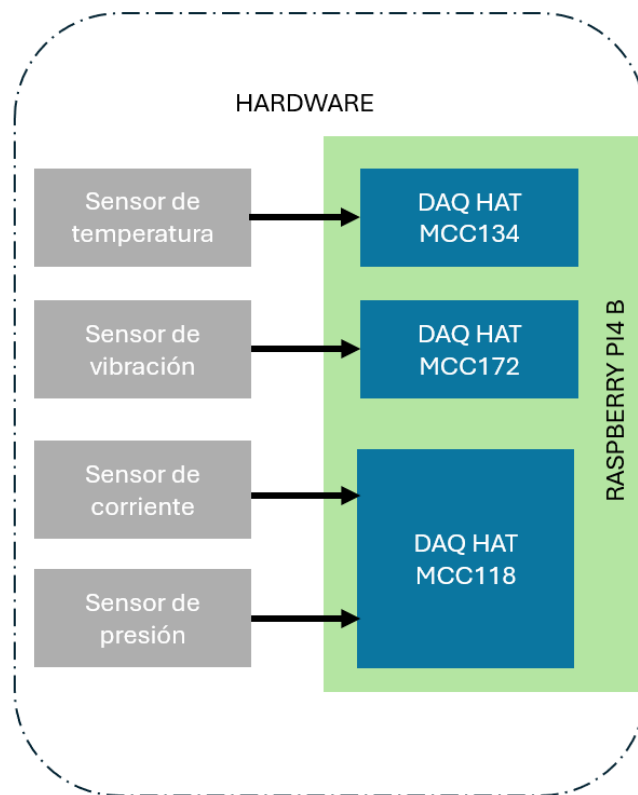


Figura 18 Arquitectura de Hardware

Fuente: elaboración propia. Año: 2024

La arquitectura de hardware está conformada por sensores, DAQ HAT MCC 118, DAQ HAT MCC 134 y DAQ HAT MCC 172 apiladas en la Raspberry PI



### 3.8.2. Arquitectura de software

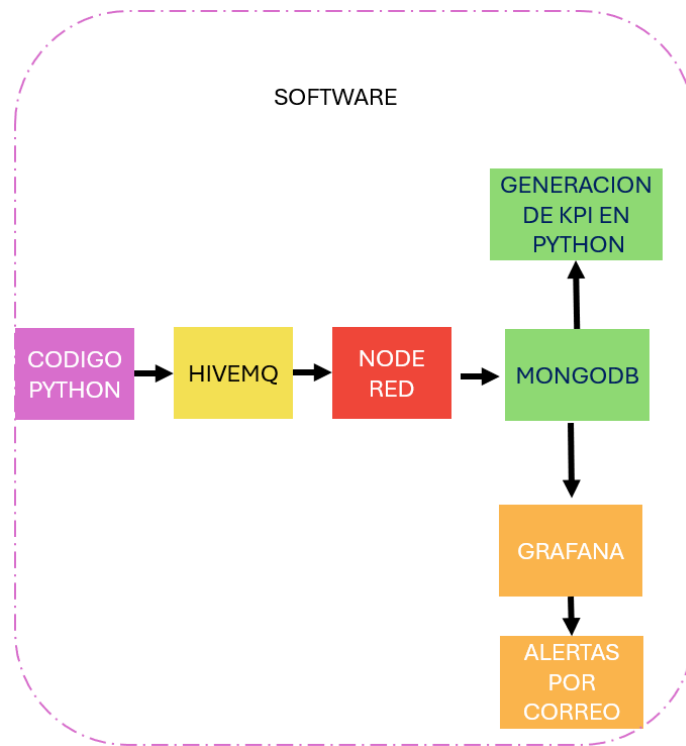


Figura 19 Arquitectura software

Fuente: elaboración propia. Año: 2024

La arquitectura del software está compuesta por diversas plataformas que operan como servicios en la nube, incluyendo Hive MQ Cloud, MongoDB Atlas y Grafana Cloud. Además, Node-RED, que puede ejecutarse en un servidor local, ya sea en una Raspberry Pi o en cualquier ordenador, gestiona y orquesta los flujos de datos de manera eficiente. Además de integrar un sistema de notificación informado a los usuarios sobre eventos importantes, como fallos o parámetros fuera de rango.



## 4. CAPÍTULO: PROTOCOLO MQTT

Este proyecto involucra diversos sistemas y protocolos, así como tecnologías de información y operación, las cuales en conjunto permiten que el proyecto se lleve a cabo. Uno de los protocolos más significativos es MQTT, el cual permite la adquisición de datos de manera remota.

### 4.1. DEFINICIÓN DE MQTT

MQTT es un protocolo de mensajería basado en estándares, o conjunto de reglas para la comunicación de dispositivos o conectar estos a internet. Estandarizado por OASIS e ISO, el protocolo de publicación / suscripción de servidor cliente. Es ligero, abierto, sencillo y está diseñado para ser fácil de implementar proporcionando una forma escalable y fiable de transmitir y recibir datos a través de una red con recursos restringidos y un ancho de banda limitado. En la actualidad MQTT es utilizado por muchas empresas para conectar gran cantidad de dispositivos a internet, entornos restringidos como la comunicación en contextos de máquina a máquina (M2M) e internet de las cosas (IoT), donde se requiere una pequeña huella de código y/o el ancho de banda de la red es escaso.

Se describe como ligero porque fue diseñado para ser simple, eficiente y no requiere muchos recursos. Se creó con un enfoque para enviar pequeñas cantidades de datos a través de redes poco confiables con ancho de banda y conectividad limitado. En comparación con otros protocolos de comunicación, MQTT contiene una huella de código pequeña, una baja sobrecarga y un bajo consumo de energía, sobresaliendo cuando se transfieren datos a través de cable en comparación con protocolos como HTTP. Esto lo hace ideal para su uso en dispositivos con potencia de procesamiento, memoria y duración de la batería ilimitada, como sensores y otros dispositivos IoT

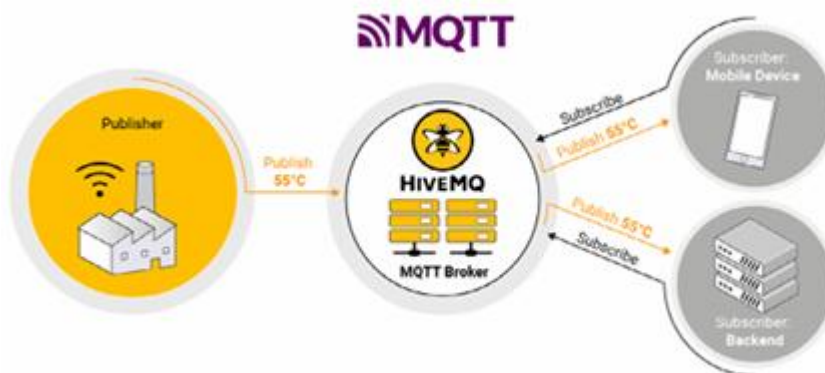


Figura 20 Protocolo MQTT

Fuente: <https://www.hivemq.com>. Año: 2024



## 4.2. RAZONES DE POR QUÉ EMPRESAS UTILIZAN MQTT

- Requiere recursos mínimos, ya que es liviano y eficiente:MQTT minimiza el ancho de banda de la red y los recursos informáticos necesarios para los clientes en el intercambio de datos.
- Admite mensajería bidireccional entre el dispositivo y el broker:permite que los dispositivos envíen y reciban datos desde el servidor ,lo que permite el intercambio de datos bidireccional con otros componentes en la red.
- Puede escalar a millones de dispositivos conectados:permite soportar millones de dispositivos en un ecosistema de IoT o IIoT
- Admite la entrega confiable de mensajes: especifica diferentes niveles de calidad de servicio QoS para garantizar una entrega confiable de mensajes.
- Almacenamiento en bufer de mensajes y reanudación de sesiones:admite sesiones persistentes entre dispositivos y servidores, lo que mejora la confiabilidad de los mensajes al garantizar que los mensajes se entreguen a los clientes incluso después de las desconexiones
- Funciona bien en redes poco fiables.
- Seguridad habilitada: admite el cifrado TLS y protocolos de autenticación para la verificación de los clientes.

### 4.2.1. Prácticas de seguridad en MQTT

MQTT puede ser seguro si se implementan medidas adecuadas como cifrado, autenticación y control de acceso. Sin estas protecciones, puede ser vulnerable a ataques y fugas de información.

- Uso de TLS/SSL (MQTT, puerto 8883): protege la comunicación con cifrado,
- autenticación y autorización: implementar autenticación con usuarios seguros y roles de acceso mediante OAuth, JWT o certificados.
- Control de accesos: configurar permisos para que solo los dispositivos autorizados publiquen o se suscriban a temas específicos.
- Firewall y reglas de red: restringir conexiones al bróker solo desde dispositivos y redes confiables.



### 4.3. FUNCIONAMIENTO Y CONCEPTOS

El servicio de MQTT está formado por los corredores (broker) MQTT y los clientes MQTT, donde el broker MQTT es un intermediario entre los emisores y los receptores, enviando mensajes a los destinatarios adecuados. Los clientes MQTT publican mensajes en el broker y otros clientes se suscriben a temas específicos para recibir mensajes. Cada mensaje mqtt incluye un tema (tópicos) y los clientes se suscriben a los temas (tópicos) de interés. El broker MQTT mantiene una lista de suscriptores y utiliza para entregar los mensajes a los clientes asociados a esos tópicos.

Un broker MQTT también puede almacenar en bufer los mensajes de los clientes desconectados, lo que garantiza una entrega fiable de los mensajes incluso en condiciones de red poco fiables, para lograr esto, MQTT admite tres niveles diferentes de calidad de servicio (QoS) para la entrega de mensajes: 0 (como máximo una vez), 1 (al menos una vez) y 2 (exactamente una vez)

Si bien la mayoría de los corredores comerciales de MQTT ahora admiten MQTT 5, algunos servicios en la nube administrados por IoT aún admiten principalmente MQTT 3.1.1.

#### 4.3.1. Clientes MQTT

Los clientes MQTT publican un mensaje en un intermediario MQTT y otros clientes MQTT se suscriben a los mensajes que desean recibir. Las implementaciones de clientes MQTT requieren una huella mínima, por lo que son muy adecuados para la implementación en dispositivos pequeños y restringidos y son muy eficientes en sus requisitos de ancho de banda.

Existe una gran cantidad de bibliotecas cliente MQTT de código abierto que están disponibles en una variedad de lenguajes de programación. HIVE MQ proporciona bibliotecas de cliente MQTT, que están diseñadas para simplificar la implementación de clientes MQTT ofreciendo a los usuarios funcionalidad, rendimiento, seguridad y confiabilidad de primer nivel algunos de los lenguajes de programación compatibles son: C#, C++, JAVA, WebSockets, python.



### 4.3.2. Broker MQTT

Los broker MQTT reciben los mensajes publicados y envían el mensaje a los clientes suscriptores. Un mensaje MQTT contiene un tema (tópico) de mensaje al que se suscriben los clientes MQTT y los intermediarios MQTT utilizan estas listas de suscripción para determinar los clientes MQTT que recibirán el mensaje.

Los Broker MQTT satisfacen diferentes necesidades, como servicios en la nube de código abierto, comerciales y administrados, por ejemplo HiveMQ ofrece dos ediciones comerciales: HiveMQ Profesional e HiveMQ Enterprise, también ofrece HiveMQ Cloud, un servicio MQTT en la nube administrada, e HiveMQ Community Edition, que es una versión de código abierto. Además HiveMQ ofrece un broker MQTT integrado en HiveMQ Edge, un software de conectividad de código abierto para casos de uso de IIoT.

#### 4.3.2.1. Pasos para comunicación broker MQTT con el cliente MQTT

- Paso 1: instalar el broker MQTT
- Paso 2: conectar los clientes MQTT al broker MQTT
- Paso 3: publicar mensajes MQTT/suscribirse a un tópico MQTT

### 4.3.3. Tópicos, suscripciones, QoS y sesiones persistentes

#### 4.3.3.1. Modelo de mensajería MQTT: temas y suscripciones

El modelo de mensajería MQTT se basa en temas (tópicos) y suscripciones. Los tópicos son cadenas en las que se publican los mensajes y a las que se suscriben. Los temas son jerárquicos y pueden contener varios niveles separados por barras, como una ruta de acceso de archivo como se muestra a continuación

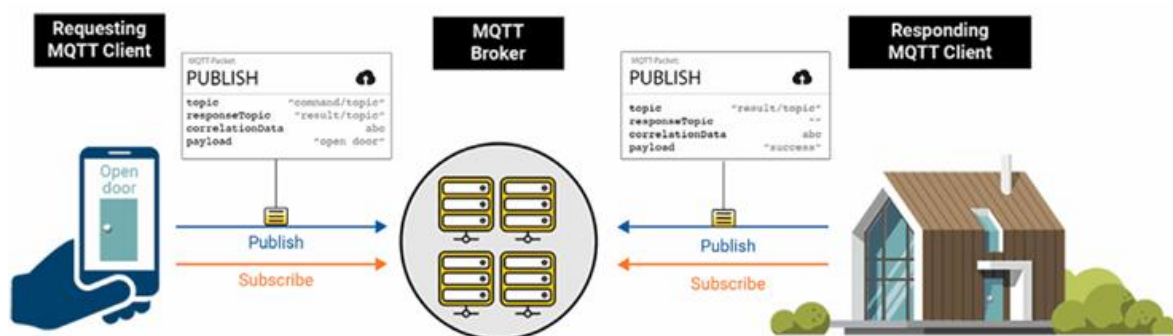


Figura 21 Temas y suscripciones MQTT

Fuente: <https://flexbool.com/>. Año: 2024



#### 4.3.4. Descripción de los niveles de calidad de servicio (QoS)

MQTT admite tres niveles de calidad de servicio(QoS), a continuación se presentará la descripción de cada nivel

- QoS 0: este nivel proporciona una entrega como máximo una vez, donde los mensajes se envían sin confirmación y puede perderse, este es el nivel más bajo de QoS y se utiliza normalmente en situaciones en las que la pérdida de mensajes es aceptable o en las que el mensaje no es crítico
- QoS 1: este nivel proporciona una entrega de al menos una vez, donde los mensajes se confirman y se vuelven a enviar si es necesario, con QoS 1, el editor envía el mensaje al intermediario y espera la confirmación antes de continuar. Si el broker no responde dentro de un tiempo establecido, el editor vuelve a enviar el mensaje, este nivel de QoS se utiliza habitualmente en situaciones en las que la pérdida de mensajes es inaceptable, pero la duplicación del mensaje es tolerable.
- QoS 2: este nivel proporciona una entrega “exactamente una vez”, donde los mensajes se confirman y se reenvían hasta que el suscriptor los recibe exactamente una vez, QoS 2 es el nivel más alto de QoS y se utiliza normalmente en situaciones en las que la pérdida o duplicación de mensajes es completamente inaceptable, Con QoS 2 el editor y el broker participan en un proceso de confirmación de dos pasos, donde el broker almacena el mensaje hasta que el suscriptor lo recibe y lo reconoce. Este nivel se utiliza normalmente para mensajes críticos, como transacciones financieras o alertas de emergencia.

Es importante tener en cuenta que los niveles más altos de QoS suelen requerir más recursos y pueden dar lugar a un aumento de la latencia y de tráfico de red. Como resultado, es importante elegir el nivel de QoS adecuado en función de las necesidades específicas de su aplicación.

MQTT admite la persistencia de mensajes, lo que garantiza que los mensajes no se pierdan en caso de fallo de la red o del servidor



#### 4.3.5. Persistencia de mensajes para la comunicación IoT confiable

La persistencia de mensajes es una característica muy importante en este protocolo, ya que garantiza que los mensajes no se pierdan en caso de fallo de la red o del servidor, en MQTT la persistencia se logra almacenando los mensajes en el broker hasta que se entreguen al suscriptor.

MQTT proporciona tres tipos de persistencia de mensajes:

- No persistente: Esta es la opción predeterminada en MQTT en la que los mensajes no se almacenan en el servidor y se perderá si falla el broker o la red. Adecuado para situaciones en la que el mensaje nos es crítico.
- Persistente en cola: En este modo, los mensajes se almacenan en el servidor hasta que se entregan al suscriptor, si el suscriptor no está disponible, los mensajes se ponen en cola hasta que el suscriptor se vuelva a conectar, este método es útil cuando el suscriptor no siempre está conectado a la red, o si el suscriptor necesita recibir todos los mensajes, incluso si se envían cuando el suscriptor está sin conexión.
- Persistente con confirmación: este modo proporciona el nivel mas alto de persistencia de mensajes. En este modo, los mensajes se almacenan en el servidor hasta que se entregan al suscriptor, y el suscriptor debe acusar recibo del mensaje. Si el suscriptor no acusa recibo, el mensaje se vuelve a enviar hasta que el suscriptor acuse recibo. Este modo es útil cuando es fundamental garantizar que el suscriptor reciba y procese los mensajes.

Para configurar la persistencia de mensajes en MQTT, el software de intermediario utilizado para manejar las conexiones MQTT debe admitir la opción de persistencia elegida. La configuración se puede realizar a través de los archivos de configuración del broker o a través de su interfaz web.

Es importante destacar que la persistencia de mensajes conlleva una compensación en términos de rendimiento y almacenamiento, por lo tanto, es importante definir el nivel de persistencia adecuado en función de los requisitos específicos de la aplicación.



#### 4.3.6. Arquitectura de publicación/suscripción de MQTT (pub/sub)

La arquitectura pub/sub es un patrón de mensajería en arquitectura de software. Permite la comunicación entre diferentes componentes o sistemas de forma desacoplada. En la arquitectura de Pub/Sub hay editores que generan mensajes y suscriptores que reciben esos mensajes. Sin embargo, esta arquitectura se puede implementar utilizando varios protocolos o tecnologías.

La publicación / suscripción presenta tres dimensiones de desacoplamiento para eficiencia óptima:

- Desacoplamiento de espacio: El editor y suscriptor no necesitan conocerse (por ejemplo, no hay intercambio de dirección IP y puerto)
- Desacoplamiento de tiempo: No es necesario que el editor y el suscriptor se ejecuten al mismo tiempo.
- Desacoplamiento de sincronización: No es necesario interrumpir las operaciones de ambos componentes durante la publicación o recepción.

##### Desacoplamiento de Pub/Sub en protocolo MQTT

La arquitectura de Pub/Sub ofrece una alternativa única a los modelos tradicionales de cliente-servidor. En el enfoque de solicitud-respuesta, el cliente se comunica directamente con el punto de conexión del servidor, lo que crea un cuello de botella que ralentiza el rendimiento, por otro lado, el modelo de publicación / suscripción desvincula al editor del mensaje de los suscriptores. El editor y el suscriptor no saben que el otro existe. Como tercer componente, un corredor se encarga de la conexión entre ellos. Este desacoplamiento produce un proceso de comunicación más rápido y eficiente.

MQTT desacopla especialmente al editor y al suscriptor, lo que significa que solo necesitan conocer el nombre del host /IP y el puerto del agente para publicar o recibir mensajes. Además, MQTT permite el desacoplamiento por tiempo, lo que permite que el broker almacene mensajes para clientes que no están en línea.

##### Deben existir dos condiciones para almacenar mensajes:

- El cliente debe haberse conectado con una sesión persistente.
- El cliente debe suscribirse a un tema (tópico) con una calidad de servicio mayor a 0.

Una de las ventajas más significativas de esta arquitectura Pub /Sub es su capacidad para filtrar todos los mensajes entrantes y distribuirlos correctamente a los suscriptores, lo que elimina la necesidad de que el editor y el suscriptor se conozcan mutuamente.



### 4.3.6.1. Función de filtrado de mensajes

En MQTT, el filtrado de mensajes se realiza mediante temas y filtros de temas con comodines. Los suscriptores se suscriben a temas específicos o utilizan comodines para recibir mensajes.

- Comodín de nivel único (+): Representa exactamente un nivel en la jerarquía del tema. Ejemplo: sensores+/temperatura abarca sensores/living/temperatura y sensores/cocina/temperatura.
- Comodín de nivel múltiple (#): Representa cero o más niveles. Ejemplo: sensores/# abarca todos los temas bajo sensores/, como sensores/living/temperatura y sensores/puertas/entrada.

El broker reenvía los mensajes a los suscriptores que tienen filtros de temas coincidentes

## 4.4. ESTRUCTURA Y FORMATO DE PAQUETES MQTT

Los paquetes de control representan la unidad básica de transferencia de datos. Estos paquetes se intercambian entre servidores y clientes MQTT para realizar diversas funciones, incluida las suscripción a temas y la publicación de mensajes.

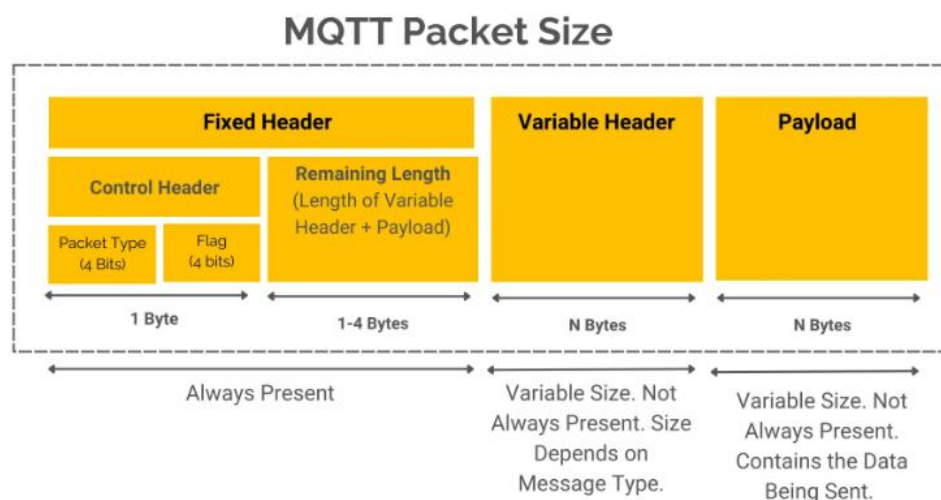


Figura 22 Estructura y formato de paquete MQTT

Fuente: <https://www.hivemq.com/>. Año: 2024





Carga útil (payload): es el mensaje o los datos reales que se envían.

A continuación, se muestra como los diferentes paquetes de control utilizan estos componentes

- CONNECT: utiliza los tres componentes
  - Encabezado fijo: Lo identifica como un paquete CONNECT
  - Encabezado variable: contiene el nombre del protocolo, la versión y los indicadores de conexión
  - Carga útil: incluye ID de cliente, tema de voluntad, nombre de usuario y los indicadores de conexión
- PUBLISH:
  - Encabezado fijo: lo identifica como un paquete PUBLISH
  - Encabezado variable: contiene el nombre del tema y el identificador del paquete.
  - Carga útil: el mensaje real que está publicando

Por ejemplo: un paquete que publica una lectura de temperatura de 25.5 en la trama a/b

Con un identificador de paquete de 1 es el siguiente:

```
Fixed Header: 0011 0000 | 00010100
Variable Header: 00000011 | 'a/b' | 00000001
Payload: "temperature:25.5"
```

Figura 25 Ejemplo paquete MQTT

Fuente: Elaboración propia. Año: 2024

En el encabezado fijo, representa el tipo de mensaje PUBLISH, son las marcas y(10 en decimal) es la longitud restante. 0011 0000 00001010

- SUSCRIBE:
  - Encabezado fijo: lo identifica como un paquete SUSCRIBE
  - Encabezado variable: contiene el identificador del paquete
  - Carga útil: Lista de temas a los que suscribirse y su QoS
- PINGREQ: utiliza solo el encabezado fijo
  - Envía un mensaje vacío solo para verificar si el servicio está funcionando
- DISCONNECT: utiliza solo el encabezado fijo
  - Informa al broker que está mudando



#### 4.4.1. PAQUETES DE CONTROL MQTT

MQTT define 15 tipos de paquetes de control, que se intercambian entre los clientes y los corredores para diversas operaciones, como la suscripción a temas y la publicación de mensajes

FOR CONNECTING/DISCONNECTING	FOR PUBLISHING	FOR SUBSCRIBING/UNSUBSCRIBING
CONNECT	PUBLISH	SUBSCRIBE
CONNACK	PUBACK	SUBACK
DISCONNECT	PUBREC	UNSUBSCRIBE
AUTH (MQTT 5.0 only)	PUBREL	UNSUBACK
PINGREQ	PUBCOMP	
PINGRESP		

Figura 26 Paquetes de control

Fuente: <https://www.hivemq.com/>. Año: 2024

##### 4.4.1.1. MQTT CONNECT y MQTT CONNACK

El paquete CONNECT es enviado por un cliente MQTT, como un sensor, una placa Raspberry PI, ESP32, etc. para iniciar una conexión con el agente MQTT. Incluye información como el ID de cliente, nombre de usuario, la contraseña y varios indicadores de conexión, una vez solicitada la conexión el servidor responde con un paquete CONNACK para indicar el resultado de este intento a la conexión.



#### 4.4.1.2. MQTT PUBLISH

Este paquete se utiliza para enviar mensajes desde el cliente al servidor. Además, se utilizan otros 4 paquetes para reconocer los mensajes QoS 1 y 2.

Indicadores de control

- Indicador DUP(duplicado)se utiliza en el paquete PUBLISH. Indica si el mensaje es duplicado o se vuelve a publicar. Se establece en 1 cuando el mensaje se vuelve a entregar debido a la falta de comunicación
- Indicador QoS:se utilizan en el paquete PUBLISH. Utiliza dos bits e indica el nivel de QoS del mensaje.
- Indicador RETAIN: se utiliza en el paquete PUBLISH. Indica si el intermediario debe retener el mensaje. Cuando se establece en 1,el corredor almacenara el mensaje y lo enviara a los futuros suscriptores
- Indicadora sesión limpia:se utiliza en el paquete CONNECT. Indica si el cliente desea establecer una sesión persistente o no
- Will flag:se utiliza en el paquete CONNECT. Indica si un mensaje de testamento debe almacenarse en el servidor
- Indicador de nombre de usuario: se utiliza en el paquete CONNECT e indica si hay un nombre de usuario presente en la carga útil
- Indicador de contraseña: se utiliza en el paquete CONNECT e indica si hay una contraseña presente en la carga útil

#### 4.4.1.3. MQTT SUBSCRIBE, UNSUBSCRIBE, SUBACK Y UNSUBACK

El paquete SUBSCRIBE permite al cliente suscribirse a temas específicos. Por ejemplo, un PLC SIEMENS (cliente) se suscribe a un tema “temperatura/actualizaciones”. Por el contrario, el paquete UNSUSCRIBE se utiliza para cancelar estas suscripciones. El servidor envía paquetes SUBACK y UNSUBACK para confirmar los resultados de estas solicitudes de suscripción n y cancelación de suscripción, respectivamente.

- El broker envía SUBACK en respuesta a un paquete SUBSCRIBE, confirmando la suscripción y especificando los niveles de QoS concedidos para cada tema.
- UNSUBACK es enviado por el broker en respuesta a un paquete UNSUBSCRIBE, conformando la cancelación de la suscripción

#### 4.4.1.4. MQTT DISCONNECT

Si es necesario finalizar la comunicación, ya sea debido a un apagado rutinario o aun erro, el cliente y el servidor pueden enviar un paquete DISCONNECT para cerrar la conexión de red.



## 5. ADQUISICIÓN REGISTRO Y SUPERVISIÓN DE DATOS

Este capítulo se analizarán las condiciones claves, las variables de medición más relevantes y el flujo de datos necesarios para una gestión eficaz.

### 5.1. CONSIDERACIONES PARA LA APLICACIÓN

#### 5.1.1. Teorema de Nyquist-Shannon

El teorema de Nyquist-Shannon es un principio fundamental en la teoría de la señal y el procesamiento de señales, que establece las condiciones bajo las cuales una señal analógica puede ser muestreada sin pérdida de información. Este describe que una señal continua puede ser reconstruida de manera exacta a partir de sus muestras, es muestreada una frecuencia al menos el doble de su frecuencia máxima. Este concepto es conocido como frecuencia de muestreo de Nyquist.

Consideraciones de aplicación:

- Frecuencia de la señal: Sea una señal analógica con una frecuencia máxima de  $f_{max}$ , que es la mayor frecuencia presente en la señal.
- Frecuencia de muestreo: Para poder reconstruir la señal sin pérdida de información, debemos muestrearla a una frecuencia  $f_s$ , tal que:

$$f_s \geq 2 \cdot f_{max}$$

Esto significa que la frecuencia de muestreo debe ser al menos el doble de la frecuencia máxima de la señal.

- Condición de Nyquist: la frecuencia mínima de muestreo  $f_s$  es conocida como frecuencia de Nyquist:

$$f_{Nyquist} = 2 \cdot f_{max}$$

- Si se cumple con la frecuencia de muestreo inferior a la de Nyquist, ocurrirá el fenómeno de aliasing, lo que significa que algunas frecuencias de la señal original se solapan, haciendo imposible la reconstrucción exacta de la señal.

### 5.2. PUNTOS DE MEDICIÓN

#### 5.2.1.1. Termocuplas

Estos sensores permitirán adquirir los datos de temperatura en un intervalo de tiempo determinado mediante el módulo DAC HAT 134, esto nos permitirá posteriormente almacenar el base de datos y supervisar las condiciones en la que se encuentra el equipo. Las termocuplas definidas se escogen en función de la aplicación y posibles partes críticas.



### 5.2.1.2. Puntos clave para la medición

- Cojinetes y rodamientos, motores, sistemas de lubricación: TIPO K (níquel-cromo/níquel-aluminio)
  - Rango de temperaturas: -200 °C a 1370 °C
  - Estas ofrecen un buen rango de temperatura y son económicas, estas piezas pueden experimentar un aumento de temperatura debido a la fricción en donde está termocupla puede detectar cualquier anomalía.
- Sellos y componentes sensibles a la temperatura: TIPO T (cobre-Constantán).
  - Rango de temperaturas: -200 °C a 350 °C.
  - Son precisas a bajas temperaturas y son adecuadas para monitorear zonas que requieren alta sensibilidad y son útiles en aplicaciones donde los componentes pueden estar sujetos a variaciones térmicas rápidas
- Líneas de vapor: TIPO K (níquel-cromo-silicio/níquel-silicio)
  - Rango de temperatura: -200 °C a 1300 °C
  - Estas termocuplas son altamente resistentes a la corrosión y la oxidación, lo que las hace ideales para sistemas ambientales agresivos.

### 5.2.1.3. Determinación de tiempo de muestreo

En el monitoreo de temperatura se debe considerar el cómo obtener muestras discretas de esta variable continua en el tiempo, como es el caso de la temperatura. Quiriendo convertir esta variable continua a una secuencia de datos discretos, el cual será posteriormente procesado por el DAC HAT MCC 134 y enviados en forma discreta mediante el protocolo MQTT hacia la nube.

#### Consideraciones para determinar el tiempo de muestreo:

- Señal de temperatura: esta variable continua en función del tiempo en donde se determinó una frecuencia máxima  $f_{max}$  en condiciones normales de 1/120 Hz.
- Frecuencia de muestreo: según el teorema, la frecuencia de muestreo debe ser al menos el doble de la frecuencia máxima de los cambios de temperatura, en este caso:

$$f_s \geq 2 \cdot f_{max}$$
$$f_s \geq 2 \cdot 1/120Hz$$
$$f_s \geq 1/120hz$$
$$f_s \geq 120seg$$



Tabla 3 Puntos de medición de temperatura

Puntos de medición	TERMOCUPLA	TIEMPO DE MUESTREO	RANGO DE OPERACIÓN
MOTOR	TIPO K	120seg	20°C-80°C
SOPORTES	TIPO K	120seg	30°C-80°C
REDUCTORES(EXTERNA)	TIPO K	300seg	30°C-70°C
LÍNEA DE VAPOR	TIPO N	60seg	190°C

Fuente: Elaboración propia. Año: 2024

### 5.2.2. Sensores de presión

Tabla 4 características del transmisor de presión A-10

CARACTERÍSTICAS DEL TRANSMISOR DE PRESIÓN A-10	
RANGO DE MEDICIÓN	0-16 Bar
NO LINEALIDAD	0.25% a 0.5%
SALIDA	3 hilos 0- 10 V
ALIMENTACIÓN	8-30VDC
CONSUMO	8 mA
CONEXIÓN ELÉCTRICA	Conector angular DIN 175301-803A
CONEXIÓN A PROCESO	G1/4 B EN 837 ACERO INOX/COBRE

Fuente: elaboración propia. Año: 2024

Tabla 5 Puntos de medición de presión

Puntos de medición	Sensor	TIEMPO DE MUESTREO	RANGO DE OPERACIÓN
Línea de vapor	TRANSMISOR DE PRESIÓN A10	60 seg	6 Bar
Lubricación en la caja reductora	TRANSMISOR DE PRESIÓN A10	600 seg	2- 5 Bar

Fuente: elaboración propia. Año: 2024



### 5.2.3. Vibraciones

Los estándares de vibración en máquinas rotativas son cruciales para garantizar el funcionamiento seguro y eficiente de los equipos. Estos estándares ayudan a definir los límites aceptables de vibración y proporcionan directrices para el monitoreo de la condición de las máquinas, como motores, bombas, generadores, turbinas y compresores, entre otros.

La vibración en las máquinas rotativas puede ser indicativa de varios problemas, como desbalance, desalineación, desgaste de rodamientos o fallos en otros componentes. Los estándares de vibración proporcionan límites que ayudan a identificar cuándo las vibraciones alcanzan niveles que podrían comprometer la integridad del equipo.

#### 5.2.3.1. Principales Estándares y Guías de Vibración

##### ISO 10816: Normas de vibración para máquinas rotativas

- La ISO 10816 es uno de los estándares más utilizados para la evaluación de las vibraciones en máquinas rotativas. Esta norma define los límites de vibración de acuerdo con el tipo de máquina, el tamaño y la aplicación.
- ISO 10816 se clasifica en varias partes que cubren diferentes tipos de maquinaria:
  - ISO 10816-1: Clasificación general para máquinas con potencias nominales mayores a 0.5 kW.
  - ISO 10816-3: Para máquinas de baja potencia (hasta 15 kW), tales como motores eléctricos.
  - ISO 10816-4: Para máquinas de gran potencia (más de 300 kW).

Los valores de vibración se clasifican según la velocidad de vibración RMS (Root Mean Square) en tres frecuencias principales:

- Vibraciones de baja frecuencia (0-100 Hz): Relacionadas con componentes como el desbalance o la desalineación.
  - Vibraciones de frecuencia intermedia (100-1000 Hz): Asociadas con problemas de rodamientos o defectos estructurales.
  - Vibraciones de alta frecuencia (>1000 Hz): Relacionadas con fallas en los rodamientos, engranajes o defectos en las piezas pequeñas.
- ISO 7919: Vibraciones de máquinas rotativas montadas sobre bases rígidas
    - La ISO 7919 proporciona límites específicos para la vibración en máquinas rotativas, especialmente para aquellas montadas sobre bases rígidas, como turbinas y generadores.
    - Los niveles de vibración se especifican en términos de desplazamiento, velocidad y aceleración.



Esta norma establece límites para la aceleración en unidades de  $m/s^2$  o  $g$  (gravedad), la velocidad de vibración en  $mm/s$  o  $in/s$  y el desplazamiento en micrómetros ( $\mu m$ ).

- API 610: Estándares para bombas centrífugas
  - La API 610 es un estándar específico para bombas centrífugas y define los límites de vibración permitidos en diferentes condiciones de operación.
  - Es particularmente útil para las bombas en aplicaciones de procesos industriales, como las utilizadas en plantas petroquímicas.
  - En términos generales, la API 610 especifica que las vibraciones de las bombas deben mantenerse por debajo de valores como  $3.5 mm/s$  RMS para evitar daños a los componentes críticos.
- BS 4675: Estándares británicos de vibración
  - El estándar BS 4675 también proporciona guías sobre los límites de vibración en máquinas rotativas, similar a los estándares de la ISO, y se enfoca en la evaluación de la vibración de motores y generadores en instalaciones industriales.

### 5.2.3.2. Valores de vibración aceptables según ISO 10816:

Los límites de vibración aceptables para máquinas rotativas pueden variar dependiendo del tipo de máquina y su aplicación, pero a continuación se muestran los valores típicos establecidos en la ISO 10816 para la velocidad de vibración en términos de  $mm/s$  RMS para diferentes tipos de máquinas:

Nivel de vibración	de	Descripción	Velocidad de vibración ( $mm/s$ RMS)
Nivel (Excelente)	A	Máquinas funcionando en condiciones ideales	$\leq 2.8$
Nivel (Buena condición)	B	Operación normal, sin problemas significativos	$2.8 - 4.5$
Nivel (Condición aceptable)	C	Comienza a notarse la vibración, pero no hay riesgo inmediato	$4.5 - 7.1$
Nivel (Condición pobre)	D	Se debe investigar, posible fallo en los componentes	$7.1 - 11.4$
Nivel (Falla inminente)	E	Alta probabilidad de daño a los componentes o fallas graves	$> 11.4$

Tabla 6 Valores de vibración aceptables según ISO 10816

Fuente: <https://dspanalytic.com/> Año: 1995



### 5.2.3.3. Valores típicos según la frecuencia de vibración:

Los valores de vibración también pueden variar según la frecuencia de la señal de vibración, generalmente medida en aceleración ( $m/s^2$ ) o desplazamiento ( $\mu m$ ). A continuación, se indican los niveles típicos de vibración en máquinas rotativas:

1. Máquinas pequeñas (motores pequeños, ventiladores, etc.):
  - Desplazamiento: hasta 0.5 mm
  - Velocidad: hasta 5 mm/s RMS
  - Aceleración: hasta 2  $m/s^2$
2. Máquinas medianas (bombas, compresores):
  - Desplazamiento: hasta 1 mm
  - Velocidad: hasta 7 mm/s RMS
  - Aceleración: hasta 5  $m/s^2$
3. Máquinas grandes (turbinas, generadores):
  - Desplazamiento: hasta 2 mm
  - Velocidad: hasta 10 mm/s RMS
  - Aceleración: hasta 8  $m/s^2$

### 5.2.3.4. Tipos de defectos que las vibraciones pueden detectar:

1. Desbalance: Causa vibraciones a la frecuencia de rotación (1X RPM).
2. Desalineación: Provoca vibraciones a la frecuencia de rotación y frecuencias armónicas.
3. Desgaste de rodamientos: Produce frecuencias características que varían dependiendo del tipo de rodamiento.
4. Fallos en engranajes: Pueden generar vibraciones a frecuencias altas, especialmente en las frecuencias características de los dientes del engranaje.

Tabla 7 Puntos de medición de vibración

Puntos de muestreo	sensor	frecuencia de muestreo	ejes
Motor 40HP	AC102-1A	1KHz	triaxial
Caja reductora	AC102-1A	1KHz	triaxial
Soportes estructurales	AC102-1A	500hz	uniaxial
Soporte de rodamientos	AC102-1A	500hz	triaxial

Fuente: elaboración propia. Año: 2024



## 5.2.4. CORRIENTE

Tabla 8 Puntos de medición de corriente

Puntos de muestreo	Sensor	Tiempo de muestreo	corrientes de operación
Motor 40HP	OPCT16ATL	60seg	In=60A Ip=451A-Io=28.4A
Motor bomba 1HP	OPCT16ATL	300seg	In=1.98A- Ip=9.9A-Io=1.3A
Motor tornillo alimentador 10HP	OPCT16ATL	60seg	In=14.7A- Ip=121A-Io=6.9A

Fuente: elaboración propia. Año: 2024

## 5.3. ESPECIFICACIONES TÉCNICAS DE LOS SENSORES

### 5.3.1. Sensores de temperatura

Termocupla tipo K

Termopar Tipo K, sigue siendo el termopar más utilizado para aplicaciones industriales. Está concebido principalmente para atmosferas neutras. De hecho, se debe tener especial cuidado a la hora de proteger el sensor cuando se utiliza en otro ambiente distinto. La temperatura máxima en continuo se aproxima a los 1100 °C y en períodos cortos de tiempo alcanza los 1200 °C. Por encima de los 800 °C la oxidación causa des calibración de forma más pronunciada.

Tabla 9 características termocupla tipo K

Especificación	Detalle
Tipo de termocupla	K (Níquel-Cromo)
Rango de temperatura	-200 °C a 1.260 °C (dependiendo del material de protección)
Precisión	±1,5 °C o ±0,4% (según la norma IEC 60584, Clase 1)
Respuesta	Rápido, ideal para aplicaciones de cambio rápido de temperatura.
Material del cable	Níquel-Cromo (90% níquel, 10% cromo)
Durabilidad	Alta en condiciones normales, pero limitada en ambientes con alta corrosión

Fuente: <https://www.tc-sa.es/>. Año: 2024



Tabla 10 Datos técnicos termocupla tipo K

Combinación de Conductores		Generadas por grado celsius con la unión de referencia a 0°C $\mu V$			aproximado de temperatura de la unión de medida		Tolerancias de salida de termopar a IEC 60584-1			
Conductor +	Conductor -	100°C	500°C	1000°C	Continuo	Puntual	Tipo	Clase 1	Clase 2	Clase 3
Níquel - Cromo También conocido como: Chromel™, Thermokanthal KP®, NiCr, T1®, Tophel™	Níquel - Aluminio (magnetic) También conocido como: Ni-Al, Alumel™, Thermokanthal KN™, T2™, NiAl™	42	43	39	0 a +1100°C	-180 a +1350°C	Rango de Temperatura Tolerancia Rango de Temperatura Tolerancia	-40°C a +375°C $\pm 1.5^\circ C$ 375°C a 1000°C $\pm 0.004 \cdot  t $	-40°C a +333°C $\pm 2.5^\circ C$ 333°C a 1200°C $\pm 0.0075 \cdot  t $	-167°C a +40°C $\pm 2.5^\circ C$ -200°C a -167°C $\pm 0.015 \cdot  t $

Fuente: <https://www.tc-sa.es/>. Año: 2024

### Termocupla tipo N

Termopar Tipo N, etiquetado como el revolucionario sustituto del termopar tipo K (el más común para uso industrial), pero solventando sus desventajas. El tipo N (Nicrosil-Nisil) muestra una mayor resistencia a la deriva relacionada con la oxidación a temperaturas altas y a otras inestabilidades propias del Tipo K en particular, pero también, hasta cierto punto, en relación con los otros termopares de metal común. Por lo tanto, puede alcanzar temperaturas más altas que el tipo K (1280°C y mayores para periodos cortos de tiempo).

Tabla 11 Datos técnicos termocupla tipo N

Combinación de Conductores		Cambio aproximado de las F.E.M. Generadas por grado celsius con la unión de referencia a 0°C $\mu V$			Rango de funcionamiento aproximado de temperatura de la unión de medida		Tolerancias de salida de termopar a IEC 60584-1			
Conductor +	Conductor -	100°C	500°C	1000°C	Continuo	Puntual	Tipo	Clase 1	Clase 2	Clase 3
Níquel - Cromo - Silicio También conocido como: Nicrosil	NÍQUEL - SILICIO - MAGNESIO También conocido como: Nisil	30	38	39	0 a +1150°C	-270 a +1300°C	Rango de Temperatura Tolerancia Rango de Temperatura Tolerancia	-40°C a +375°C $\pm 1.5^\circ C$ 375°C a 1000°C $\pm 0.004 \cdot  t $	-40°C to +333°C $\pm 2.5^\circ C$ 333°C a 1200°C $\pm 0.0075 \cdot  t $	-167°C a +40°C $\pm 2.5^\circ C$ -200°C a -167°C $\pm 0.015 \cdot  t $

Fuente: <https://www.tc-sa.es/>. Año: 2024

### 5.3.2. sensor de presión WIKA A-10

De construcción compacta, puede integrarse en una amplia gama de diseños de máquinas. Los numerosos rangos de medición, así como los rangos de medición especiales, contribuyen a su aplicabilidad universal.

Tabla 12 Datos técnicos sensor de presión wika A-10

<b>RANGO DE MEDICIÓN</b>	0-16 Bar
<b>NO LINEALIDAD</b>	0.25% a 0.5%
<b>SALIDA</b>	3 hilos 0-10V
<b>ALIMENTACIÓN</b>	8-30VDC
<b>CONSUMO</b>	8mA
<b>CONEXIÓN ELÉCTRICA</b>	Conector angular DIN 175301-803A
<b>CONEXIÓN A PROCESO</b>	G1/4 B EN 837 ACERO INOX/COBRE

Fuente: <https://www.wika.cl/>. Año: 2024



### 5.3.3. Sensor de corriente

El OPCT16ATL 100/3,3V es un sensor de transformador de corriente de núcleo dividido.

Tabla 13 Datos técnicos sensor de corriente

Rango de medición	0 a 100 A
Voltaje de salida	3,3 V
Corriente de salida	50 mA
Precisión	±0,5%
Respuesta temporal	50 ms
Conexión	Pinza amperimétrica de tipo abierto (10A - 2,5 mA)

Fuente: <https://es.aliexpress.com/>. Año: 2024

### 5.3.4. Sensor de vibración

El acelerómetro AC102-1A es un sensor diseñado para medir la aceleración en aplicaciones industriales y científicas. Este tipo de dispositivo convierte la aceleración física en una señal eléctrica que puede ser utilizada para diversas funciones de monitoreo y análisis.

Tabla 14 Datos técnicos sensor de vibración

Características técnicas	Estándar	Métrico
Número de pieza	AC102	M/ø M8/AC102
Sensibilidad (±10%)		100 mV/g
Respuesta de frecuencia (±3dB)	30-810.000 cpm	0,5-13500 Hz
Rango dinámico		± 80 g, pico *Vfuente ≤ 5 V, polarización de 12 V
<b>Eléctrico</b>		
Tiempo de asentamiento		<2,5 segundos
Fuente de voltaje		18-30 VCC
Excitación de corriente constante		2-10 mA
Ruido espectral a 10 Hz		14 µg/√Hz
Ruido espectral a 100 Hz		2,3 µg/√Hz
Ruido espectral a 1000 Hz		2 µg/√Hz
Impedancia de salida		<100 ohm
Tensión de salida de polarización		10-14 VCC
Aislamiento de casos		>10 <sup>8</sup> ohm
Peso	3,2 onzas	90 gramos
Material de la caja		Acero inoxidable 316L
Rosca de montaje		Orificio roscado ciego de 1/4- 28
Conector (no integral)		2 pines MIL- C-5015
Frecuencia resonante	1.380.000 cpm	23000 Hz
Par de montaje	2 a 5 pies- libras.	2,7 a 6,8 Nm
Herrajes de montaje incluidos	Perno de 1/4- 28	Perno adaptador M6x1 o M8x1,25
Certificado de calibración		CA10

Fuente: <https://www.ctconline.com/>. Año: 2024



## 5.4. FLUJO DE DATOS

El flujo de datos permite estructurar los movimientos de la información desde su origen hasta su destino dentro del sistema de monitoreo, es decir, desde que se recopilan hasta que se almacenan o procesan para visualización de estos. El flujo de datos es fundamental en este proyecto debido a que, sin él, no podríamos supervisar continuamente las variables e indicadores de interés. Este flujo de datos se desglosa en las siguientes etapas clave:

### 5.4.1. Adquisición de Datos:

Mediante DAQ HAT de MCC se adquieren los datos, estas son apiladas en Raspberry Pi 4 y direccionadas mediante puentes, las variables adquiridas son procesadas y acondicionadas en las propias MCC las cuales cuentan memorias (información crucial para el funcionamiento y operación), convertidores ADC y en el caso de MCC134 Y MCC 172 microcontroladores propios. La comunicación con Raspberry Pi se lleva a cabo mediante los pines GPIO utilizando los protocolos I2C o SPI.

Para variables de temperatura se utiliza DAQ HAT MCC 134, para señales de voltaje de hasta +- 10 V (corriente aC y presión) se utiliza DAQ HAT MCC 118 y para señales de vibración se utiliza DAQ HAT MCC 172.

Para realizar la adquisición de datos utilizando DAQ HAT de MCC y una Raspberry pi mediante Python, se deben seguir ciertos pasos como:

#### 5.4.1.1. Habilitar puerto GPIO e interfaz I2C y SPI

Habilitar estas interfases es esencial para instalar y usar estos dispositivos DAQ HAT, ya que los protocolos de comunicación I2C y SPI son utilizados para la transferencia eficiente de información entre Raspberry Pi y el hardware de adquisición. Si estos puertos no están habilitados, el dispositivo no podrá establecer la comunicación adecuada

- I2C: es un bus de datos serial de dos hilos, usado para la comunicación a corta distancia, en el proyecto se usa para la comunicación entre Raspberry PI con la memoria HAT EEPROM
- SPI: Es un bus de comunicación más rápido en comparación con I2C y es ideal para una alta transferencia de datos.

En el sistema operativo de Ubuntu se pueden habilitar las interfaces I2C y SPI con los siguientes comandos:

Instalar herramientas I2C

```
sudo apt update
sudo apt install i2c-tools
```

habilitar I2C a través de la herramienta de configuración del sistema

```
sudo apt install raspi-config
sudo raspi-config
```

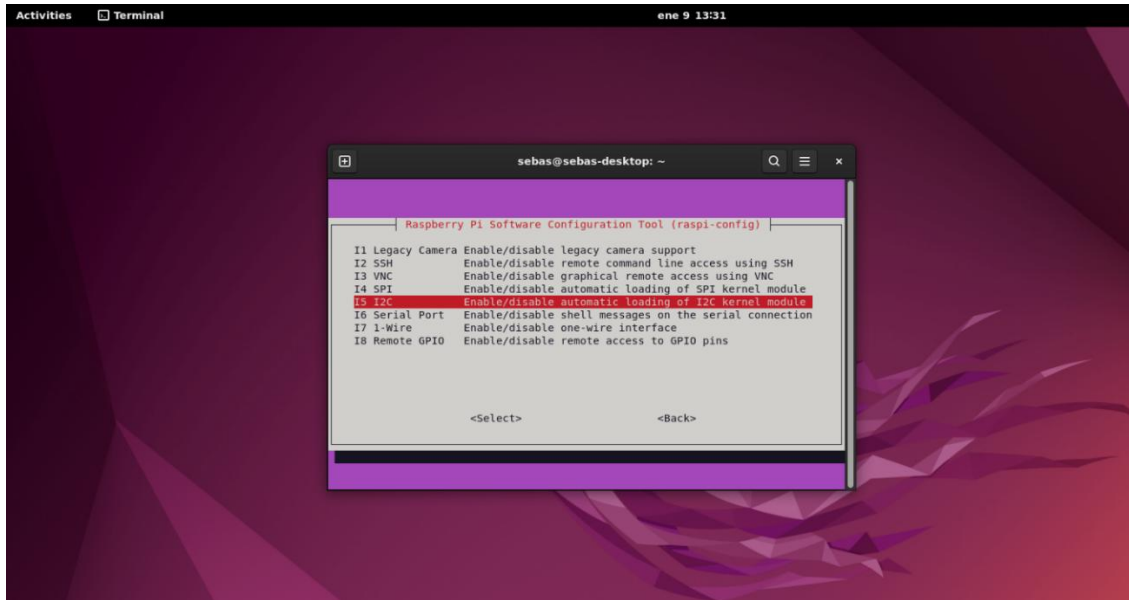


Figura 27 Habilitación de puertos en ubuntu

Fuente: Elaboración propia. Año: 2024

#### 5.4.1.2. Instalación de DAQ HAT, librerías y código de lectura de variables

Ver ANEXO A y B



### 5.4.2. Transmisión de datos

Los datos recopilados mediante un código en lenguaje Python ejecutando diferentes funciones como: de conexión al bróker MQTT y de lectura de datos desde las DAQ HAT (sistema de adquisición de datos). Son enviados al bróker de HiveMQ Cloud mediante cifrado TLS utilizando el puerto 8883, autenticación y autorización, configurando credenciales y permisos. HiveMQ Cloud es considerado una plataforma como servicio (PaaS) en lo que es computación en la nube.

Los datos en HiveMQ Cloud son adquiridos por Node RED una Plataforma de desarrollo basada en flujos que actúa como Cliente MQTT. Además de ser utilizada para desplegar y ejecutar aplicaciones en la nube, Node-RED se suscribe a los tópicos de interés para recibir actualizaciones. Para ello, se configura un nodo MQTT ingresando la dirección del servidor MQTT de HiveMQ Cloud y las credenciales como el nombre de usuario y la contraseña.

Editar nodo mqtt in

Eliminar Cancelar Hecho

Propiedades

Servidor: 6e10b8f0f10841239ecf63a7847028f4.s1.eu.hivemq.cloud:8883

Acción: Suscríbete a un solo tema

Tema: termocupla1/temp

CdS: 1

Salida: un objeto JSON

Nombre: termocupla1

Habilitado

Figura 28 Suscripción Node-red a HIVE MQ

Fuente: elaboración propia. Año: 2024



Editar nodo mqtt in > **Editar nodo mqtt-broker**

Eliminar Cancelar Actualizar

**Propiedades**

Nombre: Nombre

**Conexión** Seguridad Mensajes

Servidor: 6e10b8f0f10841239ecf63a7847028f4.s1.eu.hi Puerto: Mensajes

Conectar automáticamente

Utilizar TLS Añadir nuevo tls-config...

Protocolo: MQTT V3.1.1

ID Cliente: Dejar en blanco para auto generado

Mantener activo: 60

Sesión  Usar sesión limpia

Habilitado 5 En todos los flujos

Figura 29 suscripción Node-red a HIVE MQ 2

Fuente: elaboración propia. Año: 2024

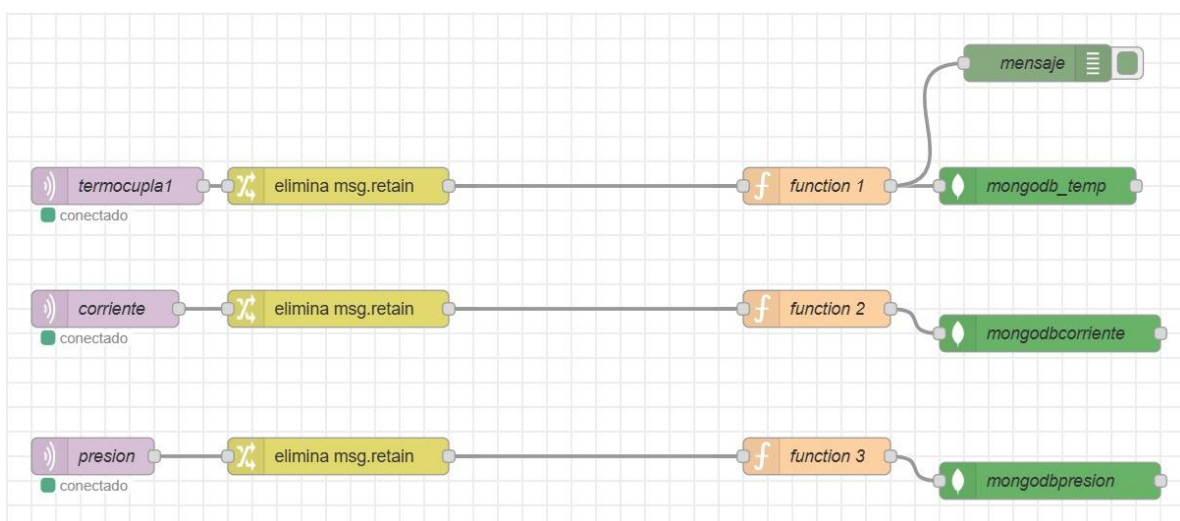


Figura 30 Enlace de datos Node-red

Fuente: elaboración propia. Año: 2024



### 5.4.3. Procesamiento de datos

#### 5.4.3.1. Procesamiento de datos Node-RED

Los datos integrados en la plataforma Node-RED son procesados mediante transformación, filtrado y validación utilizando los nodos Función y Change. Estos nodos permiten modificar y reorganizar el contenido del mensaje, así como transformar los datos a formato JSON e incluir un valor de fecha y hora del flujo.

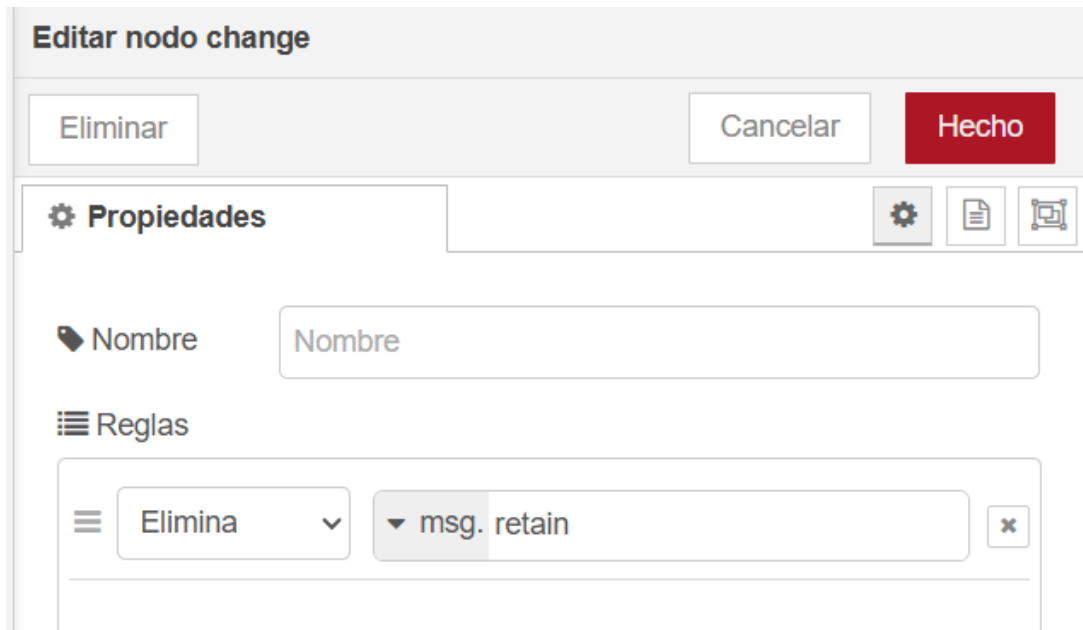


Figura 31 Procesamiento de datos 1

Fuente: elaboración propia. Año: 2024

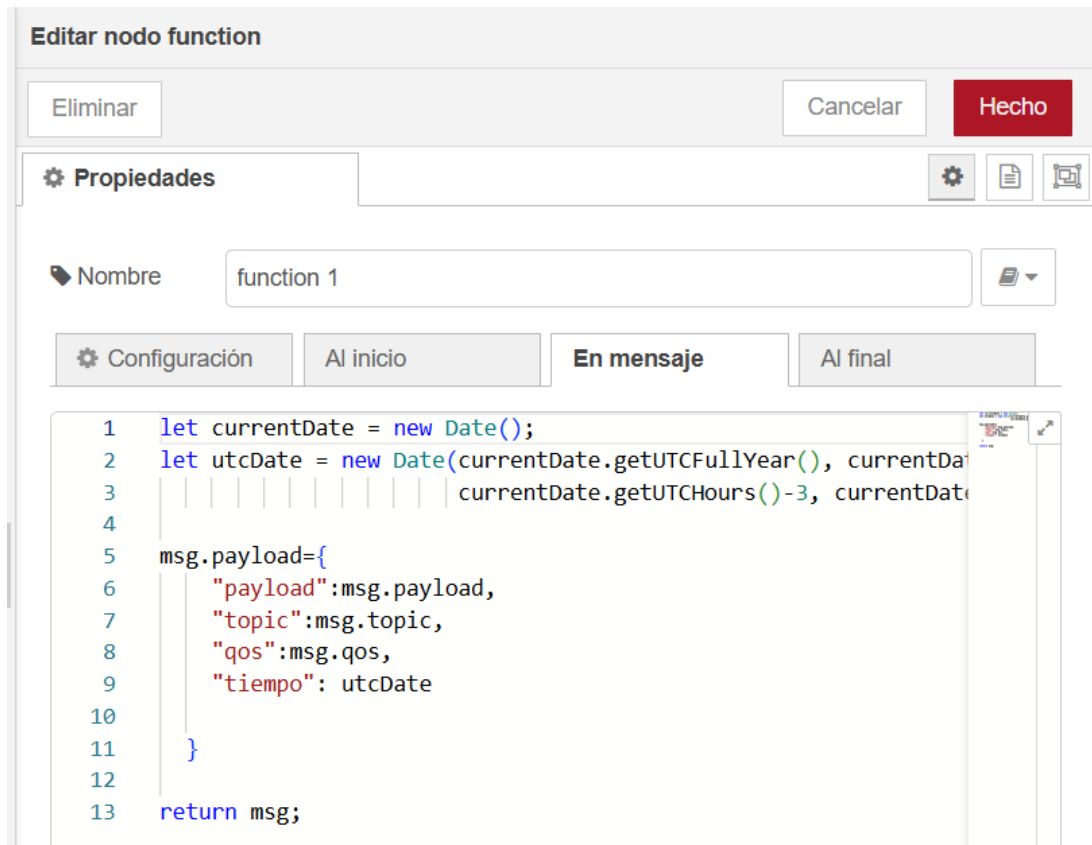


Figura 32 Procesamiento de datos 2

Fuente: elaboración propia. Año: 2024

#### 5.4.3.2. KPIs de mantenimiento en Python

El procesamiento de los datos proporcionados por Node-RED hacia la base de datos de MongoDB Atlas se realiza a partir de diferentes pasos diseñados para calcular y almacenar KPIs relacionados con el tiempo de operación y las detenciones de la máquina, ya sea por fallas o detenciones programadas. El proceso se basa en analizar registros almacenados en la base de datos de MongoDB y calcular KPIs de MTTF (tiempo medio hasta la falla), MTTR (tiempo medio de reparación) y MTBF (tiempo medio entre fallas) utilizando el código que se describe a continuación y tomando las siguientes consideraciones:

- Los datos se extraen de la base de datos de MongoDB Atlas ingresando la URL de conexión seleccionando la colección correspondiente para el procesamiento de los datos.
- La colección de interés es 'corriente' ya que permite determinar si el motor principal del equipo está en funcionamiento o se encuentra detenido.
- De los datos procesados se determina la ventana de operación del equipo calculando el tiempo total del procesamiento de dichos datos.
- Se calcula el tiempo en que el equipo este detenido consultando si el equipo esta energizado mediante la variable de corriente. Esta debe ser cercana a 0A.
- Se registra el número total de detenciones del equipo.
- Se calculan los KPIs de mantenimiento.
- Se almacenan los datos de 'tiempo maquina detenida', 'ventana de operación', 'fallas detectadas', 'MTTF', 'MTTR' Y 'MTBF' en una DB de 'KPIs de mantenimiento'.
- La evaluación se realiza cada 30 segundos



```
operadores > app5.py > calcular_tiempos_payload_bajo_5
1 from pymongo import MongoClient
2 from datetime import datetime, timedelta
3 import time
4
5 def calcular_tiempos_payload_bajo_5():
6     uri = "mongodb+srv://sebastianromerob:sebal23@cluster0.obfr6.mongodb.net/test?retryWrites=true&w=majority"
7     client = MongoClient(uri)
8
9     try:
10        db = client['test']
11        coleccion = db['corriente']
12
13        # Crear o seleccionar la base de datos 'MTTF' y la colección 'resultados'
14        monitoreo = client['test']
15        resultados_collection = monitoreo['KPIs']
16
17        # Realizar la consulta para obtener todos los registros de 'payload' con tiempo
18        registros = list(coleccion.find(
19            {'payload': {'$exists': True}}, # Obtiene todos los registros con 'payload'
20            {'_id': 0, 'tiempo': 1, 'payload': 1}
21        ).sort('tiempo', 1)) # Ordena por tiempo en orden ascendente
22
23        if not registros:
24            print("No se encontraron registros.")
25            return
26
27        # Calcular el periodo total de tiempo de todos los registros
28        tiempo_inicio_total = registros[0]['tiempo']
29        tiempo_fin_total = registros[-1]['tiempo']
30
31        # Calcular la duración total de todos los registros en segundos
32        duracion_total_periodo = (tiempo_fin_total - tiempo_inicio_total).total_seconds()
33
34        # Convertir la duración total a horas
35        duracion_total_periodo_horas = duracion_total_periodo / 3600 # Convertir segundos a horas
36
37        duracion_total = 0
38        tiempo_inicio = None
39        cambios_umbral = 0 # Contador para cambios de umbral
40        payload_anterior = None # Para rastrear el cambio de umbral
41
42        # Recorrer los registros para calcular el tiempo donde payload es menor a 1
43        for i, registro in enumerate(registros):
44            payload = registro['payload']
45            tiempo_actual = registro['tiempo']
46
47            # Si payload es menor a 1 y no hemos registrado un tiempo de inicio
48            if payload < 5 and tiempo_inicio is None:
49                tiempo_inicio = tiempo_actual
50
51            # Si payload es mayor o igual a 1 y tenemos un tiempo de inicio
52            elif payload >= 5 and tiempo_inicio is not None:
53                tiempo_fin = tiempo_actual
54                if isinstance(tiempo_inicio, datetime) and isinstance(tiempo_fin, datetime):
55                    duracion = (tiempo_fin - tiempo_inicio).total_seconds()
56                    duracion_total += duracion
57                tiempo_inicio = None
58            cambios_umbral += 0 # Cambio de umbral de menor a mayor a 1
59
```

Figura 33 Código para KPIs

Fuente: elaboración propia. Año: 2024



```
operadores > app5.py > calcular_tiempos_payload_bajo_5
5 def calcular_tiempos_payload_bajo_5():
60     # Detectar el cambio de umbral (de mayor a menor a 1)
61     if payload_anterior is not None:
62         if payload_anterior < 5 and payload >= 5:
63             cambios_umbral += 0 # Cambio de umbral de menor a mayor a 1
64         elif payload_anterior >= 5 and payload < 5:
65             cambios_umbral += 1 # Cambio de umbral de mayor a menor a 1
66
67     payload_anterior = payload # Actualizamos el valor anterior de payload
68
69     # Si el último registro sigue siendo menor a 5, calcula hasta el último registro
70     if tiempo_inicio:
71         tiempo_fin = registros[-1]['tiempo']
72         if isinstance(tiempo_inicio, datetime) and isinstance(tiempo_fin, datetime):
73             duracion = (tiempo_fin - tiempo_inicio).total_seconds()
74             duracion_total += duracion
75
76     # Convertir la duración total a horas
77     duracion_total_horas = duracion_total / 3600 # Convertir segundos a horas
78
79     # Evitar división por cero si no hubo cambios de umbral
80     if cambios_umbral == 0:
81         MTF = 0
82         MTR = 0
83         MTBF = 0
84     else:
85         MTF = (duracion_total_periodo_horas - duracion_total_horas) / cambios_umbral # TIEMPO MEDIO HASTA LA FALLA
86         MTR = (duracion_total_horas / cambios_umbral) # TIEMPO MEDIO DE REPARACION
87         MTBF = MTF + MTR # TIEMPO MEDIO ENTRE FALLAS
88
89     # Redondear a 5 decimales para los resultados
90     MTF = round(MTF, 5)
91     MTR = round(MTR, 5)
92     MTBF = round(MTBF, 5)
93
94     print(f"Duración total maquina detenida a 5: {duracion_total_horas:.5f} horas")
95     print(f"ventana de operacion : {duracion_total_periodo_horas:.5f} horas")
96     print(f"fallas detectadas: {cambios_umbral}")
97     print(f"Tiempo medio de reparación (MTR): {MTR:.5f} horas")
98     print(f"Tiempo medio hasta la falla (MTF): {MTF:.5f} horas")
99     print(f"Tiempo medio entre fallas (MTBF): {MTBF:.5f} horas")
100
101     # Guardar el resultado en la base de datos 'test'
102     KPIs = {
103         'timestamp': datetime.utcnow(),
104         'tiempo maquina detenida': round(duracion_total_horas, 5),
105         'ventana de operacion': round(duracion_total_periodo_horas, 5), # Agregar este valor
106         'fallas detectadas': cambios_umbral,
107         'MTF': MTF,
108         'MTR': MTR,
109         'MTBF': MTBF
110     }
111     resultados_collection.insert_one(KPIs)
112
113     finally:
114         client.close()
115
116     # Bucle que ejecuta la función cada 30 segundos
117     while True:
118         calcular_tiempos_payload_bajo_5()
119         time.sleep(30) # Esperar 30 segundos antes de la siguiente ejecución
120
```

Figura 34 Código para KPIs 2

Fuente: elaboración propia. Año: 2024

```
_id: ObjectId('678671563597c5fcbd9ceeba')
timestamp: 2025-01-14T14:14:46.329+00:00
tiempo maquina detenida: 0.38222
ventana de operacion: 0.71417
fallas detectadas: 3
MTF: 0.11065
MTR: 0.12741
MTBF: 0.23806
```

Figura 35 KPIs de mantenimiento

Fuente: elaboración propia. Año: 2024



#### 5.4.4. Almacenamiento de datos

Una vez los datos son procesados en node red mediante los nodos prescritos, se utiliza el nodo de mongodb3 para insertar los datos en formato BSON (Binary JSON) en documentos, este nodo debe configurarse para insertar los datos en la plataforma de base de datos como servicio (DBaaS) de MongoDB Atlas por medio de los siguientes campos:

- **MongoDB Server:** Es la URL proporcionada por MongoDB Atlas a través de la herramienta Connect para acceder a los datos. Se copia la cadena de conexión: “mongodb+srv://sebastianromerob:<db\_password>@cluster0.obfr6.mongodb.net/”, reemplazando <db\_password> por la contraseña correspondiente. A continuación, se ingresa el nombre de la base de datos, como “test”, y se añade una sección adicional para manejar la conexión y las operaciones: ?retryWrites=true&w=majority&appName=Cluster0. Finalmente, se asigna un nombre al servidor y se ingresan el usuario y la contraseña correspondientes.

Editar nodo mongodb3 in > Editar nodo mongodb3

Eliminar

Propiedades

URI: mongodb+srv://sebastianromerob:seba123@cluster0.obfr6.mongodb.net/test?retryWrites=true&w=majority&appName=Cluster0

Name: sebastianromerob

Username: sebastianromerob

Password: .....

Connection Options: Stringified JSON

Parallelism Limit: -1

Figura 36 configuración de nodo MONGODB3

Fuente: elaboración propia. Año: 2024

- Se ejecuta como un servicio externo. Se ingresa el nombre de la colección donde se almacenarán los datos, junto con la operación de inserción para agregar datos en dicha colección. Finalmente, se asigna un nombre al nodo.



Editar nodo mongodb3 in

Eliminar

**Propiedades**

Service: External service

Server: sebastianromerob

Collection: dato\_temp

Operation: insert

Name: mongodb\_temp

Figura 37 configuraciones de nodo MONGODB3 2

Fuente: elaboración propia. Año: 2024

Adicionalmente, existe la posibilidad de interactuar con la base de datos de MongoDB Atlas usando la interfaz gráfica de MongoDB Compass ingresando la cadena conexión de Atlas en una nueva conexión de Compass.

cluster0

Manage your connection settings

URI ⓘ Edit Connection String

mongodb+srv://sebastianromerob:seba123@cluster0.obfr6.mongodb.net/

Name: cluster0 Color: No Color

Favorite this connection  
Favoriting a connection will pin it to the top of your list of connections

> Advanced Connection Options

How do I find my connection string in Atlas?  
If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the cluster to which you wish to connect.  
[See example](#)

How do I format my connection string?  
[See example](#)

Cancel Save Connect Save & Connect

Figura 38 Ingreso de cadena de conexión desde mongodb compass

Fuente: elaboración propia. Año: 2024

Clúster de MongoDB Atlas: Un clúster de MongoDB Atlas es un conjunto de instancias de base de datos que trabajan juntas para proporcionar escalabilidad, alta disponibilidad y redundancia. MongoDB Atlas permite crear y gestionar clústeres de manera completamente gestionada en la nube, sin tener que preocuparse por la infraestructura subyacente.



**test.corriente**  
STORAGE SIZE: 72KB LOGICAL DATA SIZE: 48.48KB TOTAL DOCUMENTS: 584 INDEXES TOTAL SIZE: 52KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

Generate queries from natural language in Compass

Filter Type a query: { field: 'value' } [Reset] [Apply] [Options]

```
{ "_id": ObjectId("6781d4900cefc165bc9826de"), "payload": 36.26, "topic": "vol1/temp", "qos": 1, "tiempo": "2025-01-11T02:16:48.000+00:00" }
```

```
{ "_id": ObjectId("6781d4950cefc165bc9826e1"), "payload": 36.36, "topic": "vol1/temp" }
```

1-20 of many results [PREVIOUS] [NEXT]

Figura 39 Documentos en base de datos “test” colección “corriente” en mongo db atlas

Fuente: <https://www.mongodb.com>. Año: 2024

**Compass**

My Queries

CONNECTIONS (6) [Search connections]

cluster0

- MTTF
- admin
- config
- demo
- local
- test
  - KPIs
  - corriente
  - dato\_temp
  - presion

cluster0 > test > presion

Documents 3 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

[ADD DATA] [EXPORT DATA] [UPDATE] [DELETE]

```
{ "_id": ObjectId("6789c73e2714d64a9c61b9c9"), "payload": 6.85, "topic": "vol2/pres", "qos": 1, "tiempo": "2025-01-17T02:58:06.000+00:00" }
```

```
{ "_id": ObjectId("6789c7432714d64a9c61b9cb"), "payload": 6.85, "topic": "vol2/pres", "qos": 1, "tiempo": "2025-01-17T02:58:11.000+00:00" }
```

Figura 40 Documentos en la base de datos “test” colección “presión” en mongo db compass

Fuente: <https://www.mongodb.com>. Año: 2024



### 5.4.5. Visualización y monitoreo

una de las etapas más importantes en el monitoreo es la visualización y supervisión de los datos, ya que permite comprender e interpretar los resultados y tomar decisiones. en este proyecto se utiliza grafana Cloud correspondiente a un Software como servicio en Cloud computing la cual tiene las siguientes características:

- Monitoreo gestionado: ofrece una plataforma de monitoreo completamente gestionada. En donde se puede recopilar almacenar y visualizar métricas de aplicaciones.
- Sin infraestructura que gestionar: no se necesita administración de servidores o infraestructura. grafana labs se encarga del mantenimiento y la escalabilidad de la plataforma.
- Integración de múltiples fuentes de datos: en este proyecto se integra MongoDB Atlas
- Alerta y notificaciones: se permiten configurar alertas para notificar si se superan ciertos umbrales o si ocurren eventos importantes.
- Escalabilidad automática: permite escalar la cantidad de métricas que se almacenan y procesan sin tener que gestionar la infraestructura.
- Paneles preconfigurados: ofrece dashboard preconfigurados para monitorizar aplicaciones y servicios, lo que facilita la puesta en marcha.

#### 5.4.5.1. Instalación de fuente de datos

Para conectar grafana a una base de datos de mongodb se instala un plugin (complemento específico) el cual permite interactuar en tiempo real con los datos de mongo DB y consultar visualizar y alertar sobre los datos.

- Conexión y autenticación  
se conecta a MongoDB mediante una cadena de conexión disponible en métodos de conexión de MongoDB Atlas `mongodb+srv://cluster0.obfr6.mongodb.net/test` y luego definir la autenticación mediante credenciales asignadas a MongoDB al crear la cuenta

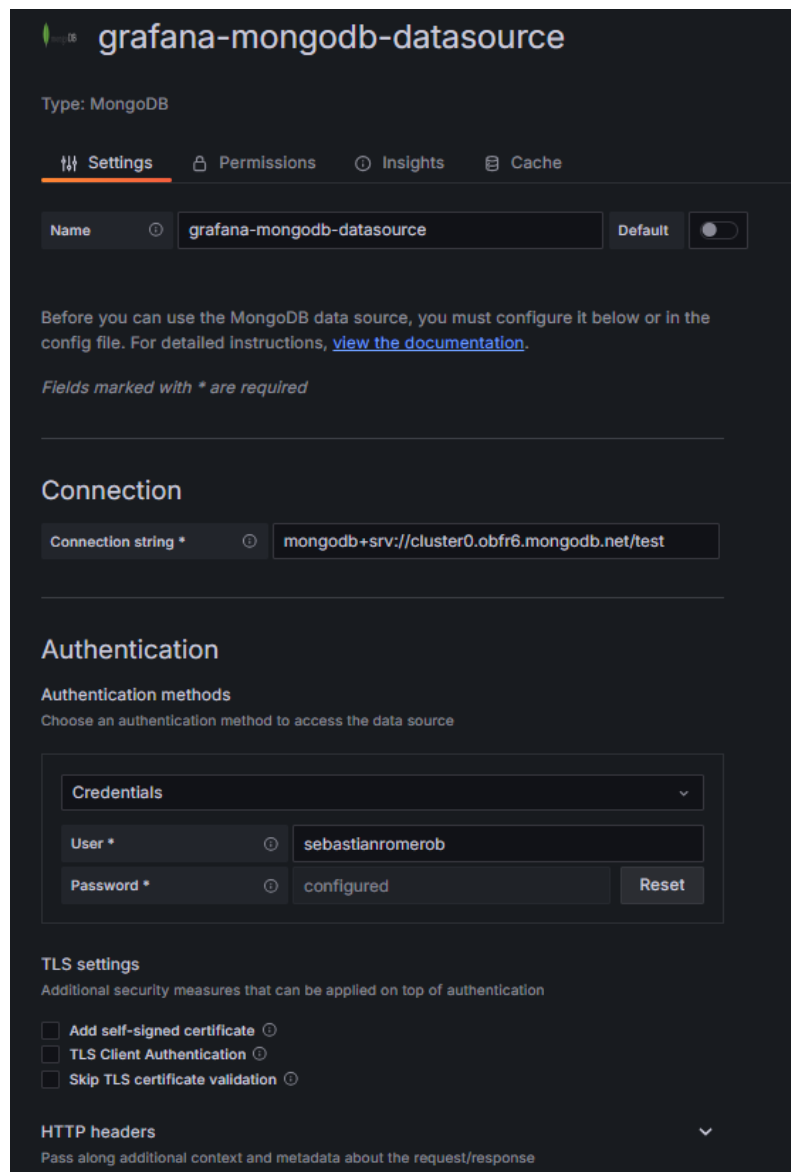


Figura 41 configuraciones de base de datos para Grafana

Fuente: <https://grafana.com/>. Año: 2024

- Ajustes adicionales: configuración opcional que se puede configurar para tener más control sobre el origen de datos de MongoDB mediante la validación de sintaxis de consulta, límite de filas de respuesta de Back-end o una conexión de origen de datos privados (PDC) y MongoDB.

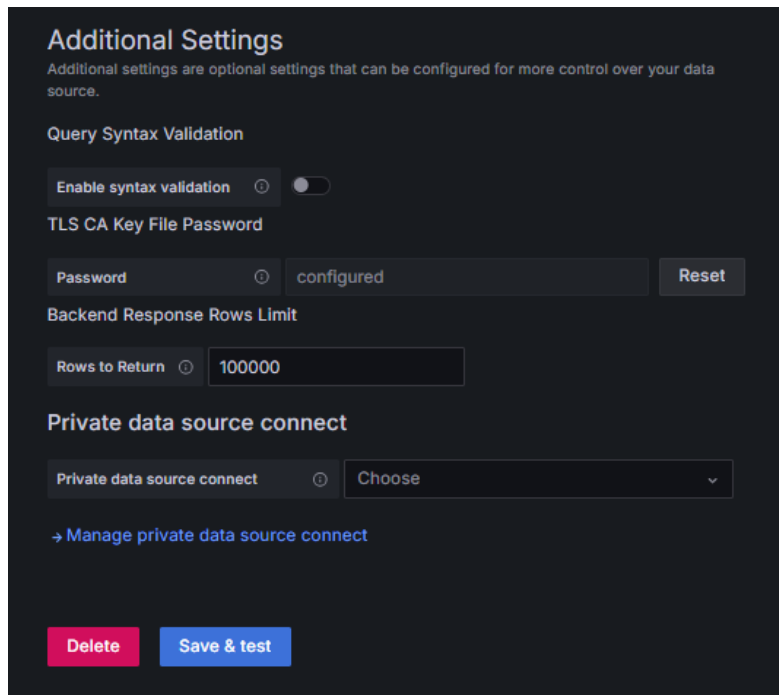


Figura 42 Ajustes adicionales base de datos Grafana

Fuente: <https://grafana.com/>. Año: 2024

### 5.4.5.2. Arquitectura de Dashboard

El siguiente dashboard muestra gráficos de series temporales, gráficos de barras (bar gauge), notificaciones de alarma y medidores (gauge), herramientas que facilitan el monitoreo y la supervisión de las variables del proceso y los KPIs de mantenimiento. Además, ofrece integración con la base de datos MongoDB Atlas, lo que permite una interactividad eficiente en las consultas, historial de datos y se adapta a las necesidades específicas de análisis.

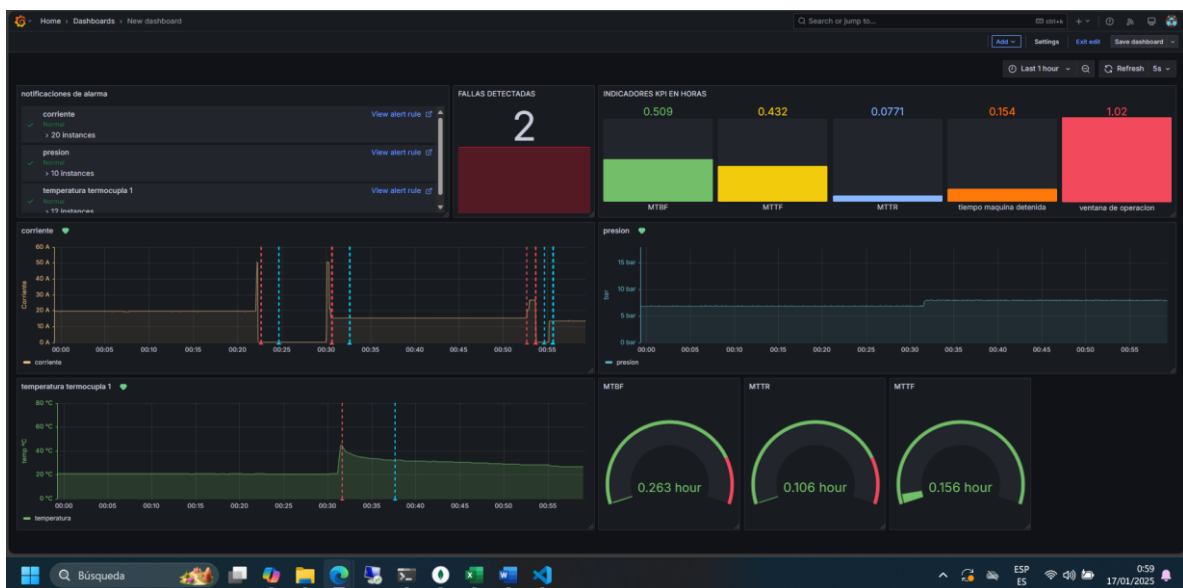


Figura 43 Dashboard grafana

Fuente: <https://grafana.com/>. Año: 2025

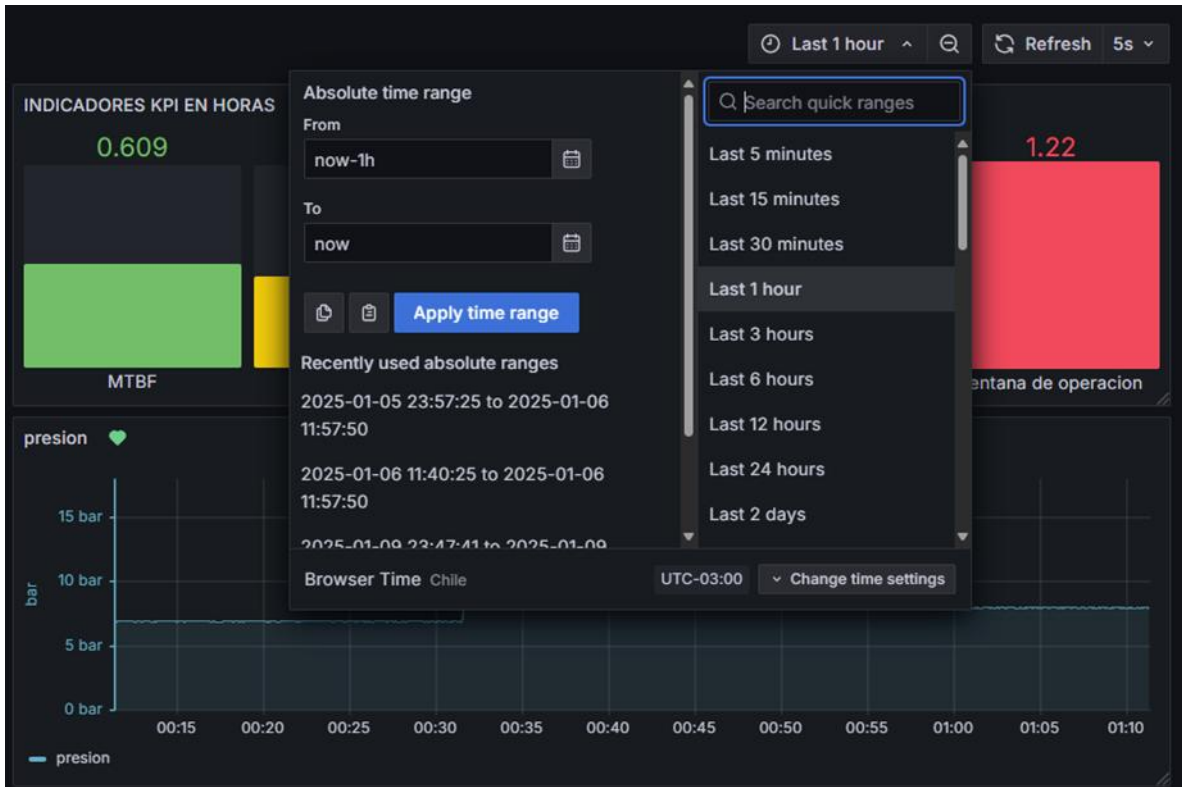


Figura 44 Ajuste de rango tiempo para visualizar los datos en el panel

Fuente: <https://grafana.com/>. Año: 2025

### 5.4.5.3. Consultas y transformación de datos para MongoDB

#### • Consultas:

Grafana proporciona un editor de consultas de MongoDB que admite la misma sintaxis que el Shell de MongoDB (MongoDB) pero con las siguientes limitaciones:

- Solo puede ejecutar un comando o consulta por consulta
- Solo se admiten comandos de lectura *find* *aggregate*
- ISOdate es el único constructor de objetos admitido

La operación *find* en MongoDB se utiliza para buscar documentos dentro de una colección o recuperar documentos específicos, mientras que la operación *aggregate* permite realizar un procesamiento de datos más avanzado, como obtener resúmenes y transformaciones complejas. Esto incluye operaciones como agrupar datos, calcular promedios, máximos, sumas, entre otros.

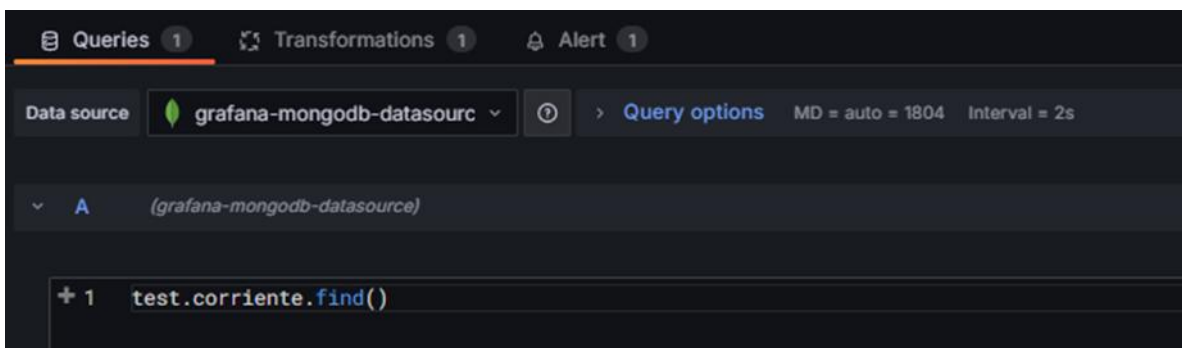


Figura 45 Operación find para buscar documentos dentro de la base de datos

Fuente: <https://grafana.com/>. Año: 2025



- **Transformaciones:**

Las transformaciones permiten manipular los datos devueltos por una consulta antes de que el sistema aplique la visualización con las transformaciones se puede:

- Cambiar el nombre de los campos
- Unir series temporales/datos similares a SQL
- Realizar operaciones matemáticas en consultas
- Usar la salida de una transformación como entrada para otra transformación

*En ocasiones, el sistema no puede representar gráficamente los datos transformados. Cuando eso suceda, haga clic en el botón de alternancia sobre la visualización para cambiar a una vista de tabla de los datos. Esto puede ayudarte a entender el resultado final de tus transformaciones.*

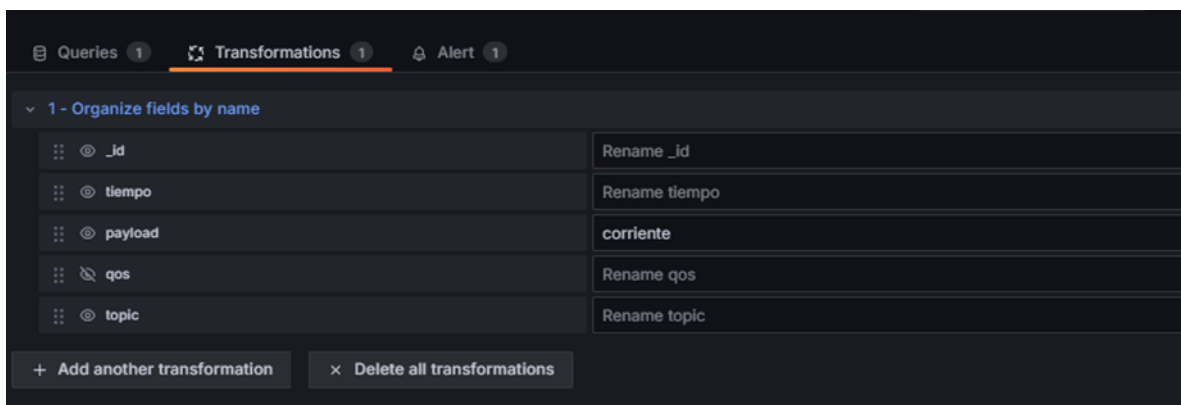


Figura 46 Funciones transformación

Fuente: <https://grafana.com/>. Año: 2025

Las funciones de transformación son útiles para modificar los datos antes de visualizarlos, estas funciones son útiles cuando se requiere realizar tareas como filtrar, agrupar, ordenar o realizar cálculos sobre los datos directamente en Grafana sin la necesidad de modificar la consulta en la base de datos algunas funciones son:

- Rename by regex (Renombrar por expresión regular)
  - Descripción: Permite cambiar el nombre de las series de tiempo utilizando expresiones regulares. Es útil cuando deseas aplicar un nombre más claro a las métricas o series.
  - Uso: Si tienes varias series con nombres complejos, puedes usar esta transformación para renombrarlas según un patrón específico.
  - Ejemplo: Cambiar el nombre de las series que contienen el texto "CPU" por "CPU Usage".



- Add field from calculation (Agregar campo a partir de cálculo)
  - Descripción: Permite agregar un nuevo campo calculado a los datos basados en operaciones aritméticas entre campos existentes.
  - Uso: Puedes realizar cálculos sobre los valores de los campos, como sumas, restas, multiplicaciones, o divisiones.
  - Ejemplo: Si tienes campos de "ventas" y "costos", puedes agregar un campo nuevo "ganancias" como la diferencia entre ambos.
- Organize fields (Organizar campos)
  - Descripción: Permite cambiar el orden de los campos o eliminar campos que no sean necesarios para la visualización.
  - Uso: Organiza los datos de manera más lógica y limpia, mostrando solo los campos que son relevantes.
  - Ejemplo: Si estás trabajando con una tabla que tiene varias columnas, puedes mover las columnas para tener un orden más coherente o eliminar las que no quieras mostrar.
- Filter by name (Filtrar por nombre)
  - Descripción: Permite filtrar las series de tiempo según su nombre o etiqueta.
  - Uso: Es útil cuando tienes múltiples series de datos y solo deseas mostrar aquellas que coinciden con un nombre específico.
  - Ejemplo: Si tienes series con nombres como "CPU" y "Memoria", puedes filtrar solo la serie "CPU" para mostrarla en el panel.
- Filter by value (Filtrar por valor)
  - Descripción: Filtra los datos basados en los valores de un campo específico.
  - Uso: Permite mostrar solo los registros que cumplen con ciertas condiciones (por ejemplo, valores mayores que un umbral).
  - Ejemplo: Mostrar solo los valores donde el "usage" es mayor a un cierto porcentaje.
- Group by (Agrupar por)
  - Descripción: Permite agrupar los datos por un campo específico (como por tiempo, por etiqueta, etc.), lo que es útil para hacer agregaciones o resúmenes.
  - Uso: Agrupar los datos para realizar operaciones agregadas, como sumas, promedios, máximos, etc.
  - Ejemplo: Agrupar las métricas de uso de CPU por hora para obtener un promedio de uso de CPU cada hora.



- Sort by (Ordenar por)
  - Descripción: Permite ordenar los datos según un campo específico, ya sea en orden ascendente o descendente.
  - Uso: Es útil cuando deseas ordenar las series de tiempo según sus valores o etiquetas.
  - Ejemplo: Ordenar los datos de las métricas de uso de memoria de mayor a menor.
- Join by field (Unir por campo)
  - Descripción: Permite unir múltiples series de datos por un campo común.
  - Uso: Puedes combinar datos provenientes de diferentes consultas en un solo conjunto de datos, basándote en un campo compartido entre ellas.
  - Ejemplo: Si tienes dos fuentes de datos diferentes, como "CPU" y "Memoria", puedes unirlos por el campo de tiempo o de etiquetas para tener una vista consolidada.
- Transpose (Transponer)
  - Descripción: Cambia las filas por columnas en los datos, es decir, invierte la disposición de los datos para que las series de tiempo se muestren como columnas y las etiquetas como filas.
  - Uso: Se utiliza cuando quieres cambiar la estructura de presentación de los datos, especialmente en tablas.
  - Ejemplo: Cambiar una tabla con filas de diferentes métricas en columnas que representan las métricas.
- Add time column (Agregar columna de tiempo)
  - Descripción: Agrega una columna de tiempo a los datos, útil si tus datos no contienen explícitamente un campo de tiempo.
  - Uso: Si tienes datos que no incluyen un campo de fecha, esta transformación agrega una columna de tiempo con los valores de tiempo apropiados.
  - Ejemplo: Para una tabla con datos sin marca de tiempo, puedes agregar la fecha y hora actual como un campo adicional.
- Remove fields (Eliminar campos)
  - Descripción: Elimina uno o varios campos de los resultados de la consulta.
  - Uso: Es útil cuando tienes datos con campos innecesarios que no deseas mostrar en el panel.
  - Ejemplo: Eliminar el campo de identificación de serie si no es relevante para la visualización.
- Normalize field (Normalizar campo)
  - Descripción: Permite normalizar los valores de un campo específico dentro de un rango determinado (por ejemplo, entre 0 y 1).



- Uso: Es útil cuando se quiere escalar los valores de un campo para compararlos de manera más efectiva.
- Ejemplo: Normalizar los valores de uso de CPU para que siempre estén entre 0 y 1, independientemente de su escala original.
- Add field from regex (Agregar campo a partir de regex)
  - Descripción: Permite extraer información de un campo utilizando expresiones regulares y crear un nuevo campo basado en esos datos extraídos.
  - Uso: Ideal cuando tienes un campo con datos estructurados (como texto o URL) y quieres extraer una parte de esos datos.
  - Ejemplo: Extraer el nombre de un host de una URL y crear un nuevo campo con ese valor.

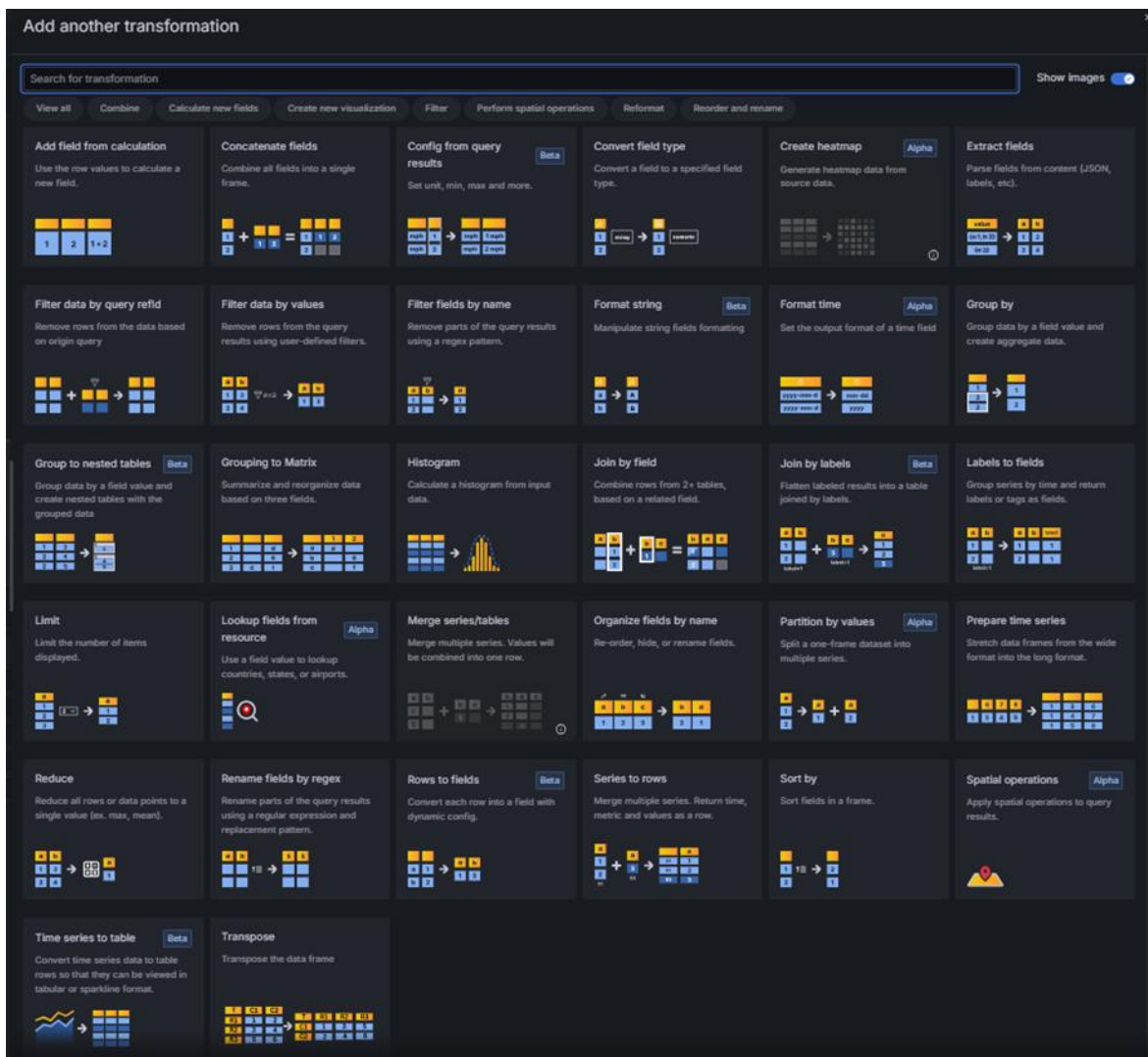


Figura 47 Funciones de transformación

Fuente: <https://grafana.com/>. Año: 2025



#### 5.4.5.4. Configuración de alarmas y notificaciones

La configuración de alarmas y notificaciones en Grafana es un proceso importante para monitorear y recibir alertas cuando las métricas del equipo superan ciertos umbrales o se presentan condiciones críticas, con Grafana se establecen alertas directamente en los paneles y se puede enviar notificaciones a través de múltiples canales, en este caso usaremos el correo electrónico. A continuación, se explica el proceso de configuración de alarmas y notificaciones:

1. Agregar y configurar canal de notificación: en este paso se agrega un canal de notificación de Email para recibir alertas por correo electrónico y luego se configura para asignar un nombre de alerta e introducir la dirección de correo electrónico.

Figura 48 configuración canal de notificación  
Fuente: <https://grafana.com/>. Año: 2025



2. Configurar la condición de alerta: se configura la frecuencia de evaluación (en el caso de prueba se configuró cada 30 segundos) las condiciones para la activación de la alerta, por ejemplo (la variable, la condición y el umbral que activara la alerta).

**Edit rule**

**1. Enter alert rule name**  
Enter a name to identify your alert rule.

Name  
corriente

**2. Define query and alert condition**  
Define query and alert condition [Need help?](#) Advanced options

A grafana-mongodb-datasource Options a minute, MD = 43200, Min. Interval = 30s Set as alert condition Run query

```
1 test.corriente.aggregate({
2   {
3     $sort: { _id: -1 }
4   },
5   {
6     $limit: 10
7   }
8 })
```

Add query

**Rule type**  
Select where the alert rule will be managed. [Need help?](#)

Grafana-managed Data source-managed

The alert rule type cannot be changed for an existing rule.

**Expressions**  
Manipulate data returned from queries with math and other operations.

**B Threshold** Alert condition  
Takes one or more time series returned from a query or an expression and checks if any of the series match the threshold condition.

Input C  
IS ABOVE 20  
Custom recovery threshold

**C Reduce** Set as alert condition  
Takes one or more time series returned from a query or an expression and turns each series into a single number.

Input A  
Function Last Mode Strict

**D Classic condition (legacy)** Set as alert condition  
Takes one or more time series returned from a query or an expression and checks if any of the series match the condition. Disables multi-dimensional alerts for this rule.

Conditions WHEN last() OF A IS ABOVE 20

Add expression Preview

Figura 49 Consulta y condición de la alerta

Fuente: <https://grafana.com/>. Año: 2025



3. La evaluación de la alarma requiere definir un período durante el cual la condición debe cumplirse para activar la alerta. En este caso, la alerta se dispara de inmediato tan pronto como se cumple la condición. Además, se configura la notificación especificando el canal de contacto y el mensaje que se envía.

4. Set evaluation behavior  
Define how the alert rule is evaluated. [Need help?](#)

Evaluation group and interval  
temp or [+ New evaluation group](#)

All rules in the selected group are evaluated every 1m. [↗](#)

Pending period  
Period during which the threshold condition must be met to trigger an alert. Selecting "None" triggers the alert immediately once the condition is met.

0s  
**None** 1m 2m 3m 4m 5m

Pause evaluation [↗](#)

[Configure no data and error handling](#)

5. Configure notifications [Advanced options](#)

Select who should receive a notification when an alert rule fires.

Recipient  
Notifications for firing alerts are routed to a selected contact point. [Need help?](#)

Alertmanager: grafana

Contact point  
SEBASTIAN ANTONIO ROMERO BUSTOS [View or create contact points](#)

Muting, grouping and timings (optional) [-](#)

6. Configure notification message  
Add more context to your alert notifications. [Need help?](#)

Summary (optional)  
Short summary of what happened and why.

variable fuera de rango [↗](#)

Description (optional)  
Description of what the alert rule does.

Enter a description... [↗](#)

Runbook URL (optional)  
Webpage where you keep your runbook for the alert.

https://

Dashboard and panel  
[New dashboard](#) [corriente](#) [↗](#) [↗](#) [↗](#)

[+ Add custom annotation](#)

Figura 50 configuraciones de evaluación inmediata y mensaje de notificación

Fuente: <https://grafana.com/>. Año: 2025



- Una alerta se activa al exceder los valores establecidos, realizando una evaluación automática, enviando notificaciones por correo y permitiendo identificar el estado de alerta en el dashboard.



Figura 51 estados de alerta de métrica de corriente

Fuente: <https://grafana.com/>. Año: 2025

**Firing** corriente [View alert](#)

**Summary**  
variable fuera de rango

**Description**  
se ha sobrepasado la corriente nominal

**Values**  
Estado de alerta=1 In=24.42

**Labels**

alertname	corriente
_id	ObjectID("67e8c133eb43572f64b27918")
grafana_folder	GrafanaCloud

[Silence](#) [View dashboard](#) [View panel](#)

Observed 33s before this notification was delivered, at 2025-03-30 03:57:40 +0000 UTC

Figura 52 Notificación al correo electrónico de alerta para métrica de corriente.

Fuente: <http://mail.google.com/mail/u/0>



### 5.5. Puertos y enlaces para acceso a plataformas.

- Node-Red: puerto 1880
- 8883: utilizado para las conexiones seguras de MQTT a través de TLS/SSL
- HiveMQ: <https://console.hivemq.cloud/clusters/>
- MongoDB: <https://cloud.mongodb.com/>
- Grafana: <https://usuario.grafana.net/>
- Dashboard de Grafana: <https://usuario.grafana.net/d/ee6io2gc7miv4d/new-dashboard?orgId=1&from=now-15m&to=now&timezone=browser&refresh=5s>

Para acceder base de datos y dashboard se debe iniciar sesión con cuenta correspondiente.



## 6. Conclusión

El desarrollo de un sistema de monitoreo remoto para el equipo cocedor en la industria pesquera representa un avance significativo en la optimización de las operaciones y la eficiencia del mantenimiento. Al integrar tecnologías avanzadas como Raspberry Pi, DAQ HAT MCC, HiveMQ, Node-RED, MongoDB Atlas y Grafana Cloud, se logra una solución escalable, flexible y accesible que permite la recolección, análisis y visualización en tiempo real de datos críticos.

Este sistema no solo facilitaría la detección temprana de posibles fallas, sino que también contribuiría a mejorar la calidad del proceso al identificar problemas antes de que se conviertan en fallas críticas. La incorporación de indicadores KPI de mantenimiento potenciaría la capacidad de tomar decisiones informadas, garantizando la continuidad operativa y permitiendo una gestión más efectiva del mantenimiento predictivo, lo que resulta en menores costos y un mayor tiempo de actividad.

Con la implementación adecuada de MTTR y MTBF se espera reducir el tiempo de inactividad en un 15-30%. Esta mejora se sustenta en estudios de confiabilidad y el mantenimiento basado en condición (CBM) el cual es desarrollado por la ISO55000 sobre gestión de activos la cual desarrolla investigaciones sobre el mantenimiento basado en condición (CBM), al identificar problemas de manera preventiva y agilizar las reparaciones.

Por otro lado, al aumentar el MTBF y reducir el MTTR, la disponibilidad del equipo puede mejorar en un rango del 10-25% de acuerdo con la fórmula  $\text{Disponibilidad} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$ , dependiendo de la criticidad del equipo y las condiciones iniciales.

Además, se espera que con el monitoreo de MTTF combinado con MTTR puede ayudar a planificar las reparaciones y reducir los costos de mantenimiento correctivo en un 20-40%. Esta mejora se basa en el análisis comparativo de los costos de mantenimiento correctivo versus el mantenimiento predictivo. Los estudios realizados por la Society for Maintenance & Reliability Professionals (SMRP) sobre el impacto del mantenimiento basado en monitoreo de condición respaldan esta reducción de costos. Otro beneficio esperado al implementar el sistema es la optimización del uso de repuestos, ya que, la predicción de fallas basada en MTTF puede disminuir el uso no planificado de repuestos en un 15-20% al evitar reparaciones de emergencia, donde el uso de repuestos no planificados suele ser significativamente más alto.

En esencia, este enfoque integrado tiene el potencial de mejorar la eficiencia de los equipos, además de contribuir a fortalecer la competitividad y fomentar una operación más sostenible.



## 7. Bibliografías

ANARAYA ALBORNOZ & EMMA GIL. Los 3 tipos principales de mantenimiento más utilizados para optimizar las intervenciones. [en línea]. <<https://www.appvizer.es/revista/organizacion-planificacion/gestion-mantenimiento/tipos-de-mantenimiento.>> [consulta: 29 de mayo de 2020]

AVRAM PILTCH. Raspberry Pi 4: Review, Buying Guide and How to Use. [en línea]. <<https://www.tomshardware.com/reviews/raspberry-pi-4>>. [consulta: 17 de septiembre de 2022]

EMAIMT BLOG POSTS. Tiempo medio hasta el fallo (MTTF). [en línea]. <<https://www.emaint.com/es/blog-what-is-mean-time-to-failure-mttf/>> [consulta: 19 de junio de 2024]

EQUIPO DE GRAFANA LABS. Fundamentos de Grafana. [en línea]. <<https://grafana.com/tutorials/grafana-fundamentals/>> [consulta: 21 de noviembre de 2024]

EQUIPO DE HIVEMQ. Introducción al protocolo MQTT – Fundamentos de MQTT. [en línea]. < <https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt/>>. [consulta: 14 de febrero de 2024]

EQUIPO DE HIVEMQ. Seguridad y confianza de HiveMQ. [en línea]. < <https://www.hivemq.com/solutions/technology/hivemq-security/> >. [consulta: 14 de febrero de 2024]

EQUIPO DE HIVEMQ. Fundamentos de seguridad de MQTT. [en línea]. < <https://www.hivemq.com/mqtt/mqtt-security-fundamentals/>>. [consulta: 25 de mayo de 2015]

INFRASPEAK TEAM. Indicadores de mantenimiento: 7 KPIs para una gestión eficaz. [en línea]. <https://blog.infraspeak.com/es/indicadores-de-mantenimiento/>. [consulta: 23 de marzo de 2023]

JERSON MARTÍNEZ. Qué es Grafana y primeros pasos. [en línea]. <<https://openwebinars.net/blog/que-es-grafana-y-primeros-pasos/>> [consulta: 27 de diciembre de 2021]

MONGODB, INC. Get Started with Atlas. [en línea]. <<https://www.mongodb.com/docs/atlas/getting-started/>> [consulta: 2024]

RICHARD BARANIUK. Teorema de Muestreo. [en línea]. <[https://espanol.libretexts.org/Ingenieria/Senales\\_y\\_Sistemas\\_\(Baraniuk\\_et\\_al.\)/10%3A\\_Muestreo\\_y\\_Reconstrucci%C3%B3n/10.02%3A\\_Teorema\\_de\\_Muestreo](https://espanol.libretexts.org/Ingenieria/Senales_y_Sistemas_(Baraniuk_et_al.)/10%3A_Muestreo_y_Reconstrucci%C3%B3n/10.02%3A_Teorema_de_Muestreo)>. [consulta: 30 de octubre de 2022]



## ANEXO A

### Instalación de DAQ HAT a las raspberry Pi

- Apagar la Raspberry Pi.
- Localiza los 4 separadores.



Figura 53 Separador raspberry

Fuente: <https://www.raspberrypi.com/>. Año: 2024

- Fijar los 4 separadores a la raspberry Pi insertando la parte roscada macho a través de los 4 orificios de las esquinas de la raspberry Pi desde la parte superior y asegurando con tuercas incluidas desde la parte inferior.
- Instale el receptáculo 2\*20 con cables extendidos (MCC proporciona un Samtec SSQ-120-03-T-D o equivalente) en el cabezal GPIO de raspberry Pi presionando la parte hembra del receptáculo en los pines del cabezal, teniendo cuidado de no doblar los cables del receptáculo. El receptáculo de 2\*20 tiene el siguiente aspecto:

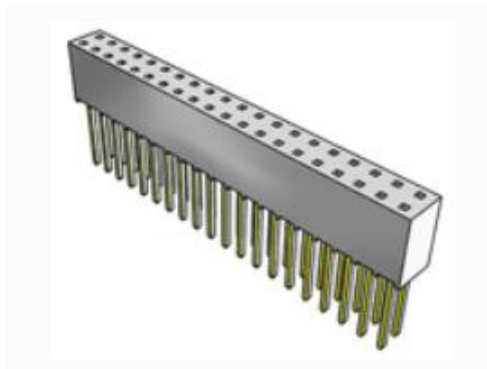


Figura 54 Receptáculo 2\*20

Fuente: <https://www.raspberrypi.com/> . Año: 2024

- El HAT debe estar en la dirección 0. Eliminar los puentes de las ubicaciones de encabezado de dirección A0-A2.



- Insertar la placa HAT en los cables del receptáculo 2\*20 de modo que los cables entren en los orificios en la parte inferior de la placa HAT y salgan a través del conector 2\*20 en la parte superior de la placa HAT. los 4 orificios de montaje en las esquinas del tablero deben alinearse con los separadores. Deslice el tablero HAT hacia abajo hasta que descansa sobre los separadores.
- Conectar todo el cableado de campo deseado a la placa instalada: no se podrá acceder a los terminales de tornillo una vez que se instalen placas adicionales sobre ella.
- Instale los 4 separadores de la placa adicional insertando la parte roscada macho a través de los 4 orificios de las esquinas de la placa HAT instalada y enroscarlas en los separadores de abajo
- Instale el siguiente receptáculo 2\*20 con cables extendidos en los cables del receptáculo de 2\*20 anterior presionando la parte hembra del nuevo receptáculo en los cables del receptáculo anterior
- Instalar los puentes de dirección apropiados en las ubicaciones de encabezado de dirección A0-A2 en la nueva placa HAT. Siempre debe haber una placa HAT en la dirección 0.

Dirección	A0	A1	A2	Configuración del puente
0				
1	Y			
2		Y		

Figura 55 Puentes DAQ HAT

Fuente: <https://www.raspberrypi.com/>. Año: 2024



- Insertar la nueva placa HAT en los cables del receptáculo 2\*20 de modo que los cables entren en los orificios en la parte inferior de la placa HAT y salgan a través del conector 2\*20 en la parte superior de la placa HAT. Los 4 orificios deben alinearse con los separadores.
- Repetir los pasos 7 al 11 para cada tablero que se va a agregar.
- Insertar los tornillos incluidos a través de los orificios de montaje en el tablero HAT superiores en los orificios roscados en los separadores y apretar ligeramente.



## ANEXO B

### Instalación de librerías

El proyecto este alojado en <https://github.com/mccdaq/daqhats>.

La instalación de las librerías consta de los siguientes pasos:

- Encender la Raspberry Pi e iniciar sesión, abrir una ventana de terminal si utiliza la interfaz gráfica.
- Actualice su lista de paquetes:

```
sudo apt update
```

- Actualizar los paquetes instalados y reiniciar

```
sudo apt full-upgrade
```

```
sudo reboot
```

- Instalar git

```
sudo apt install git
```

- Descargar este paquete en su carpeta de usuario con git:

```
cd ~
```

```
git clone https://github.com/mccdaq/daqhats.git
```

- Compilar e instalar la biblioteca y las herramientas compartidas. El instalador detectara las EEPROM de la placa HAT y guardara el contenido si es necesario

```
cd ~/daqhats
```

```
sudo ./install.sh
```



## Escritura de código para leer los datos desde la DAQ HAT

```
1 from __future__ import print_function
2 from time import sleep
3 from daqhats import mcc134, mcc118, HatIDs, HatError, TcTypes
4 import paho.mqtt.client as mqtt
5 import ssl
6 import threading
7
8 # Iniciar cliente MQTT
9 client = mqtt.Client()
10 client.tls_set(tls_version=ssl.PROTOCOL_TLS)
11
12 # Ajuste de usuario y contraseña
13 client.username_pw_set("TrabajoT", "Proyecto4")
14
15 #conexion con broker
16 try:
17     client.connect("6e10b8f0f10841239ecf63a7847028f4.s1.eu.hivemq.cloud", 8883)
18     print("Conectando al broker HIVE MQ CLOUD")
19 except Exception as e:
20     print(f"Error de conexion: {e}")
21     exit(1)
22
23 def on_connect(client, userdata, flags, rc):
24     print("Conectado con HIVEMQ CLOUD: " + str(rc))
25     client.subscribe("#") # Suscripcion de topicos
26
27 def on_message(client, userdata, msg):
28     print(f"Mensaje recibido en {msg.topic}: {msg.payload.decode()}")
29
30 client.on_connect = on_connect
31 client.on_message = on_message
32
33
34 # configuracion de muestreo y canales de daq
35 delay_between_reads_daq1 = 30 # Retraso de 5 segundos para DAQ 1 (mcc134)
36 delay_between_reads_daq2 = 5 # Retraso de 0.5 segundos para DAQ 2 (mcc118)
37 channel = 0 # Canal 0 de mcc134
38 chan = 7 #canal 7 de mcc118
39 chan1 = 6 #canal 6 de mcc118
40
41 # Evento para detener los hilos de lectura
42 stop_event = threading.Event()
```

Figura 56 Código lectura de datos 1

Fuente: elaboración propia. Año: 2024



```
44 # Funcion para leer de DAQ 1 (mcc134) y publicar
45 def leer_daq1(hat, channel):
46     while not stop_event.is_set(): # Mientras no se haya activado el evento de detencion
47         value = hat.t_in_read(channel)
48         canal0_termopar1 = "{:.2f}".format(value)
49         client.publish("termocupla1/temp", canal0_termopar1, qos=1)
50         sleep(delay_between_reads_daq1)
51
52 # Funcion para leer de DAQ 2 (mcc118) y publicar
53 def leer_daq2(hat1, chan, chan1):
54     while not stop_event.is_set(): # Mientras no se haya activado el evento de detencion
55         value1 = hat1.a_in_read(chan)
56         canal7_corriente1 = "{:.2f}".format(value1 * 10)
57         client.publish("vol1/temp", canal7_corriente1, qos=1)
58
59         value2 = hat1.a_in_read(chan1)
60         canal6_presion = "{:.2f}".format(value2 * 4)
61         client.publish("vol2/pres", canal6_presion, qos=1)
62         sleep(delay_between_reads_daq2)
63
64 def main():
65     """
66     Esta funcion se ejecutara automaticamente cuando el modulo se ejecute directamente.
67     """
68     tc_type = TcTypes.TYPE_K # Cambiar al tipo de termopar deseado
69
70     try:
71         # Obtener instancia del objeto seleccionado
72         address = 0
73         hat = mcc134(address)
74         address1 = 1
75         hat1 = mcc118(address1)
76         hat.tc_type_write(channel, tc_type)
77
78     try:
79         client.loop_start() # Inicio de bucle MQTT
80
81     # Crear y ejecutar los hilos para leer y publicar datos
82     hilo_daq1 = threading.Thread(target=leer_daq1, args=(hat, channel))
83     hilo_daq2 = threading.Thread(target=leer_daq2, args=(hat1, chan, chan1))
84
```

Figura 57 Código lectura de datos 2

Fuente: elaboración propia. Año: 2024

```
84 def main():
85     # Iniciar los hilos
86     hilo_daq1.start()
87     hilo_daq2.start()
88
89     # Mantener el hilo principal activo mientras los hilos se ejecutan
90     while True:
91         try:
92             sleep(1) # Mantener el programa en ejecucion
93         except KeyboardInterrupt:
94             # Si se detecta un Ctrl+C, activar el evento para detener los hilos
95             print("Interrumpido por el usuario. Deteniendo los hilos...")
96             stop_event.set() # Señal para que los hilos terminen
97             break # Salir del bucle principal
98
99     # Esperar a que los hilos terminen
100     hilo_daq1.join()
101     hilo_daq2.join()
102
103     finally:
104         client.loop_stop() # Detener el bucle de MQTT
105         client.disconnect() # Desconectar el cliente MQTT
106
107     except (HatError, ValueError) as error:
108         print('\n', error)
109
110 if __name__ == '__main__':
111     # Solo se ejecutara cuando el modulo se llame directamente
112     main()
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Conectando al broker HIVE MQ CLOUD
Conectado con HIVEMQ CLOUD: 0
Mensaje recibido en termocupla1/temp: 26.69
Mensaje recibido en vol1/temp: 44.16
Mensaje recibido en vol2/pres: 17.25
Mensaje recibido en vol1/temp: 44.57
Mensaje recibido en vol2/pres: 17.25
```

Figura 58 Código lectura de datos 3

Fuente: elaboración propia. Año: 2024

